SOFT COMPUTING

Computational Intelligence, Theory and Applications

Advances in Soft Computing

Editor-in-chief
Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series
can be found on our homepage:
springeronline.com

Henrik Larsen, Janusz Kacprzyk, Sławomir Zadrozny, Troels Andreasen, Henning Christiansen (Eds.)
Flexible Query Answering Systems, 2000
ISBN 3-7908-1347-8

Robert John, Ralph Birkenhead (Eds.)
Developments in Soft Computing, 2001
ISBN 3-7908-1361-3

Frank Hoffmann, Mario Köppen, Frank Klawonn, Rajkumar Roy (Eds.)
Soft Computing: Methodologies and Applications, 2005
ISBN 3-540-25726-8

Bernd Reusch (Ed.)
Computational Intelligence, Theory and Applications, 2005
ISBN 3-540-22807-1

Bernd Reusch
(Ed.)

Computational Intelligence, Theory and Applications

International Conference
8th Fuzzy Days in Dortmund, Germany,
Sept. 29–Oct. 01, 2004 Proceedings

Springer

Professor Dr. Bernd Reusch

Department of Computer Science 1
University of Dortmund
FB 8 Ektrotechnik
FB Informatik
LS Informatik I
Postfach 50 05 00
44221 Dortmund
Germany
E-mail: Bernd.Reusch@udo.edu

Library of Congress Control Number: 2005925479

ISSN print edition: 1615-3871
ISSN electronic edition: 1860-0794
ISBN-10 3-540-22807-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-22807-1 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable for prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media
springer.com
© Springer-Verlag Berlin Heidelberg 2005
Printed in The Netherlands

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: by the authors and TechBooks using a Springer LaTeX macro package
Cover design: Erich Kirchner, Heidelberg

Printed on acid-free paper SPIN: 11016205 89/TechBooks 5 4 3 2 1 0

Preface

For the 8th time since 1991 we invite researchers to participate in the Dortmund Fuzzy-Days. I am very glad that our conference has established itself as an international forum for the discussion of new results in the field of Computational Intelligence. Again all papers had to undergo a thorough review: each one was judged by five referees to guarantee a solid quality of the programme.

From the beginning of the Fuzzy-Days on, Lotfi A. Zadeh felt associated with the conference. I would like to express my gratitude for his encouragement and support and I am particularly glad that he once again delivers a keynote speech. Much to my pleasure Janusz Kacprzyk, Jong-Hwan Kim, Enrique Trillas and Eyke Hüllermeyer have also agreed to present new results of their work as keynote speakers.

With Prof. Jong-Hwan Kim of the Korean Advanced Institute of Science and Technology (KAIST) a colleague takes part in this year's Fuzzy-Days to whom I am indebted because he has inspired me and my colleagues to turn our attention towards the exciting field of Robot Soccer. Within a few years Robot Soccer has become an accepted test field for robotic research and the development of Multi Agent Systems. The importance of Computational Intelligence for these applications is evident. We address this topic not only with a dedicated session, but we will also organise a robot soccer match to demonstrate to the participants the entertaining quality of robot soccer as well as the challenging scientific problems.

I wish to thank all participants of the Dortmund Fuzzy-Days for their commitment to the conference and the organisers, namely Mrs Ulrike Lippe, for the excellent job they did. Last but not least, I am obliged to the German research council for their valuable financial support.

July 2004 *Bernd Reusch*

Contents

Neuro-Fuzzy Systems

Fuzzy Mathematics

Fuzzy Optimization

Poster Contributions

Evolutionary Algorithms

An Evolutionary Algorithm for the Unconstrained Binary Quadratic Problems

István Borgulya

University of Pécs, Faculty of Business and Economics, H-7621 Pécs, Rákóczi út 80, Hungary.
borgulya@ktk.pte.hu

Abstract. In this paper a new evolutionary algorithm (EA) is described for the unconstrained Binary Quadratic Problem, which is to be used with small, medium and large scale problems as well. This method can be divided into two stages, where each stage is a steady-state EA. The first stage improves the quality of the initial population. The second stage uses concatenated, complex neighbourhood structures for the mutations and improves the quality of the solutions with a randomized k-opt local search procedure. The bit selection by mutation is based on an explicit collective memory (EC-memory) that is a modification of the flee-mutation operator (Sebag et al. 1997). We tested our algorithm on all the benchmark problems of the OR-Library. Comparing the results with other heuristic methods, we can conclude that our algorithm belongs to the best methods of this problem scope.

Keywords: Binary quadratic programming; Large-size problems; Evolutionary algorithm.

1 Introduction

The general formulation the unconstrained binary quadratic programming problem (BQP) is the following:

$$\text{Max } f(x) = x^T \, Q \, x + c^T \, x$$

where $x \in \{0, 1\}^n$, $Q \in R^{nxn}$ is an n × n symmetric rational matrix.

BQP has a central role in combinatorial optimization. A large number of problems can be formulated as maximization of quadratic real values function in 0-1 variables. For that reason, BQP has been referred to as "the mother of all the combinatorial problems" (Bjorndal et al. 1995). E.g. BQP is equivalent to many classical combinatorial optimization problems such as maximum cut, maximum clique, maximum vertex packing, and maximum vertex cover. As important application we can see e.g. machine scheduling, capital budgeting, financial analysis and molecular conformation problems.

I. Borgulya: *An Evolutionary Algorithm for the Unconstrained Binary Quadratic Problems*, Advances in Soft Computing **2**, 3–16 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

The techniques which can be used to find the exact optimal solution are the branch and bound and branch and cut methods (e.g. (Helmberg and Rendl 1998), (Horst et al.2000), Pardalos and Rodgers 1990)). We can also use linearization techniques as well (e.g. the reformulation-linearization technique (Sherali and Adams 1998)), which converts nonlinear mixed-integer or zero-one problems into linear ones. This technique can be used not only to construct exact solution algorithms, but also to design powerful heuristic procedures.

Generally problems of sizes larger then n=100 cannot be solved in an acceptable time. Since most real-world problems are large-size problems, heuristics are used to find good solutions within a reasonable time (Glover et al. 2002). A large number of heuristic methods have been developed for solving the BQP. Various heuristic methods are also frequently used, such as one-pass heuristics (Glover et al. 2002), simulated annealing (SA) (Beasley 1999), (Katayama and Narihisa 2001), tabu search (TA) (Beasley 1999), (Glover et al. 1998), evolutionary algorithm (EA) and versions (e.g. genetic algorithm (GA), evolutionary strategy (ES)) (Lodi et al. 1999), (Merz and Freisleben 1999), scatter search (Glover 1997), memetic algorithms (MA) (Merz and Katayama 2001), as well as differently iterated search techniques (e. g. the parthenogenetic algorithms (PA) (Katayama and Narihisa 2001) or various subgradient-type methods (Shor 1998).

In this paper, we present a new heuristic method to solve the BQP. This heuristic is an EA that consists of 2 consequent stages. The first stage improves the quality of the initial population. The second stage uses concatenated, complex neighbourhood structures for the mutations, improves the quality of the solutions with a randomized *k-opt* local search procedure and uses a special filter and restart technique. The bit selection by mutation is based on an explicit collective memory (EC-memory) that is a modification of the flee-mutation operator (Sebag et al. 1997). The efficacy of the method was studied on the OR-Library benchmarks: the small, medium and large scale problems were all successfully solved. Comparing the results to others methods, we can state that our algorithm belongs to the best heuristic's methods of this problem scope.

In section 2, we describe our EA in general. Section 3 includes implementation details of our EAs. In Section 4, we present our computational experience, and we compare our results with other heuristic's results. Section 5 contains concluding remarks.

2 The principle of the new evolutionary algorithm

2.1 The structure of the algorithm

Hybrid EAs are frequently used for solving combinatorial problems. These methods improve the quality of the descendent solution for example with the application of a local search procedure, SA, or TS. The constitution of these

systems corresponds to an extension of an EA: for instance a local search procedure is applied at every step of the EA cycle.

The new EA unlike former hybrid EAs based on a single stage, uses a 2-stage algorithm structure in order to speed up convergence and to produce higher quality results. The first stage is a quick "preparatory" stage that is designated to improve the quality of the initial population. The second stage is a hybrid EA with some special operators.

Let us discuss the 2 EAs (stages) in more detail:

1. The first stage forms some solutions at random and then tries to improve them by randomly generating descendents. The descendent may replace the most similar one of the former solutions.
2. The second stage is a hybrid ES. The algorithm uses two different recombination operations, and concatenated, complex neighbourhood structures for the mutations. The recombination operation is a uniform or single-point recombination or otherwise simple copy-making. In selecting the parents, priority is given to the best, highest objective/fitness function value: the algorithm selects the fittest solution with 0.5 probability and another solution with $0.5/t$ probability (where t is the size of the population). By mutation we applied varying number of *bit-flip* and a special *bit-flip* (*bit-flip-flop).* We form the neighbourhood structure using: *some bit-flip-flops + some bit-flips.*
   The quality of the solutions is improved with a local search procedure. We applied the randomized *k-opt local search* (Merz and Katayama 2001). Finally in order to keep the diversity of the population we use a *filter* and a *restart* procedure. The *filter* selects only the best of the solutions close to each other, the other ones are deleted. The *restart* begins the second stage again, if the fittest solution didn't change in the last generations. It replaces the weakest solutions with new ones (70% of the population), and it applies the local search procedure on a part of the new individuals.

2.2 EC-memory

There are many variants of EC-memory methods that memorises the past events and/or past successes of the evolution process. We choose to adopt the method of (Sebag et al. 1997) that memorises the past failures of evolution through a virtual individual, the virtual loser. Let us see the principle of this method.

Let X, Y, Z and T be individuals, where X has a high fitness, and Y, Z and T all have a low fitness. Let the average of Y, Z and T be noted VL for virtual loser. The probability of mutating bit i in individual X should reflect how much it discriminates $X_i$from VL, that is, it should increase with $1 - |VL_i - X_i|$.

The EC memory is updated by relaxation from a fraction of the worse individuals in the current population. If α denotes some relaxation factor, *it* is the number of generation and dVL is the average of these worse individuals,

$$VL^{it+1} = (1\text{-}\alpha)\ VL^{it} + \alpha\ dVL\ (e.g.\ \alpha{=}0.2).$$

(Sebag et al. 1997) use a flee-mutation operator based on VL. In this mutation operator each bit to mutate is selected by tournament and the probability of mutating of one bit depends on both the bit itself and the individual at hand.

In our algorithm we use the virtual loser too. We don't apply the flee-mutation operator, but in our mutation structure we use the virtual loser by bit selection. We use two variants of the bit selection:

- The bit to mutate is selected randomly and its probability of mutating depends on the quantity p_i.($p_i = 1 - |VL_i - X_i|$.)
- Only the bits with the highest p_i values are selected and their probability of mutating depends on the quantity p_i.

Applied to the same problem, different bit selection mutations generally succeed with different efficacies. Even though their concurrent random application might increase computational cost with some of the problems, on average it increases efficacy. Therefore our algorithm randomly selects between the two types of bit selections for each mutation.

The new algorithm constructed this way, named EBQP (Evolutionary algorithm for the BQP) may be used both for small, medium and large scale BQP. Its structure and function are different from the former methods used to solve the BQP, as it uses an evolutionary framework, and it applies a special transformation for the mutations.

The EBQP got its final form after a number of assays. We tested the *1-opt* and *k-opt* local search procedures, and tested the mutation operation with and without the EC-memory. The test results made us conclude that the usage of the randomized k-opt local search yields 10 times better results (more accurate) than the 1-opt local search. With the new mutation operator based on EC-memory the efficiency of the algorithm proved to be better than without EC-memory: on the benchmark set the quality of the solutions was improved by 6-7 %, and the computational cost (measuring the average running time) was decreased by 25 %.

3 The new algorithm

3.1 The characteristics of the EAs

The main functions and characteristics of the EAs are the following:

Initial population. The same population and individuals are used in all stages. The first individuals of the P population are randomly generated from S. These are the first "solutions".

Fitness function. The algorithm uses the objective function f(x) as fitness function.

Selection operator. In the first stage descendents are randomly selected from S, without the application of any further operators (recombination, mutation). In the second stage the algorithm selects two different parents from the population: the first of them is the most appropriate solution with 0.5 probabilities.

Recombination operator. In the second stage the algorithm chooses from two options:

- With 0.8 probability it chooses the two parents from the mating pool and the descendent is constructed with uniform or single-point recombination.
- With 0.2 probability it chooses the best parent from the mating pool and the descendent is built by copy-making.

Mutation operator. In the second stage the algorithm has two options:

- After discrete or single-point recombination it applies the mutation operator with p_{rm} probabilities. The value of p_{rm} is adjusted by the user, and it is low by easy and high by difficult problems (e.g. p_{rm} is 0.05 or 0.95)
- After copy-making it applies the mutation operator.

The applied transformations are the following: *bit-flip* a single bit (variable) is flipped randomly in the descendent; *bit-flip-flop* if there are two randomly chosen bits, the i*th* and j*th* variables in the descendent having different values, the bits i and j will be flipped.

The *bit-flip* is to be considered as the random succession of sequential flips. The path of the flips follows the hypotenuses of the hypercube of n dimensions, and it gets from one vertex to another one step by step through the neighbouring vertexes. At each step the Hamming distance between the vertexes is 1. The *bit-flip-flop* is a special variant of the *bit-flip* transformation. Its path follows the diagonal lines of the hypercube, and instead of the neighbouring vertexes it flips to the transversal points. At each step the Hamming distance between the vertexes is 2.

The complex, multiple transformation is as follows:

$$some\ bit-flip-flops + some\ bit-flips$$

where the *bit-flip-flop* transformation is executed only with the probability of 0.5, and the *bit-flip* and *bit-flip-flop* transformations are executed varying times.

Our algorithm applies the two variants of bit selection with 0.5-0.5 probability. Therefore a *bit-flip-flop* transformation is executed when the probability of mutation for the first selected variable is higher than p_i (p_i= 1-($VL_i - X_i$)). At the *bit-flip* transformation, the chosen variable will also be flipped by the probability of p_i.

```
Begin
  calculate gains g_i for all I in {1,…,n}
  repeat
    x_prev=x, G_max=0, G=0,C={1,…,n}
    repeat
        Generate a random permutation RP[] of the set {1,…,n}
        For j=1 to n
            k=RP[j]
            if g_k>0 then
              G=G+g_k,G_max=g
              x_k=1-x_k, x_best=x, update gains g_i for all i
              C=C/{k}
            fi
        end for
        find j with g_j=max_{i∈C} g_i
        G=G+g_i
        If G>G_max then G_max=G, x_best=x fi
        x_j=1-x_j,C=C/{j}, update gains g_i for all i
    until C=∅
    if G_max>0 then x=x_best else x=x_prev fi
  until G_max≤0
end
```

Fig. 1. The *randomized k-opt-local-search* algorithm (Merz and Katayama 2001)

The virtual loser is defined after the termination of the first stage. It is periodically updated by using the weakest individuals. In the updating procedure we use 20% of the population

Local-search. In the EBQP we applied the *randomized k-opt-local-search* algorithm (Merz and Freisleben 2002), (Merz and Katayama 2001). The algorithm is shown in Figure 1, x is a given solution vector of length n and g is a gain vector of length n, that is stored with x. The kth gain of g denotes the cost of the neighbour solution, when a single bit k is flipped in the current solution:

$$gk = qkk\,(\overline{xk} - xk) + \sum_{i=1, i\neq k}^{n} qik\,(\overline{xk} - xk)$$

$$where\ \ \overline{xk} = 1 - xk$$

Moreover, g is efficiently updated each time a bit is flipped. The gains g_i do not have to be recalculated each time. Instead, it is sufficient to calculate the difference of the gains Δg_i :

$$g_i^{'} = \begin{cases} -g_i \;\; if \;\; i = k \\ g_i + \Delta g_i(k) \;\; otherwise \end{cases}$$
$$with \;\; \Delta g_i(k) = 2q_{ik}(\bar{x}_i - x_i)(x_k - \bar{x}_k)$$

Reinsertion. In the first stage, the algorithm compares the descendent with the most similar solution. If the descendent is better then the former solution, it is replaced by the descendent. In the second stage the algorithm compares the descendent with the best parent from the mating pool, or if the number of the individuals is less then the population size, it increases the number of the individuals (after restart).

Filtering. We added a check to the EBQP (in every kn*th* generation). In order to keep the diversity of the population we select only the best of the solutions close to each other, the other ones are deleted (x and x' are close to each other if the $d^H(x, x')$ is less than a predefined value, e.g. $n/4$). This filtering speeds up the convergence, too. The new individuals substituting the deleted ones are generated from the old individuals with 10 *bit-flip* mutation.

Restart strategy. If no new best individual in the population was found for more than 20 generations, the EBQP begins the second stage with an other population. The individuals excepting the best 30 % of the population are deleted, and new individuals are generated. The first *kn* individuals are generated in the same way as the descendents and one individual is generated in a generation. The rest of the new individuals are generated from the old individuals with 10 *bit-flip* mutations in the next kn*th* generation. So the population size will be full again at the time of the next check (at the next kn*th* generation).

Stopping criteria.The algorithm is terminated if the running time (in CPU seconds) is more than a prefixed time limit.

3.2 The main steps of the algorithm

Let us introduce the following notations:

- Let us denote the 2 stages as EA1 and EA2.
- Let the population of the it^{th} generation be denoted by P(it), and let $x_1, x_2, \ldots, x_t$ be the individuals of the P(it) population.
- Let us denote the complex neighbourhood transformation based on the virtual loser (*some bit-flip-flops+some bit-flips)* of the q descendent by *Nhs(q).* The two transformations are executed randomly, maximum min($n/2$, 50) times. The probability of applying the first transformation is 0.5.
- The measure of the similarity of the two solutions x and z is given by *H(x, z)= 1/(1 +* $d^H(x, z)$) where $d^H(x, z)$ is the Hamming distance of the solutions.
- Let us denote the procedure, which filters out and deletes the close solutions ($d^H(x, x') < n/4$) by *Filter.* At the end, the procedure generates new solutions to replace the deleted ones.

Parameters

6 parameters affect the run of the algorithm:

- t – the size of the population.
- *itt* – a parameter of the second stage. If the number of iterations (it) reaches *itt*, the second stage begins.
- *kn* – a parameter which determines the timing of checks. We use the *Filter* and *Restart* procedure only at every kn*th* iteration.
- m – a parameter of the *k-opt* local search. The outer repeat loop can run only m times.
- p_{rm} – a parameter of the second stage. The probability of the mutation operator after recombination.
- *timelimit* – a parameter for the stopping condition. The procedure is finished if the running time (in CPU seconds) is more than *timelimit*.

Variables:
it – the number of the iterations.

Procedure EBQP(t, *itt*, *kn*, p_{rm}, *timelimit*, opt, optp)
it:=0
Let $x_i \in\{0,1\}^n$ (i=1,. . .,t), P(it) ← $\{x_1\ ,\ \ldots,\ x_t\ \}$.
Compute $f(x_1)$, . . . , $f(x_t)$.
Do *itt* **times**
Selection of a $q \in S$.
Compute $f(q)$.
Let $H(q, x_{z:})$:=$max_j H(q, x_j); j, z \in$ {1,2,. . . ,t}

If $f(q) > f(x_z)$ **then** $x_z := q$ **fi**.
it:=*it*+1, P(it) ← P(*it*-1).
od.
Initial value of VL
/* second stage:
Repeat
Do *kn* **times**
Two parents x_i, x_j selection
Generation of the q descendant.
Nhs(q). Local search.
Compute $f(q)$.
If |P(it)|<t **then** P(it)= P(it) ∪ q
else
if $f(q) > f(x_i)$ **then** $x_i := q$ **fi**.
fi
it:=*it*+1, P(it) ← P(*it*-1).
od.

Update of VL
Filter, Restart .
until "running time" > *timelimit.*
optp= the best x individual, opt=f(optp)
end

3.3 Details of the implementation

When describing the algorithm, some heuristic solutions and the shortcuts of the computations were not described. Let us see them now one-by-one.

Speeding-up the computation.

The algorithm speeds up calculating the value of the objective function values faster. The base of the reduction of the computing time is the possibility, that the recombination and the mutation operators change only a certain fragment of the variables. The descendents are produced by some consequential bit transformations from one of the parents. Therefore it is enough to know the f(x) function value of the x parent only, and the f (x') function value of the x' descendent could be constructed from the former one by bit transformations. In summary, the algorithm calculates the f(x') function values from f(x) by using the g_k gain values of consequential bit transformations.

Speeding-up the local search.

To reduce the running time of the randomized *k-opt* local search (Figure 1), the terminal condition of the inner repeat loop can be modified so that the loop is terminated if there were no new x_{best} values for more than 50 iterations and the outer repeat loop can repeat only 1 or 5 times (the m value is 1 by easy and 5 by difficult problems).

Heuristic in the mutation

The EBQP executes the first element of the *bit-flip-flop + bit-flip* with the given probability, and the two transformations are repeated by a mutable number of times. After testing numerous possible alternatives, using the test problems from the OR-Library, we found *bit-flip-flop* transformation to be the most efficient at 0.5 probabilities. Defining the number of the transformations arouses a more complex problem. To demonstrate its complexity we can choose a parthenogenetic algorithm (Katayama and Narihisa 2001) as an example. This algorithm uses a parameter dependent on the task for the maximal number of bit transformations in a mutation. We found that the best results were attained at different parameter values at different task groups, with the algorithm proving most efficient when the maximum number was between 40 and 80.

We did not wish to choose the maximal number of the transformations as a parameter. Instead, we searched for a maximal number giving a result of average acceptability, but applicable at each task. After testing several options, two and fifty proved to be as the most apt number of maximal bit transformations. At the 5-10 element problem groups glov500, B1000 and B2500 of the OR-Library the average relative percentage deviation of the solution from the best known solution was individually calculated (The test problems are detailed in Section 4). Subsequently, the results were averaged for both the groups and individual tasks. Based on the comparative tests, we found 50 bit transformations to be the most apt.

4 Solutions of test problems

Test problems

We tested the EBQP with the benchmark set of the OR-Library (http://www.ms.ac.uk/info.html). Although we managed to solve each individual BQP problem in the OR-Library, we wish to show only three benchmark sets. This are two sets from Beaslay with 1000 and 2500 dimensions and the set from (Glover et al. 1998) with 500 dimension (notation: B1000, B2500, glov500). In each sets there are 5 or 10 instances.

Parameter selection

To achieve a quick and accurate solution we need appropriate parameter values. Studying some of the more complex problems of the benchmark set we analyzed the process of the convergence how the parameter values were affecting the convergence, the finding of the global optimum and the speed of the calculation.

So we analyzed the size of the population which should be used in EBQP to achieve a good trade-off between the probability of finding best-known solutions and the necessary run-time to do so. In general, best behaviour was obtained with a size of population (t) between 30 and 60. The size of the population of 30 was found appropriate for the small - medium ($\leq$1000) and the size of the population of 60 was found appropriate for larger (>1000) number of dimensions.

As a next step, we studied the frequency of checks (kn parameter) and the p_{rm} probabilities of the mutation. At the time of the check several functions are executed. This means running the VL update, the *Filter* and eventually the *Restart* procedure. It is inexpedient to run them often, because then the VL looses efficacy and we waste too much time on filtering. We found that the frequency of the checks is optimal around every 10 generations (kn=*10*). There were many possible values found for the p_{rm} probabilities. We found, that for the half of the test problems, namely the easier tasks p_{rm} =0 was

suitable. In the other half of the problems, especially the difficult and 2500 dimension problems $p_{\mathrm{rm}} = 1$ was more appropriate. We can further refine parameter settings with the $m$parameter of the *k-opt* local search. The m value can be chosen 1 by easy and 5 by difficult problems.

Finally, we studied the number of iterations of the first stage. We found the first stage to be necessary, because improving the first solutions is beneficial to the quality of the results and usually decreases the scatter of the result. However, 30 iterations are enough for improving the initial population.

Computation results

We tested the algorithm on the test problems from the OR-Library. The algorithm managed to find solutions which are either best known or within 0.03 % of the best known solutions for 100% of the test problems. Table 1 includes the results of the chosen test-problems. The table presents mean values calculated from a number of runs: all problems (tasks) were run 10 times. We used a predefined time limit of 60 seconds for each instance of the set glov500, 300 seconds for each of B1000 and 2000 seconds for each of B2500 (The EBQP was implemented in Visual Basic and ran on a Pentium 4 1.8 GHz with 256 MB RAM).

The Table 1 and 2 show the average solution values for large scale problems. In Table 1 we give the problem name, the task number in the problem group, the best known solution, the average relative percentage deviation of the solution from the best known solution (AD). The Table 2 shows the average running time in seconds to the best solutions.

Table 1. Average solution values for the large scale problems

Task	Glov500		B1000		B2500	
	Best known	AD	Best known	AD	Best known	AD
1	61194	0	371438	0.0064	1515944	0
2	100161	0	354932	0	1471392	0.0240
3	138135	0	371236	0	1414192	0
4	172771	0	370675	0	1507701	0
5	190507	0	352760	0.0015	1491816	0
6			359629	0	1469162	0
7			371193	0	1479040	0.0025
8			351994	0.0042	1484199	0.0011
9			349337	0.0011	1482413	0.0009
10			351415	0	1483355	0.0231

Comparative results

As for comparison, we chose different methods: the TS and SA heuristics by Beasley (TS_B, SA_B) [1], the SA heuristic by Katayama and Narihisa

Table 2. Average running time for the large scale problems

task	Glov500	B1000	B2500
1	27.6	62	821
2	14.0	22	1825
3	26.3	84	697
4	37.5	170	634
5	10.5	152	945
6		95	1509
7		115	1612
8		210	844
9		241	1254
10		151	1288

(SA_KN), the genetic local search by Merz and Freisleben (GLS_MF), the parthenogenetic algorithm by Katayama and Narihisa (PA), and the memetic algorithm by Merz and Katayama (MA).

The comparison was encumbered by the use of various programming languages, operating systems and computers. Only one appropriate aspect of comparison could be found, namely the average relative percentage deviation of the solution from the best known solution, so our table of comparison (Table 3) is based on the results of comparable accuracies. (The TA_B and SA_B program ran on Silicon Graphics, R4000 CPU with 100 MHz. The GLS_MF ran on Pentium II PC, 300MHz. The SA_KN algorithm was implemented in C and ran on a Sun Ultra 5/10, UltraSPARC-IIi 440 MHz under OS Solaris 7. The PA and MA programs ran on Sun a Ultra 5/10 UtlraSPARC-IIi 440MHz, too).

We compared the average results. Analyzing the results of Table 3 we can confirm that the MA algorithm has the best, and the PA has the second best results in general. In case of the B2500 problems the PA has the best, and at the B1000 problems the MA has the best results. The EBQP has also good results: his result is the second at the B1000 problems, and the third at the B2500 problems.

Regarding that the MA and PA are the most effective methods of BQP; we can conclude that the EBQP also belongs to the best available methods. The results of the EBQP we can improve probably. Chosen a more appropriate recombination operator, or other local search procedure, probably we can improve the quality of the solution, and we can reduce the running time. The choice of the p_{rm} probability and the m parameter of the *k-opt* local search we can automize too. With the help of competing subpopulations we can use different p_{rm} and m values parallel, and the algorithm searches the most appropriate values.

Table 3. Comparative results for large scale problems

Method	Glov500	B1000	B2500
	AD	AD	AD
SA_B (best) (Beasley 1999)	0.0720	0.0860	0.0640
TS_B (best) (Beasley 1999)	0.0020	0.0370	0.0920
GLS_MF (Merz and Freisleben 1999)	-	0.0290	0.0610
SA_KN (Katayama and Narihisa 2001)	0.0220	0.0290	0.0210
EBQP	0	0.0013	0.0051
PA (Katayama and Narihisa 2001a)	0	0.0022	0.0002
MA (Merz and Katayama 2001)	0	0.0000	0.0004

5 Summary

In this paper we presented a new heuristic algorithm, named EBQP for solving the BQP problem. The concept of the EBQP differs from the previous ones: the structure of the new algorithm is an evolutionary framework consisting of 2 stages, for the solution of BQP we introduce a problem-specific mutation operator and we use a *k-opt* local search procedure. So we use a complex neighbourhood structure for the mutations, where we can concatenate different neighbourhoods, and the bit selection by mutation is based on an explicit collective memory (EC-memory). The progression of the algorithm is influenced by 6 parameters, more of which have the same value for the different problems.

We can conclude that the EBQP was successfully tested with different kinds of BQP. The algorithm managed to find solutions which are either best-known or within 0.03% of the best known solutions for 100% of the benchmark problems of the OR-Library. Comparing the results with other heuristic methods, we can conclude that the EBQP belongs to the best evolutionary algorithms of this problem scope.

Acknowledgments

The Hungarian Research Foundation OTKA T 042448 and the ERFP-DD2002-HU-S-01 project supported the study.

References

1. Beasley JE (1999) Heuristic algorithm for the unconstrained binary quadratic programming problem, Working Paper, Imperial College.
2. Bjorndal MH, Caprara A, Cowling PI, Della Crosce F, Lourenço H, Malucelli F, Orman AJ, Pisinger D, Rego C, Salazar JJ (1995) Some thoughts on combinatorial optimization. European Journal of Operational Research 83: 153-170.

3. Glover F (1997) A template for scatter search and path relinking. In: Hao JK, Lutton E, Ronald E, Schoenauer D, Snyers D (eds) Artificial Evolution, Lecture Notes in Computer Science, 1363. Heidelberg, pp 13-54.
4. Glover F, Kochenberger GA, Alidaee B (1998) Adaptive memory tabu search for binary quadratic programs, Management Science. 44. (3): 336-345.
5. Glover F, Alidaee B, Rego C, Kochenberger GA (2002) One-pass heuristics for large-scale unconstrained binary quadratic problems, European Journal of Operational Research 137: 272-287.
6. Helmberg C, Rendl F (1998) Solving quadratic (0,1)-problem by semidefinite programs and cutting planes, Mathematical Programming 82: 388-399.
7. Horst R, Pardalos PM, Thoai NV (2000) Introduction to Global Optimization 2^{nd} Edition, Kluwer Ac. Pub. Dordrecht
8. Katayama K, Narihisa H (2001) Performance of simulated annealing-based heuristic for the unconstrained binary quadratic programming problem, European Journal of Operational Research 134: 103-119.
9. Katayama K, Narihisa H (2001a) On Fundamental Design of Parthenogenetic Algorithm for the Binary Quadratic Programming Problem, Proceeding of the 2001 Congress on Evolutionary Computing. Seoul. Vol. 1: 356-363.
10. Lodi A, Allemand K, Liebling M (1999) An evolutionary heuristic for quadratic 0-1 programming, European Journal of Operational Research 119: 662-670.
11. Merz P, Freisleben B (1999) Genetic algorithm for binary quadratic programming, in: Proceeding of the 1999 international Genetic and Evolutionary Computation Conference, Morgan Kaufmann, Los Altos, CA, pp 417-424.
12. Merz P, Freisleben B (2002) Greedy and Local Search Heuristics for Unconstrained Binary Quadratic Programming. Journal of Heuristics, vol. 8, no.2, pp 197-213.
13. Merz P, Katayama K (2001) A Hybrid Evolutionary Local Search Approach for the Unconstrained Binary Quadratic Programming Problem. Tech. Rep., Department of Computer Science, University of Tübingen, Germany. Accepted for publication in Bio Systems.
14. Pardalos PM, Rodgers GP (1990) Computational aspects of a branch and bound algorithm for quadratic zero-one programming, Computing 45: 131-144.
15. Sebag M, Schoenauer M, Ravisé C (1997) Toward Civilized Evolution: Developing Inhibitions. In: Bäck T (ed): Proc. of the 7^{th} International Conference on Genetic Algorithm. Morgan Kaufmann Pub. San Francisco, pp 291-298.
16. Sherali HD, Adams WP (1998) A reformulation-linearization technique for solving discrete and continuous nonconvex problems, Kluwer Academic Publishers, Dordrecht
17. Shor NZ (1998) Nondifferentiable Optimization and Polynomial Problems, Kluwer Academic Publishers, Dordrecht

Application of Genetic Algorithms by Means of Pseudo Gradient

Boriana Vatchova M. Sc.

Institute of Computer and Communication Systems – BAS, "Akad. G.Bonchev" str. bl. 2, 1113 Sofia, Bulgaria

Keywords: genetic algorithms, populations, chromosome, fitness function, crossover operator, mutation and selection, pseudo-gradient.

Introduction

Genetic algorithms (GA) and their capabilities for solving optimization problems have been thoroughly investigated recently in a number of application areas. These algorithms represent a class of stochastic algorithms based on methods of biological evolution. The chromosome is one of the smallest particles in the kernel of every cell and it defines the genetics of living organisms. Nature has the ability of finding suitable chromosomes through natural selection. Analogically, genetic algorithms can find optimal solutions by means of rational computational iterations [1, 2, 3, 4].

The iterations associated with genetic algorithms can be represented through a series of populations of arguments of the optimization task. The latter are called "computer implemented chromosomes" or only "chromosomes". Every new population of chromosomes (values of arguments of the optimization task) can be derived analogically to biological procedures in nature, known as crossover, mutation and selection.

For the first two procedures, there aren't routine methods for a reasonable choice. In this case, methods for random search are applied and selection is based on an operation for comparing the values of the fitness functions for the separate chromosomes in populations. Besides this, genetic algorithms usually remove chromosomes with lower values of their fitness function. [5]

This is a drawback of these algorithms because the removal of such chromosomes implies some loss of information received so far.

The purpose of this paper is to keep this kind of information for a limited number of iterations, as it is the case in nature.

B. Vatchova: *Application of Genetic Algorithms by Means of Pseudo Gradient*, Advances in Soft Computing **2**, 17–24 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

For the purpose of more objective assessment of the new chromosomes, the information received from previous generations of chromosomes, including the ones with lower values of their fitness function has to be kept. When hereditary information is used successfully from previous generations of chromosomes, it is normal to expect an acceleration of the speed of genetic algorithms due to a reduction of the number of iterations.

1 Essence of the Method

In most genetic algorithms, the initial population is generated through random selection. After a limited number of initial iterations based on crossover and mutation operators, an assessment is done for the less perspective sub-domains of definition area and their rapid discharge from chromosomes. These chromosomes are replaced by chromosomes in the more perspective sub-domains. It is possible to generate logical rule in the algorithm which will determine the future concentration of chromosomes in the perspective sub-domains by calculating the values of the pseudo gradient of optimization function. It is assumed in this case that the chromosomes in the next populations will be concentrated in sub-domains of all local optimums. In this way, an additional genetic algorithm must ensure that the chromosomes leave the sub-domains in which local optimums are not perspective candidates for a global optimum.

2 Pseudo-Gradient Function

Every population of chromosomes in genetic algorithms is based on a random choice which includes crossover probability P_C and mutation probability P_M.

This peculiarity is taken from nature but the probability for a reasonable choice is not very high, especially in the case of mutation. This implies a low speed of movement of chromosomes to the optimum of the function.

The choice of every next generation of chromosomes has to be made wisely. This can be done by means of an additional operator which can direct chromosomes very fast to a local, and after that, to the global optimum. This operator will calculate the gradient of the function on the basis of available information about previous populations.

Given the gradient of a function grad f(x,y,z,...), after a certain number of populations, e.g. 3 or 4, the next population is regenerated from the following pseudo grad algorithm:

$$\begin{aligned} x_{s+1} &= x_s + k_s \mathrm{grad}^* f_s(x_s, y_s, z_s, \ldots) \\ y_{s+1} &= y_s + k_s \mathrm{grad}^* f_s(x_s, y_s, z_s, \ldots) \\ z_s + 1 &= z_s + k_s \mathrm{grad}^* f_s(x_s, y_s, z_s, \ldots) \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \end{aligned} \tag{2.1}$$

where

$$\begin{aligned} \text{grad}^* f_s(x_s, y_s, z_s, \ldots) = [(\Delta f_s^*(x_s, y_s, z_s, \ldots)/\Delta x_s, \\ \Delta f_s^*(x_s, y_s, z_s, \ldots)/\Delta y_s, \Delta\, f_s^*(x_s, y_s, z_s, \ldots)/\Delta z_s), \ldots]^T \end{aligned} \tag{2.2}$$

is "pseudo gradient" of a fitness function:

$$\Delta f_s(x_s, y_s, z_s) = f_s(x_s, y_s, z_s) - f_{s-1}(x_{s-1}, y_{s-1}, z_{s-1}) \tag{2.3}$$

$$\Delta x_s = x_s - x_{s-1}, \Delta y_s = y_s - y_{s-1}, \Delta z_s = z_s - z_{s-1}$$

The symbol "T" means transposition of a vector, s is the current iteration of the chromosomes $x_s, y_s, z_s, \ldots$ and k_s is the variation step of the chromosomes;

By means of the operator for assessment of the pseudo gradient, the choice of new chromosomes is done on purpose and not randomly. The values of the pseudo gradient function are calculated on the basis of the finite differences of the variation of chromosomes and the corresponding finite differences of their fitness function. The calculation of the pseudo gradient function for the above statements does not match classical definition for a gradient (some chromosomes of some operations do not have variation).

The "pseudo gradient" is a vector whose components are calculated on the basis of the information about the variation of the fitness function and the chromosomes for the last 3–4 populations (iterations). For example, for the last 3 populations (marked as 1, 2, 3), the elements of the vector gradient are calculated by means of the finite variations of the fitness function and the corresponding chromosomes for the last 3–4 populations (iterations).

For the last three populations, the values of pseudo gradient vector are calculated by means of the finite variations of the fitness function and the corresponding chromosomes between populations 2–1, 3–1, 3–2.

$$\begin{aligned} \text{grad}_{21} f(x, y) &= [\Delta f_{21}/\Delta x_{21}, \Delta f_{21}/\Delta y_{21}]^T \\ \text{grad}_{31} f(x, y) &= [\Delta f_{31}/\Delta x_{31}, \Delta f_{31}/\Delta y_{31}]^T \\ \text{grad}_{32} f(x, y) &= [\Delta f_{32}/\Delta x_{32}, \Delta f_{32}/\Delta y_{32}]^T \end{aligned} \tag{2.4}$$

where $\text{grad}_{21} f(x, y), \text{grad}_{31} f(x, y)$ and $\text{grad}_{32} f(x, y)$ are local gradients of the pseudo gradient grad*f(x,y) of the function f(x,y).

The pseudo gradient $\text{grad}^* f_s(x_s, y_s)$, which is derived from inherited information about the last three generations of chromosomes, can be derived as the average arithmetic value of the elements of local gradients in accordance with expressions (2.4).

The proposal method is a combination between genetic algorithms and gradient methods. In this case, the joint application of these two methods is superior to most situations when they are used on their own due to the fact that their individual drawbacks are well inhibited.

Classical gradient methods are usually based on sequential and independent variation of the arguments of the function that is to be optimized. In

this case, the iterations may take a long time. As opposed to this, the random search method is based on similar operations that are carried out simultaneously, and is therefore much faster.

When new generations of chromosomes are derived randomly, no hereditary information is used for the purposeful variation of the new values of the function that is to be optimized.

This variation is obtained by calculating the pseudo gradient for newly generated chromosomes using expression (2.1).

3 Application of the Method

An example is shown below for finding the optimal solution of a function with two arguments f(x,y).

In this case, the function is the fitness function of chromosomes. It is presented graphically in the figure and possesses one local optimum and a global optimum. Following the approach presented above, the definition domain of the function is separated in two sub-domains. Each of these subdomains possesses a centre around which one initial chromosome is located as shown with the point in the figure. Each chromosome is noted with values of arguments x and y.

a_{0_s}	a_{1_s}	a_{2_s}	a_{3_s}	a_{4_s}	b_{0_s}	b_{1_s}	b_{2_s}	b_{3_s}	b_{4_s}

Here $a_{0_s}, a_{1_s}, \ldots, a_{4_s}$ and $b_{0_s}, b_{1_s}, \ldots, b_{4_s}$ are binary symbols 0 and 1, and they form together a binary coded definition for one chromosome from population $s = 0, 1, 2, 3, \ldots$. The junior bits of the elements x and y of the chromosomes are a_{0_s} and b_{0_s}. The binary symbols are "artificial genes" and carry the values of the characteristics of the chromosomes. The mutation has to be done mainly with junior bits.

When the definition domain of the function f(x,y) is disintegrated from the sub-domain that is uniformly distributed over the whole definition area, the approach above has to be followed. The number of chromosomes for the initial population is 49 and after a limited number of populations it is reduced when the chromosomes start moving towards local optimums of the fitness function. The local optimum in this case is slightly over 35, $f(x = 20, y = 22) \approx 35$ and the global optimum is $f(x = 15, y = 18) \approx 40$ (see fig. 1).

There is one sub-domain in the figure that is denoted by a square, in whose centre the chromosome H (x=8 y=12) is located. The artificial chromosome is presented by the binary values of x and y, and it is entered in the computer program as a structure of x, y:

0	0	0	1	0	0	0	1	1	0

← x →← y →

$$0.2^0 + 0.2^1 + 0.2^2 + 1.2^3 + 0.2^4 = 8$$
$$0.2^0 + 0.2^1 + 1.2^2 + 1.2^3 + 0.2^4 = 12$$

The fitness function of this chromosome is 23, f (8, 12) = 23.

For the initial populations of this chromosome and its "mutants" presented in the figure by small circles, the crossover with chromosomes and its "mutants" can form the neighborhood sub-domains. Once iteration has been executed by means of the crossover and mutation operators, all fitness functions are calculated for every active chromosome in associated sub-domain. The pseudo gradient of the optimization function is then calculated for each chromosome of the current population. After the generation of the first 3–4

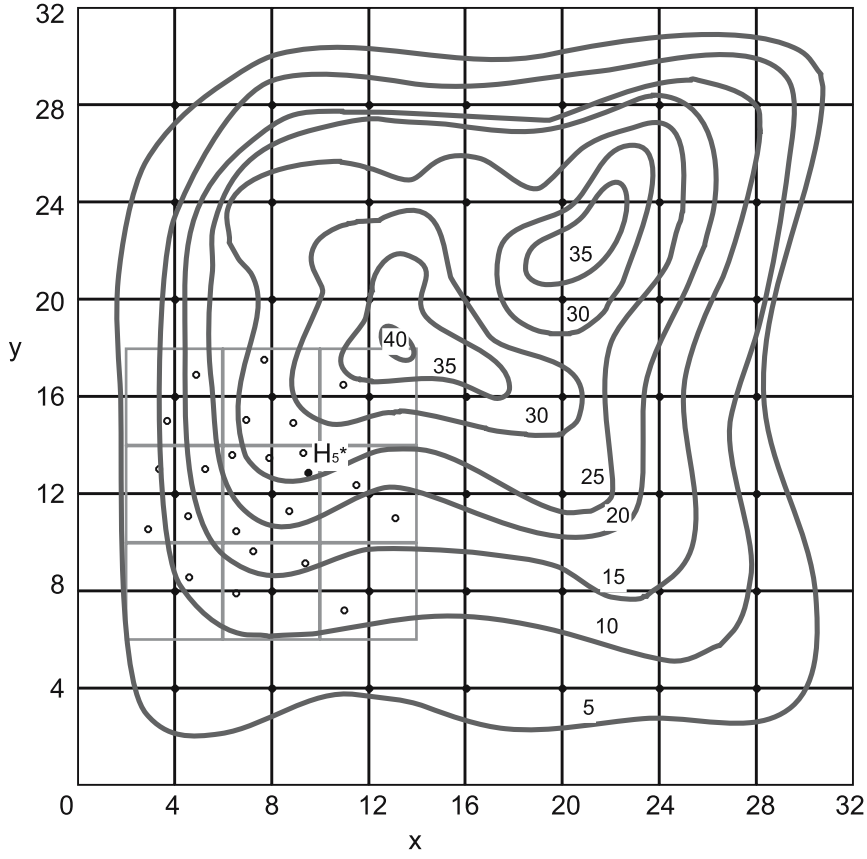


Fig. 1. Iteration procedure for finding extremum of function by means genetic algorithm and pseudo gradient

populations, a data base is created for calculating pseudo gradients with sufficient reliability. They generate the next population of chromosomes. From this example, it can be seen that after the fitness functions for the new population have been compared, a significant number of chromosomes are will not appear to be perspective and they will removed. These are the chromosomes from the peripheral subdomains, as it is shown in the figure.

The chromosomes that are removed are replaced with new chromosomes by means of crossover and mutation operators. This increases the concentration of chromosomes in sub-domains where the values of the fitness function are meaningful. Thus, the genetic algorithm becomes faster due to the higher concentration of chromosomes in the sub-domains of the global optimum.

4 Numerical Example

Lets the object which is investigated to be in the random state, expressed through artificial chromosome $\mathrm{H}_1(\mathrm{x}_1 = 8; \mathrm{y}_1 = 12)$ with fitness function $\mathrm{f}(\mathrm{H}_1) = 23$. During crossover and mutation probabilities and selection of genetic algorithms, the object passes in to state $2(\mathrm{H}_2(\mathrm{x}_2 = 9, \mathrm{y}_2 = 11.2);\ \mathrm{f}(\mathrm{H}_2) = 22)$, after that in to state $3(\mathrm{H}_3(\mathrm{x}_3 = 8, \mathrm{y}_3 = 13.5), \mathrm{f}(\mathrm{H}_3) = 26.5)$, state 4 $(\mathrm{H}_4(\mathrm{x}_4 = 7; \mathrm{y}_4 = 9), \mathrm{f}(\mathrm{H}_4) = 16)$ and e.g. After selection operator has been implemented it has being fallen away state $\mathrm{H}_4(\mathrm{f}(\mathrm{H}_4) = \min)$. The presented state $\mathrm{H}_3\mathrm{f}(\mathrm{H}_3)) = 26,5$ is the most suitable, but there isn't guarantee to increase the value of fitness function in to next implementation of crossover and mutation probabilities.

When it is executed random selection of the basic genetic algorithm is enough the state 5 $(\mathrm{H}_5(\mathrm{x}_5 = 6.5, \mathrm{y}_5 = 13.6; \mathrm{f}(\mathrm{H}_5) = 22.5)$, but this state is less suitable from H_3. It is considered that the next population is derived from the above state, during crossover and mutation probabilities. The basic variable state of the object will evaluate towards local and after that global extreme, but the achievement of this is hardly and long.

By means of calculation of "pseudo gradient" on the basic of initial states $\mathrm{H}_1, \mathrm{H}_2, \mathrm{H}_3, \mathrm{H}_4$ is achieved purposeful choice of the state $\mathrm{H}^*{}_5$, which guaranties higher fitness of the state H5. The parameters of "pseudo grad" function are calculated accordance variables of coordinates and the corresponding fitness function.

For the transition $\mathrm{H}_1 \rightarrow \mathrm{H}_2$:

$$\Delta \mathrm{f}_{21} = |22 - 23| = 1 \tag{4.1}$$

$$\Delta \mathrm{x}_{21} = |9 - 8| = 1$$

$$\Delta \mathrm{y}_{21} = |11,2 - 12| = 0,8$$

$$\mathrm{grad}_{21}\mathrm{f}(\mathrm{x}, \mathrm{y}) = [1/1 = 1 \quad 1/0,8 = 1,25]^{\mathrm{T}}$$

For the transition $H_2 \rightarrow H_3$

$$\Delta f_{32} = |26,5 - 22| = 4,5 \tag{4.2}$$

$$\Delta x_{32} = |8 - 9| = 1$$

$$\Delta y_{32} = |13,5 - 11,2| = 2,3$$

$$\mathrm{grad}_{32}\mathrm{f(x,y)} = [4,5/1 = 4,5 \quad 4,5/2,3 = 1,95]^{\mathrm{T}}$$

For the transition $H_3 \rightarrow H_4$

$$\Delta f_{43} = |16 - 26,5| = 10,5 \tag{4.3}$$

$$\Delta \mathrm{x}_{43} = |7 - 8| = 1$$

$$\Delta \mathrm{y}_{43} = |9 - 13,5| = 4,5$$

$$\mathrm{grad}_{43}\mathrm{f(x,y)} = [10,5/1 = 10,5 \quad 10,5/4,5 = 2,33]^{\mathrm{T}}$$

The "pseudo gradient" of fitness function is calculated as average value of the values of local grad functions $\mathrm{grad}_{21}\mathrm{f(x,y)}, \mathrm{grad}_{32}\mathrm{f(x,y)}, \mathrm{grad}_{43}\mathrm{f(x,y)}$:

$$\mathrm{grad}^{*}\mathrm{f(x,y)} = [(1 + 4,5 + 10,5)/3(1,25 + 1,95 + 2,33)/3]^{\mathrm{T}} = \tag{4.4}$$

$$= [5,33 \quad 1,84]^{\mathrm{T}}$$

The new state of the object $\mathrm{H^{*}}_5$ will receive through purposeful variable of the coordinates according expression (2.1):

$$\mathrm{X}^{*}_5 = \mathrm{x}_1 + 0,5.5,3 = 8 + 0,5.5,33 = 10,65 \tag{4.5}$$
$$\mathrm{Y}^{*}_5 = \mathrm{y}_1 + 0,5.1,84 = 12 + 0,5.1,84 = 12,92$$

where variation step of the chromosomes $\mathrm{K_s} = 0,5$.
The fitness function for object $\mathrm{f^{*}}_5(\mathrm{H_5}^{*}) = 26$

5 Conclusion

Combining of the methods of GA and pseudo gradient decreases the number of experiments, as a result large number of arguments, which are changed at the same time. When it is used "pseudo gradient" is achieved directly looking for the extreme. The numerical example follows that after three casual iterations have been implemented of the basic genetic algorithm is achieved absolutely logical variation of the object towards local extreme. In this case not only the direction and also the speed of the coordinates of the states are modified purposeful accordance of the coordinates for the looking extreme.

References

1. Gegov E. (2003). Methods and Applications of Computer Intelligence in the Information Technologies and Control of Systems. Sv.Ivan Rilski Press.
2. Goldberg, D.E (1989) Genetic Algorithms in Search. Optimization and Machine Learning, Addision-Wesley, Reading, MA.
3. Holland, J.H. (1975) Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor..
4. Mitchel, M. (1996). An Introduction to Genetic Algorithms. MIT Press, Cambridge, MA
5. Zbigiew Michalewcz (1992) Genetic Algorithms + Data Structures = Evolution Programs. Extended Edition, Springer-Verlag.

Optimization by Island-Structured Decentralized Particle Swarms

Juan F. Romero and Carlos Cotta

Dept. of Lenguajes y Ciencias de la Computación, University of Málaga
ETSI Informática (office 3.2.49), Campus de Teatinos
E-29071 - Málaga - SPAIN
ccottap@lcc.uma.es

Abstract. This work explores the utilization of the island-model within the context of Particle Swarm Optimization (PSO). The well-known notions of decentralized evolutionary algorithms are extended to this context, resulting in the definition of a multi-swarm. The influence that different parameterizations of the model, namely, the number of swarms, their interconnection topology, the policy for selecting particles to migrate, and the policy for accepting incoming particles is studied. Four continuous optimization problems are used for this purpose. The experimental results indicate that a moderate number of swarms arranged in a fully-connected topology provide the best results.

1 Introduction

Particle Swarm Optimization (PSO) is one of the most impetuously emerging paradigms of evolutionary computation (EC). Initially proposed by Kennedy and Eberhart [1], PSO is inspired in the processes that give rise to the creation of shoals of fishes, flock of birds, or swarms of insects [2, 3, 4]. A PSO algorithm is based on using a population (or *swarm*) of particles that traverse the search space. These particles have a local memory of the best point of the search space they have visited, and have a tendency to return to it. Similarly, a global memory of the best solution visited by any particle of the swarm –with the subsequent attraction for all particles– is used. As a consequence, social behavior emerges; the swarm starts exploring the search space, and whenever a particle discovers a new incumbent, it collectively moves to the region around it.

This appealing scheme has revealed itself as a competitive approach for optimization, owing to its simplicity and relatively low computational cost. Several extensions have been proposed in order to further improve its performance. Examples of such extensions are the consideration of attraction/repulsion forces among particles [5], and the hybridization with local

J.F. Romero and C. Cotta: *Optimization by Island-Structured Decentralized Particle Swarms*, Advances in Soft Computing **2**, 25–33 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

search or other EC mechanisms [6, 7]. In this work, we propose the use of multiple cooperative swarms, each of which tries to solve the original problem in partial isolation following the philosophy of island-model evolutionary algorithms [8]. The novelty of this approach relies in attaining swarm cooperation via migration of memory states, i.e., regularly synchronizing the memory states of some swarms. The parameters and strategies governing this cooperation will be analyzed below. As it will be shown, this multi-swarm approach can lead to improved optimization performance.

2 Background

A PSO algorithm can be regarded as a hybrid between population-based search and trajectory-based search. As in the latter, paths across the search space are constructed by individual agents (the particles); as in the former, these agents interact. This interaction is not direct as in genetic algorithms though. On the contrary, it takes place indirectly through the use of a shared memory. More precisely, let $\mathcal{S} \triangleq \mathcal{D}^n$ be the search space, where $\mathcal{D}$ is some domain. As it is typically the case, we will assume $\mathcal{D} \equiv \mathbb{R}$, i.e., the search is conducted in a $n-$dimensional continuous space. Let $f : \mathcal{S} \longrightarrow \mathbb{R}$ be the objective function (to be minimized, without loss of generality). A swarm is a collection of μ particles $p_1 \cdots p_\mu$. Each particle p_i is specified by a pair $(\mathbf{x}_i, \mathbf{v}_i)$, where $\mathbf{x}_i \in \mathbb{R}^n$ is the position of the particle (a point in the search space), and $\mathbf{v}_i \in \mathbb{R}^n$ is the velocity of the particle. This latter value indicates the trajectory that the particle will follow. Initially, both positions and velocities are selected at random. Subsequently, the particles evolve using the following equations:

$$\mathbf{v}_i(t+1) = \omega \cdot \mathbf{v}_i(t) + \alpha_1 \cdot \left[\mathbf{x}_i^l(t) - \mathbf{x}_i(t)\right] + \alpha_2 \cdot \left[\mathbf{x}^g(t) - \mathbf{x}_i(t)\right] \quad (1)$$

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t) \quad (2)$$

where

$$\mathbf{x}_i^l(t) = \operatorname{argmin}\left\{f\left(\mathbf{x}_i(\tau)\right) \mid 0 \leqslant \tau \leqslant t\right\} \quad (3)$$

is the local memory of the best point visited by particle p_i, and

$$\mathbf{x}^g(t) = \operatorname{argmin}\left\{f\left(\mathbf{x}_i^l(t)\right) \mid 0 \leqslant i \leqslant \mu\right\} \quad (4)$$

is the global memory of the best solution found by the swarm. Thus, each particle has a local tendency to return to its best known location. Simultaneously, all particles are attracted by the best solution known by the swarm (hence, the emergence of collective behavior). The coefficients α_1 and α_2 are typically selected at random in each step to introduce some variability in the velocity behavior. As to ω, it is a parameter termed *inertia weight*, used to control the balance between exploration (high ω) and exploitation (low ω).

3 An Island Model of PSO

As it is the case in most populational metaheuristics, convergence to a suboptimal solution can be problematic in PSO, i.e., the swarm may find a local optimum with a large basin of attraction, and get trapped in it. In the context of classical EC techniques such as genetic algorithms, this problem is treated in several ways: on one hand, once it is detected it can be remedied by triggering high mutation rates, or restarting the population; on the other hand, it can be prevented by using for example decentralized populations [9], i.e., using multiple interconnected subpopulations that evolve in partial isolation (as opposed to a single *panmictic* population). We will precisely translate this latter approach to PSO.

A multi-swarm is defined as a triplet $\langle \mathcal{P}, \mathcal{T}, \mathcal{C} \rangle$, where $\mathcal{P} = \{P_1, \cdots, P_m\}$ is a collection of m particle swarms, $\mathcal{T} : \mathcal{P} \rightarrow 2^{\mathcal{P}}$ is the topology of the multi-swarm (that is, how the different swarms are interconnected), and $\mathcal{C}$ is a communication policy. The latter is a four-tuple $\langle \nu, \xi, \sigma, \psi \rangle$, where $\nu \in \mathbb{N}$ indicates the communication frequency, $\xi \in \mathbb{N}$ indicates the number of particles that will migrate from one swarm to another, σ is a function for selecting the migrating particles, and ψ is a replacement function used to decide what to do with incoming particles. We have considered two alternatives for σ, that we call `RANDOM` and `BEST`. The former refers to randomly selecting a particle for migrating; the latter corresponds to migrating the particle whose local memory contains the current incumbent of the swarm. As to ψ, two options are possible: `ALWAYS` (the worst particle of the swarm is replaced by the incoming particle), and `BETTER` (the replacement of the worst particle is done only if it is worse than the incoming one). The functioning of the multi-swarm would thus be as follows:

1. **repeat**
   a) **parfor** $i \in [1..m]$ **do**
      i. Iterate swarm P_i for ν steps.
      ii. **parfor** $P_j \in \mathcal{T}(P_i)$ **do**
         A. **send** particles in $\sigma(P_i)$ **to** P_j.
      iii. **foreach** P_j such that $P_i \in \mathcal{T}(P_j)$ **do**
         A. **receive** particles s_j **from** P_j.
         B. **insert** particles s_j **in** P_i **using** ψ.
   until TerminationCriterion is met.

We have considered two choices of $\mathcal{T}$: `RING`, and `COMPLETE`. The former is $\mathcal{T}(P_i) = \left\{P_{(iMODm)+1}\right\}$, i.e., swarms are arranged in a unidirectional ring; the latter is $\mathcal{T}(P_i) = \mathcal{P}$, i.e., each swarm broadcasts its particles to the remaining swarms. This complete topology ensures a global synchronization of the best position identified in the multi-swarm. The dissemination of this global incumbent is slower in the ring model, since it has to jump gradually from swarm to swarm. The two topologies are illustrated in Figure 1.

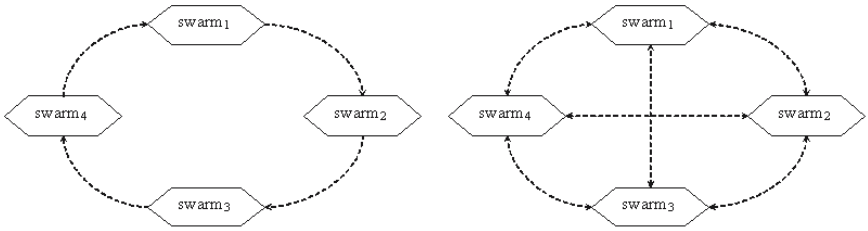


Fig. 1. Swarms are arranged in a unidirectional ring (`RING` topology – left) or in a clique (`COMPLETE` topology – right).

4 Experimental Results

The experiments have been done with four continuous functions: Sphere, Rastrigin, Rosenbrock and Griewank. Table 1 shows the mathematical expressions of these functions. The model has been run with 400 particles organized in 1, 2, 4, 8 and 16 swarms, for a total number of 500,000 evaluations of the objective function. In all cases, we have considered the values $\nu = 50$ and $\xi = 1$. Also, particles are always initialized at a corner of the search space, so as to make harder the optimization task. Finally, the inertia weight ω decreases linearly during the run within certain limits. Table 2 shows the particular parameters for each function.

Table 1. Test functions used in the experimentation.

Sphere	$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$	
Rastrigin	$f(\mathbf{x}) = nA + \sum_{i=1}^{n} \left[x_i^2 - A\cos(\theta x_i)\right])$	$A = 10, \theta = 2\pi$
Rosenbrock	$f(\mathbf{x}) = K\sum_{i=1}^{n-1} \left(x_{i+1} - x_i^2\right)^2 + \sum_{i=1}^{n} \left(x_i - 1\right)^2$	$K = 100$
Griewank	$f(\mathbf{x}) = K\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$K = \frac{1}{4000}$

Table 2. Parameters for each test function.

	n	variable range	initial range	max v_i	ω
Sphere	40	$[-100, 100]$	$[50, 100]$	100	$(0.9, 0.3)$
Rastrigin	40	$[-5.12, 5.12]$	$[2.56, 5.12]$	5.12	$(0.9, 0.3)$
Rosenbrock	40	$[-100, 100]$	$[15, 30]$	100	$(0.9, 0.3)$
Griewank	100	$[-600, 600]$	$[300, 600]$	600	$(0.7, 0.4)$

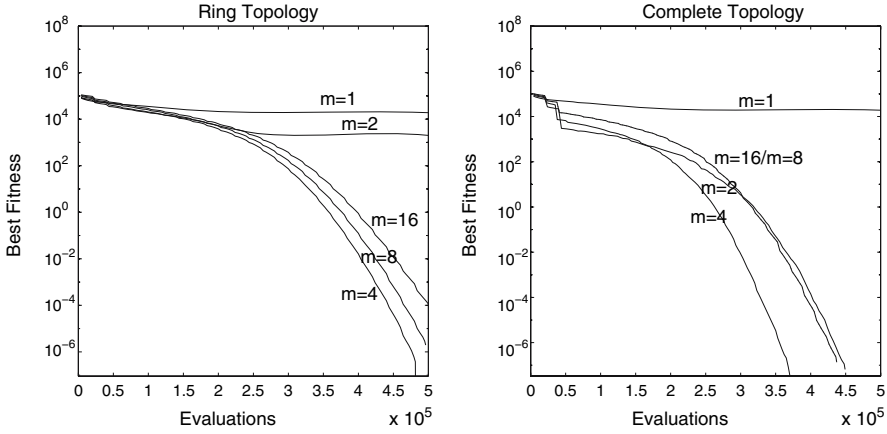


Fig. 2. Evolution of fitness (averaged for 30 runs) for different number of swarms in the Sphere function. The left graph corresponds to the `RING` topology, and the right one to the `COMPLETE` topology. Notice the use of a log-scale in the Y-axis.

The initial tests are devoted to assess the influence of the migrant selection and replacement policies. To this end, 30 runs of the multi-swarm algorithm have been performed for each combination of values of $\mathcal{T}$ (topology), σ (migrant selection), ψ (replacement), and m (number of swarms). A run is considered successful if a solution better than a certain threshold (10^{-6} for Sphere, Rastrigin, and Griewank, and 40 for Rosenbrock) is obtained. The number of successful runs for each problem and configuration is shown in Table 3. As it can be seen, `BEST` migrant selection tends to be better than `RANDOM` selection. The difference between `ALWAYS` replacement and `BETTER` replacement is not so marked. Nevertheless, it seems clear that the best combination is `ALWAYS`+`BEST`, especially in connection with the `COMPLETE` topology (whose superiority over `RING` is very evident).

Subsequent experiments thus concentrate on σ=`BEST`, and ψ=`ALWAYS`. This means that synchronization of swarm incumbents (global or local by diffusion according to the chosen topology) is performed every ν steps. Figures 2 through 5 show the outcome of the experimentation for these problems. Consider firstly the results for the Rastrigin function (Figure 3). As it can be seen, the mono-swarm is incapable of advancing to the optimal solution ($f_{opt} = 0$), and gets stagnated at some suboptimal region. However, the algorithm can progress toward the optimal when the particles are structured in more swarms (at least 2 for the `COMPLETE` topology, and more than 2 in the case of the `RING` topology). Partial isolation thus helps avoiding local optima. Notice that in general, a high number of swarms results in lower convergence, since the different swarms do not coordinate their efforts between synchronization events, and hence diversify their search. Progress is also quicker for the `COMPLETE` topology than for the `RING` topology. In this problem, global synchronization of swarm incumbents makes the search focus in the more promising regions

Table 3. Number of successful runs (out of 30) of the multi-swarm for different configurations.

$\mathcal{T}$	σ	ψ	m	Sphere	Rastrigin	Rosenbrock	Griewank
RING	RANDOM	BETTER	1	5	8	8	0
			2	29	28	10	0
			4	30	30	16	0
			8	20	30	6	0
			16	0	30	1	0
		ALWAYS	1	5	8	8	0
			2	26	27	10	0
			4	30	30	9	0
			8	19	30	5	0
			16	0	30	0	0
	BEST	BETTER	1	5	8	8	0
			2	27	27	12	0
			4	30	30	7	0
			8	30	30	6	0
			16	0	30	3	0
		ALWAYS	1	5	8	8	0
			2	24	24	6	0
			4	30	30	4	0
			8	22	30	9	0
			16	0	30	0	0
COMPLETE	RANDOM	BETTER	1	5	8	8	0
			2	29	30	12	2
			4	30	30	18	7
			8	29	30	5	7
			16	23	30	11	1
		ALWAYS	1	5	8	8	0
			2	30	30	23	19
			4	30	30	2	22
			8	30	30	0	22
			16	29	30	3	0
	BEST	BETTER	1	5	8	8	0
			2	29	30	21	0
			4	30	30	20	0
			8	30	30	16	15
			16	25	30	6	9
		ALWAYS	1	5	8	8	0
			2	30	30	26	0
			4	30	30	23	23
			8	30	30	0	26
			16	30	30	11	13

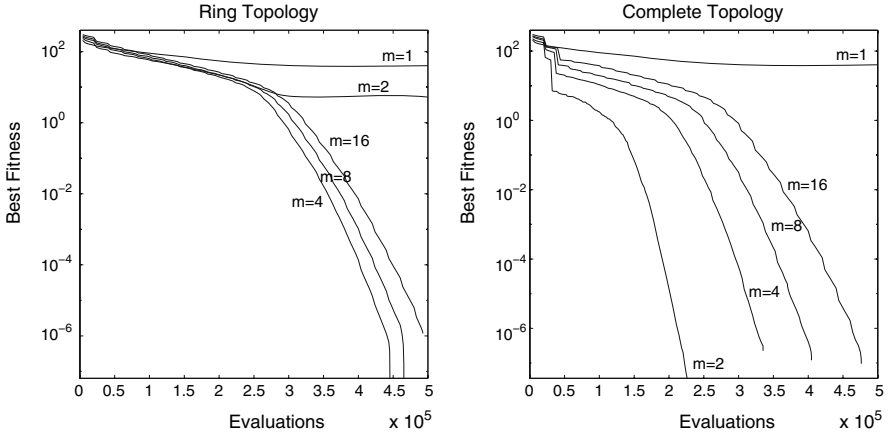


Fig. 3. Evolution of fitness (averaged for 30 runs) for different number of swarms in the Rastrigin function. The left graph corresponds to the `RING` topology, and the right one to the `COMPLETE` topology. Notice the use of a log-scale in the Y-axis.

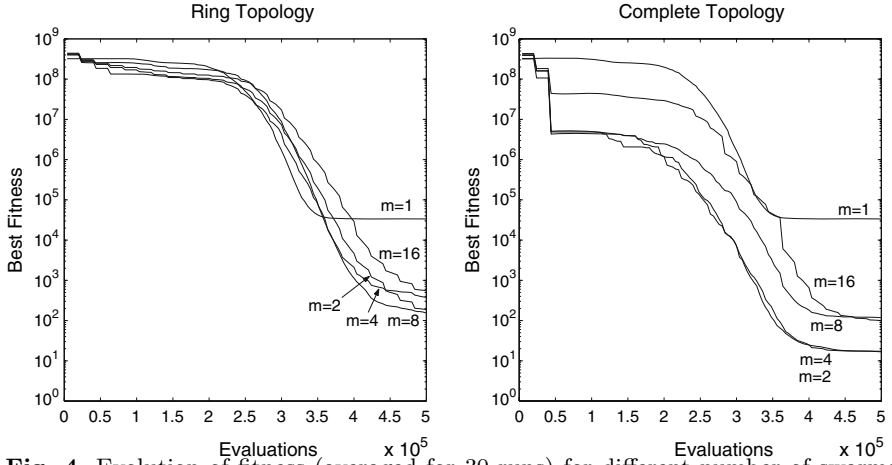


Fig. 4. Evolution of fitness (averaged for 30 runs) for different number of swarms in the Rosenbrock function. The left graph corresponds to the `RING` topology, and the right one to the `COMPLETE` topology. Notice the use of a log-scale in the Y-axis.

at a faster pace than synchronization by diffusion does. The latter is a more exploratory strategy, due to the lower coupling of swarms. The behavior in the Sphere function is essentially the same as described before for the Rastrigin function (cf. Figure 2). This outcome is consistent with the fact that both functions are non-epistatic.

The results are somewhat similar for the Rosenbrock function (Figure 4). Again, the mono-swarm cannot advance too much, and ends up at a low-quality suboptimal solution. The multi-swarms can progress further though. As for the Rastrigin function, the `COMPLETE` topology converges faster (and to better solutions) than the `RING` topology. In this case, the configuration

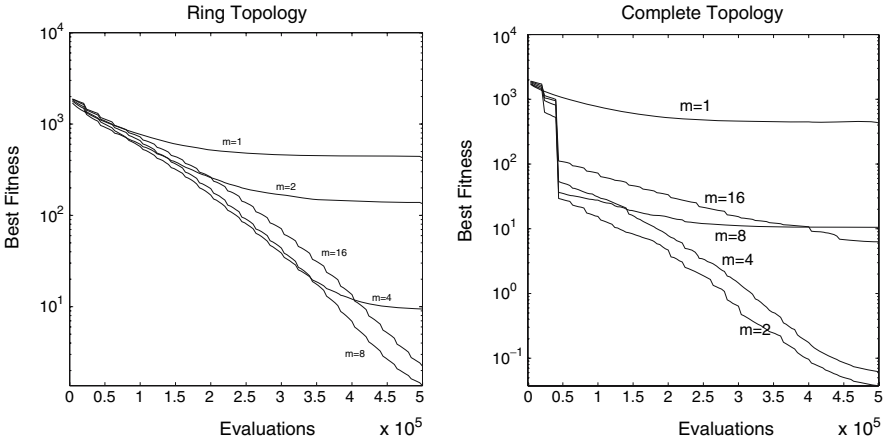


Fig. 5. Evolution of fitness (averaged for 30 runs) for different number of swarms in the Griewank function. The left graph corresponds to the `RING` topology, and the right one to the `COMPLETE` topology. Notice the use of a log-scale in the Y-axis.

with 4 swarms performs at least as good as 2 swarms. This owes to the higher difficulty of the problems, that causes a shift in the ideal balance between exploration and exploitation. In this sense, notice the results for the Griewank function (Figure 5). The gradual diffusion of local incumbents in the `RING` topology requires a higher number of swarms for achieving the same results than an intermediate number of swarms provide in the `COMPLETE` topology. In this latter case, the smallest swarms result in lower performance, probably due to not strong enough exploitation.

5 Conclusions

An island-model of PSO has been proposed and evaluated here. The results obtained have been encouraging since multi-swarms have been capable of outperforming panmictic swarms. Regarding the parameters of the model, the complete topology seems to provide better results than the ring topology. As to the number of swarms, an intermediate value (2–4) seems to be more adequate. This latter setting is obviously connected to the precise swarm sizing chosen; in this sense, more experiments with diverse swarm sizes will be done in the future to ascertain the influence of this parameter.

In addition to this latter issue, future work will also consider more complex models in which local search is embedded in individual swarms –i.e., *memetic swarms*, cf. memetic algorithms [10]– as well as the deployment of the algorithm on distributed systems. The algorithm is inherently amenable for parallel computing, and hence can greatly benefit from the use of these techniques.

Acknowledgement

The second author is partially supported by Spanish MCyT and FEDER under contract TIC2002-04498-C05-02.

References

1. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the Fourth IEEE International Conference on Neural Networks, Piscataway NJ, IEEE Press (1995) 1942–1948
2. Kennedy, J., Eberhart, R.: The particle swarm: Social adaptation in information processing systems. In Corne, D., Dorigo, M., Glover, F., eds.: New Ideas in Optimization. McGraw-Hill IK (1999) 379–387
3. Eberhart, R., Shi, Y.: Particle swarm optimization: development, applications and resources. In: Proceedings of the 2001 Congress on Evolutionary Computation, Piscataway NJ, IEEE Press (2001) 81–86
4. Kennedy, J., Eberhart, R.: Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco CA (2001)
5. Riget, J., Vesterstrøm, J.: A diversity-guided particle swarm optimizer – the ARPSO. Technical Report 2002-02, EVALife Project Group, Department of Computer Science, Universit of Aahrus (2002)
6. Krink, T., Løvbjerg, M.: The lifecycle model: Combining particle swarm optimisation, genetic algorithms and hillclimbers. In Merelo, J., Adamidis, P., Beyer, H.G., Fernández-Villacañas, J.L., Schwefel, H.P., eds.: Parallel Problem Solving From Nature VII. Volume 2439 of Lecture Notes in Computer Science., Berlin, Springer-Verlag (2002) 621–630
7. Angeline, P.: Using selection to improve particle swarm optimization. In: Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, IEEE Press (1998) 84–89
8. Tanese, R.: Distributed genetic algorithms. In Schaffer, J., ed.: Proceedings of the Third International Conference on Genetic Algorithms, San Mateo, CA, Morgan Kaufmann Publishers (1989) 434–439
9. Alba, E., Tomassini, M.: Parallelism and evolutionary algorithms. IEEE Transactions on Evolutionary Computation **6** (2002) 443–462
10. Moscato, P., Cotta, C.: A gentle introduction to memetic algorithms. In Glover, F., Kochenberger, G., eds.: Handbook of Metaheuristics. Kluwer Academic Publishers, Boston MA (2003) 105–144

Directed Mutation by Means of the Skew-Normal Distribution

Stefan Berlik

Dortmund University, Department of Computer Science I, 44221 Dortmund, Germany.
stefan.berlik@uni-dortmund.de

Abstract. Directed mutation can improve the efficiency of processing many optimization problems. The directed mutation operator presented in this paper is based on the Skew-Normal Distribution. It is the first one that is not defined by case differentiation. Its expectation as well as its variance are convergent for all degrees of skewness and random number generation is simple and fast. An appropriate recombination scheme is given, and experimental results using this directed mutation are presented.

Keywords: Skew-normal distribution, evolutionary algorithm, directed mutation, mutation operator.

1 Introduction

The main idea of the directed mutation is to mutate with random numbers that lie preferably on the side the optimum is expected. This implies expected values unequal to zero and contrasts the classical mutation operators. Using this method the optimization strategy can adopt the most promising direction over the generations. To do so a customizable skewed distribution is needed.

2 Directed Mutation

To be able to do directed mutation one needs to generate skew distributed random numbers. Thus one has to introduce expected values unequal to zero. Of course this means that the mutation operator is not compliant to standard evolution strategy any longer postulating an expected value of zero, so mutating around the origin. On the other hand it has to be ensured that the expected value is convergent, thereby forcing the mutation operator to continue mutating near by the origin. It should be pointed out that convergence

S. Berlik: *Directed Mutation by Means of the Skew-Normal Distribution*, Advances in Soft Computing **2**, 35–50 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

in the symmetrical case only is not enough. Also for all skewed cases convergence is needed. This demand is violated by mutation operators based on the Ξ-distribution proposed by Hildebrand [5]. Diverging expected values caused by increasing skewness parameters here can lead to wide jumps.

Of even greater interest is the variance. It can be seen as a measure for the mutation strength and is a strategy parameter in most evolution strategies. Because of this it should not be modified by the skewness parameter. In the ideal case the variance is a constant, independent of the skewness parameter. At least convergence is necessary and a small spread between minimal and maximal value is desired to limit the impact of the skewness on the mutation strength. Again, the Ξ-distribution violates this demand.

3 Skew-Normal Distribution

The class of distributions that is used to build the directed mutation operator is closely related to the skew-normal (SN, hereafter) distribution introduced by Azzalini [1]. Its density function is defined as

$$f(z;\lambda) = 2\phi(z)\Phi(\lambda z) \qquad z \in \mathbb{R}\,, \tag{1}$$

where ϕ and Φ represents the probability density function (p.d.f.) and the cummulative distribution function (c.d.f.) of the standard Normal density, respectively. λ is a real parameter that controls the skewness, where positive (negative) values indicate positive (negative) skewness. In the case $\lambda = 0$ the SN density gets back to the Normal density (see Fig. 1). With $Z \sim SN(\lambda)$ one denotes a random variable (r.v.) that has density (1). The SN class enjoys remarkable properties in terms of mathematical tractability. Some results found by Azzalini [1] will be given in the following.

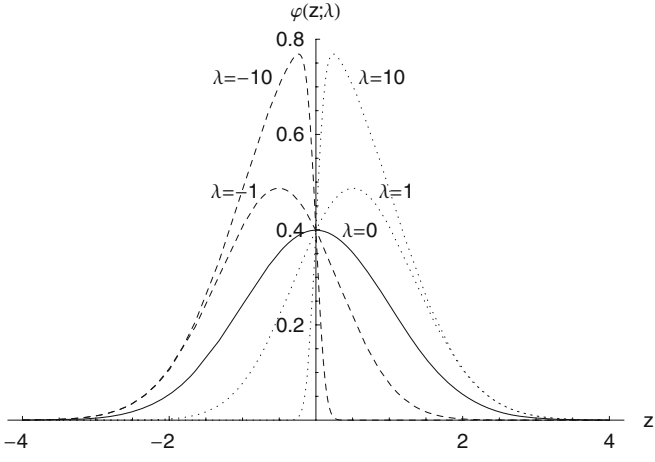


Fig. 1. The density functions SN(−10), SN(−1), SN(0), SN(1), and SN(10)

3.1 Moments

The first four Moments are given as

$$E(Z) = b\delta \,, \tag{2}$$

$$V(Z) = 1 - (b\delta)^2 \tag{3}$$

$$\gamma_1(Z) = \frac{1}{2}(4-\pi)\mathrm{sign}(\lambda)\left(\frac{\big(E(Z)\big)^2}{V(Z)}\right)^{3/2} \tag{4}$$

$$\gamma_2(Z) = 72(\pi-3)\left(\frac{\big(E(Z)\big)^2}{V(Z)}\right)^{2} \tag{5}$$

where

$$b = \sqrt{\frac{2}{\pi}} \quad \text{and} \quad \delta = \frac{\lambda}{\sqrt{1+\lambda^2}} \,.$$

$\gamma_1(Z)$ and $\gamma_2(Z)$ denote the third and fourth standardized cumulants. As desired, both expectation and variance converge. The limits are

$$\lim_{\lambda\to\pm\infty}\big(E(Z)\big) = \mathrm{sign}(\lambda)\sqrt{\frac{2}{\pi}} \tag{6}$$

$$\lim_{\lambda\to\pm\infty}\big(V(Z)\big) = 1 - \frac{2}{\pi} \,. \tag{7}$$

Their graphs are depicted in Figs. 2 resp. 3. One can see that the variance is convergent, but still spreads about 0.64. To make the variance constant, a linear transformation has to be applied to the SN distributed r.v. leading to the standardized SN distribution.

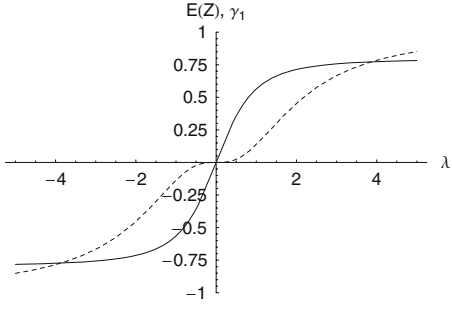


Fig. 2. The expectation (solid) and skewness (dashed) of a SN(λ) distributed random variable

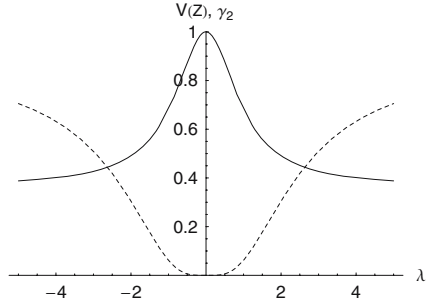


Fig. 3. The variance (solid) and kurtosis (dashed) of a SN(λ) distributed random variable

3.2 Random Variate Generation

A r.v. Z with density (1) can be generated by an acceptance-rejection method. Therefore sample X and Y from Φ' and ϕ, respectively until the inequality $X < \lambda Y$ is satisfied. Then put $Z = Y$. On average, two pairs (X, Y) are necessary to generate Z.

4 Standardized Skew-Normal Distribution

The transformation that has to be applied obviously depends of the skewness parameter λ. Taking into account that $V(a+bZ) = b^2V(Z)$, $s^2V(Z) \stackrel{!}{=} 1$ and (3) lead to

$$s = \frac{1}{\sqrt{V(Z)}} = \frac{1}{\sqrt{1-(b\delta)^2}} = \sqrt{\frac{\pi(1+\lambda^2)}{\pi+(\pi-2)\lambda^2}}\ . \tag{8}$$

4.1 Probability Density Function

If F is the distribution function of a random variable Z, then $aZ+b$ has distribution function $F\big((z-b)/a\big)$, provided $a > 0$. If the corresponding densities exist, they are $f(z)$ and $\frac{1}{a}f\big((z-b)/a\big)$. Thus the density of the Standardized Skew-Normal distribution (SSN) takes the following form

$$f(z;\lambda) = \frac{2}{s}\phi\left(\frac{z}{s}\right)\Phi\left(\frac{\lambda z}{s}\right) \qquad z \in \mathbb{R}\ . \tag{9}$$

Due to the standardization the densities are widened and flattened, see Fig. 4.

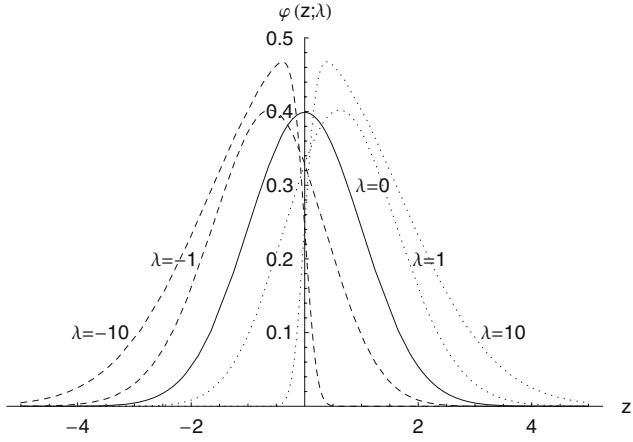


Fig. 4. The density functions SSN(−10), SSN(−1), SSN(0), SSN(1), and SSN(10)

4.2 Moments

With (8), $E(a + bZ) = a + bE(Z)$, and $V(a + bZ) = b^2V(Z)$ one can deduce the first four moments of SSN distribution from the moments of the SN distribution (2)-(5):

$$E(Z) = sb\delta \,, \tag{10}$$

$$V(Z) = 1 \,, \tag{11}$$

$$\gamma_1(Z) = \frac{1}{2}(4-\pi)\big(E(Z)\big)^3 \,, \tag{12}$$

$$\gamma_2(Z) = 2(\pi-3)\big(E(Z)\big)^4 \tag{13}$$

where

$$b = \sqrt{\frac{2}{\pi}} \quad \text{and} \quad \delta = \frac{\lambda}{\sqrt{1+\lambda^2}} \,.$$

The limits of the SNN class are

$$\lim_{\lambda\to\pm\infty}\big(E(Z)\big) = \operatorname{sign}(\lambda)\sqrt{\frac{2}{\pi-2}} \tag{14}$$

$$\lim_{\lambda\to\pm\infty}\big(V(Z)\big) = 1 \,. \tag{15}$$

Graphs of the first four moments of the SSN are shown in Figs. 5 resp. 6.

5 Building the Directed Mutation Operator

The SSN distribution is a sound basis to construct a directed mutation operator. Two things are sill missing. The self adaptation of the skewness parameters has to be designed and an appropriate recombination scheme has to be build.

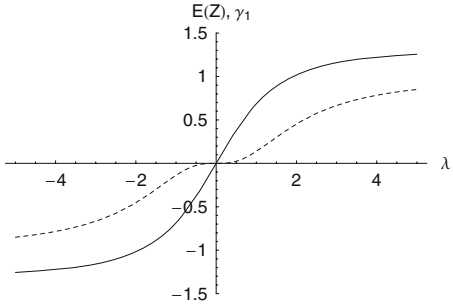


Fig. 5. The expectation (solid) and skewness (dashed) of a SSN(λ) distributed random variable

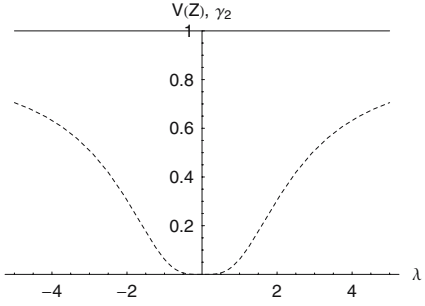


Fig. 6. The variance (solid) and kurtosis (dashed) of a SSN(λ) distributed random variable

5.1 Operators for the Mutation of the Strategy Parameters

Mutation of the mutation strengths is done as usual with the log-normal operator [6]. The actual mutation strength is multiplied with random number ξ that is obtained by an exponential transformation of a normally distributed random number

$$\tilde{\sigma} := \xi\sigma, \quad \xi := e^{\tau N(0,1)} \; . \tag{16}$$

The variable τ is an exogenous strategy parameter and called learning parameter. To prevent deadlocks from overfitted skewness damping has to be introduced into the mutation operator for the skewness parameters. This can be done via a damping factor d which leads to the damped log-normal operator

$$\tilde{\lambda} := d\xi\lambda, \quad \xi := e^{N(0,1)}, \quad 0 \leq d \leq 1 \; . \tag{17}$$

The variable d is also an exogenous strategy parameter. The right choice of d depends on the individual optimization problem. Experimental results show that $d \approx 0.95$ is a good choice for most problems.

5.2 Coupled Recombination

To use directed mutation one also has to consider the recombination operator (cf. e.g. [4]. The statement by Bäck [2] that independent recombination on object variables and strategy parameters is justified by experimental observations could not be approved for the use of directed mutations. When doing so all directed mutation variants yield significantly worse results compared to the classical variants.

The reason for this could be seen in the higher grade of localization that arises from the togetherness of object variable, mutation strength, and skewness parameter. In fact they have to be treated as a unit. Thus coupled recombination assures that the strategy parameters are chosen from the same parent where the object variable at hand is taken from when recombining parents. In the case of intermediate recombination the same weight ought to be used for the different components.

6 Experimental Studies

6.1 Analyzed Mutation Operators

Six mutation operators have been analyzed. Three of them realize directed mutation:

- *Standardized Skew-Normal mutation*
  The operator presented in this paper has been used with coupled recombination. A damping factor of 0.95 has been used when mutating the skewness parameters.
- *Simple directed mutation*
  This operator is defined in sections and has convergent expectation and variance [3]. Also here coupled recombination and a damping factor of 0.95 have been used.
- *Asymmetrical mutation*
  This was the first operator that introduced directed mutation to the field of evolutionary algorithms. Like the simple directed mutation it is defined in sections but has divergent expectation and variance. The operator has been used as proposed by Hildebrand [5], i.e. with classical recombination and skewness parameters being mutated the same way the mutation strengths are.

The other three operators are the classical ones:

- *Simple mutation with one mutation strength*
  Only one mutation strength is used that is applied in turn to every object variable.
- *Simple mutation with n mutation strengths*
  n mutation strengths are used, a separate mutation strength for each object variable.

- *Correlated mutation*
  Besides the control of the n mutation strengths, correlated mutation allows to rotate the coordinate pane arbitrarily by supporting a full covariance matrix.

6.2 Test Functions

Seven well-known test functions [7] have been examined, see Table 1. All functions are high dimensional problems and have here been treated with 30 dimensions. Except for function f_7 they all are unimodal. This is due to the fact that the local behavior of the mutation operator should be studied. The number of local mimima in f_7 increases exponentially with the function dimension. All but function f_6 are continuous functions.

Table 1. The test functions used in the experimental studies. n is the dimension of the function, f_{min} its minimum value, and $S \subseteq \mathbb{R}^n$ the object variable space

Test function	n	S	f_{min}
$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	$[-100, 100]^n$	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$[-10, 10]^n$	0
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	30	$[-100, 100]^n$	0
$f_4(x) = \max\{\lvert x_i \rvert, 1 \leq i \leq n\}$	30	$[-100, 100]^n$	0
$f_5(x) = \sum_{i=1}^{n-1} \left[100 \left(x_{i+1} - x_i^2 \right)^2 + (x_i - 1)^2 \right]$	30	$[-30, 30]^n$	0
$f_6(x) = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$	30	$[-100, 100]^n$	0
$f_7(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right) - \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + e$	30	$[-32, 32]^n$	0

6.3 Experimental Setup

The experiments have been done using a (30,200)-ES with self-adaptive standard deviations. The same initial populations were used. All object variables have been set by random. The mutation strengths haven been set to 0.1, the skewness and correlation parameters have initial values of 0. All experiments have been carried out 50 times.

6.4 Experimental Results

According to the following Figs. 7–20 the SSN mutation outperforms the other mutation operators. Very significantly this is the case for the functions f_4 and f_5. One can also see that the simple directed mutation performs nearly as good, even slightly better for the functions f_1,f_2, and f_7. SSN and the simple directed mutation form the group of best performing operators. The third directed mutation operator, i.e. the asymmetrical mutation clearly performs worse. Compared to the classical variants it yields no gain.

f_1 Sphere Model

$$f_1(x) = \sum_{i=1}^{30} x_i^2$$

$$-100 \leq x_i \leq 100, \quad \min(f_1) = f_1(0, \ldots, 0) = 0$$

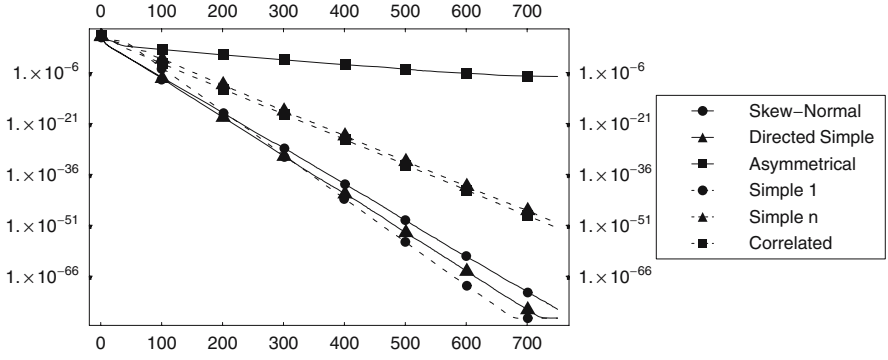


Fig. 7. Averages of the results

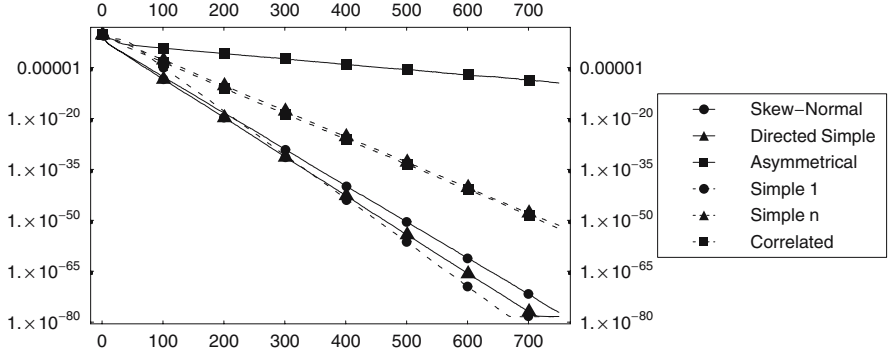


Fig. 8. Medians of the results

f_2 Schwefel's Problem 2.22

$$f_2(x) = \sum_{i=1}^{30} |x_i| + \prod_{i=1}^{30} |x_i|$$

$$-10 \leq x_i \leq 10, \quad \min(f_2) = f_2(0, \ldots, 0) = 0$$

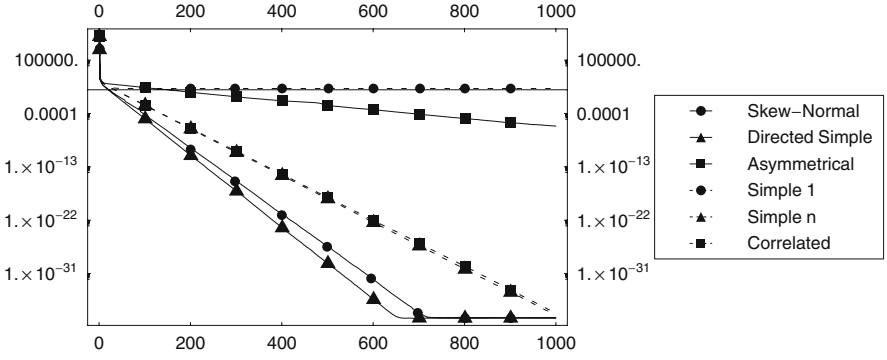


Fig. 9. Averages of the results

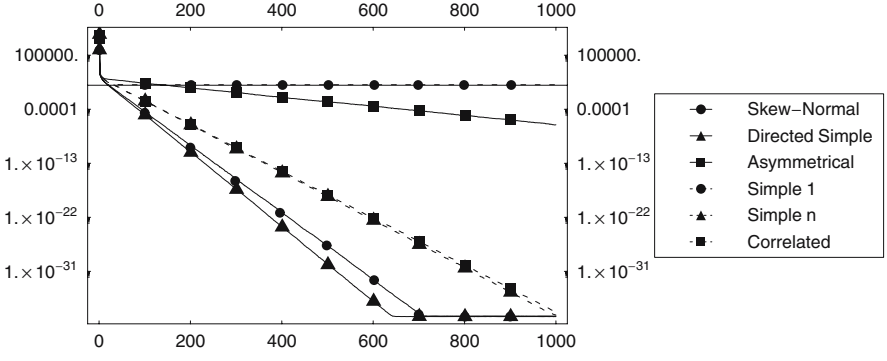


Fig. 10. Medians of the results

f_3 Schwefel's Problem 1.2

$$f_3(x) = \sum_{i=1}^{30} \left(\sum_{j=1}^{i} x_j \right)^2$$

$$-100 \leq x_i \leq 100, \quad \min(f_3) = f_3(0, \ldots, 0) = 0$$

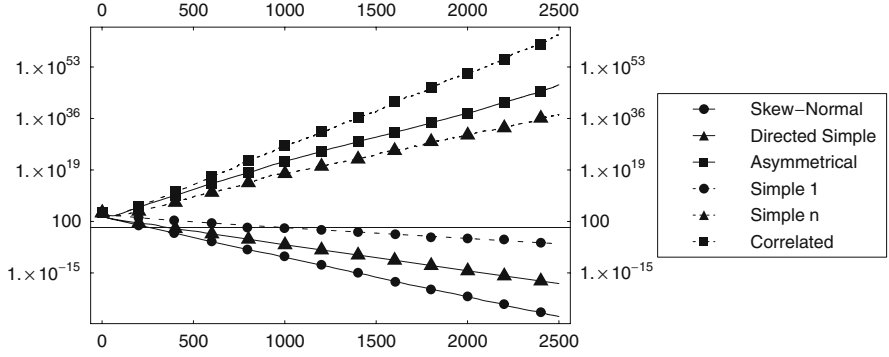


Fig. 11. Averages of the results

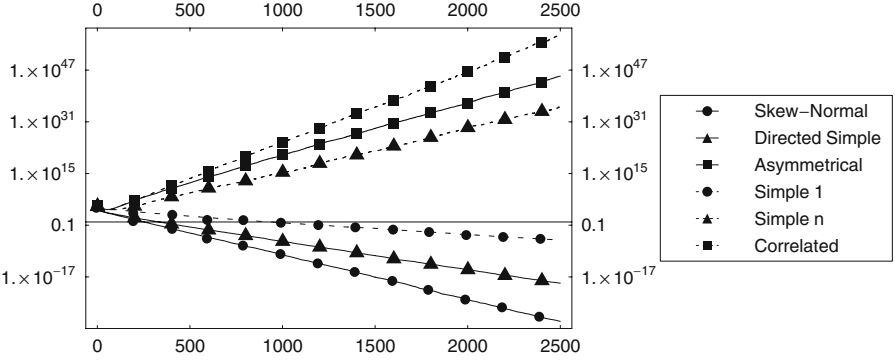


Fig. 12. Medians of the results

f_4 Schwefel's Problem 2.21

$$f_4(x) = \max\{|x_i|, 1 \leq i \leq 30\}$$
$$-100 \leq x_i \leq 100, \quad \min(f_4) = f_4(0, \ldots, 0) = 0$$

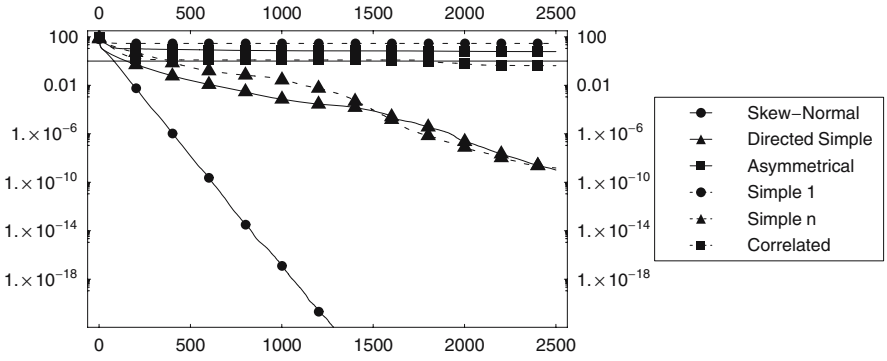


Fig. 13. Averages of the results

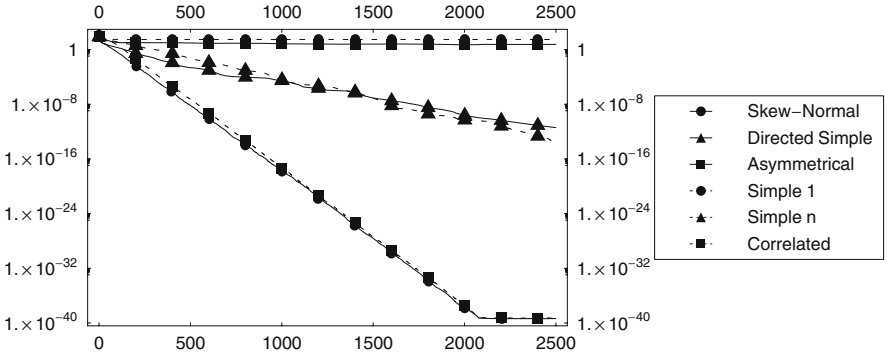


Fig. 14. Medians of the results

f_5 Generalized Rosenbrock's Function

$$f_5(x) = \sum_{i=1}^{29} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$$
$$-30 \le x_i \le 30, \quad \min(f_5) = f_5(1, \ldots, 1) = 0$$

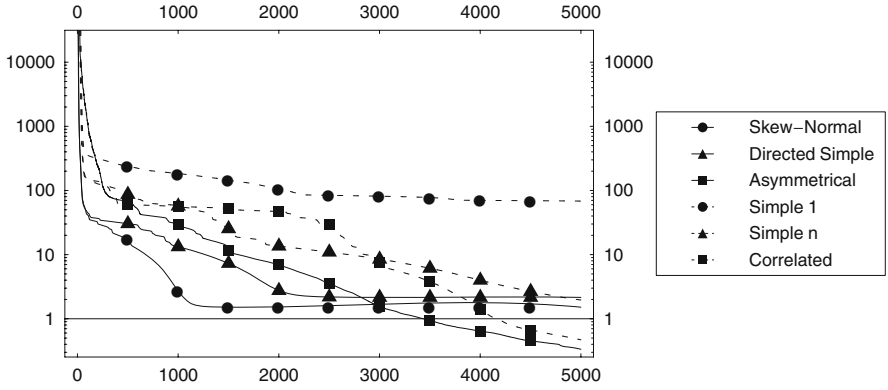


Fig. 15. Averages of the results

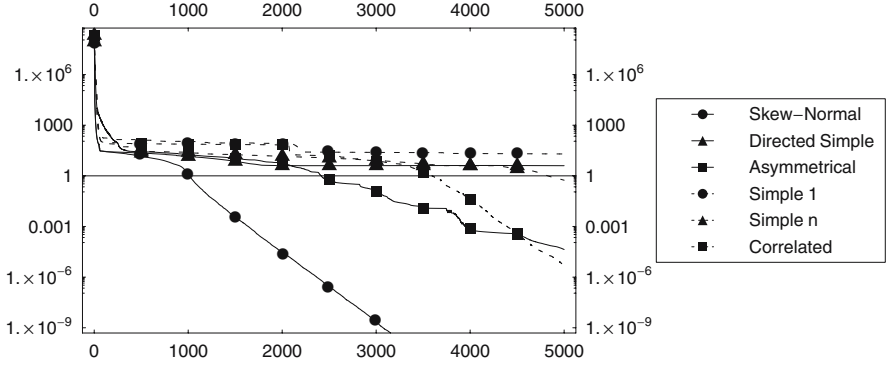


Fig. 16. Medians of the results

f_6 Step Function

$$f_6(x) = \sum_{i=1}^{30} (\lfloor x_i + 0.5 \rfloor)^2$$

$$-100 \leq x_i \leq 100, \quad \min(f_6) = f_6(0, \ldots, 0) = 0$$

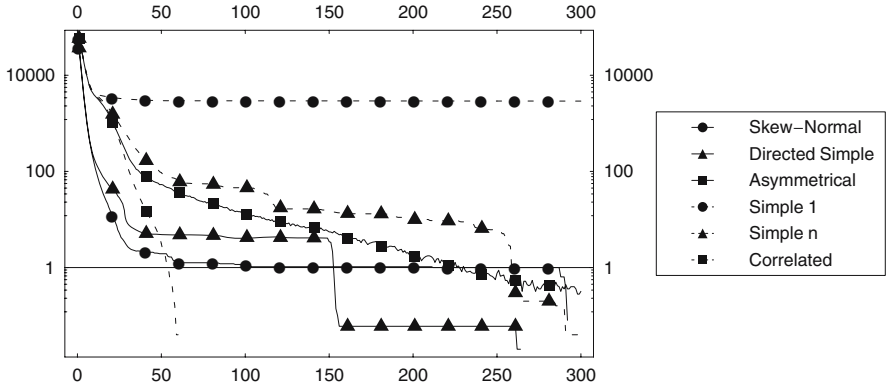


Fig. 17. Averages of the results

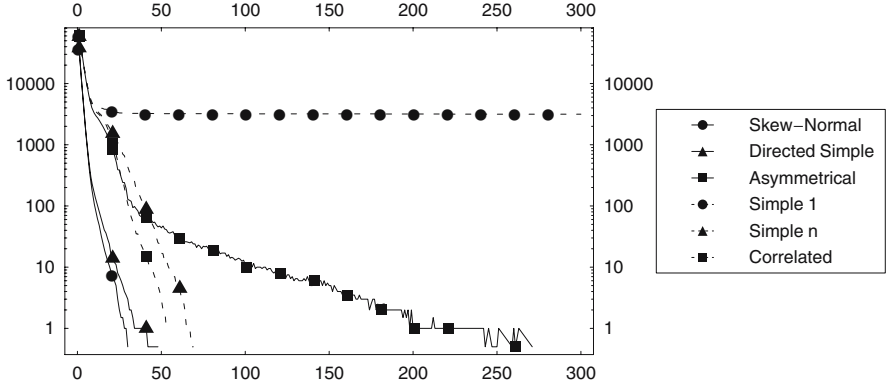


Fig. 18. Medians of the results

f_7 Ackley's Function

$$f_7(x) = -20\exp\left(-0.2\sqrt{\frac{1}{30}\sum_{i=1}^{30} x_i^2}\right) - \exp\left(\frac{1}{30}\sum_{i=1}^{30}\cos(2\pi x_i)\right) + 20 + e$$
$$-32 \le x_i \le 32, \quad \min(f_7) = f_7(0,\ldots,0) = 0$$

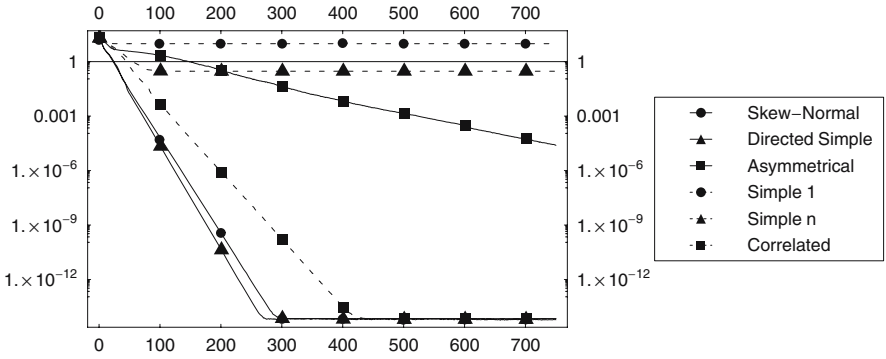


Fig. 19. Averages of the results

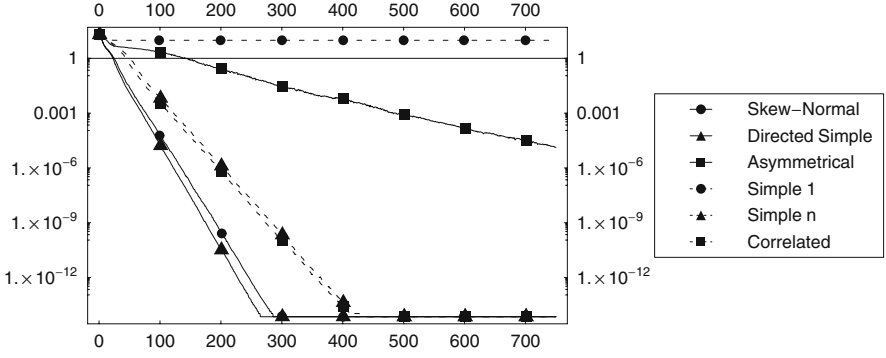


Fig. 20. Medians of the results

7 Conclusions

With the directed mutation by means of the SN distribution a mutation operator is given that clearly outperforms the other mutation operators. It is the only directed mutation with the density function not being defined by case differentiation. Its expectation as well as its variance are convergent for all degrees of skewness. Random number generation is simple and fast. Taking into account that the algorithm itself is quit fast, e.g. compared to the correlated mutation, the use of the directed mutation might be quite beneficial for many problems.

References

1. A. Azzalini. A class of distributions which includes the normal ones. *Scand. J. Statist.*, 12:171-178, 1985.
2. T. Bäck. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York, 1996.
3. S. Berlik. A directed mutation framework for evolutionary algorithms. In R. Matoušek and P. Ošmera, editors, *Proceeding of the 10th Int.Conf. on Soft Computing, MENDEL*, pages 45-50, 2004.
4. D. B. Fogel. Recombination. In T. Bäck, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, chapter C3.3, pages C3.3:11-C3.3.13. Institute of Physics Publishing, Bristol and Oxford University Press, Oxford, 1997.
5. L. Hildebrand. *Asymmetrische Evolutionsstrategien*. PhD thesis, Universität Dortmund, 2001.
6. H.-P. Schwefel. Adaptive Mechanismen in der biologischen Evolution und ihr Einfluβ auf die Evolutionsgeschwindigkeit. Technical report, Technical University of Berlin, 1974.
7. H.-P. Schwefel. *Evolution and Optimum Seeking*. John Wiley & Sons, New York, 1995.

Rule-Based Fuzzy Inference

Smooth Extensions of Fuzzy If-Then Rule Bases

Thomas Vetterlein

Department of Computer Science I, Dortmund University, 44227 Dortmund
Thomas.Vetterlein@uni-dortmund.de

Abstract. In order to extend fuzzy if-then rules bases, we propose to make use of a method which has been developed for the interpolation of crisp data – the multivariate spline interpolation. Among the various possibilities of how to accomplish the necessary generalisations, we describe here the probably simplest method: We apply spline interpolation to fuzzy data which itself is approximated by vectors of a finite-dimensional real linear space.

1 The Problem

We consider in this paper the question how to extend a fuzzy if-then rule base to a total function, requiring that this function is, in some reasonable sense, as smooth as possible. Roughly speaking, we assume to be in a situation of the following kind: We are given two variables both of which may take sharp or unsharp values in a bounded subset of a finite-dimensional real Euclidean space; the first of these variables uniquely determines the second one; and we know about this dependence only from a few special cases. The problem is then how to determine a function which maps the whole domain of the first variable to the domain of the second one, thereby comprising not only the known cases, but also minimising a parameter which measures in some reasonable sense the function's curvature.

Leaving the smoothness requirement aside, the problem is well-known, and various methods have been proposed. Before comparing our approach to already existing ones, we shall first specify the formal background of our considerations, so as to be able to clarify the idea of this paper.

We shall work with what could be called standard fuzzy sets according to [DiKl]. However, rather than working with functions from a base set to the real unit interval, we prefer to have fuzzy sets defined level-wise. Besides, all fuzzy sets will be assumed to be contained in one fixed bounded region of an $\mathbb{R}^p$.

T. Vetterlein: *Smooth Extensions of Fuzzy If-Then Rule Bases*, Advances in Soft Computing **2**, 53–59 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

Definition 1.1. *By a domain, we mean a regularly closed, convex, bounded subset of $\mathbb{R}^p$ for some $p \geq 1$. Let Ω be a domain, and let $\mathcal{K}(\Omega)$ be the set of non-empty convex and closed subsets of Ω. We partially order $\mathcal{K}(\Omega)$ by the subset relation $\subseteq$; and we endow $\mathcal{K}(\Omega)$ with the topology induced by the Hausdorff metric dn d_H.*

Let [0,1] be the real unit interval, endowed with the natural ordering and usual topology. A fuzzy vector in Ω is then meant to be a decreasing and left-continuous function v: $[0,1] \to \mathcal{K}(\Omega)$; by $\mathcal{F}(\Omega)$, we denote the set of all fuzzy vectors in Ω. We partially order $\mathcal{F}(\Omega)$ pointwise; and we endow $\mathcal{F}(\Omega)$ with the metric $d(v,w) = \sup_{\alpha\in[0,1]} d_H(v(\alpha), w(\alpha))$, $v, w \in \mathcal{F}(\Omega)$.

Clearly, $\mathcal{F}(\Omega)$ may be identified with those elements of $\mathcal{E}^n$ from [DiKl] whose support is within the domain Ω [DiKl], Proposition 6.1.6/7].

We will embed $\mathcal{F}(\Omega)$ in the usual way into a function space; see e.g. [DiKl]. In what follows, S^{p-1} denotes the unit sphere of $\mathbb{R}^p$, and $(\cdot,\cdot)$ is the usual scalar product in $\mathbb{R}^p$; $p \geq 1$.

Definition 1.2. *Let $\Omega \subseteq \mathbb{R}^p$ a domain, and let $v : [0,1] \to \mathcal{K}(\Omega)$ be a fuzzy vector in Ω. We call*

$$s_v : [0,1] \times S^{n-1} \to \mathbb{R},\ (\alpha, e) \to \sup\{(r,e) : r \in v(\alpha)\}$$

the support function of v.

Moreover, let $L_\infty([0,1] \times S^{n-1})$ be the linear space of bounded real-valued functions on $[0,1] \times S^{n-1}$, endowed with the supremum norm. Let $L_\infty([0,1] \times S^{n-1})$ be pointwise partially ordered.

Proposition 1.3. *Let $\Omega \subseteq \mathbb{R}^p$ be a domain. Then the mapping $\mathcal{F}(\Omega) \to L_\infty([0,1] \times S^{n-1}), v \to s_v$ is injective, isometric, and order preserving.*

Moreover, $s \in L_\infty([0,1] \times S^{n-1}$ is the support function of some fuzzy vector $v \in \mathcal{F}(\Omega)$ iff (i) for all $\alpha \in [0,1], e, e_1, e_2 \in S^{n-1}, \lambda_1, \lambda_2 \geq 0$ such that $e = \lambda_1 e_1 + \lambda_2 e_2$, we have $s(\alpha, e) \leq \lambda_1 s(\alpha, e_1) + \lambda_2 s(\alpha, e_2)$, (ii) for all $e \in S^{n-1}, s(\cdot, e)$ is decreasing and left-continuous, and (iii) $v \leq s\Omega$.

Note that under (iii), we considered Ω as a crisp element of $\mathcal{F}(\Omega)$ in the usual way.

By Proposition 1.3, we may identify $\mathcal{F}(\Omega)$ with a closed subset of the Banach function space $L_\infty([0,1] \times S^{n-1})$.

We next have to specify which functions between sets of fuzzy sets are taken into account for interpolation.

Definition 1.4. *Let $\Xi \subseteq \mathbb{R}^m$ and $\Upsilon \subseteq \mathbb{R}^n$ be domains, and let $\mathcal{X} \subseteq \mathcal{F}(\Xi)$. Then a function $f : \mathcal{X} \to \mathcal{F}(\Upsilon)$ is called fuzzy if f may be extended to a function $\overline{f} : \mathcal{F}(\Xi) \to \mathcal{F}(\Upsilon)$ which preserves the order, that is, for which $\overline{f}(v) \leq \overline{f}(w)$ whenever $v \leq w$ for $v, w \in \mathcal{F}(\Xi)$.*

In case that $\mathcal{X}$ is finite, we call f a fuzzy if-then rule base.

It is now possible to formulate the aims towards which we are working. Assume first that our data is crisp, that is, that we are given finitely many pairs from a bounded, closed subset Ω of $\mathbb{R}^m$ and from $\mathbb{R}^n$. Then a degree of smoothness of a function $\Omega \to \mathbb{R}^n$ is given by the integral of the norm of the second derivative over Ω; under certain assumptions, this value exists and we are lead to a uniquely determined interpolating function.

Now, in the fuzzy case, we wish to calculate functions from $\mathcal{F}(\Xi)$, viewed as a subset of the Banach space $L_\infty([0,1]\times S^{m-1})$, to $\mathcal{F}(\Upsilon)$, viewed as a subset of $L_\infty([0,1]\times S^{n-1})$. Proceeding analogously to the crisp case leads apparently to difficult requirements. First of all, an interpolating fuzzy function must possess its second Fréchet derivative; second, its norm must be integrable with respect to some measure on $\mathcal{F}(\Xi)$; third, this measure should be a metrically invariant one.

Unfortunately, this program fails: a metrically invariant measure does not exist on $\mathcal{F}(\Xi)$. A way out of this first difficulty is to restrict $\mathcal{F}(\Xi)$ in a way such that a measure of this kind does exist.

On the other hand, the program simplifies dramatically if we replace the spaces $L_\infty([0,1]\times S^{p-1})$ by finite-dimensional ones – simply by restricting the function domain to a finite subset. This is how we proceed in Section 3.

2 Known Approaches

Like in many areas of the theory of fuzzy sets, also in the present one a quite active research is to be noted. We would like to mention three directions– those we know about. It is (i) the logical approach; (ii) the usage of fuzzy relations; (iii) interpolation based on linearity notions. Let a fuzzy if-then rule base $(u_1, v_1), \ldots, (u_k, v_k)$ of pairs from $\mathcal{F}(\Xi) \times \mathcal{F}(\Upsilon)$ for domains Ξ and Υ be given, and let us in this section adopt the usual notion of a fuzzy set as a function from a base set to [0,1].

(i) First of all, an entry (u_i, v_i) of the rule base may be considered as a proposition like "if u_i then v_i" and may be formalised on the base of a logical calculus. For instance, some version of Hájek's Basic Predicate Logic [Haj] may be used. The difference to our setting is easily stated: In the logical framework, we investigate what is expressed by the rule base as it stands, not taking into account what is not derivable. Put into the language of logics, we may say that it is our aim to properly *extend* a given rule base, that is, adding statements which are not part of the information provided by the rules. To this end, we work so-to-say "horizontally" – by using features of the base set –, and not "vertically" – by considering various ways of how to connect truth values. – However, we would like to mention the work of Nov́ak, see e.g. [Nov], where the "horizontal" viewpoint is used also in logics; namely, certain logical connectives are defined which do refer to the structure of the base set.

(ii) Concerning the second point, it is clear that there are in principle lots of possibilities of what to require from a function $f : \mathcal{F}(\Xi) \to \mathcal{F}(\Upsilon)$ such that

$f(u_1) = v_1, \ldots, f(u_k) = v_k$. A rather popular condition is that f is induced by a fuzzy relation between Ξ and Υ; this means in the simplest case that there is a fuzzy relation $R : \Xi \times \Upsilon \to [0,1]$ such that $f_R(u_i) = v_i$ for all i, where $f_R(u)(y) = \sup_{x \in \Xi}(u(x) \wedge R(x,y))$ for $u \in \mathcal{F}(\Xi)$ and $y \in \Upsilon$. It is clear that this requirement dramatically restricts the possible choices of f. In particular, f is then already determined by the fuzzy singletons, i.e. those fuzzy sets having a one-point support, because f preserves all suprema. Another point is the fact that a fuzzy relation R such that f_R maps the u_i to v_i may not exist at all; conditions are listed e.g. in [Got].

(iii) In connection to the third line of research, we should mention the well-known work of Kóczy and Hirota at the first place; see e.g. [KoHi]. His and several others' work comes closest to what we have in mind. For a review of methods developed so far, the article [Jen] can be recommended. Let us mention the idea behind the method which also is contained in [Jen]; it deals with rule bases $((u_1, v_1), \ldots, (u_k, v_k)$ of pairs of convex fuzzy sets over $\mathbb{R}$. To calculate the image $f(u)$ of some value u in the domain under the interpolating function f, one has first to determine two "neighbouring" entries u_1 and u_2; then, roughly spoken, $f(u)$ is constructed in the analogous way from v_1 and v_2 as u may be constructed from u_1 and u_2. – This approach as well as comparable ones are based on a clear geometric intuition, and their common advantage is that their technical realization is straightforward. A disadvantage is that their applicability is usually restricted; it seems that ill-behaved cases can mostly not be excluded. Besides, the transition to fuzzy sets of higher dimensions, if possible at all, requires often new ideas, like also in [JeKlKo], the paper subsequent to [Jen]. Finally, when we have a look what happens in the case that our data are crisp, we realize that we have to do with the easiest possible method; in the one-dimensional case, the methods reduce to the linear interpolation between neighbouring points.

3 Interpolation of Crisp Data – Splines

When trying to make use of methods which have been developed for the interpolation of crisp data, we have to overcome the problem that we no longer have to do with finite-dimensional spaces: even $\mathcal{F}(\Omega)$, where $\Omega \subseteq \mathbb{R}$, embeds into an infinite-dimensional space. So there are in principle two possible ways: either, we generalise the existing methods to the infinite-dimensional case; or we reduce our spaces to finite-dimensional ones. In this paper, we shall describe a method according to the second way.

We first have to review the crisp case; so assume that we are given pairs $(x_1, y_1) \ldots, (x_k, y_k) \in \mathbb{R}^m \times \mathbb{R}^n$ and that we want to determine the smoothest possible function $f : \mathbb{R}^m \to \mathbb{R}^n$ which interpolates the given data. This problem is made precise, and has its unique solution, according to the following method, developed in the multidimensional case first by Atteia [Att]. We shall use the following variant of it.

Definition 3.1. *Let $m, n \geq 1$, and let $\Omega \subseteq \mathbb{R}^n$ be bounded and regularly closed. Then set*

$$F \stackrel{\text{def}}{=} \{f : \Omega \to \mathbb{R}^n : \text{is continuous and of class } C^1, \; Df \text{ is bounded and Lipschitz continuous}, \; D^2 f \, is \, L^1\} ;$$

let $||f||\infty = \sup_{x \in \Omega} f(x)$, $||Df||_\infty = \sup_{x \in \Omega} ||Df(x)||$, *and* $||D^2 f||_1 = f_\Omega ||D^2 f(x)|| dx$; *endow F with the norm*

$$||f|| = ||f||_\infty + ||Df||_\infty + ||D^2 f||_1, f \in F .$$

Here, the derivatives $Df, D^2 f$ are understood to have $\overset{o}{\Omega}$ as their domains.

F, together with $|| \cdot ||$, is a Banach space. The key fact which we need is now the following [Att, Hol].

Theorem 3.2. *Let $m, n \geq 1$, and let $\Omega \subseteq \mathbb{R}^m$ be bounded and regularly closed. Let K be a non-empty closed, convex subset of F; and for $f \in F$, set*

$$m(f) = \int_\Omega ||D^2 f(x)|| dx .$$

Let $N = \{f \in F : m(f) = 0\}$, and $C_K = \{f \in F : f + K = K\}$. Assume that N is finite-dimensional and that $N \cap C_K = \{0\}$. Then there is a unique $f \in K$ minimising m(f).

With respect to the notation of this theorem, we see that if $x_1, \ldots, x_k \in \Omega$ such that there is no non-trivial affine function mapping all x_i to 0, and $y_1, \ldots y_k \in \mathbb{R}^n$, we may set $K = \{f \in F : f(x_1) = y_1, \ldots, f(x_k) = y_k\}$ to conclude that there is a uniquely determined function f ∈ K for which the integral over the norm of the second derivative is smallest.

4 Interpolation of Fuzzy Data

Assume now that we have to solve a problem analogous to the one of the last section, with the difference that both the data from the domain and from the range are fuzzy vectors. Our fuzzy data is, according to Section 1, represented by real-valued functions on $[0, 1] \times S^{n-1}$. To reduce the interpolation problem to a finite-dimensional one, we shall, according to the pragmatic approach announced, approximate these functions by their values in finitely many points.

Accordingly, we work with the following structure, modeled upon Definition 1.2 and Proposition 1.3.

Definition 4.1. *Let $\Omega \subseteq \mathbb{R}^p$ be a domain, let I_f be a finite subset of $[0,1]$ and S_f a finite subset of S^{p-1}. We call a function $s\colon I_f \times S_f \to \mathbb{R}$ an approximative support function w.r.t. Ω, I_f, S_f if (i) for all $e \in S_f$, there is an $x \in \mathbb{R}^p$ such that $(x,e) = s(e)$ and $(x,f) \leq s(f)$ for all $f \in S_f \backslash \{e\}$; (ii) for all $e \in S_f$, $s(\cdot,e)$ is decreasing; and (iii) $s \leq s_\Omega|_{I_f \times S_f}$.*

Moreover, let $L(I_f \times S_f)$ be the space of real-valued functions on $I_f \times S_f$, endowed with the supremum norm and the pointwise order.

In an analogous way, also the notion of an interpolating function and of a fuzzy if-then rule base is adapted from Definition 1.4.

Let us fix domains $\Xi \subseteq \mathbb{R}^m$ and $\Upsilon \subseteq \mathbb{R}^n$, and finite subsets $S_d \subseteq S^{m-1}, S_r \subseteq S^{n-1}, I_f \subseteq [0,1]$.

Definition 4.2. *Let $\mathcal{D}$ be the set of approximative support functions w.r.t Ξ, I_f, S_d, and $\mathcal{R}$ those w.r.t. Υ, I_f, S_r. Set*

$$F \stackrel{\text{def}}{=} \{f :\ \mathcal{D} \to \mathcal{R} : f \text{ is continuous and of class } C^1, \\ Df \text{ is bounded and Lipschitz continuous}, \\ D^2 f \text{ is } L^1, f \text{ preserves the order}\}\ .$$

For a subset $\mathcal{X}$ of $\mathcal{D}$, a function $f : \mathcal{X} \to \mathcal{R}$ is called fuzzy approximative if there is an $\bar{f} \in F$ extending f.

In case $\mathcal{X}$ if finite, we call f an approximative fuzzy if-then rule base.

We may now apply Theorem 3.2.

Theorem 4.3. *Let $(u_1, v_1), \ldots, (u_k, v_k) \in \mathcal{D} \times \mathcal{R}$ be an approximative fuzzy if-then rule base such that there is no non-trivial affine function $\mathcal{D} \to L(I_f \times S_r)$ which maps $u_1 \ldots, u_k$ to 0. Set*

$$m(f) = \int_{\mathcal{D}} ||D^2 f(x)|| dx\ .$$

Then there is a unique $f\colon \mathcal{D} \to \mathcal{R}$ such that $f(u_1) = v_1, \ldots, f(u_k) = v_k$ which minimises $m(f)$.

Proof. The set K of all functions $f \in F$ such that $f(u_1) = v_1, \ldots, f(u_k) = v_k$ is convex and closed. Furthermore, by definition of an approximative fuzzy if-then rule base, K is not empty. So the claim follows from Theorem 3.2.

References

[Att] M. Atteia, Fonctions 'spline' definies sur un ensemble convexe, *Numer. Math.* **12** (1968), 192–210.

[DiKl] P. Diamond, P. Kloeden, "Metric spaces of fuzzy sets: theory and applications", World Scientific, Singapore 1994.

[Got] S. Gottwald, Fuzzy relation equations and approximate solutions. Some recent results, Proceedings of 3rd EUSFLAT conference (Zittau 2003), 6–11.

[Haj] P. Hájek, "Metamathematics of fuzzy logic", Kluwer Acad. Publ., Dordrecht 1998.

[Hol] R. B. Holmes, "Geometric functional analysis and its applications", Springer-Verlag, New York and Berlin 1975.

[Jen] S. Jenei, Interpolation and extrapolation of fuzzy quantities revisited – an axiomatic approach, *Soft Comput.* **5** (2001), 179–193.

[JeKlKo] S. Jenei, E. P. Klement, R. Konzel, Interpolation and extrapolation of fuzzy quantities - the multiple-dimensional case, *Soft Comp.* **6** (2002), 258–270.

[KoHi] O. Kóczy, K. Hirota, Approximate reasoning by linear rule interpolation and general approximation, *Int. J. Approx. Reasoning* **9** (1993), 197–225.

[Nov] V. Novak, A horizon shifting method of linguistic hedges for approximate reasoning, Proceeding I of the 5th IEEE conference (New Orleans 1996), 423–427.

Pre-validation of a Fuzzy Model

Farida Benmakrouha[1]

[1] Computer Science Department INSA - IRISA, 20 Avenue Buttes de Coësmes, CS 1431 35043 Rennes Cedex

1 Introduction

Many papers propound algorithms for extraction of knowledge from numerical data. But, few works have been developed for design of experiments and datum plane's cover.

This paper tries to measure the impact of datum plane's cover on the outcome of a fuzzy inference system. We propose a measure used to pre-validate a fuzzy model.

This pre-validation takes place after design of the inference system. So, when the model is not pre-validated, we have not to carry out the next steps, optimisation and validation.

The structure of the paper is as follows : In the second section, we present the inference system. In the next section, we analyse a measure of datum plane's coverage. We propound, using this measure, a pre-validation of the fuzzy model.

In section 4, we emphasize, with two examples, the importance of this criterion to estimate the performance of a fuzzy inference system.

2 The Inference Fuzzy System

In a zero-order Takagi-Sugeno model (T-S model) [3], the rule basis are of the form :

$$\begin{array}{c}\textbf{If } x_1 \text{ is } A_{1i} \textbf{ and } x_2 \text{ is } A_{2i} \ldots \textbf{ and } x_p \text{ is } A_{pi} \\ \textbf{then } y \text{ is } C_i\end{array} \tag{1}$$

where X_j are input variables, A_{ij} are linguistic labels on the universe of discourse Y represented by a fuzzy set μA_{ij}, y is the inferred output and C_i are constant values. The output y is, given an input vector $x_1, \ldots, x_p$:

F. Benmakrouha: *Pre-validation of a Fuzzy Model*, Advances in Soft Computing **2**, 61–64 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

$$y(x)^{i=1} = \alpha_i \frac{\sum^n (x) \times C_i}{\sum_{j=1}^n \alpha_j(x)} \tag{2}$$

where the truth value of rule i $\alpha_i(x) = \mu A_{i1} \ldots \mu A_{ip}$ and n the number of rules. This truth value is the product of the membership degree of x to A_{ip} The conclusions C_i can be initialized by an algorithm described in [1]. But, when data don't cover the whole input space, some rules are never learned and methods for determining conclusions don't work very well.

3 Pre-validation of a fuzzy system

We suppose that there exists a learning set $\mathcal{L} = \{(x_i, d_i)\}$, where X_i is an input vector and d_i, the corresponding output.

We assume also that the desired function f is defined in

$$V = [a_1, b_1] \times [a_2, b_2] \times \ldots \times [a_p, b_p]$$

Usually, to validate a fuzzy inference system, the mean square error (MSE) is calculated on a test set. If the MSE exceeds a threshold, then training is made, using a gradient method. This consists in modifying C_i at each presentation of examples from the error $(y(\mathbf{x}_i) - d_i)$.

Unfortunately, in case of model's invalidation, we can't determine rules never learned that cause the gap between the model and the real system. Moreover, if there is an insufficient coverage of datum plane, training and finer splitting of input space are inefficient and useless.

With this criterion proposed below, we estimate the datum plane's coverage and we are able to isolate unactivated rules. Then, partial remodelling of the fuzzy inference system is possible.

For each rule, we define intervalls

$$I_i = \{x | \alpha_i(x) > \beta\}$$

On some conditions, these intervalls are disjoined and formed a partion of V. We note

P_i = probability associated to I_i under the hypothesis H_0 (X follows an uniform distribution)

N_i = number of x such that $\alpha_i(x) > \beta$

$$U = \sum_{i=1,m} \frac{(N_i - mP_i)^2}{mP_i}$$

(m is the number of data).
n is the number of rules.

We apply the χ^2 test on U. If the hypothesis is accepted, the fuzzy model is pre-validated, otherwise, it is not.

4 Experiments

We compare, using two examples, output inferred by fuzzy systems :

- when datum plane is "sufficiently" covered
- when datum plane is not well covered

The mean squarre error, performance index of the approximate system, is calculated on measures of the test set(V), before optimization step.

We take P= 0.90, the threshold to reject hypothesis $H0(U_P = 117.4)$ with $m - 1 = 99$ degrees of freedom and $\beta = 0.3$.

4.1 First Example

This example has been shown in [4].

$$y = \frac{sin(x)}{x}$$

In this paper, the authors propose a method to improve Takagi-Sugeno' approximation, under assumption that datum plane is covered. If this assumption no longer holds, we see that this method can't work, since it implies that, before optimization, the fuzzy system gives acceptable results.

Table 1. First Table

n	Training set	Test interval	MSE	U
7	[0.2 3.5]	[0.2 11.0]	0.117	131.01
7	[0.2 11.0]	[0.2 11.0]	0.002	10.67

4.2 Second example

This example has been shown in [2]:

$$g(x_1, x_2) = 1.9(1.35 + e^{x1}\sin(13(x_1{-}.6)^2) \\ e^{-x2}\sin(7x_2)) + 4\ . \tag{3}$$

Table 2. Second Table

n	Training set	Test interval	MSE	U
49	[0.2 0.6]	[0.0 1.0]	1.07	475.08
49	[0.5 1.0]	[0.0 1.0]	0.97	349.0

5 Conclusion

We have proposed a measure for pre-validation of a fuzzy inference system. When the model is not pre-validated, we have not to carry out next steps, particularly optimization step.

We have shown that this criterion is a good measure for datum plane's coverage.

From these experiments, we remark that coverage rate is correct under two conditions :

- (i) datum plane and test interval are not disjoined.
- (ii) the ratio number of rules / card(T) is acceptable.

We also note that training is useless when the model is not pre-validated.

References

1. Glorennec PY., "Quelques aspects analytiques des systèmes d'inférence floue", *Journal Européen des Systèmes Automatisés*, Vol. 30, n° 2–3, 1996.
2. Hwang J.N., Lay S.R., Maechler M., Martin D., Schimert J. , "Regression modeling in backpropagation and projection pursuit learning", *IEEE Trans. Neural Networks*, Vol. 5 n° 3, May 1994
3. Takagi T., Sugeno M., "Fuzzy identification of systems and its applications to modeling and control", *IEEE Trans, on SMC*, SMC–15(1), Janv 1985.
4. J.Gutierrez, F. Fernandez, S. Guadarrama, E. Trillas , "A step towards conceptually improving tagaki-sugeno's approximation",*IPMU'2002*, III:pp 1789–1794, July 2002

Multiresolution Fuzzy Rule Systems

Ricardo Ñanculef[1], Carlos Concha[1], Claudio Moraga[2], and Héctor Allende[1]

[1] Department of Informatics, Federico Santa María University, Valparaíso, Chile
[jnancu,cconcha,hallende]@inf.utfsm.cl
[2] Department of Computer Science, University of Dortmund, 44221 Dortmund, Germany
claudio.moraga@udo.edu

Abstract. This paper describes the modelling of fuzzy rule systems using a multiresolution strategy that handles the problem of granularization of the input space by using multiresolution linguistic terms. Models of different resolutions are chained by antecedents because linguistic terms of a level j are obtained by refinements of linguistic terms of a superior level $j+1$. The models can also be chained by consequents using aggregation procedures. The family of models are called Multiresolution Fuzzy Rule Systems.

A metasemantics based on linguistic operators is proposed for the interpretation of the refinements as a rule specialization. Interesting models result allowing local refinement of rules that preserve the semantic interpretation.

Keywords: Fuzzy Rule Systems, Rules Hierarchies, Learning Algorithms, Multiresolution Analysis.

1 Introduction

Fuzzy logic provides an inference morphology that enables approximate human reasoning capabilities to be applied to knowledge-based systems [6]. Making use of this framework, fuzzy rule-based systems establish a map from a feature space to an output space by aggregation of fuzzy IF-THEN rules each one representing a partial knowledge of the universe of discourse. Rigorous theoretical works have proved that several types of fuzzy models are universal approximators in the sense that these fuzzy models could approximate any real continuous function to any degree of accuracy on a compact domain [2], [8]. The problem with this results is that they are non-constructive and therefore the practical construction of fuzzy models,the model identification problem, requires more solid foundations.

[1]This research was partially supported by Research Grant Fondecyt 1040365 and Research Research Grant DGIP-UTFSM 240425. Partial support was also received from the German Ministry for Education and Research under Grant CHL 03-Z13.

R. Ñanculef et al.: *Multiresolution Fuzzy Rule Systems*, Advances in Soft Computing **2**, 65–79 (2005)
www.springerlink.com © Springer-Verlag Berlin Heidelberg 2005

Basically, there are two ways to solve the identification problem: from the expert knowledge or learning from sample data. The former exploits the fuzzy sets capability to represent linguistic terms, while the latter utilizes the universal approximation property of these systems. Usually the two approaches cooperate in a hybrid system that use heuristic information codified in a form of linguistic variables to generate an initial model structure and then learning procedures for tuning the free parameters.

The task of building a fuzzy system involves at least the following three subtasks:

- *A Granularization Task*, i.e., to identify an appropriate partition of the input and output spaces. This procedure can be viewed as determination of a collection of subspaces $\mathbf{C} = \{V_1, V_2, \ldots, V_r\}$ whose organization (in form of a rule-system) allows us to handle the complexity of the entire universe of discourse by means of local pieces of information.
- *Variables Morphology Setting*, i.e., to select the morphology of the linguistic variables taking part in the problem modelling. For example we can choose trapezoidal functions as reference functions for each linguistic term and put them together with a normalization constraint over the partition structure. The morphology of these variables determines the way of representing the knowledge contained in the subspaces of information generated by the granularization structure and together they distribute a set of elastic constraints over the universe of discourse.
- *Selection of an Inference Mechanism*, i.e. to choose a fuzzy inference procedure that allows us to propagate the partial acquired knowledge and obtain new knowledge from it in a coherent and "common sense" way that mimics human reasoning mechanisms.

When doing this we must preserve the interpretable capacity of the model as well as reasonable approximation capabilities. The first claims that the system must be able to explain how it reaches its decisions while the last, that these decisions must be good.

Our approach to fuzzy modelling consists in using a granularization strategy based on the complexity decomposition that the multiresolution analysis makes over a data space. Seminal ideas in this field are introduced in [4], [5] and [16]. The space decomposition procedure automatically induces the morphology of the linguistic variables by refinements procedures; or viceversa, the choice of a basic suitable mother function induces the overall granularization structure. The resulting fuzzy model is a stratified set of rules, each level representing a given resolution. Metasemantic operators will give a semantic interpretation of global and partial refinements.

Let us begin by introducing the multiresolution framework.

2 Multiresolution Framework

Given a function or signal we can think of representing it on a function space generated by some basis functions. The function can be viewed as composed of a local background that is representable in terms of these basis functions and some fluctuations or details on top of it that are loosed by the approximation on this space. The distinction between the representable part and the details is given obviously by the approximation capacity of the function space. We can name this degree of discernibility a resolution level *j*. If we are able to establish an order over a set of approximation spaces in terms of their resolution, we can imagine progressively increasing the resolution level adding finer details to the coarser description, providing a successively better approximation to the function. We will say that a function is more complex that another if we need a greater level of resolution for representing the first one. The hierarchy established for the approximation spaces is then a hierarchy for the complexity that they can handle. The mathematical framework that puts these ideas in order is the multiresolution analysis. The references [14], [7] contain an introduction to the concept of multiresolution analysis in the context of wavelet theory. We state here the following definition[1] [10], [3].

Definition 1. *A multiresolution analysis (M.A.) of* $L^2(\Re)$ *consist of a sequence of embedded closed subspaces* $\{V_j\}_{j\in\mathbf{Z}}$ *de* $L^2(\Re)$ *such that the following conditions hold* $\forall j \in Z$

1. $V_j \subset V_{j+1}$,
2. $\cup_{j\in\mathbf{Z}}$ *is dense in* $L^2(\Re)$ *and* $\cap_{j\in\mathbf{Z}} = 0$,
3. $f(t) \in V_j \Leftrightarrow f(2t) \in V_{j+1}$ *(dilation)*
4. $f(t) \in V_0 \Leftrightarrow f(t-k) \in V_0$ *(traslation invariance)*
5. $\exists$ *a function* $\varphi(t) \in V_0$ *with a non-vanishing integral, called scaling function such that the collection* $\{\varphi(t-k)\}_{k\in\mathbf{Z}}$ *is a Riesz basis of* V_0.

The third condition is the key requirement of dyadic refinement. It says that all spaces $\{V_j\}$ are scaled versions of the central space V_0 and defines the scales of all these subspaces. A key consequence of this property is that since $\varphi \in V_0 \subset V_1$ must exist a sequence $\{h_k\} \in l^2(Z)$ such that the scaling function satisfies

$$\varphi(t) = 2\sum_k h\varphi(2t-k) \tag{1}$$

This functional equation [2] is known as the refinement equation. It is immediate that the collection of functions $\{\varphi_{jk}/k \in Z\}$ with $\varphi_{jk} = 2^{j/2}\varphi(2^j t - k)$

[1] In the following we will refer to one-dimensional multiresolution analysis of $L^2(\Re)$ that can be extended to more dimensions and to more general spaces.

[2] When the basis of functions is not orthogonal it is necessary define this equation in terms of a dual basis.

and a scaled family $\{\breve{\varphi}_{jk} = c_j \varphi_{jk}\}_{jk}$ are both Riesz bases of V_j. The particular scaling factor $c_j = 2^{j/2}$ in the definition of φ_{jk} keeps the l^2-norm of the functions equal to one in every scale. In other contexts, fuzzy modelling for example, may be desirable that other norms, such as the supremum norm, keeps constant. In these cases we choose the normalization coefficient c_j appropriately.

The difference W_j between two successive spaces V_j and V_{j+1} are called the complementary space of V_j. We write $V_{j+1} = V_j \oplus W_j$. As well as the scaling function defines a family of bases for all the approximation spaces V_j it is would be very useful to have a compact representation of their complementary spaces W_j. In this context the wavelet function is introduced. A function ψ is a wavelet if the collection of functions $\{\psi(t-k); k \in Z\}$ is a Riesz basis of W_0. In this case the set of functions $\{\psi_{jk} = 2^{j/2}\psi(2^j t - k); k \in Z\}$ is a Riesz basis of W_j and the family $\{\psi_{jk}; j, k \in Z\}$ is a Riesz basis of $L^2(\Re)$.

Given such function ψ, the wavelet spaces W_j are usually enough for carry out the multiresolution representation. In many other problems however, the scaling function have analytic and operational properties that make preferable to work with the spaces V_j explicitly. For example, the scaling function is usually non-negative while for the mother wavelet it is not the case. Moreover, when we find to protect the transparency of the approximation of a function at different levels of resolution, it is preferable to work explicitly in the spaces V_j more than only on the wavelet spaces.

Wavelet spaces provides a convenient way of handle the details information that we gain or loose when we change the resolution level. The convenience of the representation is in the good analytic and computational properties of the so called wavelet transforms. Let P_{j+1} be the projection of $f(t)$ on the space V_{j+1}. Then we have

$$P_{j+1}f = f_{j+1} = P_j f + d_j = \sum_k c_{j,k}\varphi_{j,k}(t) + \sum_k d_{j,k}\psi_{j,k}(t) \tag{2}$$

The wavelet transform is an algorithm to compute the coefficients $c_{j,k}$ and $d_{j,k}$ from the coefficients $c_{j+1,k}$ and viceversa. In the orthogonal case, the transform takes the form

$$\begin{array}{l} c_{j,l} = \sum_k h_{k-2l} c_{j+1,k} \\ d_{j,l} = \sum_k g_{k-2l} c_{j+1,k} \end{array} \tag{3}$$

This transform can be inverted as

$$c_{j+1,k} = \sum_l h_{k-2l} c_{j,l} + \sum_l g_{k-2l} d_{j,l} \tag{4}$$

The first direction of equations (from finer to coarser) is called decomposition or analytic process. The second (from coarser to finer) is named reconstruction or synthesis. The analytic and synthetic processes are equally simple (or complex) only if we are looking for a multiresolution representation of a function

having a perfect knowledge of the dyadic points that refinement equations need. Of course when our proposes involve learning an unknown function from a set of examples we cannot demand such exhaustive and homogeneous knowledge.

3 Multiresolution Fuzzy Rules Systems

In this section we study the use of the space decomposition strategy provided by multiresolution analysis for building fuzzy rule systems.

3.1 Multiresolution Takagi-Sugeno Hierarchy

Let us observe the definition of Multiresolution Analysis again. In each space V_j we can reach an approximation to a function $f(t)$ by a decomposition in terms of the basis functions of this space, i.e.

$$f \approx P_j f = f_j = \sum_k c_{j,k} \varphi_{j,k}(t) \tag{5}$$

where P_j is the projector on the space, $\{\varphi_{j,k}; k \in Z\}$ is the base of the space and $c_{j,k}$ are the decomposition coefficients. We want to use a Takagi-Sugeno fuzzy system to represent the approximation in this space. The Takagi Sugeno system was introduced by [15], [9] and is described by a set of the following if-then rules

$$F = \left\{ F^{(k)} \right\}_k = \left\{ \texttt{If} \;\; t \;\; \texttt{is} \;\; A^k \;\; \texttt{then} \;\; y = f_k(t) \right\}_k \tag{6}$$

where A^i stands for linguistic terms in antecedent while $y_i = f_i(x)$ is a crisp function in the consequent, usually constants or n-order polynomials.

When we are modelling these type of systems we need to specify a partition strategy of the input space that gives us the regions of operation for the local models that implement the consequents of the local rules. The region of operation of the i-th model is determined by the support of the (multidimensional) fuzzy relation $\mu_{A^{(i)}}(t)$ implementing the antecedent of the corresponding rule. An adequate partition of the inputs space is a key process for the performance of the fuzzy model because it establishes the way in which the knowledge about the problem is organized. We propose to use the multiresolution decomposition strategy to determine the subspaces where the local models defined by each rule must be located. It can be achieved by relating a model F_j to each space V_j.

The approximation given by the Takagi-Sugeno system (6) can be related to the multiresolution approximation (5) if we identify the decomposition coefficients $c_{j,k}$ with the crisp function consequent $f_i(x)$ and the family $\{\varphi_j(\cdot - k); k \in Z\}$ with fuzzy sets A^i. We denote F_j the fuzzy system corresponding to V_j and write

$$F_j = \left\{F_j^{(k)}\right\}_k = \{ \texttt{If} \;\; x \;\; \texttt{is} \;\; \varphi_{j,k} \;\; \texttt{then} \;\; y = f_{j,k} = c_{j,k}\}_k \tag{7}$$

whose operational output, choosing the classical aggregation method, is

$$y_{0,j} = \frac{\sum_k \varphi_{j,k}(t) c_{j,k}}{\sum_k \varphi_{j,k}(t)} \tag{8}$$

If the normalization factor $\sum_k \varphi_{j,k}(t)$ is equal to 1, the output of (7) and the multiresolution output (5) are the same. Otherwise the results differ only by the normalization operation, which obeys to a difference between the fuzzy modelling approach and the multiresolution analysis approach. While in the fuzzy modelling approach each rule gives an expert opinion (complete answer) about the approximated function by a local map, the multiresolution analysis only calculates projections (partial answers) on some basis functions. In the first case the basis functions act as similarity measures of the function with the local approximation while in the second they are pieces of the function. Of course, the two approaches can become compatible and operationally equivalent if the rule experts are taken as augmented approximators of the decomposition coefficients $c_{j,k}$.

$$F_j = \left\{F_j^{(k)}\right\}_k = \{ \texttt{If} \;\; x \;\; \texttt{is} \;\; \varphi_{j,k} \;\; \texttt{then} \;\; y = \tilde{c}_{j,k}\}_k \tag{9}$$

where $\tilde{c}_{j,k} = c_{j,k} \sum_k \varphi_{j,k}(t)$. Now we state the following definition:

Definition 2. *A **Multiresolution Takagi-Sugeno Hierarchy** is an ordered sequence of fuzzy models* $\ldots F_j \subset F_{j+1} \subset \ldots$ *each one defined by (7) and where the relation* $\subset$ *is defined in terms of the existing order relation among their corresponding multiresolution spaces* $\ldots V_j \subset V_{j+1} \subset \ldots$.

For the model defined above each rule will be defined by a dilation and translation of the scaling function $\varphi(\cdot)$ and a crisp consequent. Then, the morphology of the linguistic terms involved in the modelling of the problem as well as the induced partition of the universe of discourse will be fully specified by the properties of the selected scaling function, now named mother linguistic term. Of course, the selection of an appropriate scaling function is crucial because it defines the behavior of the fuzzy model. For example the overlapping degree of $\{\varphi_{jk}(\cdot); k \in Z\}$ (fuzzy sets from the premises in the scale j) defines the degree of smooth switching between models from consequents.

Selected the mother linguistic term $\varphi(\cdot)$ the linguistic terms $\{\varphi_{j+1,k}(\cdot); k \in Z\}$ of a level $j+1$ are related to the linguistic terms $\{\varphi_{j,k}(\cdot); k \in Z\}$ of level j by the refinement equation of multiresolution analysis

$$\varphi_{j+1,l}(t) = 2 \sum_k \varphi_{j,k}(\cdot t - l) \tag{10}$$

As the set of functions on level $j+1$ is obtained by squeezing the set of functions on level j, the former have a major capacity to discriminate fine behaviors. In each dimension the number of information pieces (basis functions) is duplicated and then each segment is covered by the double of entities available for describing the behavior in this region. For this reason we say that rules implemented on level j are specialized rules of rules on the higher level $j+1$. The equation (10) is the key procedure for obtain higher level linguistic terms for refined lower level ones.

The order established in definition (2) is a complexity hierarchy for the fuzzy models F_j because each one implements a coarser or finer partition of the input space given by the resolution j, determining the structural complexity of the model. We can put this hierarchy operative defining the output of each level $y_{j,0}$ as (8) and the way in that they are agregated. The most natural way of doing it is thinking in an weighted average of the partial results $y_{j,0}$ proportional to the activation of the corresponding level generated by an input x. The total activation of a level j given a input x is $\sum_k \varphi_{j,k}(x)$ and then the total output of a hierarchy can be given by

$$y_0 = \frac{\sum_j \sum_k \varphi_{j,k}(x) c_{j,k}(x)}{\sum_j \sum_k \varphi_{j,k}(x)} \tag{11}$$

The aggregation operation (11) seems natural. However we can also follow Yager [1], [17] in the sense of establishing a relation of preference for the rules. Suppose we have the collection X of all rules corresponding to all the resolution levels. Then we can define a relation S over $X \times X$ where $S(x,y) = 1$ says that x is preferable to y. If the relation $S(x,y) = 1$ and $S(y,x) = 1$ means that x and y are equally preferable. Then the ordering generating the Multiresolution Takagi-Sugeno Hierarchy can be $S(\varphi_{i,k}, \varphi_{j,l}) = \lceil \max(i,j)/\min(i,j) \rceil$ that considers the resolutions as a rule-clustering principle, leading the rules of the same resolution in the same pseudo-equivalence class. Of course the relation $S(\varphi_{i,k}, \varphi_{j,l}) = \lceil \min(i,j)/\max(i,j) \rceil$ generates essentially the same hierarchy. For hierarchies built using this relation Yager proposes to calculate the output $\tilde{y}_{0,j}$ in the layer j and use the interlevel aggregation operator HEU defined as

$$\tilde{y}_{0,j} = HEU(\tilde{y}_{0,j-1}, y_{0,j}) = \tilde{y}_{0,j-1} + (1 - \alpha_{j-1} y_{0,j}) \tag{12}$$

Where $\tilde{y}_{0,j}$ and $y_{0,j}$ are the inter-layer aggregated and non aggregated output of layer j, respectively, while $\alpha_{j-1} = \max_k[\varphi_{j-1,k}(x)]$. In particular, if some level j obtains a membership close to one in some rule, the process of adding information slows and possible stops. Then, it is important to define where we starts to aggregate: from the high resolution levels or from the lower which is equivalent to select $S(\varphi_{i,k}, \varphi_{j,l}) = \lceil \min(i,j)/\max(i,j) \rceil$ or $S(\varphi_{i,k}, \varphi_{j,l}) = \lceil \max(i,j)/\min(i,j) \rceil$ respectively.

Up to here we are making use of the spaces V_j but not the complementary spaces W_j. Unfortunately, the wavelet functions are not suitable for using them as fuzzy sets implementing linguistic terms because in the majority of the cases they are not non-negative and have a very irregular behavior. Then we cannot think in a complementary fuzzy residual model catching the approximation details non representable in terms of the model F_j. However the wavelet spaces W_j can be useful to handle the error in switching between different scales.

3.2 Multiresolution Takagi-Sugeno System

In the proposed system the layers are chained by the antecedents from the refinement of multiresolution analysis but each one computes their output independently. Different resolution levels provide different views of the problem, more local or more global, approximating more local or global tendencies. If we think of the consequents of each rule as local experts we could say that each expert gives an opinion more specific or locally supported if it takes as a base of decision a more reduced section of the space, i.e., it works at a more fine resolution level. Then, experts working at different horizons of view (resolutions) provided useful information for solving the problem.

However, it is natural to think that the consequents $f_{j,k}(t) = c_{j,k}$ should be related in some operative way. We can chain the different resolution levels by the lifting scheme imposed by wavelet transform.The non-normalized output of the system F_i is

$$y_{0,i} = \sum_k \varphi_{i,k}(t) c_{i,k} \tag{13}$$

As this approximation to $f(t)$ is corresponding with the approximation provided by (5) in the space V_j, each consequent $c_{j,k}$ could be obtained from the system F_{j+1} by the wavelet transform as

$$c_{j,l} = 2 \sum_k h_{k-2l} c_{j+1,k} \tag{14}$$

We define now a multiresolution fuzzy rule system considering the output dependency relation defined by the last equation

Definition 3. *1. Each Takagi-Sugeno System F_i with output (13) is a **Multiresolution Takagi-Sugeno System** named root model.*

*2. Each root model F_i and a finite ordered sequence of models $F_{i+1}, \ldots F_{i+I}$ form a **Multiresolution Takagi-Sugeno System** if all the consequents of the models $F_i, \ldots F_{i+I-1}$ satisfy the lifting equation (14) and the model F_{i+I} is a typical Takagi-Sugeno model whose consequents are functions of t (ususally these functions are constants or simple parametric models of local support)*

The system of rules takes the form

$$
\begin{array}{l}
(F_i)\ \texttt{If}\ x\ \texttt{is}\ \varphi_{i,1}\ \texttt{then} \\
\quad (F_{i+1})\ \texttt{If}\ x\ \texttt{is}\ \varphi_{i+1,1}\ \texttt{then} \\
\quad\quad (F_{i+2})\ \texttt{If}\ x\ \texttt{is}\ \varphi_{i+2,1}\ \texttt{then} \\
\quad\quad\quad \cdots \\
\quad\quad\quad (F_{i+I})\ \texttt{If}\ x\ \texttt{is}\ \varphi_{I,1}\ \texttt{then}\ c_{I,1} \\
\quad\quad\quad\quad\quad\quad \texttt{If}\ x\ \texttt{is}\ \varphi_{I,2}\ \texttt{then}\ c_{I,2} \\
\quad\quad\quad \cdots \\
\quad\quad\quad\quad\quad \texttt{If}\ x\ \texttt{is}\ \varphi_{i+2,2}\ \texttt{then} \\
\quad\quad\quad \cdots \\
\quad\quad \cdots
\end{array}
\tag{15}
$$

The action of the lifting scheme leading the consequents $c_{j,k}$ from the consequents $c_{j+1,k}$ can be viewed as a (lifting) aggregation operator that can be defined directly from the coefficients h_l of the scaling function

$$c_{j,l} = 2\sum_k h_{k-2l} c_{j+1,k} \tag{16}$$

Alternatively we can define an operator $\bigoplus$ that projects the output $y_{0,j+1}$ of the rule system F_{j+1} over the corresponding rule system F_j. From the properties of the scaling function we have

$$
\begin{gathered}
c_{j,l} = 2\sum_k \varphi_{j+1}(t-k) \left\langle \varphi_j(t-l), \sum_k c_{j+1,k}\varphi_{j+1}(t-k) / \sum_k \varphi_{j+1}(t-k) \right\rangle \\
c_{j,l} = \bigoplus_{j,l} y_{0,j+1}
\end{gathered}
\tag{17}
$$

3.3 Locally Refined Systems

A problem with the proposed Multiresolution Fuzzy Rule Systems is that F_{j+1} duplicates the number of rules of the system F_j and if we want to use it in a learning scheme we expect an explosive growing of rules while the system is refined. Fortunately, the aggregation operation (16) that defines the dependence of a rule in R_k^j from rules in $j+1$ has a non-null effect only over the support of the linguistic term $\varphi_{j,k}(t)$. If it had a compact support we can restrict the dependence over a finite number of linguistic terms in the finer resolution level. The family of n-th order B-Splines for example, is $n+1$-compact and the dependence of the rule R_k^j is limited to n+2 rules of the finer system F_j.

This motivates the definition of locally refined systems, where only a few refined rules of the level $j+1$ are activated (and built) for implementing some rules of the level j that depend of them. We will consider subsets of rules $\tilde{F}_i^K$ of F_i selecting only some set of shifts K of the mother linguistic term $\varphi_{i,0}$ of F_i.

Definition 4. *A **Partial Multiresolution Takagi-Sugeno Hierarchy** is an ordered sequence of subsets of rules* $\ldots \tilde{F}_j^K \subset \tilde{F}_{j+1}^K \subset \ldots$ *where* $\tilde{F}_i^K$ *is built selecting from* F_i *only some set of shifts* K *of the mother linguistic term* $\varphi_{i,0}$*. The relation* $\subset$ *is defined in terms of the existing order relation among their corresponding multiresolution systems* $\ldots F_j \subset F_{j+1} \subset \ldots$*.*

The last hierarchy of systems can be made operative by defining the same outputs and aggregation strategies considered for the (Global) Takagi-Sugeno Hierarchy with the restriction that we take only the rules indexed by shifts k contained in the corresponding set K_i. Now we define the partial refined version of the Multiresolution Takagi-Sugeno System, where the layers are linked also by the consequents.

Definition 5. *Each Subset of Rules* $\tilde{F}_i^K$ *of a Takagi-Sugeno System* F_i *corresponding with* V_i *is a **Partial Multiresolution Takagi-Sugeno System** on the level* i *if*

1. *There exist two disjoint subsets of* K*,* K_1 *and* K_2 *where each existing consequent* $c_{i,k}$*,* $k \in K_1$ *is a simple function of the input, such a constant or an* n*-th order polynomial and each* $c_{i,k}$*,* $k \in K_2$ *is obtained from a Partial Multiresolution Takagi-Sugeno System on the level* $i+1$ *by the lifting equation (14).*
2. *The number or embedded Partial Multiresolution Takagi-Sugeno Systems* $\tilde{F}_{i+1}^{K_{i+1}}, \tilde{F}_{i+2}^{K_{i+2}}, \ldots$ *is finite.*
3. *Each rule contained in the embedded Partial Multiresolution Takagi-Sugeno Systems* $\tilde{F}_{i+j}^{K_{i+j}}$*,* $j = 1, 2, \ldots I$ *have consequents required by a rule of* $\tilde{F}_{i+j-1}^{K_{i+j-1}}$ *for implementing its consequent* $c_{i+j-1,k}$*,* $k \in K_{i+j-1,2}$*. In other words each active consequent in* $\tilde{F}_{i+j}$ $j = 1, 2, \ldots I$ *participates in the refinement procedures.*

Then, we only implement refined linguistic terms in $\tilde{F}_{j+1}$ if are needed for implementing consequents in the system $\tilde{F}_j$. The consequents $c_{i,k}$, $k \in K_1$ are called simple consequents and $c_{i,k}$, $k \in K_2$ multiresolution consequents. The last level $\tilde{F}_I$ of an embedded structure of Partial Multiresolution Takagi-Sugeno Systems, containing only simple consequents is called terminal level and the first $\tilde{F}_i$ is named root level if it computes

$$y_0^i = \frac{\sum_{k \in K} \varphi_{i,k}(t) c_{i,k}}{\sum_i \varphi_{i,k}(t)} \tag{18}$$

The other levels, all having at least one multiresolution consequent, are called internal levels.

3.4 Linguistic Splines

A well known family of scaling functions suitable for interpretation as fuzzy sets are the B-splines introduced in [12] jointly with the cardinal spline interpolation theory. The zeroth-order B-Spline is the box function $\beta^0(t) = \chi_{[0,1]}$ wich is a usual first choice when designing linguistic terms and generates the Haar-Multiresolution analysis. The B-splines of higher order can be defined by a repetitive convolution of the zeroth-order B-spline $\beta^0(t)$

$$\beta^n(t) = \beta^0 * \overset{n+1}{\ldots} * \beta^0(t) \tag{19}$$

The family $\{\beta^n(x-k)\}_{k\in Z}$ has many interesting properties [12], [13]: for example, they are compactly supported. In any interval, only $n+1$ translations of $\beta^n(t)$ are not null. They can also be normalized to guarantee that the supremum is 1 or that the system $\{\beta^n(x-k)\}_{k\in Z}$ is a partition of the unity.

The B-Splines family generate the n-th order multiresolution analysis $\{V_j^n\}_{j\in Z}$ where each V_j^n is defined as the closed span of $\beta^n(t)$. Since the B-Splines forms a basis for the splines, the linguistic variables implemented by B-Spline linguistic terms are called Linguistic Splines [11]. For this family the refinement equation takes the form [7].

$$\beta^n(t) = 2^{-n} \sum_{k=0}^{n+1} (-1)^k \binom{n}{k} \beta^n(2t-k) \tag{20}$$

3.5 Semantic Interpretation of Global and Partial Refinements

In Multiresolution Fuzzy Rule Systems each fuzzy set implementing a linguistic term in a rule is generated by squeezing a basic linguistic term. If the original morphology have a meaning it seem natural to associate to the refined terms a meaning related to the original semantics. For this meaning deduction procedure we will use the term metasemantics [11]. A study of metasemantic operators to relate the meaning of refined linguistic terms from non-refined ones is presented in [11] for hat functions. Now we generalize the results for B-Splines of any order.

The key operators are the linguistic modifier [18] "`very`" denoted $v(\cdot)$ and the metasemantic operator "`between`" denoted $b(\cdot,\cdot)$. As proposed by Zadeh [18], $v(\cdot)$ takes a fuzzy set implementing a linguistic term named "`w`" and build the linguistic term "`very w`" implemented by a fuzzy set with the half of the support and the same core. The operator $b(\cdot,\cdot)$) for two morphologically equivalent fuzzy sets build a morphologically equivalent fuzzy set centered at the middle point between the original terms i.e., for two fuzzy sets $T(t-\alpha)$ and $T(t-\beta)$, $\alpha \leq \beta$ its effect is $b(T(t-\alpha), T(t-\beta)) = T(t-(\alpha+\beta)/2)$.

In the case of general spline linguistic variables the same metasemantic structure can be applied. The splines linguistic variables $\{\beta^n_{jk}(t); k \in C \subset Z\}_j$ of odd order have linguistic terms centered at the points $C^o_j = \{p_k = k/2^j; k \in Z\}$, j denoting the resolution level, while the even order are centered at points $C^e_j = \{p_k = 2k + 1/2^{j+1}; k \in Z\}$. When we refine all the linguistic terms on a dimension of level j we obtain new terms on level j which have the half of the original support and cover more densely the space of the linguistic variable. In the case of odd order B-Splines if on the first scale we have a term centered at $c \in C^o_j$ corresponding to a shift k, on the new scale we always obtain a new term centered at c that correspond to the shift $2k$ in C^o_{j+1} and additional terms appearing centered at the middle points between points of C^o_j, corresponding to shifts $2k+1$ or $2k-1$ in C^o_{j+1}. Then if the original terms $\beta^n_j(t-k+1)$, $\beta^n_j(t-k)$ and $\beta^n_j(t-k-1)$ have labels "`u`", "`v`" and "`w`" respectively, the new terms $\beta^n_{j+1}(t-2k+1)$ $\beta^n_{j+1}(t-2k)$ and $\beta^n_{j+1}(t-2k)$ can be interpreted as "`between very u and very v`","`very v`", and "`between very v and very w`" respectively.

In the case of even order B-Splines the core points of j are not core points in the refined level $j+1$ because the structure of center points C^o_j is generated by odd multiplies of $1/2^{j+1}$ that cannot be simplified to other odd factor by the denominators. Then we build a modified set of functions $\tilde{\beta}_{j+1,k}(t) = \beta_{j+1,k}(t-1/2)$ which have core points at $k/2^{j+1}, k \in Z$. When k is an odd integer the last core points match with the core points of the family $\beta_{j,k}(t)$ and when k is a even integer the corresponding linguistic terms are centered between the first ones. Then we can apply the metasemantics applied in the case of odd order for interpreting the meaning of $\tilde{\beta}_{j+1,k}(t)$ in terms of the labels of $\tilde{\beta}_{j,k}(t)$. If the terms $\tilde{\beta}^n_j(t-k+1)$, $\tilde{\beta}^n_j(t-k)$ and $\tilde{\beta}^n_j(t-k-1)$ have labels "`u`", "`v`" and "`w`" respectively, the new terms $\beta^n_{j+1}(t-2k+1)$ $\beta^n_{j+1}(t-2k)$ and $\beta^n_{j+1}(t-2k)$ can be interpreted as "`u'`" = "`between very u and very v`", "`very v`", and "`v'`" = `between very v and very w` respectively. Finally the terms $\beta^n_{j+1}(t-2k)$ and $\beta^n_{j+1}(t-2k)$ can be interpreted from the modified family $\tilde{\beta}_{j+1,k}(t)$ as "`between u' and very v`" and "`between very v and w'`" respectively.

If we work with finite domain linguistic variables we must scale data to an interval $[a, b]$ covered with an appropriate number of initial linguistic terms belonging to the initial model F_i. The only precaution that we take is that the right-last linguistic term is modelled with the left part of the B-spline that we are considering

$$\hat{\beta}^{n,left}_i(t) = \begin{cases} \beta^n_i(t-a) & t \geq a \\ 1 & \texttt{otherwise} \end{cases} \tag{21}$$

and similarly with the left-last term. Now, suppose they have labels "`u`" and "`v`" respectively. If the considered B-Spline is of odd order in the next scale

$i+1$ the refined extreme terms can be interpreted conventionally from the labels of the original final terms as "`very u`" and "`very v`" respectively. It seems obvious that the interpretation of the central terms remains the same. If the linguistic terms correspond to B-Splines of even order the refinement procedure do not lead to linguistic terms centered in the extreme points of the interval. We can take as left extreme term the last term whose right half is completely contained in the interval and extend its core to the left extreme of the interval. Similarly for the right extreme. It seems natural to interprete the new linguistic terms as "`very u`" and "`very v`" respectively because although the core is extended, it is made in the direction of increasing membership of the high level linguistic terms with labels "`u`" and "`v`".

The subsequent refinements from j to $j+1$ affecting the extreme linguistic terms are made from the non-corrected terms of the superior level j as if the family of functions would exist beyond the limits of the interval $[a, b]$.

In our approach the refined rules work independently or they are created to implement the consequents of the original high level rules. When we partially refine a system $\tilde{F}_j^K$ we generate all the linguistic terms needed for covering the support of the original terms. We propose interpreting this new linguistic terms with the metasemantic operators "`very`" and "`between`" as if the refinement were global although some linguistic terms corresponding to F_j do not exist in F_j^K, that is, they are not activated for operating.

If we can obtain a semantic for the linguistic terms on a scale, the metasemantic procedures exposed above works very well to obtain the meaning of refined terms of the next scale. Semantic problems can appear however when the refinements involves several scales. In this case the predicates associated to the new refined linguistic terms can loose sense because they put linguistic modifiers over already modified predicates and so on. Even in these cases the refinement procedures can be interpreted as an explanation of the decisions that the system F_j are taking. We say that a reasoning rule of F_j can conclude something if specialized neighbors rules support this decision concluding something else related to the original conclusion by the aggregation procedure (17). The metasemantic operators working on abstract labels associated to the linguistic terms involved in the refinement work for explaining how the system takes decisions of increasing complexity.

Finally we must say that when the data is purely numerical the semantic problems can be relaxed associating to each linguistic term centered at c the predicate "`around c`". Then, metasemantic operators can explain what means the refinements from the semantic point of view when passing from one scale to another.

4 Remarks

A family of models called Multiresolution Fuzzy Rule Systems was proposed for handling the problem of granularization of the input space. Interesting

models result allowing local refinement of rules for handling the problem of explosive growing of rules when the space is multidimensional. When doing this we have preserved the interpretability of the model. The structure can be viewed as a strategy for hierarchically mixing experts, probably heterogenous. Each expert provides different views of the problem, more local or more global, approximating more local or global tendencies depending of the support of the linguistic term associated to it. The structure of learning algorithms for building globally and partially refined systems has been studied beyond the scope of this paper.

References

1. L. Valverde B. Bouchon-Meunier and R. Yager (eds.), *Hierarchical representation of fuzzy if-then rules*, Springer, 1993.
2. J.J. Buckley, *Universal fuzzy controllers*, Automatica 28 (1977), 1245-1248.
3. I. Daubechies, *Orthonormal bases of compactly supported wavelets*, Communications on Pure and Applied Mathematics 41 (1988), 909-996.
4. A. de Soto, *Building a hierarchical set of fuzzy rules using wavelet theory*, Inter. Conference on Information processing and Management of Uncertainty Knowledge -Based Systems 3 (2000), 1764-1769.
5. ———, *Uniform fuzzy partitions with cardinal splines and wavelts: Getting interpretable linguistic fuzzy models*, Inter. Conference on Information processing and Management of Uncertainty Knowledge -Based Systems 3 (2002), 1-20.
6. R. Fullr, Introduction to neuro-fuzzy systems, Advances in Soft Computing, Springer-Verlag, 2000.
7. B. Jawerth and W. Sweldens, *An overview of wavelet based multiresolution analyses*, SIAM Rev. 36 (1994), no. 3, 377-412.
8. B. Kosko, *Fuzzy systems as universal approximators*, IEEE Trans. Computers 43 (1994), no. 11, 1329-1333.
9. G. Kang M. Sugeno, *Structure identification of fuzzy model*, Fuzzy Sets and Systems 28 (1988), 15-33.
10. S. Mallat, *Multifrequency channel decompositions of images and wavelet models*, IEEE Trans. Acoust. Speech Signal Process. 37 (1989), 2091-2110.
11. C. Moraga, *A metasemantics to refine if-then rules*, Proc. 34th Int. IEEE Symposium on Multiple-valued Logic (2004).
12. I.J. Schoenberg, *Contribution to the problem of approximation of equidistant data by analytic functions*, Quart. Appl. Math. 4 (1946), 4599, 112-141.
13. ———, *Cardinal spline interpolation*, Soc. for Industr. Appl. Math. (1973).
14. G. Strang, *Wavelets and dilation equations: a brief introduction*, SIAM Rev. 31(1989), no. 4, 614-627.
15. M. Sugeno T. Takagi, *Fuzzy identification of systems and its application to modelling and control*, IEEE Trans. System, Man and Cybernetics 15 (1985), no. 1, 116-132.
16. S. Tan Y. Yu, *Complementary and equivalence relationships between convex fuzzy systems with symmetry restrictions and wavelets*, Fuzzy Sets and Systems 101 (1999), 423-438.

17. R.R. Yager, *On the construction of hierarchical fuzzy systems models*, IEEE Transactions on Systems, Man and Cybernetics-PARTC:Applications and Reviews 28 (1998), no. 1, 1245-1248.
18. L. Zadeh, *Outline of a new approach to the analysis of complex system and desision process*, IEEE Trans. Systems, Man and Cybernetics 1 (1972), no. 1, 28-44.

Data Characterization through Fuzzy Clustering

Session Organiser:
Bernadette Bouchon-Meunier

Fuzzy Clustering of Macroarray Data

Olga Georgieva[1], Frank Klawonn[2], and Elizabeth Härtig[3]

[1] Institute of Control and System Research, Bulgarian Academy of Sciences
P.O. Box 79, 1113 Sofia, Bulgaria
ogeorgieva@icsr.bas.bg
[2] Department of Computer Science, University of Applied Sciences
Braunschweig/Wolfenbuettel, Salzdahlumer Str. 46/48, D-38302 Wolfenbuettel, Germany
f.klawonn@fh-wolfenbuettel.de
[3] Institute for Microbiology, Technical University of Braunschweig, Spielmannstr. 7, D-38106 Braunschweig, Germany
e.haertig@tu-bs.de

1 Introduction

The complete sequence of bacterial genomes provides new perspectives for the study of gene expression and gene function. DNA array experiments allow measuring the expression levels for all genes of an organism in a single hybridization experiment.

Computational analysis of the macroarray data is used extensively to extract groups of similarly expressed genes. The aim is to organize DNA array data so that the underlying structures can be recognized and explored. The gene groups identified as clusters are searched for genes known to be involved in similar biological processes, implying that genes of unknown functions may be involved in the same processes. Commonly used computational techniques include hierarchical clustering, K-means clustering and self-organizing maps. These share many features, particularly the distance metric, which measures the relationship between samples or genes in the data space formed by the expression values. The output from different clustering algorithms usually depends more on the type of distance metric used than on any other factor.

The central limitation of most of the commonly used algorithms is that they are unable to identify genes whose expression is similar to multiple, distinct gene groups, thereby masking the relationships between genes that are coregulated with different groups of genes in response to different conditions. For example, the K-means clustering partitions genes into a defined set of discrete clusters, attempting to maximize the expression similarity of the genes in each cluster assigning each gene to only one cluster, obscuring the relationship between the conditionally coregulated genes. The recently implemented heuristic variant of Fuzzy C-Means (FCM) clustering [4] shows the advantage

O. Georgieva et al.: *Fuzzy Clustering of Macroarray Data*, Advances in Soft Computing **2**, 83–94 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

of the fuzzy clustering technique as a valuable tool for gene expression analysis as it presents overlapping clusters, pointing to distinct features of each gene's function and regulation.

The paper presents a methodology to deal with macroarray data analysis. Each step of the data processing is described in detail. First, the crucial preprocessing step on the raw data to transform the data set into an appropriate and reliable form for clustering is applied. Secondly, subtractive clustering is implemented in order to obtain a good initial data partition. The obtained cluster centres are used to initialize the fuzzy C-means clustering algorithm by which the optimal values of the cluster centres and partition matrix are obtained. The partition matrix is used to determine the gene groups that belong to a given cluster with a prescribed membership degree. The proposed procedure is applied to macroarray data of *B. subtilis*. The obtained results show that informative clusters are obtained. The extracted gene clusters are overlapping, pointing to distinct aspects of the gene function and regulation.

2 Data Set

Here we used data obtained from the soil bacterium *B. subtilis* grown under different growth conditions. *B. subtilis* is able to grow in the absence of oxygen using nitrate as alternative electron acceptor or fermentation processes. The switch between aerobic and anaerobic metabolism in *B. subtilis* is regulated mainly at the transcriptional level. To investigate the global changes in gene expression under various anaerobic conditions we used DNA macroarrays containing DNA fragments of all 4107 genes of *B. subtilis*. We analysed mRNA from cells grown aerobic, anaerobic with nitrate, anaerobic with nitrite and under fermentative conditions. When the mRNA levels were compared during exponential growth, several hundred genes were observed to be induced or repressed under the various conditions tested. The data of the macroarrays are obtained by autoradiography using phosphorimaging and the intensities representing the expression levels are transformed into a table. In the obtained numerical table each row corresponds to a gene and each column to one growth condition analyzed.

The considered macroarray data set is a data matrix that consists of the expression levels (ratios) of 4107 genes of *B. subtilis* defining each matrix raw. The cells have been carried out on four different environment condition namely aerobic (A), fermentative (B), nitrite (C) and nitrate (D). The intensities obtained in each experiment are organized in the columns. The ratio in each environment condition has been measured twice. Thus, the original macroarray matrix $Z = (z_{kj})(k = 1, \ldots, N, j = 1, \ldots, n)$ is a large sized data matrix with $N = 4107$ rows and $n = 8$ columns.

Two distinct data sets of the type described above have been obtained in different time instants. The first data set, named identification data set, was used for data partition and determination of the searched gene groups. The second one, named the validation data set, was used only for verification of the obtained clusters.

3 Fuzzy Clustering of *B. subtilis* Macroarray Data

In order to identify the gene clusters and simultaneously to evaluate the level of relationships of genes that are coregulated with different groups, the Fuzzy C-Means clustering technique [2, 5] is applied. It is an objective function-based clustering method. It aims at minimizing an objective function that indicates a kind of fitting error of the clusters to the data. The underlying objective function for most of the clustering algorithms is:

$$J = \sum_{i=1}^{c} \sum_{k=1}^{N} (u_{ik})^m d_{ik}^2 \tag{1}$$

where N is the number of data points; c is the number of clusters; u_{ik} and d_{ik} denote the membership degree and the distance of the point $x_k, k = 1, \ldots, N$, to the i-th cluster prototype (centre), $i = 1, \ldots, c$, respectively, $m \in [1, \infty)$ is the weighted exponent coefficient (fuzzifier) which determines how much clusters may overlap. In order to avoid the trivial solution assigning no data to any cluster, i.e. setting all u_{ik} to zero, and to avoid empty clusters, the constraints

$$u_{ik} \in [0, 1], \qquad 1 \leq i \leq c, \quad l \leq k \leq N \tag{2}$$

$$\sum_{i=1}^{c} u_{ik} = 1, \qquad 1 \leq k \leq N \tag{3}$$

$$0 < \sum_{k=1}^{N} u_{ik} < N, \qquad 1 \leq i \leq c \tag{4}$$

are introduced. When the fuzzifier value $m = 1$ is chosen, then $u_{ik} \in \{0, 1\}$ will hold at a minimum of the objective function (1), i.e. the resulting partition will be crisp.

The parameters to be optimized are the membership degrees u_{ik} and the cluster parameters which finally determine the distance values d_{ik}. In the simplest case, a single vector named cluster centre represents each cluster v_i. For FCM clustering the distance of a data point to a cluster is simply the Euclidean distance between the cluster centre v_{ik} and the corresponding data point:

$$d_{ik}^2 = \|x_k - \upsilon_i\|^2 = (x_k - \upsilon_i)^\top (x_k - \upsilon_i) \,, \tag{5}$$

where $x_k = [x_{k1}, \ldots, x_{kn}]$ is the k-th data point defined as a vector in the feature space and $\upsilon_i = [x_{i1}, \ldots, x_{in}]$ is the i-th cluster prototype vector.

The minimization of the functional (1) represents a nonlinear optimization problem that is usually solved by means of Lagrange multipliers, applying an alternating optimization scheme [2]. This optimization scheme alternatively considers one of the parameter sets, either the membership degrees or the cluster parameters as fixed, while the other parameter set is optimized, until the algorithm finally converges.

The main problem of the macroarray data clustering is the badly structured data space that lacks well separated (distinguished) data groups. Most of the points are found to be located in one small area close to the zero value. Thus, two important problems arise in applying the FCM clustering algorithm. The first one arises from the specific characteristics of the data set mentioned above. The data set should be presented in an appropriate form in order to guarantee the reliability and authenticity of the extracted information. The second problem concerns the initialization of the FCM algorithm. In the objective function (1), the number of clusters has to be fixed in advance. Since the number of clusters is usually unknown, an additional scheme has to be applied to determine the number of clusters and their prototypes. Both problems are considered in detail and an effective solution for the macroarray data set is described.

3.1 Preprocessing Step

The preprocessing step is a preliminary step of the whole data processing (clustering) which aimed to transform the data set into an appropriate and reliable form for clustering. In most situations, the data obtained after scanning must be transformed and normalized before they can be analyzed. By applying several operations on the identification data set the data representation is improved and the quality of the clustering is increased.

One main problem of the macroarray data processing is connected to the data noise. Differences occurred quite frequently in the repeated measurements of the same gene under the same conditions. For this, an average value of the gene expression within every environment condition is determined. In case one of the expressions is zero or invisible (no measured value is provided) the other one is taken into account. Thus, in the considered particular case the data matrix is transformed from an eight to a four column matrix. This in practice means that the clusters will be searched for in a four-dimensional data space.

Since empirical evidence shows that low-intensity spots (low gene expression measurements) are more likely to be noisy, eliminating those spots is a fairly safe option [6]. However, as the most of the data values are close to the zero value it is rather difficult to distinguish the noisy values from the

informative ones. That is why only genes that have one or more zero average expression value are taken out from the data set. Eleven genes having a zero average value and by that three of them at two environment conditions were defined and removed from the considered data set.

The second specificity of the illuminated macroarray matrix is that there are so bright spots that some of the surrounding ones could not be distinguished. Their expression levels can not be measured and they are considered as invisible. Genes that have at least one invisible average expression should also be removed from the data set. However, in the particular case under consideration it two invisible expressions for one gene in one condition never occurred.

The upper extreme of the filtered group consists of genes that have large expression that differ drastically from the rest of the data. Normally, they form a small group that is far away from the other data in the data space. In this situation every clustering algorithm will separate both the large and small group of outliers, but this partition will be not informative as the important clusters are within the large amount and narrow spread data points. Genes that have an average expression of over 100 are taken out from the data set. Both cutting values for lower and upper data filtering are subjectively determined. They are subject to an expert choice depending on the concrete data configuration.

The results in many DNA macroarray experiments are ratios. The fact that the output is not symmetric, i.e. twofold change has a ratio of either 2 (up-regulation) or 0.5 (down- regulation), presents a problem for analysis because most distance metrics treat a ratio change differently depending on the direction. Thus, it is essential to convert ratio data into a form that is not sensitive to the direction of the change. The solution of this problem usually applied is a log transform of all ratio values [6]. Here, the natural logarithm was applied but all logarithm bases are equivalent for this task. By this transformation the high ratios are compressed while the small ratios increase their importance by stretching their distances. This effect is very beneficial as it increases the distinguishability among the data due to the fact that the largest amount of expression ratios lies between zero and one.

Generally, the expression ratios are obtained by a comparison to a common reference sample. For each gene the series of ratio values are relative to the expression level of that gene in the reference sample. Thus, in case the uncentred correlation matrix is used, two genes whose expression patterns are identical to each across a range of samples, but different in the reference sample, will not cluster together. Since the reference sample has nothing to do with the experiment, the expression levels should be transformed to be independent from the reference sample. This problem can be solved by subtracting the average (mean) or median log ratio for a gene from the log value for each value for that gene. Median centring the data for columns and rows is applied consequently ten times to the processed identification data set. This transformation removes certain types of biases.

The applied FCM clustering method uses the Euclidean distance metric, which imposes data normalization before clustering in order to ensure reliable clustering results.

These operations are not associative, so the order in which they are applied is very important. The series of operations for the raw data are:

1. Average of the expression values within every environment condition $Z_{N\times 8} \rightarrow Z_{N\times 4}$;
2. Filtering by removing:

$$\begin{array}{ll} z_{ki} > 100 & \text{(to high expression)} \\ z_{ki} = 0 & \text{(no expression)} \\ z_{ki} = -1 & \text{(invisible expression)} \end{array}$$

3. Log transform all values: $\log(Z_{N_1\times 4}) = (\log Z_{ki}), k = 1, \ldots, N_1, i = 1, \ldots, 4$, where N_i, is the revised number of the genes in the data set;
4. Median centring through data columns and rows consequently ten times;
5. Normalization of the data.

Figure 1 shows the data transformations carried out by the described procedure in the three (A-B-C) dimensional data space.

The data preprocessing does not change the structure of the data set. It only makes it more reliable and informative for the applied clustering algorithm.

3.2 Initialization of the Clustering Algorithm

The number of clusters and the initial partition matrix should be provided in advance for all objective function-based clustering algorithms. Since, by the clustering optimization scheme usually a local minimum is found, the provided initialization parameters are of a major importance, especially in the case of extreme large data set. An effective solution can be achieved only, if the correct number of clusters and a suitable initial data partition are defined.

The main drawback of the fuzzy clustering algorithm is the lack of an initialization procedure. There is no reliable technique for determining the number of clusters and an initial data partition. In the standard objective function-based clustering additional strategies have to be applied in order to define the number of clusters. There are two commonly applied strategies. Through the first one the clustering is carried out with different numbers of clusters and the partition is evaluated by some global validity measure like average within-cluster distance, fuzzy hypervolume or average partition density [5, 1]:

1. Average within-cluster distance (AWCD)

$$AWCD = \frac{1}{c}\sum_{i=1}^{c}\frac{\sum_{k=1}^{N} u_{ik}^{m} d_{ik}^{2}}{\sum_{k=1}^{N} u_{ik}^{m}}\ , \tag{6}$$

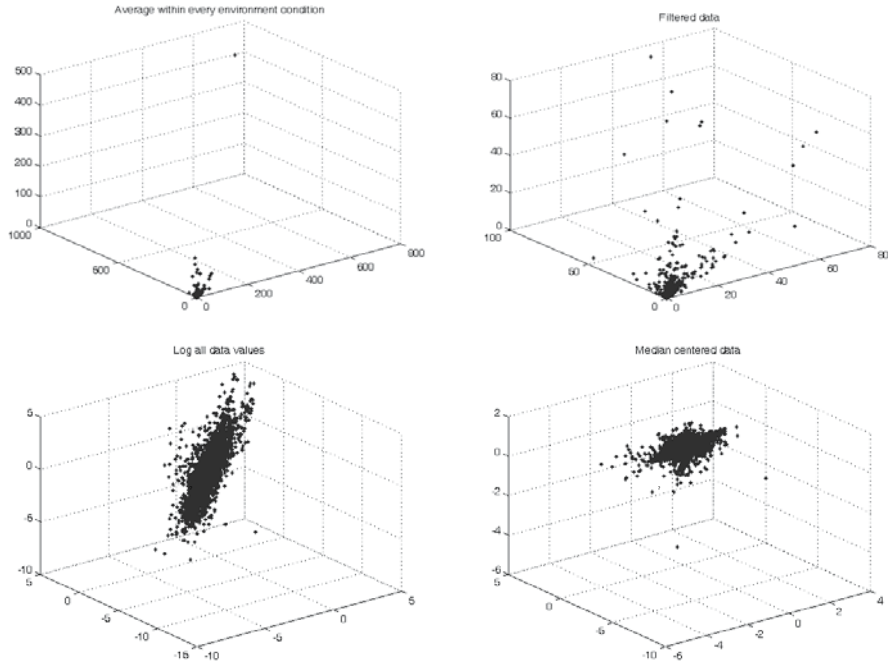


Fig. 1. Data transformations: averaging, filtering, log transform, median centred data

This measure monotonically decreases with the number of clusters c. A "knee" in the graph will indicate a suitable partition.

2. Fuzzy hypervolume (Vh)

$$Vh = \sum_{i=1}^{c} [\det(F_i)]^{\frac{1}{2}} \ , \tag{7}$$

where $F_i, i = 1, \ldots, c$, are the cluster covariance matrices. Good partitions are indicated by small values of Vh.

3. Average partition density (APD)

$$APD = \frac{1}{c} \sum_{i=1}^{c} \frac{S_i}{[\det(F_i)]^{\frac{1}{2}}} \ , \tag{8}$$

where S_i is the sum of the membership degrees of the data vectors that lie within a hyperellipsoid whose radius is the standard deviation of the cluster features:

$$S_i = \sum_{k=1}^{N} u_{ik}, \quad \text{for every k, such that} \quad (z_k - v_i) F_i^{-1} (z_k - v_i)^T < 1 \ . \tag{9}$$

Good partitions are indicated by large values of APD.

The clustering procedure is started several times for a given number of clusters and a randomly set partition matrix. The procedure is repeated for different numbers of clusters, varying from a sufficiently small to a sufficiently large amount. Best initial values are those that optimize the chosen cluster validity measures. Normally more then one cluster validity measure should be incorporated. Another strategy is based on compatible cluster merging [1] that starts with a high number of clusters and then deletes bad clusters as well as merges similar clusters together step by step. Both strategies require high computational costs.

In this paper another strategy for a good initialization of the objective function clustering has been applied. A subtractive clustering algorithm [3] is used to partition the identification data set. This algorithm determines the number of clusters and estimates their cluster centres. Then the cluster centres are used as a good initialization for the FCM clustering. As a result by the FCM the optimal cluster centres' coordinates and the partition matrix are obtained. The partition matrix is very useful as its elements are the membership degrees indicating how much each gene belongs to each cluster. So, one obtains not only a partition, but the intensity of the gene's partition. Thus, by defining some cutting level, it is easy to determine groups of those genes that belong to the clusters with desired level of membership.

There are four important parameters of the subtractive clustering algorithm that are responsible for the data partition. The cluster radius r_a is a vector that specifies a cluster centre's range of influence in each data dimension, assuming the data lie within the unit hyperbox. However, after the preprocessing step the transformed data set is fit to assuming spherical clusters, i.e. data points are bounded by a hypercube. Thus, we use the same cluster radius for all data space dimensions. A squash factor s_q is used to multiply the r_a value to determine the neighbourhood of a cluster centre within which the existence of other cluster centres are to be discouraged. The accept ratio ε_a sets the potential, as a fraction of the potential of the first cluster centre, above which another data point will be accepted as a cluster centre. The reject ratio ε_r sets the potential, as a fraction of the potential of the first cluster centre, below which a data point will be rejected as a cluster centre. As the first two parameters are more decisive to the clustering results, they are varied to obtain a good initial partition in terms of the cluster measures AWCD, Vh and APD. The remaining two factors are set to a constant value $\varepsilon_a = 0.5$ and $\varepsilon_r = 0.15$ as it is prescribed in the originally proposed algorithm [3]. The calculated cluster measures by varying first the cluster radii and constantly maintaining the squash factor and after that varying the squash factor while cluster radii have been set to a constant value, are presented in Fig. 2. It is difficult to define exactly the best values for r_a and s_q in terms of these cluster measures as there are discrepancies. We can only define an appropriate interval where the best partition is realized. The influential point of both

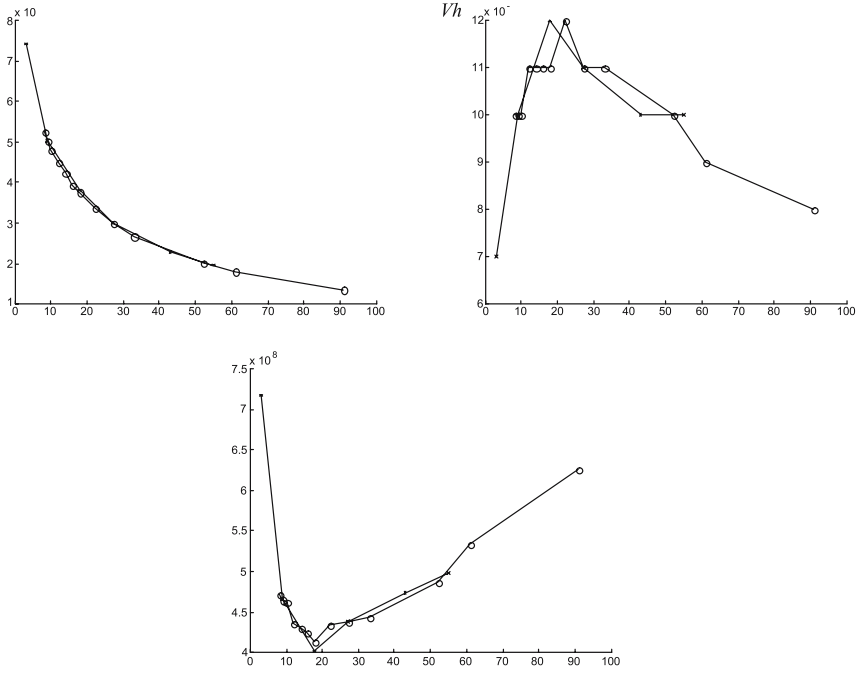


Fig. 2. Cluster measures: -o- r_a varies and $s_q = 1$; -x- $r_a = 0.05$ and s_q varies

curves in the top of Fig. 2 is defined in the interval for $c \in [20, 40]$. In the same interval for the corresponding r_a and s_q values a local minimum can be found in the curves for Vh and APD, respectively. As a reasonable compromise between the optimal values of the cluster measures and a good initial solution for FCM clustering the data partition is chosen that is obtained by the subtractive clustering with parameters $r_a = 0.05$ and $s_q = 1$.

4 Results and Discussion

By the procedure described above 27 clusters and their cluster centres are identified. They are used to calculate the initial partition matrix by applying the membership calculation formula of the FCM algorithm. By running the FCM algorithm 27 fuzzy clusters are identified such that genes having similar profiles are assigned to a cluster with highest membership degrees. The degree of membership of each gene to each cluster is given in the partition matrix. It is used to determine the clusters by cutting the membership degrees on a desired level.

The obtained results show that informative clusters are obtained. The extracted gene clusters are overlapping, which enables us to reveal distinct aspects of the gene function and regulation (Table 1). The number of the

Table 1. Number of clustered genes for different membership level cut

Cutting level of the membership degree	0.6	0.5	0.4	0.3	0.2	0.1
Number of all clustered genes	274	447	720	1180	2361	6589

clustered genes for the 0.1 cutting level is rather bigger then the number of the clustered genes. This means that a large overlapping of the obtained clusters occurs. By increasing the cutting level the number of the clustered genes decreases.

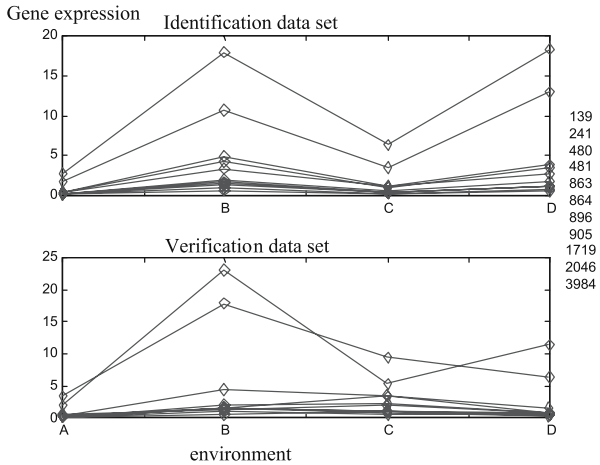


Fig. 3. Profiles of genes (given by numbers) belonging to cluster no. 17

Some typical gene clusters are presented in Figures 3–5. The expression profiles belong to genes that are assigned to the given cluster with a membership degree greater than 0.5 (upper curves in the figures). In order to verify the obtained partition the verification data set is also used. The expression profiles of the genes belonging to this cluster but extracted from the verification data set are shown in the bottom of the figures. The gene expression is changed during the time. This means that it could be expected that the profiles in both data sets could be different. The fact that the extracted genes from the identification data set are still a group with similar profiles in the verification data set proofs the reliability of the obtained partition.

Cluster 17, 26 shows the typical expression pattern: A-low, B-high, C-low and D-high and a lot of genes known to be important for anaerobic life are represented in this cluster. The genes of this cluster are almost not expressed under the A and C conditions (aerobic and anaerobic with nitrate) whereas under the conditions B and D (fermentative and anaerobic with nitrite) the expression of the genes is induced. This expression pattern shows that the

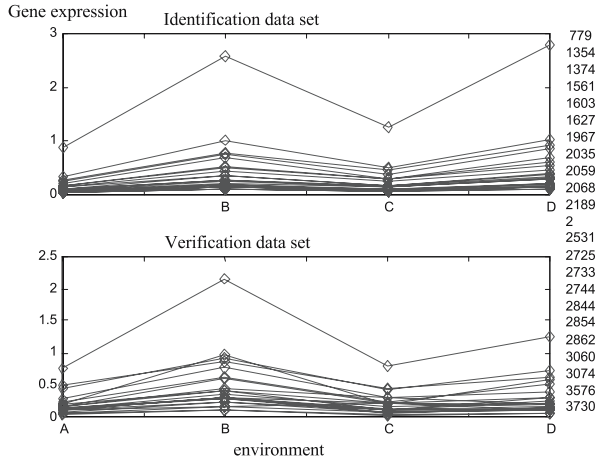


Fig. 4. Profiles of genes (given by numbers) belonging to cluster no. 26

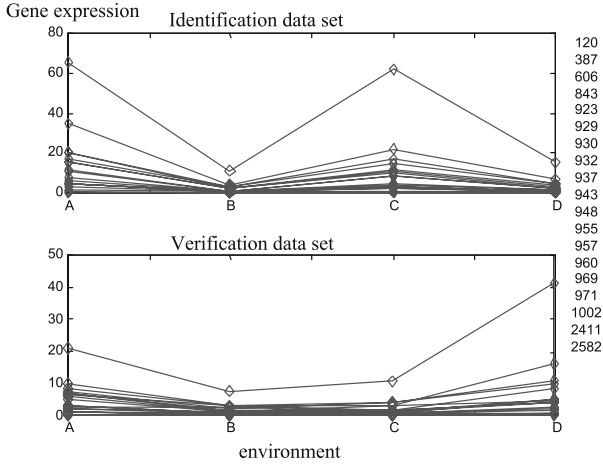


Fig. 5. Profiles of genes (given by numbers) belonging to cluster no. 19

condition A and B are more similar to each other than the anaerobic conditions C and D. Under condition A and B the electron transport chain is needed and different electron endacceptors are used: Oxygen in the case of condition A and nitrate at condition B.

Also in cluster 17 we found a lot of genes with unknown functions. Since the expression pattern is similar, we may postulate that they are also needed for anaerobic adaptation. This has to be analyzed in further experiments.

The opposite expression pattern is visible in cluster 19 (A-high, B-low, C-high, D-low), indicating that now genes are expressed that are mainly

needed for conditions A and C. Here we found a lot of genes encoding ribosomal proteins.

5 Conclusions

In this paper, we have demonstrated how fuzzy clustering can be applied to the analysis of DNA array data using *B. subtilis* as an example. Since the raw data are not suitable for clustering, especially since they contain a high proportion of noise, appropriate transformations have to applied to the data in a preprocessing step. Furthermore, we used a subtractive clustering strategy to determine the number of clusters and to find a good initialization for the fuzzy clustering step. We have used two data sets that were generated independently, but under the same conditions. We have carried out the cluster analysis only in the first data set, whereas the second data set served for verification of the clusters. Assigning the data from the verification set to the clusters derived from the first data set has shown that the clusters group genes together with similar expression characteristics despite the inherent noise.

Taking a closer look at the clusters, we could verify some known correlations between gene expression as well as find some new interesting groups of genes with unknown function which show a similar expression pattern. This might provide some further hints to their function which has to be analysed in further biological experiments.

References

1. R. Babuska. *Fuzzy Modeling for Control*. Kluwer Academic Publishers, Boston, 1998.
2. J.C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.
3. S.L. Chiu. Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems*, 2:267–278, 1994.
4. M.B. Gasch, A.P. and Eisen. Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. *Genome Biology*, 3(ll):l–22, 2002.
5. F. Höppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy Cluster Analysis*. Wiley, Chichester, 1999.
6. P.T. Spellman. Cluster analysis and display. In D. Bowtell and J. Sambrook, editors, *DNA Microarrays*, pages 569-581. Cold Spring Harbor Laboratory, Cold Spring Harbor, 2002.

Fuzzy Clustering: Consistency of Entropy Regularization

Hichem Sahbi[1,2] and Nozha Boujemaa[2]

[1] Fraunhofer-IPSI (GMD institute), Dolivostr. 15, 64293 Darmstadt, Germany.
[2] IMEDIA Research Group (INRIA), BP 105–78153. Le Chesnay Cedex, France.
{hichem.sahbi,nozha.boujemaa}@inria.fr

Abstract. We introduce in this paper a new formulation of the regularized fuzzy c-means (FCM) algorithm which allows us to set automatically the actual number of clusters. The approach is based on the minimization of an objective function which mixes, via a particular parameter, a classical FCM term and an entropy regularizer. The method uses a new exponential form of the fuzzy memberships which ensures the consistency of their bounds and makes it possible to interpret the mixing parameter as the variance (or scale) of the clusters. This variance closely related to the number of clusters, provides us with a more intuitive and an easy to set parameter.

We will discuss the proposed approach from the regularization point-of-view and we will demonstrate its validity both analytically and experimentally. We conducted preliminary experiments both on simple toy examples as well as challenging image segmentation problems.

1 Introduction

Define a training set $\{x_1, \ldots, x_N\}$ being for instance images in database categorization or colors in image segmentation. A clustering algorithm finds a function which assigns each training example x_i to one class, resulting into a final partition of C subsets. Basically, a simple hierarchical agglomeration algorithm can do this task [1, 2, 3], but sometimes the decision as whether a training example belongs to one or another cluster can be *fuzzy* and a family of algorithms dealing with fuzziness exist in the literature [4, 5, 6].

The main issue in the existing clustering methods remains setting the appropriate number of classes for a given problem. The well-studied fuzzy c-mean (FCM) algorithm [4] has proven to perform well when the application allows us to know a priori the number of clusters or when the user sets it manually. Of course, the estimation of this number is application-dependent, for instance in image segmentation it can be set a priori to the number of targeted regions.

H. Sahbi and N. Boujemaa: *Fuzzy Clustering: Consistency of Entropy Regularization*, Advances in Soft Computing **2**, 95–107 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

Unfortunately, for some applications such as database categorization, it is not always possible to predict automatically and even manually the appropriate number of classes.

Several methods exist in the literature in order to set automatically the number of classes for clustering; among them the well studied competitive agglomeration (CA) [5] and recently a new original approach based on kernel methods [7]. The former attempts to set automatically the relevant number of classes using *regularization*. The underlying objective function usually involves an FCM term which measures the fidelity of the data to the clusters and a regularization term which reduces the number of classes, i.e., the complexity. Solving such a problem implies finding the membership degrees of each training example to the different classes and assigning the former to the class which maximizes its membership. Nevertheless, the constraints on the bounds and the consistency of the membership degrees are difficult to carry. In this work, we introduce a new simple formulation which guarantees the consistency of the membership degrees and provides a solid connection and interpretation in terms of regularization.

In the remainder of this paper i stands for data indices while k, c stand respectively for a particular and a given cluster indices. Other notations will be introduced as we go along through different sections of this paper which is organized as following: in §2 we review the basic formulation of the regularized FCM while in §3 we introduce our *entropy* regularizer. In sections 4, 5, we discuss the consistency of our solution, the technical issues, limitations and in §6 we show the experimental validity of the method for 2D toy examples as well as for image segmentation. We conclude in §7 and we provide some directions for a future work.

2 A short reminder on regularized FCM

A variant of the regularized version of FCM [4] consists in the minimization problem:

$$\mathcal{J}(C,\mu) = \sum_{k=1}^{C}\sum_{i=1}^{N} \mu_{ik}\ d_{ik}^2 + \alpha(t)\ \mathcal{R}(C,\mu) \tag{1}$$

under the constraints that $\mu = \{\mu_{i1}, \ldots, \mu_{iC}\}$ is the probability distribution of the fuzzy memberships of x_i to C clusters and d_{ik} is the distance of x_i to the k^{th} cluster. Let X and Y be two random variables standing respectively for the training examples and their different possible classes $\{Y_1, \ldots, Y_C\}$. A membership degree μ_{ik} can be expressed as:

$$\mu_{ik} = P(Y = Y_k / X = x_i) \tag{2}$$

which leads to $\sum_k \mu_{ik} = 1$ since:

$$\sum_k \mu_{ik} = \sum_k P(Y = Y_k / X = x_i) = \frac{\sum_k P(Y = Y_k, X = x_i)}{P(X = x_i)} = 1 \qquad (3)$$

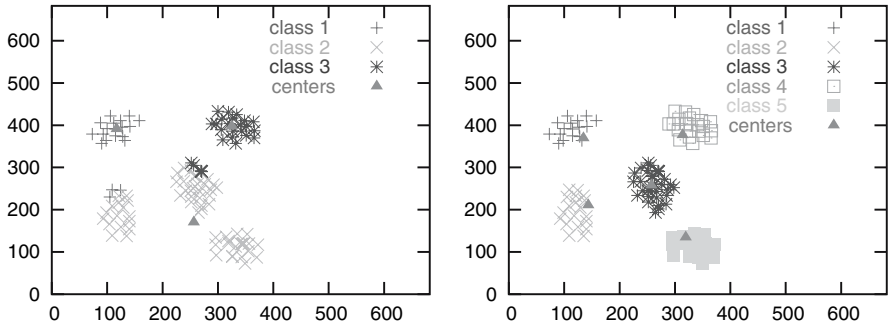


Fig. 1. *FCM clustering (i.e., $\alpha = 0$). The number of clusters C is set manually and respectively to* 3 *and* 5. *It is clear that the distribution of the data is not well captured by* 3 *centers but well captured with* 5.

The first term of the objective function (1), referred to as the FCM term, measures the fidelity of each training example to its cluster and vanishes when each example is a cluster by it self. When $\alpha = 0$, the function (1) is equivalent to the FCM formulation. The later requires the appropriate setting of the number of clusters and when this number is difficult to predict the algorithm may fail to capture the relevant distribution of the data (see. Figure 1).

The regularization term $\mathcal{R}(C, \mu)$ measures the complexity or the spread of the clusters and reaches its minimum when all the training examples are assigned to only one cluster. Among possible regularizers $\mathcal{R}(C, \mu)$ we can find the quadratic and the Kullback-Leibler functions [5, 8]. The tradeoff between the fidelity and the regularization terms makes it possible to define the optimal number of clusters automatically for a given application. The FCM and the regularization terms are mixed via a coefficient α which controls this tradeoff. In the different existing formulations, this coefficient is proportional to the ratio between the FCM term and the regularizer $\mathcal{R}(C, \mu)$ and decreases with respect to iterations:

$$\alpha(t) \sim f_\tau(t) \times O\left(\frac{\sum_{k=1}^{C} \sum_{i=1}^{N} \mu_{ik}\ d_{ik}^2}{\mathcal{R}(C, \mu)}\right) \qquad (4)$$

where f_τ is a decreasing function which can be:

$$f_\tau(t) = e^{-t/\tau} \tag{5}$$

Initially, the algorithm selects a large value of α, so the objective function will pay more attention to the regularization term and this makes it possible to decrease the number of clusters. As we go along different iterations of the clustering process, the decrease of α according to (4) ensures that the clustering process will pay more attention to the fidelity term, so the centers will be updated to minimize the distances to their training data.

The population of a cluster, defined as $pk = \frac{1}{N}\sum_i \mu_{ik}$ makes it possible to identify and remove the clusters for which the membership values are weak (i.e., below a given threshold). This threshold referred to as the *minimal population* can be set a priori depending on the application. For instance, in image segmentation, the minimal population can be related to the minimal size of a region.

3 Entropy regularization

Following the framework in [9], we consider the entropy regularization term:

$$\mathcal{R}(C,\mu) = -\frac{1}{N}\sum_{i=1}^{N} \underbrace{-\sum_{k=1}^{C} \mu_{ik}\, log_2(\mu_{ik})}_{The\ entropy\ term}$$

If the memberships of all the training examples to m clusters ($m < C$) are similar, the later overlap, $\mathcal{R}$ decreases and reaches its global minimum when $m = C$. On the other hand, when each training example is a cluster by it self, the distribution of the memberships will be peaked, so the entropies will vanish and $\mathcal{R}$ will reach its global maximum 0. Let's consider a new definition of the membership degrees as $\{\mu_{ik} = e^{-U_{ik}^2},\ \ U_{ik} \in \mathbb{R}\}$ which per construction ensures the consistency of their bounds. If we plug these memberships in the FCM term and in the regularizer $\mathcal{R}(C,\mu)$, we can show that the underlying constrained minimization problem becomes:

Minimize

$$\begin{aligned} \mathcal{J}(C,U) &= \sum_{k=1}^{C}\sum_{i=1}^{N} e^{-U_{ik}^2} d_{ik}^2 + \alpha\frac{K}{N}\sum_{i=1}^{N} \underbrace{-\sum_{k=1}^{C} e^{-U_{ik}^2} U_{ik}^2}_{The\ new\ entropy\ term} \\ \text{s.t.} \quad & \sum_{k=1}^{C} e^{-U_{ik}^2} = 1, \quad i = 1,\ldots,N \end{aligned} \tag{6}$$

where $K = log_{10}(e)/log_{10}(2)$ and $U = \{U_{ik}\}$. Using Lagrange [10, 11], the minimization problem can be written as:

Minimize

$$\mathcal{L}(C,U,\lambda) = \sum_{k=1}^{C}\sum_{i=1}^{N} e^{-U_{ik}^2} d_{ik}^2 - \alpha\frac{K}{N}\sum_{k=1}^{c} e^{-U_{ik}^2} U_{ik}^2 + \sum_i \lambda_i \left(\sum_k e^{-U_{ik}^2} - 1\right) \tag{7}$$

When the gradient of $\mathcal{L}(C,U,\lambda)$ with respect to $\{U_{ik}\}$ and $\{\lambda_i\}$ vanishes, we obtain respectively:

$$\begin{aligned} -U_{ik}^2 &= \tfrac{N}{-K\alpha}(d_{ik}^2 + \lambda_i) - 1 \\ e^{-U_{ik}^2} &= \frac{e^{\frac{N}{-K\alpha}(d_{ik}^2+\lambda_i)}}{e} \end{aligned} \tag{8}$$

and

$$\begin{aligned} \sum_c e^{-U_{ic}^2} = 1 &= \frac{1}{e}\sum_c e^{-\frac{N}{K\alpha}d_{ic}^2}\, e^{-\frac{N}{K\alpha}\lambda_i} \\ e^{-\frac{N}{K\alpha}\lambda_i} &= e\frac{1}{\sum_c e^{-\frac{N}{K\alpha}d_{ic}^2}} \end{aligned} \tag{9}$$

Now, the conditions for optimality lead to the *very simple* solution:

$$\mu_{ik} = e^{-U_{ik}^2} = \frac{e^{-(N/K\,\alpha)d_{ik}^2}}{\sum_c e^{-(N/K\,\alpha)d_{ic}^2}} \tag{10}$$

4 Consistency and Interpretation

4.1 Regularization

It is easy to see that this solution is consistent with the role of the coefficient α. When $\alpha \to \infty$, the limit of μ_{ik} will go to $1/C$, the distribution of the memberships will be uniform and the entropies will take high values, so the regularization term $\mathcal{R}$ will reach its global minimum. It follows that the centers of the C clusters $\{c_k = \frac{1}{N}\sum_i^N \mu_{ik}\, x_i, \quad k = 1,\ldots,C\}$ will converge to only one cluster. Notice that the overlapping clusters can be detected at the end of the algorithm using a simple criteria such as the distance between their centers is below a given threshold ϵ [3] or by using more sophisticated tests such as the Kullback-Leibler divergence in order to detect similar cluster distributions. Overlapping clusters are removed and replaced by only one cluster. Of course, the memberships can be updated using (10).

[3] In practice, ϵ is set to 1% the radius of the ball enclosing the data.

On the other hand, when $\alpha \to 0$, the effect of the regularization term vanishes, so each training example x_i will prefer the closest cluster as shown using (11):

$$\begin{aligned} \lim_{\alpha\to 0} e^{-U_{ik}^2} &= \lim_{\alpha\to 0} \frac{1}{1+\sum\limits_{c\neq k} e^{-(N/K\alpha)\{d_{ic}^2-d_{ik}^2\}}} \\ &= \begin{cases} 1 \; if \; d_{ik} = \min\{d_{ic},\; c \neq k\} \\ 0 \; if \; d_{ik} > \min\{d_{ic},\; c \neq k\} \end{cases} \end{aligned} \tag{11}$$

or will be a cluster by it self if the number of clusters C is equal to N.

4.2 Cardinality of the Training Set

Notice also that the size of the training set N plays an important role in the precision of our model. When $N \to \infty$, each membership term μ_{ik} will go either to 1 or 0 depending respectively on the fact that the k^{th} center is the closest or not to the training example x_i, so the degree of uncertainty decreases as the size of the training set N increases.

4.3 Scaling

If the training examples are assumed to be Gaussian distributed then it is possible to use the Mahalanobis distance which takes into account the spread of the data. For some $\sigma^2 = \alpha K/N$, we can rewrite the membership coefficients (10) as:

$$\mu_{ik} = \frac{e^{-\frac{1}{\sigma^2}(x_i-c_k)^t \sum\limits_k^{-1}(x_i-c_k)}}{\sum\limits_c e^{-\frac{1}{\sigma^2}(x_i-c_c)^t \sum\limits_c^{-1}(x_i-c_c)}} \tag{12}$$

Here c_k and $\sum_k^{-1}$ denote respectively the center and the covariance matrix of the k^{th} cluster. Now, σ acts as a scaling factor; when it is underestimated, each c_k will be a center of a Gaussian which vanishes everywhere except in c_k and each example will form a cluster by it self if $C = N$ (cf. figure 2, top-left). On the contrary, when σ is overestimated the Gaussian will be quasi-constant, so no difference will be noticed in the memberships of the training examples to the clusters and this results into one big cluster (cf. figure 2, bottom-right).

Figure (3) shows clustering results on other $2D$ toy examples using our method. It is easy to see that the number of classes is respectively 5 and 4, so setting either the variance or the number of classes and using respectively our method and FCM will lead to the same results (cf. figure 3, left and figure 1, right). However, when data live in high dimensional spaces, it may

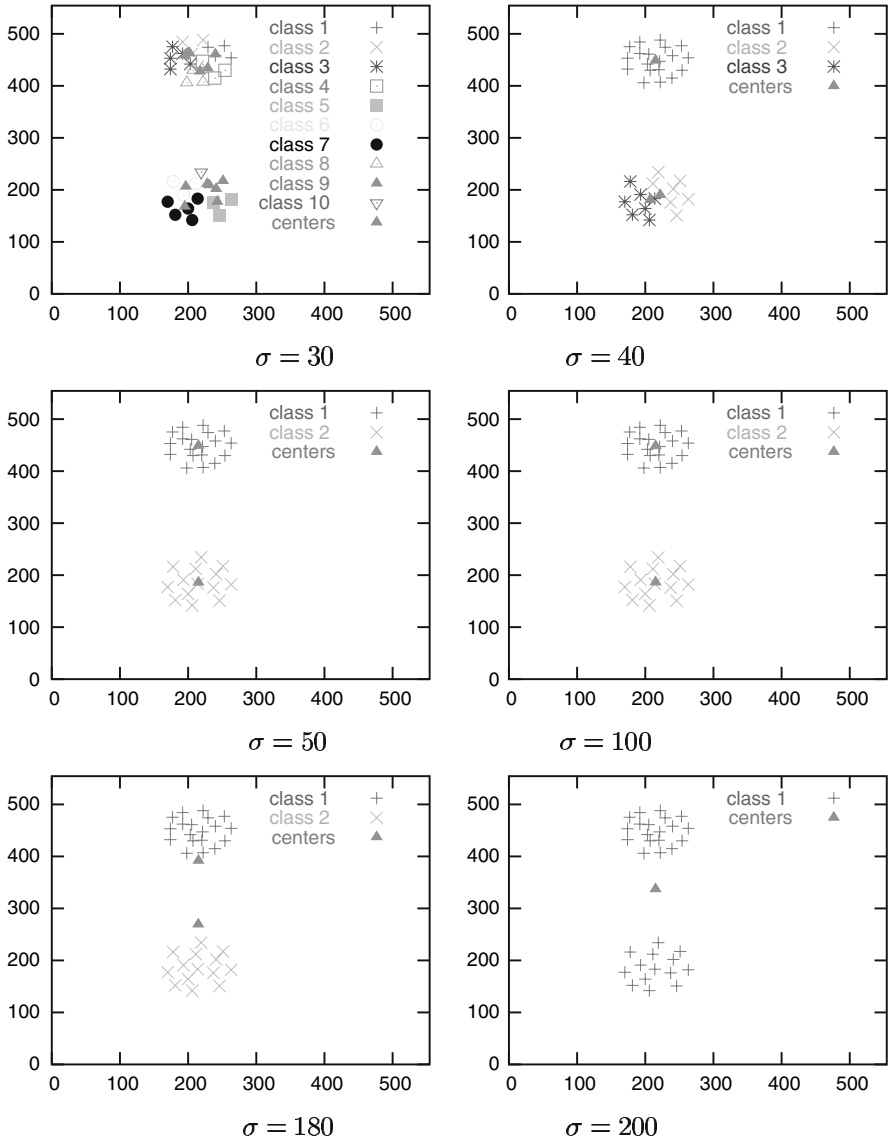


Fig. 2. *Location of the centers with respect to the value of the regularization parameter σ. As the regularization parameter increases, the membership degrees will be uniform, the regularization term decreases, so the centers overlap and the number of clusters decreases. The parameters C and N are respectively equal to* 20 *and* 33

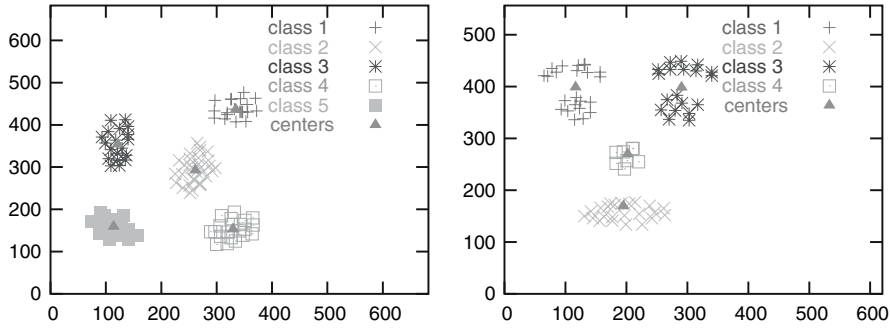


Fig. 3. *Results of the clustering algorithm using our new regularizer ($\sigma = 60$)*

be easier to predict[4] the variance rather than the number of classes as the pre-visualization of the data is usually difficult.

Figure (4, top) shows the entropy and the fidelity terms with respect to the variance σ. As expected, low variance implies high entropy, so the resulting number of classes is high (cf. figure 4, bottom-left). On the other hand, high variance implies low entropy, so the underlying number of classes decreases. The best variance σ is of course application-dependent, but it may exist a trivial σ when the classes are linearly separable.

4.4 Uniqueness of the Solution

Even though the objective function (7) is not convex (see. figure 5), we will show that the above formulation leads to a unique membership solution.
The vector of parameters $U = \{U_{ik}\}_{i,k=1}^{N,C}$ live in the Euclidean space $\mathbb{R}^p (p = C \times N)$ where $\{e_{ik}\}_{i,k=1}^{N,C}$ is assumed to be its canonical orthogonal basis. For $\{c_{ik}\} \in \{-1, +1\}^p$, we define an *n-quadrant* $S_{c_{11},\ldots,c_{NC}}$, in the span of $\{c_{ik}e_{ik}\}$, as:

$$S_{c_{11},\ldots,c_{NC}} = \left\{ U \quad \text{s.t} \quad U = \sum_{i,k=1}^{N,C} a_{ik}\, c_{ik}\, e_{ik}, \quad a_{ik} \in \mathbb{R}^+ \right\} \tag{13}$$

It is clear that:

$$\mathbb{R}^p = \bigcup_{c_{11},\ldots,c_{NC} \in \{-1,+1\}} S_{c_{11},\ldots,c_{NC}} \tag{14}$$

Uniqueness of the local solutions: the objective function (7) has a unique local minimum per n-quadrant. Indeed, when the gradient vanishes in a given n-quadrant $S_{c_{11},\ldots,c_{NC}}$, the local solution can be written:

[4]The variance may be predicted by sampling data from different classes, estimating their variances and averaging the later through these classes.

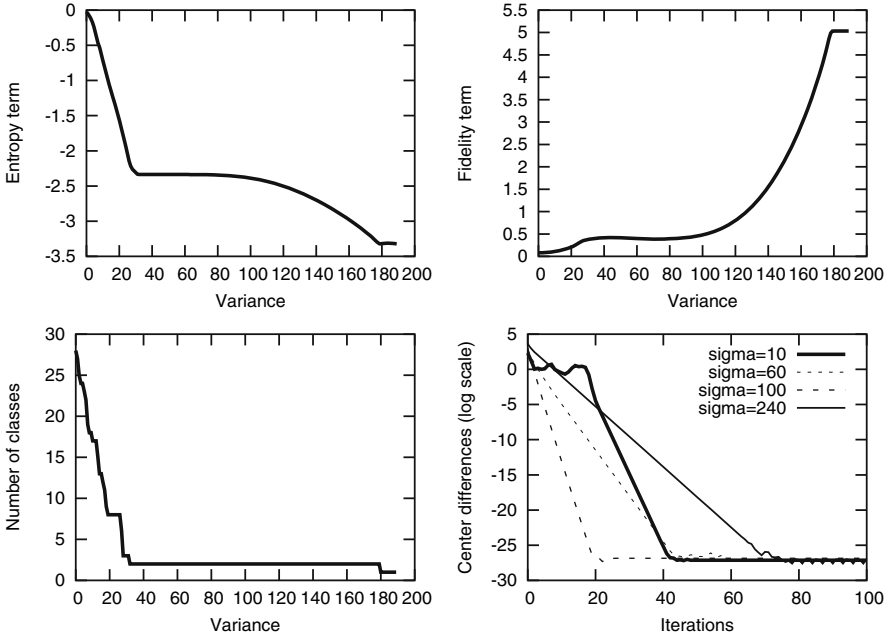


Fig. 4. *(Top) variation of the entropy and the fidelity terms with respect to the variance σ. Both the entropy and the fidelity values are shown in the log-base* 10 *scale. (Bottom-left) the decrease of the number of clusters with respect to the variance. (Bottom-right) The convergence process is shown through the localization error $\sum_k \|c_k^{(n)} - c_k^{(n-1)}\|^2$, where $c_k^{(n)}$ is the k^{th} center estimated at the n^{th} iteration. These results are related to the clustering problem shown in figure (2)*

$$U_{ik} = c_{ik}\sqrt{\frac{N}{K\alpha}(d_{ik}^2 + \lambda_i) + 1}, \quad i = 1, \ldots, N, k = 1, \ldots, C \tag{15}$$

Uniqueness of the memberships: let $\{U_{ik}\}$ and $\{U'_{ik}\}$ be two local solutions belonging respectively to two n-quadrants $S_{c_{11},\ldots,c_{NC}}$ and $S_{c'_{11},\ldots,c'_{NC}}$. It is easy to see that these two local solutions expressed as:

$$\begin{aligned} U_{ik} &= c_{ik}\sqrt{\tfrac{N}{K\alpha}(d_{ik}^2 + \lambda_i) + 1} \\ U'_{ik} &= c'_{ik}\sqrt{\tfrac{N}{K\alpha}(d_{ik}^2 + \lambda_i) + 1} \end{aligned}, \quad i = 1, \ldots, N, k = 1, \ldots, C \tag{16}$$

lead to the same value of both the objective function $\mathcal{L}(C, U, \lambda)$ and the membership degrees, i.e., $e^{-U_{ik}^2} = e^{-U_{ik}'^2}, \quad i = 1, \ldots, N, k = 1, \ldots, C$. Hence, any local solution is also global.

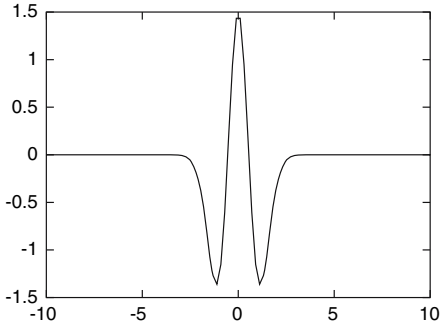


Fig. 5. *Plot of the objective function for a trivial case: one class and one training example. ($\alpha = 5, d_{11} = 0.5$). It is clear that the function is not convex but has a unique solution in each n-quadrant*

5 Discussion

It is known that the initialization process in FCM requires C to be exactly the actual number of clusters. When C random centers are selected initially, it may happen that two (or more) of them will lie in the same cluster while other separate clusters will be captured with only one center, hence FCM may result into over/under-clustering problems (cf. fig. 1, left). In contrast, we consider in our approach C to be far larger than the actual number of classes in order to guarantee that each cluster will be captured, at least, with one center. This may result into several overlapping clusters, so the underlying populations will become week[5] but at the end of the algorithm, simple tests, such as thresholding the distances between the centers or more sophisticated Kullback-Leibler divergence, can be used in order to detect and remove overlapping clusters and replace them with few others (cf. §4.1).

The use of the exponential form (10) makes the memberships relatively high for data belonging to a given cluster and vanishing for the others, so the influence of the data on the centers is local and depends of course on the chosen variance σ. Therefore, since the estimation of the centers relies strongly on the data intervening in the underlying clusters, this results into better localization of the centers as shown in figure (3, left) with respect to figure (1, right).

It may be easier for some applications to predict the variance σ rather than the number of clusters. Let's consider clustering face images by their identities. Predicting the variance σ can be achieved by sampling manually face images from some persons, estimating their variances, then setting σ to the expectation of the variance through these persons. While this setting is

[5]Many clustering techniques rely on the notion of minimal population in order to eliminate weak clusters. Usually this destroys clusters having few data even with strong memberships.

not automatic, it has at least the advantage of introducing a priori knowledges on the variance of the data at the expense of few interaction and reasonable effort from the user. Furthermore, this is easier than trying to predict the actual number of clusters, mainly for huge databases living in high dimensional spaces.

6 Applications

6.1 Toy examples

Figure (6) shows some results of our clustering algorithm on $2D$ toy examples. Different clusters are shown using various colors. The left-hand side picture shows three clusters with different cardinalities well-captured with three centers. The mid-picture shows three other clusters with three different scales (approximately 30, 60 and 90 pixels). After the application of the clustering process with $\sigma = (30 + 60 + 90)/3$, the three classes are well-captured with three centers. The right-hand side picture shows a non-trivial case, where the clusters are non-linearly separable.

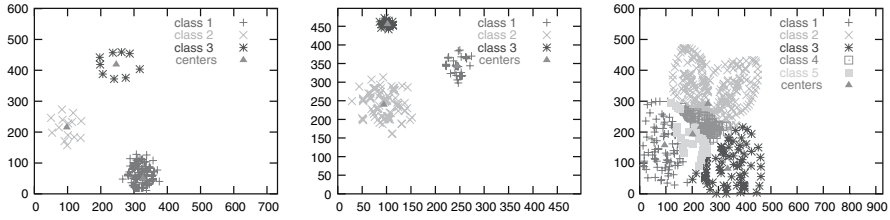


Fig. 6. *Some 2D toy examples. The variances are set respectively (from left-to-right) to* 50, 60 *and* 140

6.2 Image segmentation

Clustering methods have proven to be very useful for many applications including image segmentation and partial queries. The idea is based on clustering the distribution of colors in an image using a suitable color space such as HSV, LUV, etc. In our experiments, the color of each pixel in an image is defined as the average RGB color in a local neighborhood of 10×10 pixels; this makes the color distribution of neighboring pixels smooth and the resulting blobs homogeneous. We applied our clustering algorithm on this distribution and each color is assigned to its class center using a simple Euclidean distance (see. figure 7). One possible application of image segmentation is partial queries [12, 9] where the user selects a region of homogeneous color or texture distribution and the system will display similar images according only to the selected region.

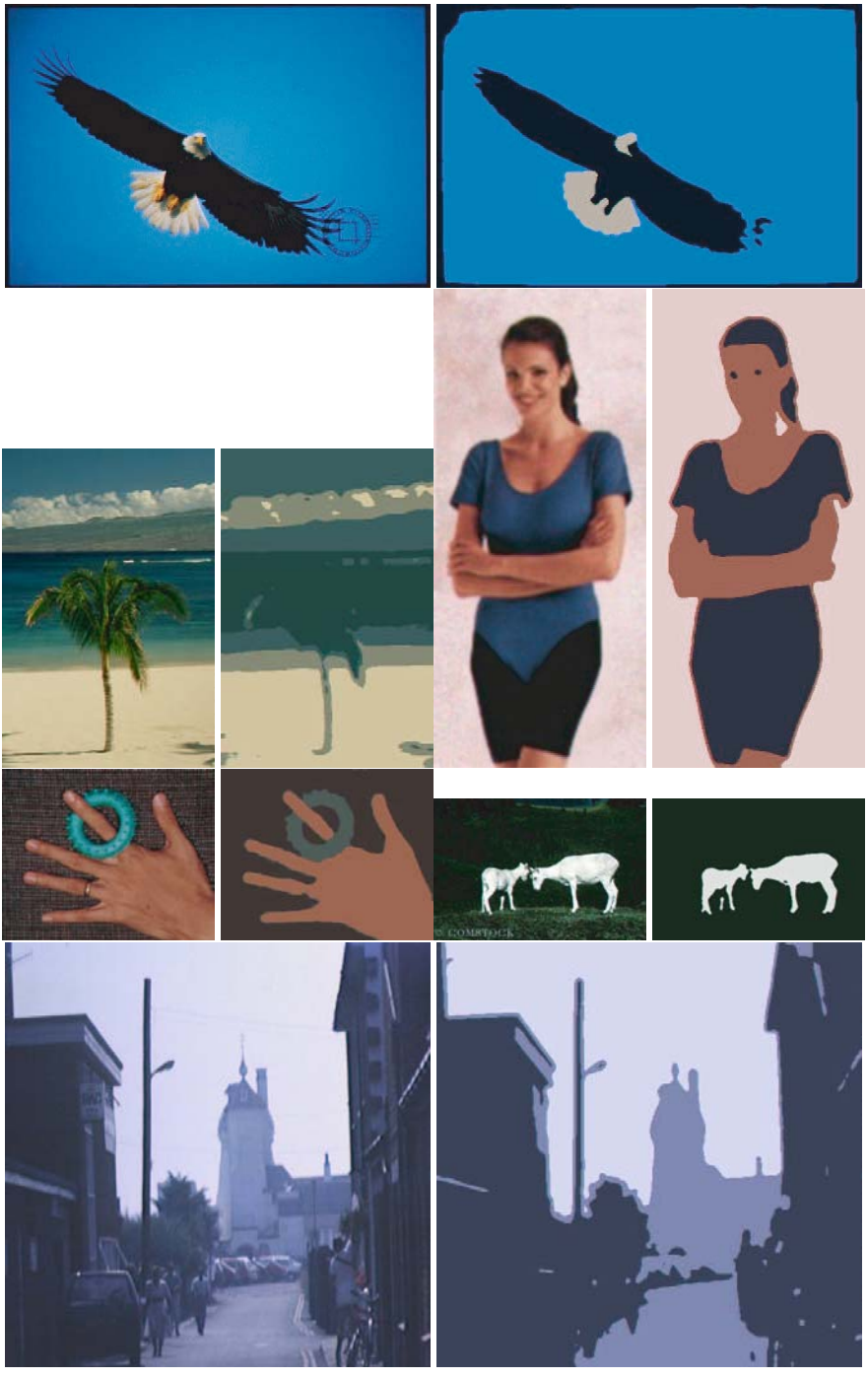


Fig. 7. *Some segmentation results. The variances are set, resp. from the top-left to the bottom-right, to* 0.2, 0.2, 0.26, 0.11, 0.3 *and* 0.2.

7 Conclusion

We introduced in this paper a new formulation of regularized FCM which is simple, consistent and easy to interpret in terms of regularization. Open issues are several; one of them is the use of isomap [13, 14] in order to capture non-linearly separable clusters. This approach relies on transductive learning which requires the interaction of the user in order to have a priori knowledges on the cluster manifolds. An other issue will be a new formulation of the objective function (6) which takes into account data with large variations in scale. Intuitively, if we consider, in the objective function (6), different mixing parameters α for different classes then it might be possible to capture clusters with large variation in scale.

References

1. C. Fraley, "Algorithms for model-based gaussian hierarchical clustering," *SIAM J. Sci. Comput*, vol. 20, no. 1, pp. 270–281, 1998.
2. A.K. Jain and R. C. Dubes, "Algorithms for clustering data. prentice hall," 1988.
3. Christian Posse, "Hierarchical model-based clustering for large datasets," *Journal of Computational and Graphical Statistics*, vol. 10, no. 3, 2001.
4. James C. Bezdek, "Pattern recognition with fuzzy objective function algorithms, new york," 1981.
5. H. Frigui and R. Krishnapuram, "Clustering by competitive agglomeration," *In Pattern Recognition*, vol. 30, no. 7, 1997.
6. Rajesh N. Dave, "Characterization and detection of noise in clustering," *In Pattern Recognition*, vol. 12, no. 11, pp. 657–664, 1991.
7. A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, "Support vector clustering, " *NIPS*, pp. 367–373, 2000.
8. Hidetomo ICHIHASHI, Katsuhiro HONDA, and Naoki TANI, "Gaussian mixture pdf approximation and fuzzy c-means clustering with entropy regularization, " *AFSS*, 2000.
9. N. Boujemaa, "Generalized competitive clustering for image segmentation," *In the 19th International Meeting of the North American Fuzzy Information Processing Society NAFIPS, Atlanta*, 2000.
10. J. J. Mor and S. J. Wright, "Optimization software guide," *Siam Publications*, 1993.
11. R. Fletcher, *Practical Methods of Optimization*, vol. 1, John Wiley & sons, New York, 1980.
12. C. Carson, M. Thomas, S. Belongie, J. Hellerstein, and J. Malik, "Blobworld: A system for region-based image indexing and retrieval, third," *Int Conf on Visual Information Systems*, 1999.
13. M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, pp. 1373–1396, 2003.
14. Ham J., D. D. Lee, S. Mika, and B. Schlkopf, "A kernel view of the dimensionality reduction of manifolds," (*110) Max-Planck-Institut fr biologische Kybernetik, Tubingen (July 2003) [Note: to appear in the Proceedings of ICML 2004*].

Fuzzy Long Term Forecasting through Machine Learning and Symbolic Representations of Time Series

Bernard Hugueney[1], Bernadette Bouchon-Meunier[2], and Georges Hébrail[3]

[1] Laboratoire Informatique de Paris 6
bernard.hugueney@lip6.fr
[2] Laboratoire Informatique de Paris 6
bernadette.bouchon-meunier@lip6.fr
[3] Ecole Nationale Supérieure des Télécommunications
georges.hebrail@enst.fr

Abstract. Time series forecasting is often done "one step ahead" with statistical models or numerical machine learning algorithms. It is possible to extend those predictive models to a few steps ahead and iterate the predictions thus allowing further forecasting. However, it is not possible to do this for thousands of data points because cumulative error tends to make the long term forecasting unreliable. Such uncertainty can be conveied by the use of fuzzy forecasting where the forecasted value is a fuzzy set rather than a number. The end-user can only appreciate the uncertainty of the forecast if the forecasting model is easy to understand. Contrary to common "black-box" models, we use symbolic machine learning on symbolic representations of time-series. In this paper, we tackle the real-world issue of forecasting electric load for one year, sampled every ten minutes, with data available for the past few years. In this context, future values are not only related to their short term previous values, but also to temporal attributes (the day of the week, holidays ...). We use a symbolic machine learning algorithm (decision tree) to extract this kind of knowledge and predict future pattern occurences. Those patterns are learnt when building a symbolic representation of the time series, by clustering episodes showing similar patterns and making the cluster a symbolic attribute of the episodes. Intra-class variations result in forecasting uncertainty that we model through fuzzy prototypes. Those prototypes are then used to construct a fuzzy forecasting easily understood by the end-user.

1 Introduction

Time series forecasting used to be concerned mostly with short-term estimations. Being short-term is relative to the number of predicted values and not to the time frame, so predicting the *yearly* values for the next ten years could be considered short-term. Time series databases are ubiquitous and often very large as it is now easy to accumulate such data: datawarehouses can nowadays be filed by sensors without human intervention. However, being able to store

B. Hugueney et al.: *Fuzzy Long Term Forecasting through Machine Learning and Symbolic Representations of Time Series*, Advances in Soft Computing **2**, 109–120 (2005)
www.springerlink.com © Springer-Verlag Berlin Heidelberg 2005

those time series of hundreds of thousands of data points is one thing, being able to predict thousands of points ahead in the future is another.

Our research was triggered by the need to forecast electric consumptions of individual customers for a whole year at *Electricite de France (EDF)*. Some customers have had their power consumption sampled every ten minutes for a few years and we strive to forecast consumptions for the next year. This goal is somewhat related to the EUnite challenge of 2001 (forecasting a month of daily maximum power consumption) but even more ambitious. Furthermore, we wish to express forecasting uncertainty with a model that is intelligible.

Our challenge is both to bring forecasting algorithms to truly data-mining scales and to use a forecasting model that makes sense to the end-users. This is done by bringing the data within the grasp of usual forecasting algorithms through data aggregation. Simple aggregations over the usual time scales (weeks or days for example) are possible but usually incur a sever loss of precision. We illustrate this by applying existing algorithms to our data set at increasing scales, from yearly aggregates to raw data sampled every ten minutes. Statistical models are the first to fail, but we show that even machine learning algorithms (whether numerical or symbolic) are not sufficient either. To overcome those limitations, we propose a novel algorithm based on a *symbolic representation* of time series, as an information preserving aggregation.

We will first present the usual forecasting methods, and the ad-hoc methods that were in use at EDF, along with their respective limitations regarding long term forecasting. We will then present our symbolic forecasting algorithm: the model, the algorithms used to compute the model and the forecasting results. The last part of this paper will also show how symbolic representations allow to express uncertainty through fuzzy forecasting.

2 Long term forecasting

Generally, long term forecasting is achieved by extending short-term forecasting statistical and machine learning models. Amongst statistical models, Box-Jenkins seasonal ARMA models and exponential smoothing with linear trend [4] and seasonality [17] are the most popular. Those models often require expert knowledge and enforce stringent constraints on the time series, hence the use of more data-driven and less restrictive machine learning algorithms such as neural networks and support vector machines.

Throughout this paper, we will use the same time series data set to illustrate our point. The data set is typical in many aspects:

- large: seven years of data, sampled every ten minutes
- noisy and with missing values
- showing patterns at different scales (yearly or daily)
- related to human activities

The last point is important because it suggests that days and weeks are relevant scales. It does not imply that the time series exhibit periodicities, but we can expect some kind of patterns at this scales. For example, there can be low comsumption days at regular intervals (for example weekends) or not (for example holidays). Figure 1 shows the raw data set consisting of more than 3.10^5 data points. At this scale, it is only possible to see the yearly pattern, however, the excerpt shown at Figure 2 exhibits some daily patterns. The last year of complete data will be used to compare the forecasting accuracy of the algorithms. Rather than to evaluate our algorithm on numerous data sets, we will explain how and why it works on our reference data set and what kind of time series one can expect to successfully forecast through symbolic representation. It is, of course, not possible to evaluate every forecasting methodology for our needs, but we display what we believe to be shortcomings of the usual statistical and machine learning (both numerical and symbolic) methodologies for such time series long term forecasting.

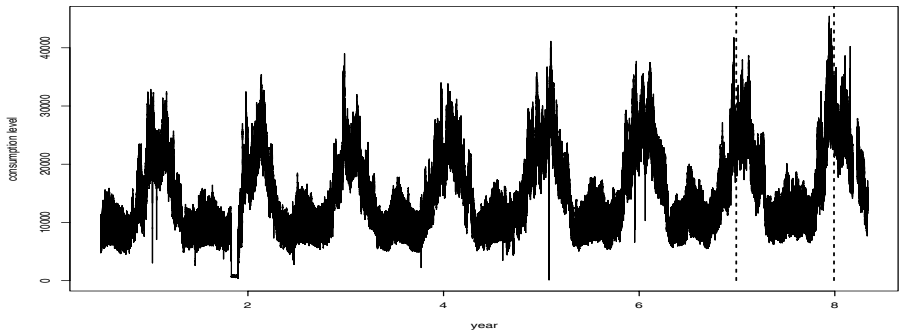


Fig. 1. Our time series: seven years sampled every ten minutes.

2.1 Iterated N-steps ahead forecasting

Principle

Time series forecasting relies on the autocorrelation of the time series. Most models are based on the following decomposition:

- trend
- seasonalities
- short-term correlation

One must first identify the trend and the seasonalities and remove them to work on stationary time series. Parameters estimation is then an optimization

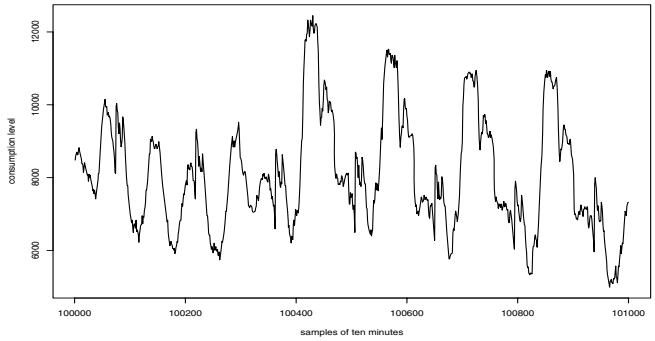


Fig. 2. Excerpt of the data set showing daily patterns.

problem than cannot always be solved optimally. For statistical models such as in the Box-Jenkins methodology, expert knowledge is used to set the parameters. Neural networks and Support Vector Machines can learn the parameters from the data but some hyperparameters must still be set.

The Holt-Winters model [15] stands as a simple statistical model taking into account the local autocorrelation (with exponentially decreasing importance), the trend and an additive seasonality. The forecast $\hat{Y}_{t+h}$ for period $t+h$ on a variable level $\hat{a}$ with a trend component $\hat{b}$ and a seasonal component $\hat{s}$ over period p is given by:

$$\hat{Y}_{t+h} = \hat{a}_t + h.\hat{b}_t + \hat{s}_t$$

As the time series display other seasonalities that the yearly one, we have tried to substract the yearly pattern computed at the monthly scale. The residuals are shown at Figure 3. Having successfully forecasted the yearly trend and seasonality with a Holt-Winters model, from now on we will focus on those residuals and try to forecast them.

According to our hypothesis on the time series, our data set exhibits less regularities at a daily scale and the Holt-Winters model fails to provide any meaningful forecasts of the residuals. Once again, this is to be expected because of the greater variability at those time scales. On Figure 3, we show that the points on a daily period can have very different values. More complex statistical models such as Basic Structural Model, which is a restricted ARIMA(0,2,2) with an additional seasonal component, are not practical because the size of the data set is out of reach of the optimization algorithms used to compute the parameters.

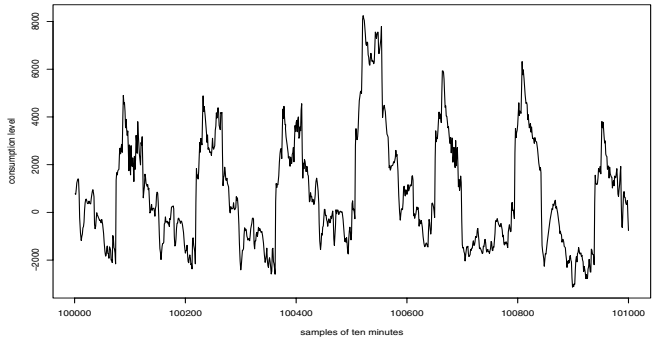


Fig. 3. Residuals with weekly and daily patterns.

Support Vector Regression

We can attempt to learn the daily and weekly patterns of the residuals with Support Vector Regressors (SVR). SVR [16] try to learn a non-linear function in the *embedded space* that maps the n previous values $y_{t-n}, ..., y_{t-1}$ to the value y_t at time t. This is done by learning a linear function in a feature space F of higher dimensionnality than the embedded space E. ϕ being the non linear mapping from E to F:

$f(x) = (w.\phi(x)) + b$ with $\phi : R^n \rightarrow F, w \in F$

ϕ is fixed and w is computed from the data by minimizing a regression risk R which is the sum of the empirical risk $R_{emp}[f]$ and a complexity term $||w||^2$:

$R = R_{emp}[f] + \lambda||w||^2 = \sum_{i=1}^{l} C(f(x_i) - y_i) + \lambda||w||^2$, where l is the sample size and C is the cost function, and λ a regularization constant.

We used Support Vector ϵ-regression which is based on a loss function with a parameter ϵ that defines a tube around the linear function in the embedded space. This defines a cost function C as:

$$C(f(x) - y) = \begin{cases} |f(x) - y| - \epsilon \; for \; |f(x) - y| > \epsilon \\ 0 \qquad\qquad otherwise \end{cases}$$

Machine learning algorithms such as Support Vector Regressors appear to be more robust than the algorithms used for the Holt-Winters parameters estimation. Thus it was possible to have meaningful forecasts at lower scales, but below the daily scale, we reached the limits of our computational resources. It was still not possible to go below daily scale for our data set, using the R binding to the award winning (EUnite 2000 and IJCNN 2001 [3] challenges) libSVM regressors implementation, without using too many gigabytes of memory. Parameters tuning (such as the value of ϵ) is tricky, or at least computationally very intensive. One could select the values in the input space to have far reaching memory without such a memory complexity. How-

ever, to meet our requirement, such a selection should be possible without user intervention.

Figure 4 shows the result of Support Vector Regressors trained at daily scale using an embedded space with a two-years memory. The resulting sum of square errors is 838.10^9 and the mean absolute error is 2994. The more commonly used mean absolute percent error is not meaningful because we are forecasting residuals around 0 (the relative error can take arbitrary large values).

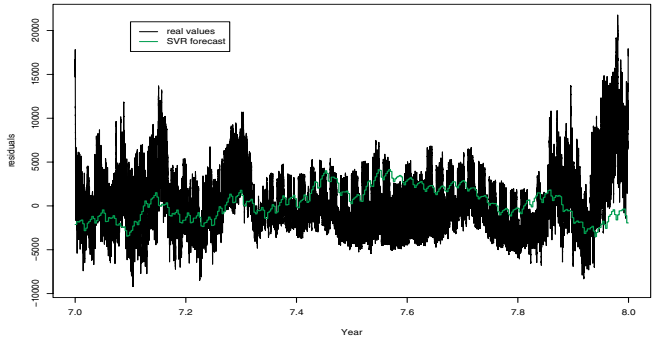


Fig. 4. Real residuals and their forecast through SVM regressors.

Limitations

We have seen that it is not possible to forecast at very precise scales with the statistical models, whether because they cannot model the changing patterns, or because of the computational complexity. As the precision grows, so does the amount of data points. Consider, for example, a weekly cycle in our data set: a week is made of 1008 points, so taking past weeks into account implies taking thousands of points into account. Those iterating one-step-ahead forecast algorithms use the newly predicted values to predict further ahead, and thus fail to take enough past values into account. This means that at some point all information from the data is forgotten by the model and nothing remains but the trend and seasonalities.

The main limitation of those approaches lies in the fact that they rely entirely on past values to predict future values. As we have seen in our case, the number of past values to take into account grows beyond our computational power.

To overcome this limitation, one should not only take past values into account, but also temporal attributes, such as the day of the week and the month of the year, because they are available throughout the year to forecast.

2.2 Taking temporal attributes into account

Using a regression tree

As an extension to the aforementioned machine learning algorithms, one can include other attributes than the previous values. Any available information can be considered: temperature, for example, is commonly used when forecasting electric consumption. For the sake of generality, we restrict ourselves to information that can be extracted from the temporal attributes, such as the day of the week and holidays. The following attributes have been used:

- Hour of the day,
- Day of the week,
- Month of the year,
- Holidays or working day,
- Week of the year,
- and one attribute for each Holidays.

We can build a regression tree [2] with (only) 17 nodes, an excerpt of which being:

1. If month in {Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct}
   a) If month in {Jan,Feb,Mar,Apr} Then $Y = 468$
   b) Else
      i. If $Hour < 0.302083$ Then $Y = -1989$
      ii. Else $Y = 488$
2. Else $Y = 1764$

Figure 5 shows the resulting forecast. It was possible to handle the whole data set at the most precise scale and the sum of square errors is 528.10^9 and the mean absolute error is 2309, which is 23% better than the forecast at daily scale that we could compute with Support Vector Regressors.

Limitations

The forecasting accuracy is limited by the fact that the data set is noisy and that local correlation is ignored. According to our hypothesis of daily patterns, illustrated by the excerpt in Figure 3, we seek to take the intra-day correlation into account. Forecasting one pattern at a time, instead of one point at a time, not only would we use the local autocorrelation but, as an added bonus, we would have orders of magnitude less data to process, which can be desirable when processing many large time series.

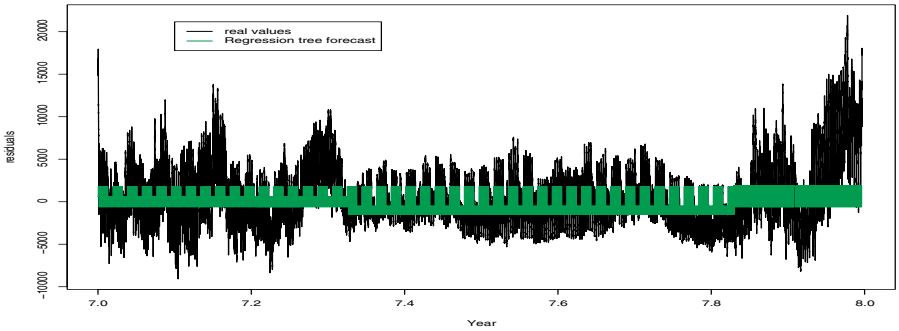


Fig. 5. Long term forecasting of the residuals with a regression tree.

2.3 An ad-hoc method

Before introducing our algorithm, it is interesting to study a very simple forecasting method that was used at EDF (somewhat) effectively to meet our goal. When facing the need to forecast as precisely as possible next year's electricity comsumption, people used the simplest forecasting "methodology" possible: next year shall be the same as this year !

Under the hypothesis that daily patterns are strongly correlated to the days of the week, the time series from the past year are shifted to match the week days of the next year. For example, 1998-01-01 was a thursday, and 1999-01-01 was a friday, so consumption for the year 1999 would be predicted to be the same as for the year 1998, starting from 1998-01-02 which is a friday, the same week day as 1999-01-01. When taking yearly linear trend into account (as we have seen in 2.1), the results can be quite good. However, this method is not reliable because:

- only one past year is used: if it was not typical the results turn out highly skewed,
- each daily pattern is predicted using only one previous daily pattern: any variability in those patterns will skew the forecast,
- there are other factors as well that should be taken into account (for example holidays).

All those drawbacks can be overcome by human intervention, which is tedious and time consuming. Our algorithm tackles those weaknesses by making use of a symbolic representation.

3 A symbolic approach to long term forecasting

3.1 Principles of the symbolic representation

Symbolic representation of times series, as we introduced it in [7] and presented in more details in [8], is a very general framework. Intuitively, symbolic representation aims at representing time series, such as the excerpt shown in Figure 3, by

- a sequence of episodes and symbols such as
  $\{([100075, 100219[, \wedge)$, $([100219, 100363[, \wedge)$
  , $([100363, 100507[, \wedge)$, $([100507, 100651[, \times)$,
  $([100651, 100795[, \times)$, $([100795, 100939[, \times)\}$
- an interpretation of each symbol: $\wedge$ would be interpreted as a propotypical day with two maxima and $\times$ as a prototypical day with one extremum.

To be useful, a symbolic representation should be of:

- minimal complexity, with regard to the number of episodes, number of symbols and interpretation of each symbol
- maximal accuracy: it should be possible to reconstruct time series from their symbolic descriptions with minimal distorsion.

Formally, we use the following definitions for a symbolic representation of time series TS with a set of episodes E, a set of symbols Λ and a function (symbolic description SD) from E to Λ:

Raw time series over a time domain D:
$TS = \{(time_i, v_i)\}_{i \in \{1,\cdots,N\}}$, with $time_i \in D$ and $v_i \in R$
Model with M episodes and K symbols:
$\Lambda = \{s_j\}_{j \in \{1,\cdots,K\}}$
$E = \{e_l = (m_l, a_l, b_l, n_l)\}_{l \in \{1,\cdots,M\}}$,
with $m_l \leq a_l \leq b_l \leq n_l$
$SD : E \to \Lambda, e_l \mapsto SD(e_l)$

The interpretation $EpisodeModel$ of the s_j symbols is such that for each (e_i, s_i), it is possible to reconstruct a pattern.

In the most generic framework, the episodes and patterns could be fuzzy. It would allow to express overlap between episodes as well as a level of confidence (or lack thereof) in accuracy of the patterns. Using fuzzy sets, as introduced by [1], to build manageable chunks of information is the idea driving *granular computing*[14]. However, for this application, we only consider what we call *regular episodes*: crisp episodes, all the same size, with no overlaps nor gaps between them. Using the previous notations, we can express this restrictions by: $b_l = n_l = m_{l+1} = a_{l+1}$.

3.2 Learning and forecasting patterns

The first step is to build a symbolic representation of the time series. For a simple set of *regular episodes*, there are only two parameters: the length of

the episodes and the offset. The offset is the instant at which the first episode begins, after the start of the time series. A complete discussion on how to compute those parameters is beyond the scope of this paper and can be found in [8]. However, an appropriate size (most of the time, the length of a day or a week), can be chosen by studying the autocorrelation function or the periodogram of the time series. For the offset, we select it so as to maximize the time series local variations around the cut-points between episodes, in order to minimize the variations inside each episode. In our example, daily episodes do not start at midnight, but at eight o'clock in the morning. It allows us to have a whole working day and a whole night in the same episode, which would be important if the nightly consumptions showed interesting patterns that should not be split across episodes. On Figure 6, we have tried each possible value for this parameter (start episodes at 00h00, 00h10, ..., 23h50): while not the global optimum, the computed offset is satistactory.

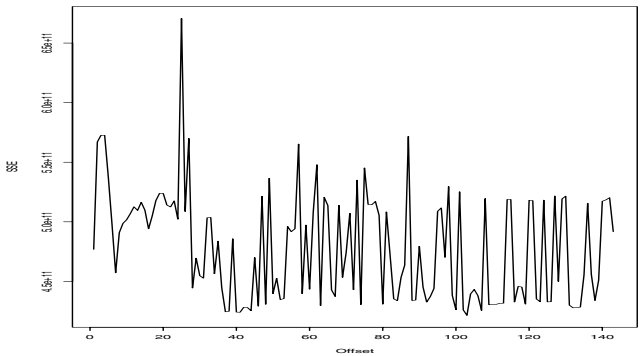


Fig. 6. Forecasting Sum of Square Errors for each of the possible values of the offset.

Once the episodes are chosen, we cluster the patterns to find some prototypical patterns. [6] is also using unsupervised clustering to build a "symbolic representation" as a preprocessing step to enhance neural networks stock market forecasts. Having a stronger hypothesis, we can use non-overlapping episodes instead of a sliding window. This allows us to avoid the pitfalls explained in [12] and having meaningful clusters for windows of a much greater size (144 points in our example instead of 2 ou 3). We use a k-means algorithm in order to minimize the sum of the intra-class sum of square errors. This error is exactly the same that would be made when replacing the time series by the interpretation of their symbolic representation: each pattern would be replaced by the prototypical pattern of its cluster.

After building a symbolic representation of the time series, it is then possible to use a decision tree algorithm to learn the relationships between at-

tributes and symbols. We use the same attributes as in 2.2 except for the hour of the day which is meaningless for daily episodes.

We split the available past days into a training set and a validation set. The parameters (clusters of the numerical unsupervised clustering algorithm and splitting criteria for the decision tree) are learnt on the training set. To set the hyperparameters (number of clusters and pruned tree size), we split the set of past days between a training set and a validation set. We exhaustively search the hyperparameters space to minimize the sum of square errors of the interpretation of the forecast symbolic representation over the validation set.

After computing the clusters and the decision tree, it is easy to do the long term forecasting. Splitting the year to forecast into episodes (with the same length and offset that were used to build the symbolic representation of past data), it is straightforward to compute the temporal attributes of those episodes. The decision tree can then forecast a symbolic representation by computing the symbol for each episode. Each symbol representing a cluster, we compute the predicted time series by choosing the prototypical pattern of the represented cluster for every episode of the forecast symbolic representation.

The results can be seen on Figure 7 (whole year) and Figure 8 (detail). This last figure clearly shows the advantage of the symbolic forecasting method: whereas a regression tree could only learn constant levels, the symbolic representation allows to describe the time series with more complex patterns. The sum of square errors is 421.10^9 and the mean absolute error is 2024 which is 12% better than the forecast at the same scale with a regression tree.

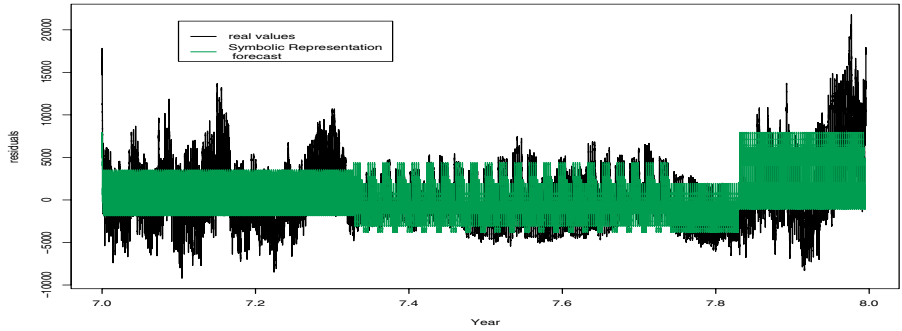


Fig. 7. Forecasting through symbolic representation (whole year).

3.3 Expressing uncertainty through fuzzy forecasting

Representing the episodes with the barycenter of their clusters enables us to use the variability of the clusters to estimate the accuracy of the represen-

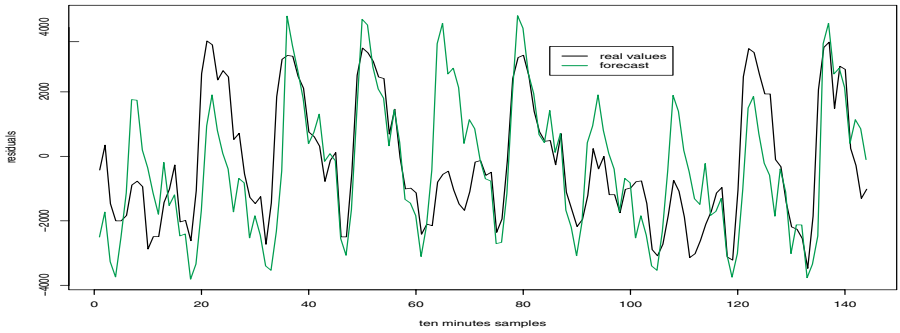


Fig. 8. Forecasting through symbolic representation (detail).

tation. As we have seen in 3.1, we defined the most generic symbolic representation framework to allow fuzzy interpretations of the symbols. End-users required fuzzy numbers that would be easy to understand. Thus, we decided to use fuzzy sets F_i, representing values encountered at time i, defined by a trapezoidal membership function d_{F_i}: $d_{F_i}(x) = \begin{cases} 0 & if \quad x < m \\ \frac{x-m}{a-m} & if \ x \in [m, a[\\ 1 & if \ \ x \in [a, b] \\ \frac{n-x}{n-b} & if \ \ x \in]b, n] \\ 0 & si \quad x > n \end{cases}$

where a (resp. b) is the lower (resp. upper) 10% bound of the set of values encountered at time i and m (resp. n) the minimum (resp. maximum) value of this set.

Figure 9 shows an excerpt of the fuzzy interpretation of the forecast symbolic representation of the time series. The time series in bold are the real values that we tried to forecast. The other four time series are, from lower to upper series:

1. $lb0$ lower bound of membership values > 0: no data point was encountered below this curve,
2. $lb1$ lower bound of membership value $= 1$: 90% of the data points were above this curve,
3. $ub1$ upper bound of membership value $= 1$: 90% of the data points were below this curve,
4. $ub0$ upper bound of membership value > 0: no data point was above this curve.

On our data set, the true data distribution verifies the expected meaning of the fuzzy interpretation:

$< lb0$	$< lb1$	$> ub1$	$> ub0$
2.3%	14.2%	9.67%	0.94%

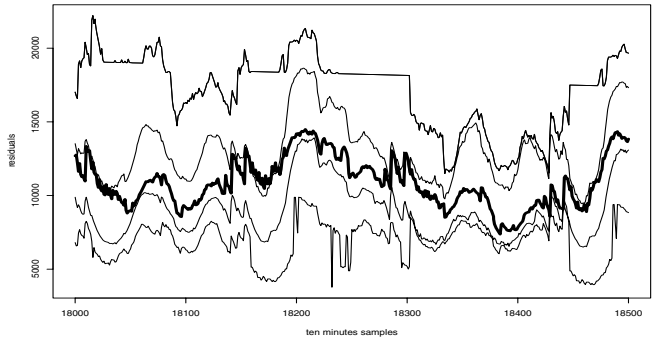


Fig. 9. Excerpt of the fuzzy forecast and the real consumption values (in bold).

4 Conclusions and future works

To effectively handle very large databases, some kind of data aggregation is often required. For time series databases, data aggregation has been used with success, mainly for indexing [10] [18] and clustering[11]. Under the hypothesis that the time series are generated by an underlying process with a finite number of states which can be inferred from temporal attributes, we have shown that it is advantageous to:

1. learn patterns
2. learn relationships between temporal attributes and patterns

We have illustrated this approach under the hypothesis of patterns of equal length. Although very restrictive, this hypothesis is true for many real-world time series databases with daily or weekly patterns. However, the principles of symbolic time series forecasting still hold for less regular patterns.

The patterns clustering could be improved with a better distance function than the euclidian distance in the k-means algorithm. Time Wrapping [19], especially when adapted to data-mining scales by [5] could be promising, assuming that the distance still allows to compute a prototype pattern by gradient descent.

Relaxing the hypothesis of regular patterns would make pattern mining a difficult problem, as explained in [12]. But clustering is not the only way to build symbolic representations of time series. [9] and [13] show how to build symbolic representations of time series using segmentation and discretization. We intend to use those symbolic representations to extend our symbolic forecasting to encompass time series exhibiting patterns at irregular intervals. Furthermore, other means of building fuzzy prototypes will have to be investigated while keeping their interpretation as straightforward as possible.

References

1. Lotfi A.Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.
2. L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and regression trees. Technical report, Wadsworth International, Monterey, CA, 1984.
3. C.-C. Chang and C.-J. Lin. Ijcnn 2001 challenge: Generalization ability and text decoding. In *IJCNN*, 2001.
4. C.Holt. Forecasting seasonals and trends by exponentially weighted moving averages. Technical report, ONR Research Memorandum, Carnegie Institute, 1957.
5. Selina Chu, Eamonn Keogh, David Hart, and Michael Pazzani. Iterative deepening dynamic time warpping. In *Second SIAM International Conference on Data Mining*, April 2002.
6. C. L. Giles, S. Lawrence, and A. C. Tsoi. Noisy time series prediction using recurrent neural networks and grammatical inference. *Machine Learning*, 44(1–2):161–183, July/August 2001 2001.
7. Georges Hebrail and Bernard Hugueney. Symbolic representation of long time-series. In *Workshop of the 4th European Conf. Principles and Practice of Knowledge Discovery in Databases, PKDD*, 2000.
8. Bernard Hugueney. *"Repr´esentations symboliques de longues séries temporelles"*. PhD thesis, LIP6, 2003.
9. Bernard Hugueney and Bernadette Bouchon-Meunier. Time-series segmentation and symbolic representation, from process-monitoring to data-mining. In B. Reusch, editor, *Computational Intelligence, Theory and Applications*, number 2206 in Lecture Notes in Computer Science, LNCS, pages 118–123. Springer-Verlag, 2001.
10. Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. Locally adaptive dimensionality reduction for indexing large time series databases. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 30(2):151–162, June 2001.
11. Eamonn Keogh and Michael J. Pazanni. An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In David Heckerman, Heikki Mannila, Daryl Pregibon, and Ramasamy Uthurusamy, editors, *Proceedings of the Forth International Conference on Knowledge Discovery and Data Mining (KDD-98)*. AAAI Press, 1998.
12. Eamonn J. Keogh, Jessica Lin, and Wagner Truppe. "clustering of time series subsequences is meaningless: Implications for past and future research". In *ICDM*, pages 115–122. IEEE Computer Society, 2003.
13. Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 2–11. ACM Press, 2003.
14. T. Y. Lin. Data mining: Granular computing approach. *Lecture Notes in Computer Science*, 1574:24–33, 1999.
15. David Meyer. Naive time series forecasting methods. *R News*, 2(2):7–10, June 2002.
16. Klaus-Robert Muller, Alex J. Smola, Gunnar Ratsch, Bernhard Scholkopf, Jens Kohlmorgen, and Vladimir Vapnik. Predicting time series with support vector machines. In *ICANN*, pages 999–1004, 1997.

17. P. Winters. Forecasting sales by exponentially weighted moving averages. *Management Science*, 6(7):324–342, 1960.
18. Byoung-Kee Yi and Christos Faloutsos. Fast time sequence indexing for arbitrary L_p norms. In Amr El Abbadi, Michael L. Brodie, Sharma Chakravarthy, Umeshwar Dayal, Nabil Kamel, Gunter Schlageter, and Kyu-Young Whang, editors, *VLDB 2000, Proceedings of 26th International Conference on Very Large Data Bases, September 10–14, 2000, Cairo, Egypt*, pages 385–394, Los Altos, CA 94022, USA, 2000. Morgan Kaufmann Publishers.
19. Byoung-Kee Yi, H. V. Jagadish, and Christos Faloutsos. Efficient retrieval of similar time sequences under time warping. In *14th International Conference on Data Engineering (ICDE'98)*, pages 201–208, February 1998.

Fuzzy Prototypes Based on Typicality Degrees

Marie-Jeanne Lesot[1], Laure Mouillet[1,2], and Bernadette Bouchon-Meunier[1]

[1] LIP6, 8 rue du capitaine Scott, 75 015 Paris, France
{Marie-Jeanne.Lesot, Bernadette.Bouchon-Meunier}@lip6.fr
[2] Thales Communications, 160 Bd de Valmy, BP 82, 92704 Colombes Cedex, France
Laure.Mouillet@fr.thalesgroup.com

Abstract. This paper considers the task of constructing fuzzy prototypes for numerical data in order to characterize the data subgroups obtained after a clustering step. The proposed solution is motivated by the will of describing prototypes with a richer representation than point-based methods, and also to provide a characterization of the groups that catches not only the common features of the data pertaining to a group, but also their specificity. It transposes a method that has been designed for fuzzy data to numerical data, based on a prior computation of typicality degrees that are defined according to concepts used in cognitive science and psychology. The paper discusses the construction of prototypes and how their desirable semantics and properties can guide the selection of the various operators involved in the construction process.

1 Introduction

Clustering [9, 8] provides a simplified representation of a dataset by decomposing it into homogeneous and distinct subgroups called clusters. Each cluster can then be described by a prototype: it is a unique individual which characterizes the data subgroup; the set of prototypes can be used to summarize the initial dataset. Our paper focuses on the prototype construction.

Most of the time, each cluster is mapped to a unique point, called its center. It can be defined in various ways, for instance as the group average – as in the k-means algorithm – or as a weighted average – as in the fuzzy c-means. This choice is based on the assumption that the average characterizes the group, which is not always the case; other points can be chosen, as for instance the median or the *Most Typical Value* [5].

Our hypothesis is that a prototype intrinsically is a *fuzzy* concept: human reasoning is not based on precise numerical values, but rather on imprecise notions. For instance, considering data describing the height of persons from various countries, it cannot be said that the typical French person is 1.6834m tall (fictitious value), which is the kind of result one would get using

M.-J. Lesot et al.: *Fuzzy Prototypes Based on Typicality Degrees*, Advances in Soft Computing **2**, 125–138 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

an average-based definition. It seems more appropriate to say that the typical French person is "rather tall". This linguistic term conveys an imprecise notion which is best modelled using the fuzzy subset framework: the latter makes it possible to formalize categories with flexible and unclear boundaries, and thus prototypes as naturally handled by human beings.

Therefore, we consider the task of building fuzzy prototypes, defined as individuals described by fuzzy attribute values. The problem is then to construct, for each group, a fuzzy set describing the typical values of the numerical data pertaining to the group.

Rifqi [16, 15] proposes a method to build fuzzy prototypes for fuzzy data, i.e. data described by a set of attributes with values defined by fuzzy sets. This method rests on a definition of typicality inspired from cognitive science and psychology: Rosch [18] showed that all members of a category do not have the same representativeness and that the typicality of an instance depends both on its resemblance to the individuals pertaining to the same group and its dissimilarity with the individuals pertaining to the other groups. Our aim is to adapt this method to crisp data.

The paper is organized as follows: section 2 describes methods that provide enriched representatives for data subgroups, as compared to point-based descriptions. Section 3 presents the method we propose to handle crisp data and it discusses the notion of prototype and its associated properties. Lastly section 4 illustrates the results obtained on a real dataset.

2 State of the Art

2.1 Enriching Point Description

A first approach to obtain rich cluster descriptions is to apply fuzzy clustering algorithms [8]: they provide fuzzy subsets for each cluster. Yet these fuzzy subsets represent an imprecise decomposition of the database and model datapoints which have unclear membership and partially belong to several clusters simultaneously. Fuzzy prototypes are to be described by other fuzzy subsets: they do not aim at describing the cluster as a whole and representing all its points, but at extracting its most typical features and defining a relevant summary to provide a characterization of the subgroup. As a consequence, prototypes are expected to be more specific fuzzy subsets than the fuzzy clusters.

Thus fuzzy clustering methods provide a distribution in addition to the cluster centers, but this distribution cannot be applied directly to define fuzzy prototypes. Likewise, other point-based prototypes can be extended to distributions, but these distributions do not describe prototypes: one can associate a gaussian distribution to the arithmetic mean; yet, it is defined in such a way that each point in the cluster has a sufficiently high probability and it does not aim at defining a representative characterizing the group itself.

Table 1. Typicality-based algorithm to build a fuzzy prototype for a cluster g [15].

Given a resemblance measure r and a dissimilarity measure d [16],

1. Compute the typicality degree of each individual x in the group g
   a) Compute the internal resemblance $R_g(x)$ as the aggregation of the resemblance of x to the other individuals pertaining to the same group: $R_g(x) = ag_{y \in g}(r(x, y))$
   b) Compute the external dissimilarity $D_g(x)$ as the aggregation of the dissimilarity of x to the individuals pertaining to the other groups: $D_g(x) = ag_{y \notin g}(d(x, y))$
   c) The typicality $T_g(x)$ is the result of the aggregation of $R_g(x)$ and $D_g(x)$
2. The prototype is constructed by aggregating the individuals that are "typical enough", i.e. individuals whose typicality degree is higher than a pre-defined threshold.

More generally, fuzzy subset elicitation methods [1, 13] are techniques that are explicitely designed to provide fuzzy subsets describing the data. Some of them involve interaction with a human expert, others are based on partitioning methods [12, 6, 7]. Many belong to the parametric framework, i.e. consist in deciding on a desired form for the membership function, e.g. trapezoidal, gaussian or fuzzy c-means like, and optimizing its parameters so as to obtain a satisfying representative; the difficulty is then to define an appropriate cost function.

The prototype construction method we propose belongs to the elicitation technique framework, but it considers a specific characterization task: it aims at building a membership function which summarizes the data subgroup and does not describe it globally. It is based on the definition of typicality degrees: the membership functions are not directly constructed from the data but from an abstraction of the data in the form of typicality degrees.

2.2 Typicality-Based Approach

The method proposed in [16, 15] to construct fuzzy prototypes for fuzzy data uses the notion of typicality defined by Rosch [18]: the typicality of a datapoint is a function of both its within-class resemblance and its dissimilarity to other classes, as detailed in table 1. In the sequel, we refer to it as *Rosch typicality*. The prototype is then defined as an aggregation of the most typical datapoints.

It is to be noted that other typicality definitions exist, as the one used by Pal *et al.* [14] or the one underlying the Most Typical Value [5]. Yet these definitions are closer to "exceptionality" coefficients than typicality degrees, insofar as they are only based on the notion of a deviation with respect to a center. They can be interpreted as internal resemblance measures, which makes them a special case of the *Rosch typicality*.

3 Typicality-Based Prototype for Crisp Data

In this section, we discuss a method based on the *Rosch typicality* for crisp data. Considering crisp data makes a major difference with fuzzy data: in the latter case indeed, a fuzzy prototype is described with the same "language" as the considered data, i.e. fuzzy sets. In the case of crisp data, the prototype has a richer description than the datapoints.

After some general remarks on the typicality approach to fuzzy prototypes, we discuss the three steps of the algorithm described in table 1, with regard to the desired semantics and properties of the prototype. At each step, we discuss the parameters and the choices that have to be made and we illustrate their consequences on the classic iris database[1] [4]. In the first place, the resemblance and dissimilarity definitions must be defined for the crisp case. Then, their aggregation to compute the typicality degree has to be studied. The last step consists in determining how typical values are aggregated so as to obtain the prototype.

3.1 General Remarks

In the following, we consider that each group can be represented by a single prototype. This assumption is valid in our clustering framework: groups correspond to clusters and are thus homogeneous enough to be associated with a single representative.

It is to be noticed that the definition of a fuzzy prototype based on typicality degrees entails the definition of three different distributions:

- the membership function which indicates the extent to which a datapoint belongs to a cluster; this distribution is only available if clusters are obtained using a fuzzy clustering algorithm;
- the typicality distribution which indicates the extent to which a datapoint is typical of a group;
- the membership function describing the prototype itself.

Relationships between these distributions can be established: for instance, it seems natural to consider that for each group and each datapoint, the typicality is lower than the membership degree to the cluster. This expresses the fact that a datapoint which has a low membership cannot be typical; it also imposes that a point which is totally typical of a group totally belongs to it.

Lastly, prior to the prototype construction, one must decide between two approaches : the algorithm can be performed either on the individual as a whole, or it can be applied attribute by attribute. Taking into account all attributes makes it possible to represent correlations between attributes, which

[1] Note that for this database, labels are provided, so a preliminary clustering phase is not necessary, we construct one prototype for each of the three classes.

Table 2. Some examples of resemblance and dissimilarity measures for two vectors x and y; $\mathrm{dist}(x, y)$ denotes a distance between x and y, Z is a normalizing factor, z a normalizing function, γ, σ, d_M and Γ are hyperparameters.

Resemblance measures, R	
scalar product	$(x \cdot y)$
distance-based	$1 - \frac{1}{Z}\mathrm{dist}(x, y)$
polynomial	$\left(\frac{(x \cdot y)}{Z} + 1\right)^{\gamma}$
gaussian	$\exp\left(-\frac{1}{2\sigma^2}\mathrm{dist}(x, y)^2\right)$
Fermi-Dirac [17]	$z\left(\frac{1}{1 + \exp\frac{\mathrm{dist}(x,y) - d_M}{\Gamma}}\right)$
Dissimilarity measures, D	
normalized distance	$\frac{1}{Z}\mathrm{dist}(x, y)$
measures depending on a resemblance measure	$1 - r(x, y)$ $\frac{1}{1 + r(x, y)}$

can provide useful information about the prototype. On the other hand, working attribute by attribute can emphasize attributes that have no typical values for a given group, and can therefore be removed from the prototype description, thereby simplifying the group description.

3.2 Similarity and dissimilarity measures

In the case of fuzzy data, the resemblance and dissimilarity measures involved in the first step of prototype construction apply to fuzzy sets. They are defined in the formal framework of comparison measures [16] as functions with values in the interval $[0, 1]$ depending on a triple characterization of the relative position of the two compared subsets A and B: they depend on their common elements (i.e. a fuzzy set measure of the intersection $A \cap B$) and on their distinctive elements (a fuzzy set measure of their set differences $A - B$ and $B - A$). For crisp data, the range of possibilities is reduced because the information on the relative positions of two points can only be measured with one quantity corresponding to a distance (or equivalently a dot product). Table 2 mentions some possibilities.

Figure 1 illustrates three types of measures, using the iris database. For visualization sake, we only consider a single attribute, the petal length (third one); the histograms represent the datapoint distribution. On the graphs on

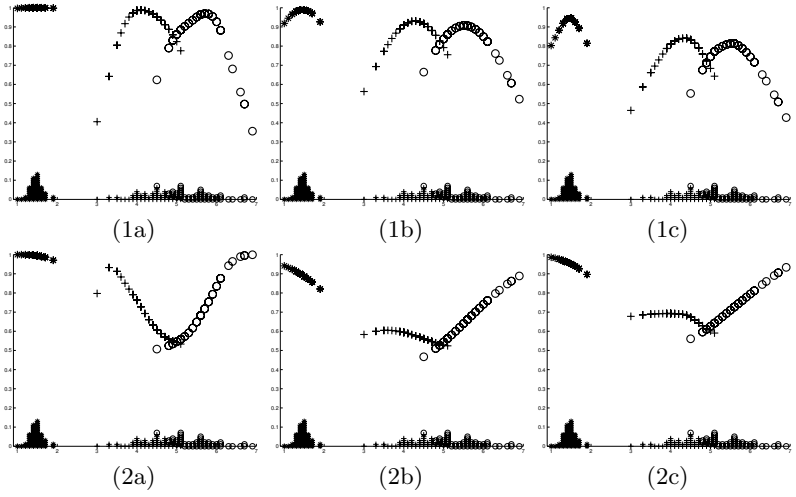


Fig. 1. Examples of comparison measures for the third attribute of the iris database. (1) Internal resemblance (a) Fermi-Dirac [17], (b) gaussian, (c) distance-based; (2) External dissimilarity (a) Fermi-Dirac, (b) gaussian, (c) normalized distance.

the first (resp. second) row, the y-axis represents the internal resemblance (resp. external dissimilarity) measure for each datapoint. One can observe that the three measures lead to different values, but usually to the same datapoint ranking within subgroups. These graphs show that the resemblance is maximal for a central point, whereas the most dissimilar points correspond to extreme values. Whichever measure is used, the middle subgroup does not appear very dissimilar as it is too close to the other clusters.

As in any learning problem, the choice of a distance measure has a major influence on the results. It seems natural to select it according to the distance used in the clustering step, but the choice may also be guided by the problem at hand, which may impose the semantics of the resemblance or the dissimilarity. Each measure has its own semantics: the Fermi-Dirac measure [17] enables the user to monitor the decrease speed as a function of the distance increase using two hyperparameters, which makes it a flexible comparison measure; in the multi-dimensional case, the polynomial resemblance takes into account correlations between attributes.

Another interesting issue is the duality of the resemblance and the dissimilarity measures. In the comparison measure framework [16], dissimilarity measures are defined independently from the resemblance measures; in particular, they are not necessarily dual, i.e. defined as the complement to 1 of each other. In the crisp case, both are defined as a function of a single quantity (a distance), thus duality seems a natural property of the couple. Yet it is still possible to define non-dual measures, the difference can simply come from

normalizing factors: two datapoints of the same class can be considered to be totally similar at a distance d_1 which differs from the distance d_2 at which points from two classes have a null dissimilarity. Such a choice implies defining a double semantics for the problem, and two different approaches to data comparison. On the contrary, duality can be justified considering the case of a datapoint x situated at an equal distance both to a point y pertaining to the same group and a point z belonging to another class: duality then expresses the fact that x is as similar to y as it is dissimilar to y.

3.3 Aggregation Operators

The second step of prototype construction consists in aggregating the internal resemblance and external dissimilarity to obtain a typicality degree. The aggregation has a strong impact on the semantics of the prototype. There exist many works on aggregation operators [3, 2], we consider here some properties which may be desired and their consequences on the aggregator choice.

Classic methods such as the mean, the Most Typical Value [5] or the probabilistic approach, only take into account data pertaining to the group to determine the group representative. They all choose the central element to represent the group, with varying definitions of "centrality". In our framework, this behavior can be obtained in the degenerate case where the aggregator only takes into account the internal resemblance and not the external dissimilarity. In this case, the typicality degree corresponds to the resemblance measure as illustrated by the first row on Fig. 1. In the same way, although this is not as intuitive, it is conceivable to choose a degenerate aggregator that only takes the external dissimilarity into account: a datapoint is considered typical if it makes it possible to conclude about the class membership; the typicality notion can then be interpreted as some kind of discrimination power (cf. second row on Fig. 1).

In real data applications, the central and the discriminative elements often coincide insofar as the center of a class is a discriminative element [18]. Yet if the prototype is constructed attribute by attribute, a non discriminative center may be encountered. The relative weight given to the centrality and to the discrimination power in the typicality degree computation can be tuned thanks to the aggregation operator.

In the sequel, we study the impact of the aggregation operator on the prototype semantics taking up the aggregator categorization from [3] which distinguishes between constant and variable attitude aggregators. We illustrate their effects on Fig. 2, considering the iris third attribute and the Fermi-Dirac comparison measures (graphs (1a) and (2a) on Fig. 1).

Constant Behavior Operators

Constant attitude operators can be categorized into three groups, distinguishing conjunctive operators, such as the logical AND, disjunctive ones, such as

the logical OR, and intermediate ones, also called tradeoff operators, such as weighted means.

Choosing a conjunctive operator (graph (a) on Fig. 2) implies that a datapoint can be considered typical if and only if it is both similar to the points of its cluster and distinct from other classes. This is a severe definition of typicality which leads to low values on average.

On the contrary, choosing a disjunctive operator implies that a datapoint can be considered typical either if it is discriminative or if it is central to the group. This leads to two kinds of typical points, as shown on graph (b) on Fig. 2: the extreme right points have a high typicality degree despite their low internal resemblance because of their high dissimilarity, the central point is typical for another reason, namely its high internal resemblance.

Lastly, tradeoff operators (graph (c) on Fig. 2) have an intermediate behavior, in particular, they possess the compensation property: a decrease in one criterion can be offset by an increase in another one for a constant result value. For instance, the extreme right point has a higher typicality degree than on graph (a) (min operator): its high dissimilarity compensates for its low resemblance.

Variable Behavior Aggregators

Variable behavior aggregators are conjunctive, disjunctive or tradeoff depending on the values that are aggregated as for instance the MICA operator [10]:

$$mica(x, y) = \max(0, \min(1, k + (x - k) + (y - k))) \tag{1}$$

where $k \in [0, 1]$ is a hyperparameter which determines the position of the change in behavior: if both values are high, i.e. higher than the neutral value k, $mica(x, y) \geq \max(x, y)$, so the operator has a disjuntive behavior; if both values are lower than k, the operator has conjunctive behavior. As the MICA operator is disjunctive for high scores and conjunctive for low scores, it possesses the full reinforcement property [2]: two low values are aggregated into an even lower value and two high values lead to an even higher result. This is illustrated on graph (d) obtained with $k = 0.6$, where many points have a typicality of 1, due to their internal resemblance and external dissimilarity higher than the threshold value k. On the contrary, for the extreme left point of the right cluster, the obtained value is lower than the one obtained with the min operator.

The symmetric sum [19] is another example of variable attitude aggregator:

$$symSum(x, y) = \frac{g(x, y)}{g(x, y) + g(1 - x, 1 - y)} \tag{2}$$

where g is a continuous, increasing, positive function satisfying $g(0, 0) = 0$. For this non-linear operator, the change in behavior occurs at threshold 0.5. For some choices of g, one also observes the full reinforcement property. Graph (e) on Fig. 2 illustrates this property with $g(x, y) = x \cdot y$.

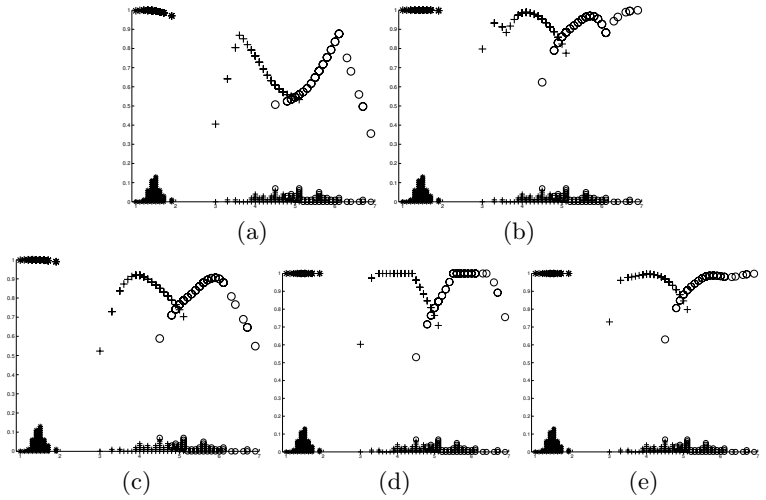


Fig. 2. Examples of aggregators applied with the Fermi-Dirac comparison measures on the 3rd attribute of the iris database. (a) $t = \min(R, D)$, (b) $t = \max(R, D)$, (c) $t = 0.7R + 0.3D$, (d) $t = mica(R, D)$ cf. (1) with $k = 0.6$, (e) $t = symSum(R, D)$ cf. (2) with $g(x, y) = x \cdot y$.

There is no way to define a priori which aggregation operator to choose: this choice depends on the considered application and the intended use of the constructed prototype.

3.4 Typical Data Aggregation

The last step consists in building the prototype itself; the objective is to define it as an aggregation of the most typical datapoints. Thus one must aggregate crisp values into a fuzzy subset, which is similar to the initial problem of building fuzzy prototypes characterizing crisp data. The difference is that the typicality degrees extracted from the initial data provide some kind of abstraction. Thus they may justify simple solutions, which are not applied to raw data, but to enriched data containing more information than the initial ones.

We propose to define two thresholds, τ_k and τ_s, that indicate the minimum typicality required to be included respectively in the prototype kernel and in the support. To perform the in-between interpolation, different solutions can be considered: one can simply use a parametric approach and build the trapezoidal membership function having the previous kernel and support. One can also perform a linear rescaling of the interval $[\tau_s, \tau_k]$ to $[0, 1]$ to get the membership degrees to the prototype. Note that in the attribute by attribute construction, both methods lead to the same prototypes.

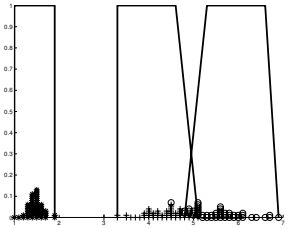


Fig. 3. Fuzzy prototype for the third attribute of the iris database, using Fermi-Dirac comparison measures and MICA aggretator.

Figure 3 illustrates the three prototypes obtained for the third attribute of the iris database. For this example, as the classes are quite homogeneous and well separated, almost all values have high membership degrees to the prototype of their class. The exceptions are located in the overlapping zone between two classes.

4 Experimental results

In this section, we consider a real two-dimensional dataset which describes the results of 150 students after two exams, shown on Fig. 4. We illustrate the construction of prototypes characterizing the various groups of students with the previously presented method. The prototypes have been constructed globally (not attribute by attribute) because there were obvious correlations between the two attributes.

Table 3 sums up the various notations and measures used in the sequel.

Clustering

The first step consists in clustering the dataset so as to decompose it into homogeneous subgroups. We applied the fuzzy c-means with an automatic choice of c based on the stability of the objective function [11], which led to $c = 5$. The results are shown on Fig. 4.

Resemblance and Dissimilarity Measures

The second step consists in computing the comparison measures. It was natural to base our resemblance and dissimilarity measures on the euclidian distance used in the clustering step, but they had to be normalized so as to take their values in $[0, 1]$. The normalization factor chosen for the dissimilarity measure indicates the distance from which two values are considered totally dissimilar; it is set to half the data diameter, the data diameter being the maximal distance between pairs of datapoints in the dataset.

Table 3. Notations and measures, where X denotes the set of examples, and $||x-y||$ the euclidian distance between x and y.

dissimilarity normalization factor	$Z_d = \frac{1}{2} \max_{x,y \in X} ||x - y||$
dissimilarity measure	$d(x,y) = \min\left(1, \frac{||x-y||}{Z_d}\right)$
cluster diameter for cluster g	$diam(g) = \max_{x,y \in g} ||x - y||$
resemblance normalization factor	$Z_r = \max_g diam(g)$
resemblance measure	$r(x,y) = \max\left(0, 1 - \frac{||x-y||}{Z_r}\right)$

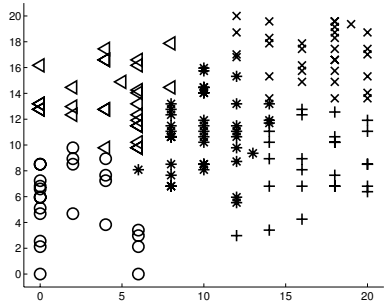


Fig. 4. Clustering results obtained on a 2D dataset. Each symbol depicts a different cluster.

The normalization factor chosen for the resemblance measure is the maximal cluster diameter, the diameter of a cluster g being defined as the maximal distance between pairs of datapoints pertaining to g.

The first two columns of Fig. 5 illustrate these choices for two clusters, the central and lower left ones (depicted by $*$ and $\circ$ on Fig. 4) and represent contour plots of the resemblance and dissimilarity measures respectively. The resemblance contours appear circular, which is due to the point distribution and the measure choice; the point maximizing the internal resemblance is the cluster average. The dissimilarity is maximal for extreme points, it is slightly higher for the left lower cluster than for the central one, which is too close to the four other clusters.

Typicality Degree Computation

We chose a variable behavior operator, the symmetric sum, introduced in (2), with $g(x,y) = x \cdot y$. This aggregator rewards datapoints that have a high internal resemblance (greater than the neutral value 0.5) and a high external dissimilarity; if both values are significantly high, they reinforce each other.

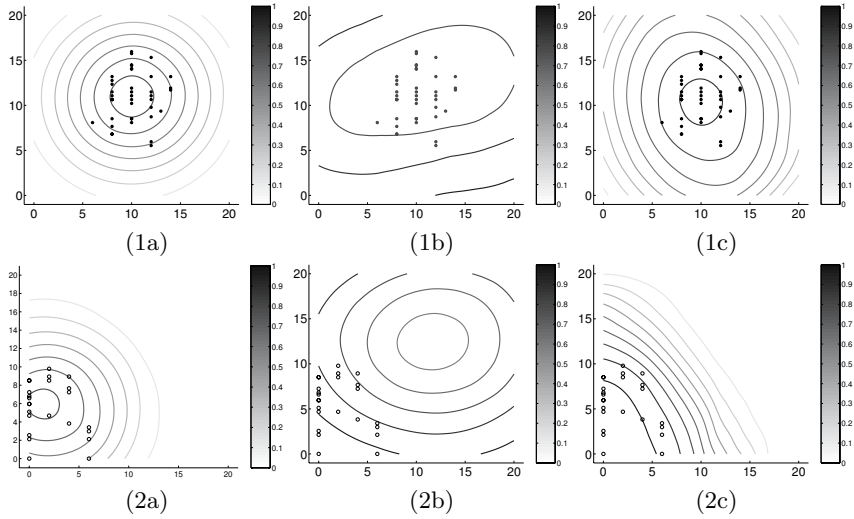


Fig. 5. Internal resemblance, external dissimilarity and typicality distribution for the central cluster (first row), depicted $*$ on Fig. 4, and the left lower cluster (second row), depicted by $\circ$.

Simultaneously, it sanctions datapoints that have a low internal resemblance (lower than 0.5) and a low external dissimilarity. In-between, a low internal resemblance can be offset by a high external dissimilarity and vice versa.

The results are illustrated on the last column of Fig. 5. It appears that the typicality distribution reflects the resemblance, modified to a certain extent by the dissimilarity influence. Thus typical values are not central values (maximizing the internal resemblance), but close values determined taking into account the external dissimilarity. These observations are consistent with the cognitive science results.

Typical Values Aggregation

The last step consists in aggregating the most typical values of each cluster into a fuzzy subset which characterizes the cluster. We consider that a datapoint whose typicality degree is higher than $\tau_k = 0.9$ totally belongs to the prototype, and we exclude from the prototype support the points having a typicality lower than $\tau_s = 0.7$. The interpolation between the so-defined kernel and support is performed through a linear rescaling of the interval $[\tau_s, \tau_k]$ on the interval $[0, 1]$.

The obtained results are illustrated on Fig. 6 which shows the contour plots of the five prototype membership functions. Their kernels contain values which simultaneously have a high internal resemblance and a low external dissimilarity. Our method enabled us to extract characterizations for the

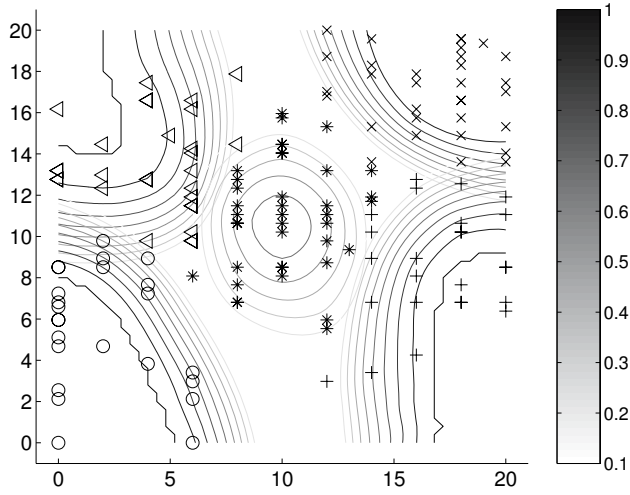


Fig. 6. Superimposition of the 5 obtained fuzzy prototypes.

five groups revealed by the clustering step. We think that these results can be improved if all the datapoints are not taken into account with the same importance in the typicality degrees computations: fuzzy clustering provides membership functions that indicate to what extent a datapoint belongs to several clusters. Intuitively, a datapoint that completely belongs to a group has to have more importance in the prototype computation than a datapoint that poorly belongs to it.

5 Conclusion

We propose a parametric fuzzy prototype construction method that consists in aggregating typical individuals to represent an homogeneous data subgroup. This method involves a certain number of parameters that must be chosen according to the desired semantics and properties of the prototype. Although these depend on the data and the application of the prototypes, it is possible to give insights on the methodology for choosing the appropriate parameters. For instance, if the prototype is used to classify new examples, its discrimination power should be emphasized, whereas if it is used to provide an explanation of the data, its centrality should be favored.

Our perspectives are to study in more detail the interrelations between the parameters and their possible redundancy. Another interesting issue is to study how the method can take into account fuzzy clusters: if they are provided by the clustering algorithm, the membership degrees to the various clusters could be used in the process, for instance to reduce the weight of datapoints that only partially belong to the cluster.

References

1. N. Aladenise and B. Bouchon-Meunier. Acquisition de connaissances imparfaites : mise en évidence d'une fonction d'appartenance. *Revue Internationale de systémique*, 11(1):109–127, 1997.
2. M. Detyniecki. *Mathematical aggregation operators and their application to video querying*. PhD thesis, Université de Paris VI, 2000.
3. V. Eude, B. Bouchon-Meunier, and E. Collain. Choix d'opérateurs d'agrégation pour l'évaluation de structures hiérarchisées. In *Proc. of IPMU'98*, pages 778–785, Paris, France, 1998.
4. R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:170–188, 1936.
5. M. Friedman, M. Ming, and A. Kandel. On the theory of typicality. *Int. Journal of Uncertainty, Fuzzyness and Knowledge-Based Systems*, 3(2):127–142, 1995.
6. A. Gyenesei. Fuzzy partitioning of quantitative attribute domains by a cluster goodness index. Technical Report 368, Turku Centre for Computer Science, 2000.
7. T.P. Hong and C.Y. Lee. Induction of fuzzy rules and membership functions from training examples. *Fuzzy Sets and Systems*, 84(1):33–47, 1996.
8. F. Höppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy Cluster Analysis, Methods for classification, data analysis and image recognition*. Wiley, 2000.
9. A. Jain, M. Murty, and P. Flynn. Data clustering: a review. *ACM Computing survey*, 31(3):264–323, 1999.
10. A. Kelman and R. Yager. On the application of a class of MICA operators. *Int. Journal of Uncertainty, Fuzzyness and Knowledge-Based Systems*, 7:113–126, 1995.
11. M.J. Lesot and B. Bouchon-Meunier. Descriptive concept extraction with exceptions by hybrid clustering. In *FuzzIEEE04*, Budapest, Hungary, 2004.
12. C. Marsala and B. Bouchon-Meunier. Fuzzy partitioning using mathematical morphology in a learning scheme. In *Proc. of the 5th IEEE Int. Conf. on Fuzzy Systems*, volume 2, New Orleans, USA, 1996.
13. S. Medasani, J. Kim, and R. Krishnapuram. An overview of membership function generation techniques for pattern recognition. *Int. Journal of Approximate Reasoning*, 19(3–4):391–417, 1998.
14. N. Pal, K. Pal, and J. Bezdek. A mixed c-means clustering model. In *FuzzIEEE97*, pages 11–21, 1997.
15. M. Rifqi. Constructing prototypes from large databases. In *Proc. of IPMU'96*, Granada, Spain, 1996.
16. M. Rifqi. *Mesures de comparaison, typicalité et classification d'objets flous : théorie et pratique*. PhD thesis, Université de Paris VI, 1996.
17. M. Rifqi, V. Berger, and B. Bouchon-Meunier. Discrimination power of measures of comparison. *Fuzzy sets and systems*, 110:189–196, 2000.
18. E. Rosch. Principles of categorization. In E. Rosch and B. Lloyd, editors, *Cognition and categorization*, pages 27–48. Lawrence Erlbaum associates, 1978.
19. W. Silvert. Symmetric summation: a class of operations on fuzzy sets. *IEEE Transactions on Systems, Man and Cybernetics*, 9:659–667, 1979.

Plenary Talk

The Power of Zadeh's Protoforms: Towards General Problem Formulations in Fuzzy Multistage Control and Group Decision Making

Janusz Kacprzyk

Systems Research Institute, Polish Academy of Sciences, Ul. Newelska 6, 01-447 Warsaw, Poland
kacprzyl@ibspan.waw.pl

Abstract. Recently, in many of his in.uential and stimulating talks, Zadeh has been advocating the concept of a protoform, which stands for a prototypical form, as a crucial tool for the formalization of human consistent reasoning, deduction capabilities of search engines, etc.

A protoform is de.ned as an abstract prototype. For instance, if we have a linguistically quanti.ed statement (which is relevant in our context), say "most experts are convinced" or "most of the competent experts are convinced", then they may lead to the following protoforms, respectively: "*Most R's* are *S*" and "*Most BR's* are *S*". Evidently, as protoforms may form a hierarchy, we can de.ne higher level (more abstract) protoforms, for instance replacing *most* by a general linguistic quantifier *Q*, we obtain, respectively: "*QR's* are *S*" and "*QBR's* are *S*", respectively. Protoforms can then be used for, e.g. devising general inference schemes, models, etc.

Basically, Zadeh points out the relevance of protoforms in the formalization of problems related to broadly perceived information technology and knowledge engineering. Examples of implementation of protoforms have been given by Kacprzyk and Zadrożny for generating families of linguistic database summaries and fuzzy queries.

In this talk we extend the area of application and relevance of Zadeh's protoform to decision making and control under fuzziness, and show – to be more specific that the use of protoforms may lead to general classes of models of fuzzy multistage (optimal) control, and group decision making and voting.

In particular, we discuss in the context of protoforms and their hierarchies a class of general multistage fuzzy control models, introduced by Kacprzyk, in which an optimal sequence of (fuzzy) controls is sought that maximizes a performance function over, for example, "most of the control stages" or "most of the earlier control stages". We show that if we use a more abstract form

J. Kacprzyk: *The Power of Zadeh's Protoforms: Towards General Problem Formulations in Fuzzy Multistage Control and Group Decision Making*, Advances in Soft Computing **2**, 141–142 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

of such a problem formulation, i.e. a more abstract protoform, then we can arrive at various problems with di.erent stage score aggregation schemes and di.erent discounting strategies.

In the context of group decision making, we consider .rst the case of a general model of group decision making under fuzzy preferences and fuzzy majority (given by a fuzzy linguistic quanti.er) proposed by Kacprzyk in which a solution (an option and/or fuzzy or non-fuzzy set of options) is sought that is "best accepted by most of the competent individuals". This new concept may lead to various fuzzy tournament base solutions as proposed by Kacprzyk and Nurmi, new measures of consensus given by Fedrizzi, Kacprzyk and Zadrożny, and was extensively developed by other people, notably Delgado, Verdegay, Herrera, etc.

We show that the replacement of such a group decision making choice function by its protoform can lead to many new classes of group decision making and consensus formation models. Notably, using some results of Kacprzyk and Zadrożny, we show that a wide majority of voting procedure (e.g. Borda, the Nurmi's minimax degree set, plurality voting, quali.ed plurality voting, approval voting, simple majority (Condorcet), Pareto, Copeland, Kacprzyk's Qminimax set, Kramer's minimax set, Condorcet-looser, Pareto inferior options, etc) can be represented by a protoform of a choice function. Presumably, this protoform may also represent other meaningful choice procedures.

The purpose of this paper is to present some classes of fuzzy multistage control and group decision making models from the point of view of Zadeh's concept of a protoform, and show that an additional insight into the existing models may be gained, and even new classes of models may be obtain that may grasp some other aspects that are di.cult to re.ect. We are sure that the power of Zadeh's protoforms may be proven in many other areas too.

Fuzzy Control

Fuzzy Logic Fluid Therapy Control System for Renal Transplantation

A. Yardimci[1], N. Hadimioglu[2]

[1] Department of Industrial Automation, School of Technical Higher Education, Akdeniz University, Antalya, Turkey
yardimci@akdeniz.edu.tr
[2] Departments of Anesthesia and Reanimation, Faculty of Medicine, Akdeniz University, Antalya, Turkey
hadimioglu@akdeniz. edu.tr

Abstract. In this study, a fuzzy logic system developed to control fluid balance during RT by changing in blood pressure (BP), heart rate (HR) and CVP value. Currently, the administration of intravascular volume is carried out primarily based on the experience and expertise of the anesthesiologist as there is no analytical method available to estimate the transplant patient's fluid level. The development of a control system to assist the anesthesiologist, so that he or she can devote attention to other tasks that can't yet be adequately automated during RT. To increase kidney viability in transplant surgery, patient safety and comfort during RT is one of the most important potential benefits of the system. For this project, 30 kidney transplantation operations, performed in Akdeniz University Organ Transplantation Center were completely followed and all physiological data recorded during last year. This data base is used for contribution of the fuzzy system membership functions and determine to base variable intervals. Also the developed system was tested with these operations records.

1 Introduction

Transplantation may be considered one sign of a country's development level. The successful melding of legal, ethical, medical, social, psychological, technological, economical, and religious aspects is mandatory for any transplantation organization. Like all similar programs in the world, transplantation activities in Turkey began with operations that broke new ground. Solid organ transplantation in Turkey began with two heart transplants in 1969, both of which, unfortunately, were unsuccessful. By the early 1970s, experimental studies on liver transplantation had already been initiated by Dr. Haberal and his team. On November 3, 1975 they performed the first renal transplantation in Turkey, with a kidney donated from mother to son. This was followed by the first cadaver kidney transplantation on October 10, 1978 using an organ supplied by the Euro Transplant Foundation.

Improved understanding and treatment of the comorbid conditions associated with ESRD and improved perioperative and intraoperative treatment of patients undergoing kidney transplantation have resulted in increased long-term graft survival, and have reduced morbidity and mortality. Kidney transplantation is

A.Yardimci and N.Hadimioglu: *Fuzzy Logic Fluid Therapy Control System for Renal Transplantation*, Advances in Soft Computing **2**, 145–158 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

becoming a possibility for wider population of high-risk patients who were, until recently, considered at prohibitive risk. Improved anaesthetic management with tight hemodynamic control and fluid management contribute to improved short- and long-term outcome of renal transplantation. Over half of the patients undergoing kidney transplant surgery have at least one comorbid condition, most commonly diabetes mellitus, hypertension, congestive heart failure (CHF), coronary artery disease (CAD), pulmonary disease, or a previous malignancy [1,2,3]. The preoperative evaluation of the kidney transplant recipient is complex, and must include screening for potential dysfunction of multiple organ systems. Better understanding of disease mechanisms, improvements in diagnostic tools, and advances in therapeutic modalities over the last several decades have resulted in a measurable reduction in perioperative morbidity and mortality for patients with uremia [4].

In a typical human being around 55% to %60 of the body weight is fluid. The fluids plays important role in maintaining a healthy body. A normal person can regulate his/her own body fluid volume to an appropriate level through daily physiological activities. However, in a patient especially under kidney transplantation, because of the fluid abnormalities usually found in patients with renal failure, balancing the body's fluid volume level is carried out by mainly the anesthesiologist. Maximal hydration during renal transplantation is of utmost importance. The new kidney must have adequate perfusion to ensure optimal function. Intraoperative fluid balance is important because adequate intravascular volume is essential for maintaining perfusion to the new kidney. There is a definite relationship between renal graft perfusion and the incidence of acute tubular necrosis. A central venous pressure line is an essential intraoperative and postoperative monitor. A central venous pressure of 10 – 15 mm Hg at the time the arterial clamps are released is recommended. Delayed primary function of the transplanted kidney has been studied extensively. There is overwhelming consensus that the "filling effect" is important for immediate graft function. Usually, anesthesiologist manage patients with chronic renal failure by keeping them "dry" to prevent fluid overload and to avoid the need for postoperative dialysis. In contrast, the transplant patient must be "wet" with a supernormal intravascular volume to ensure luxury flow to the new renal graft. Anesthesia for the uremic patient undergoing transplantation is unique in this respect, and the anesthesiologist must be aware of this extremely important difference [20,9,13]. At present, the input/output conceptual model acquired by the anesthesiologist through experience and observations is the main guideline for determining the fluid infusion rate, as there is no analytical approach available to model the dynamics of the process. The input variables of this conceptual model are physiological variables of the patient associated with the body fluid level. The output is the required liquid infusion rate.

2 RT Monitoring

Monitoring of recipients of renal transplants includes standard American Society of Anesthesiologists (ASA)–recommended 5-lead ECG-arterial blood pressure,

oxyhemoglobin saturation by pulse oximeter, end-tidal CO_2, temperature, and urinary output. Because of the fluid abnormalities usually found in patients with renal failure, the high incidence of cardiac disease, and the importance of graft perfusion and early function, central venous pressure monitoring is recommended. Monitoring of the central venous pressure lowers the risk for acute tubular necrosis and transplant failure caused by hypovolemia [11,17]. If the patient is positioned in the Trendelenburg position, the absolute values of central venous pressure may not be accurate; however, central venous pressure trends and the response of the central venous and arterial pressure to volume loading provide equally as useful clinical information. Direct arterial pressure monitoring also may be used, especially when easy access to frequent blood gas analysis is warranted [4].

Pulmonary artery catheters are seldom required during renal transplantation. Their use is limited to patients with poorly controlled hypertension, CAD with left ventricular dysfunction, valvular heart disease, and severe chronic obstructive pulmonary disease. In cases of unexplained refractory hypotension, more precise assessment of intravascular volume status can be performed by using transesophageal echocardiography. Transesophageal echocardiography can be used to differentiate between hypotension caused by hypovolemia or hypotension caused by decreased myocardial contractility (i.e., cardiac failure).

The viability of the renal allograft depends on immediate and adequate renal blood flow. High blood flow through the graft immediately on clamp release is associated with early renal function, a decreased mortality rate, and increased graft survival [7].

3 Intravascular Volume Factor Affecting Kidney Viability In Transplant Surgery

The most important intraoperative measure that improves the likelihood of immediate graft function is to maintain an adequate intravascular volume and ensure satisfactory perfusion of the transplanted kidney.

A central venous pressure of 10 mm Hg to 15 mm Hg has been recommended to maintain optimal intravascular volume. Aggressive intraoperative volume expansion is warranted because delayed renal function is associated with a 20% to 40% decrease in transplant graft survival and increased patient mortality [11,12] Postoperative acute tubular necrosis can result from inadequate hydration. Maintaining the mean pulmonary artery pressure over 20 mm Hg has been shown to reduce the occurrence of acute postoperative tubular necrosis [10]. While pursuing adequate hydration, however, the patient should be monitored for signs of fluid overload to avoid pulmonary edema, because many recipients have pre-existing cardiac disease (e.g., hypertensive cardiomyopathy, CHF, CAD).

4 Crystalloids and Colloids

Isotonic crystalloid solutions (Ringer's lactate solution and 0.9% saline solution) are very commonly used to compensate for general losses of water and electrolytes and are usually the first choice for fluid replacement. Isotonic crystalloid solutions do not contain oncotically active macromolecules. Therefore, their effect on

plasma volume expansion of approximately 200 ml for every 1000 ml administered, with an intravascular half-life of 20 to 30 min, is very limited [3] . To substitute for blood loss, crystalloid solutions must be infused in four- to fivefold greater amount, compared with colloid solutions, to exert the same volume effects. Moreover, it was demonstrated that crystalloids could not effectively restore microcirculatory blood flow in several organs in models of hemorrhagic shock. In the dynamic processes of systemic inflammatory response syndrome (SIRS) or sepsis, with increased transmembrane fluid flux and low plasma COP, fluid shift from the intravascular compartment to the interstitial compartment is promoted if crystalloids are exclusively infused. In addition to their ineffectiveness in restoring sufficient tissue perfusion, this phenomenon increases the risk for tissue edema, particularly in the lung and gut mucosa [3]. However, despite these results from experimental trials, an ongoing controversy exists regarding the use of crystalloids or colloids for adequate fluid replacement. To date, there has been no clinical trial with sufficient statistical power to compare the different fluids with respect to mortality rates.

Crystalloids have no specific nephrotoxic effects and are the basic fluids to fulfill the requirements for water and electrolytes in critically ill patients. In cases of major intravascular hypovolemia or severe sepsis, the exclusive administration of crystalloids is not appropriate because they are not able to sufficiently restore microcirculation, which is the major pathogenic factor in the development of multiple organ failure. Therefore, crystalloids should be used in conjunction with colloids to restore intravascular volume.

The controversy regarding the use of colloid or crystalloid solutions for volume resuscitation in critically ill patients has been reinitiated by the meta-analyses mentioned above. Because of their content of macromolecules, colloids are retained within the intravascular space to a much greater extent, resulting in a greater intravascular volume effect. The volume effect exerted by colloids and their volume-supporting capacity with time depend on their concentration, mol wt, molecular structure, COP, metabolism, and elimination rate.

5 Patients and Methods

The study was approved by the Hospital Ethics Committee. In Akdeniz University, in the operating room of the Organ Transplantation Center, for a year, data were collected from the patients, who underwent kidney transplantation operation, and who were administered isoflurane. For this project, 30 kidney transplantation operations, performed in Akdeniz University Organ Transplantation Center were completely followed and all physiological data recorded. All recorded operation data evaluated for the system. The data base was constructed of the records from a total of 30 patients (10 females, 20 males) who were asked permission in advance of the operation and were informed about the study. The patients mean age is 34.4 (range 15-57) yr, mean weight is 63.6 (37-85) kg and mean height is 164.4 (148-191) cm. During the operation every five minutes the BP, HR, and CVP were recorded. This data base is used for contribution of the fuzzy system membership functions and determine to base

variable intervals. Also the developed system was tested with these operations records.

6 Fuzzy Logic Fluid Infusion Rate Control System

Since the 1980s new techniques have appeared from which fuzzy-based has been applied extensively in medical systems. Although medicine is a science which isn't related to control engineering, it is being affected to such an extent that it is now possible to use available control techniques for on-line devices, especially during surgical operations and in intensive care units [16]. Nevertheless, no standard methods exist for transforming human experience into the rule base of the fuzzy inference control system. Beside this the shape of the membership functions associated with the linguistic expressions are determined by trial and error methods so that the performance index is optimised.

Because a biological process has a non-linear, time-varying structure and time-varying parameters, modelling it suggests the use of rule-based controllers like fuzzy controllers [23,24,25]. Fuzzy rule-based systems include many aspects of fuzzified values, such as the rules antecedents and consequence. The rules structure are usually of the form of *if... then*. In its basic form this type of the control is equivalent linguistically to a PI controller, and depending on the output, whether it is incremental or absolute, the controller is known as PI or PD respectively.

Blood pressure is most conveniently measured using a cuff, which should be of the right size for the patient. Systolic pressure can be detected by palpation of the brachial or radial artery or by auscultation. In anaesthetic practice, the systolic pressure has greater significance than the diastolic pressure, which isn't frequently recorded, particularly if access to the arm is difficult No "normal" blood pressure can be specified for the anaesthetized patient; in general the systolic pressure should be stable in the range 90-140 mmHg (12.0-18.7 kPa) [5,6]. But we can not say this range is valid for kidney transplant patients. Because, remember that, over half of the patients undergoing kidney transplant surgery have at least one comorbid condition, most commonly diabetes mellitus, hypertension, CHF, coronary artery disease CAD, pulmonary disease, or a previous malignancy. So, special BP, HR and CVP membership functions have to be contributed by anesthesia specialists. The membership functions and the rule base of the fuzzy logic system were determined under the inspection of specialists by abiding by the data base information.

6.1 System Structure

FuzzyTECH 5.52f MP Explorer, MPLAB 3.01 and MATLAB 6.01 programs were used for this application. But the main program has written by us.

The system structure identifies the fuzzy logic inference flow from the input variables to the output variables. The fuzzification in the input interfaces translates analog inputs into fuzzy values. The fuzzy inference takes place in rule blocks which contain the linguistic control rules. The output of these rule blocks is linguistic variables. The defuzzification in the output interfaces translates them

into analog variables. The following figure shows the whole structure of the Fuzzy Logic Liquid Infusion Rate Control System (FLLIRCS) including input interfaces, rule block and output interface. The connecting lines symbolize the data flow.

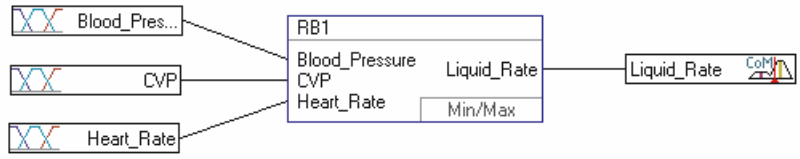


Figure 1. Structure of the FLLIRCS

6.2 Variables and membership functions

Linguistic variables are used to translate real values into linguistic values. The possible values of a linguistic variable are not numbers but so called 'linguistic terms'. Five linguistic variables and their associated membership functions are defined for each input variable, namely very_low, low, normal, high and very_high. They are also called fuzzy zones of the variables. The output variable is also divided into five fuzzy zones with the zone names zero, low, normal, high, very_high. The defined membership functions of the zones are shown in Figure 2. , 3. , 4. and 5. As mentioned before, these definitions are based on the consultation made with the anesthesia specialists.

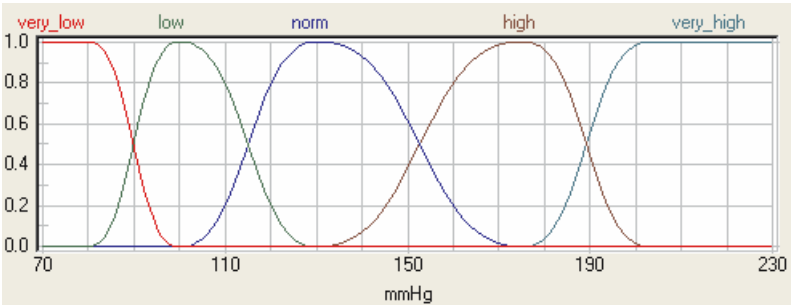


Figure 2. MBF of "Blood_Pressure"

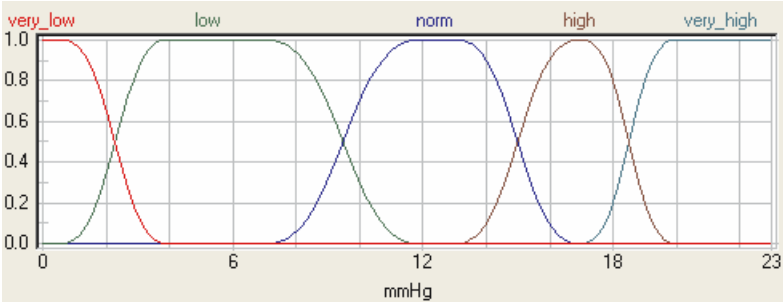


Figure 3. MBF of "CVP"

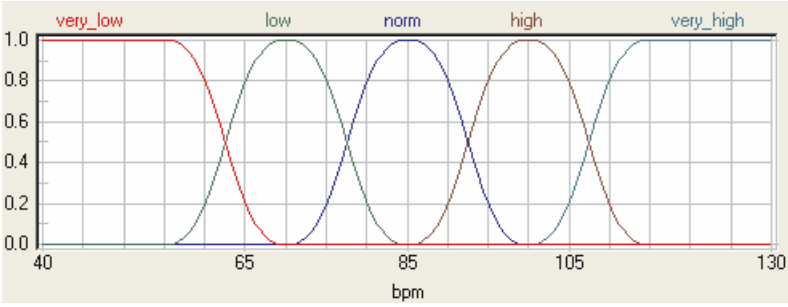


Figure 4. MBF of "Heart_Rate"

Figure 5. MBF of "Liquid_Rate"

6.3 Rule Base

The rule base contains the control strategy of a fuzzy logic system. The rules' 'if' part describes the situation, for which the rules are designed. The 'then' part describes the response of the fuzzy system in this situation. The degree of support (DoS) is used to weigh each rule according to its importance.

There are three inputs to the FLLIRCS and each has 5 fuzzy zones. Hence the number of rules in the rule base is equal to 125 (5 x 5 x 5). But, FLLIRCS Rule base includes 121 rules. 4 rules are eliminated by specialist. Eliminated rules are related to conditions which never happened. The rule base built based on the input/output data obtained from the RT operation records.

All the rules are extracted from the input/output vector obtained from the patient during RT. An input/output vector is the real data recorded for the input/output variables at an instant of time. For example, say, in the n^{th} minutes of a RT, the readings of the variables BP, HR, CVP and LR are a mmHg, b beats per minute, c mmHg and d milliliter per hour. These four values (a, b, c, d) form a vector of input/output data which represents one of the existing relationships between the input and output variables. The rule specified by an input and output vector has strength proportional to the degree of the membership of each variable to its fuzzy set. This strength is referred to as Rule Degree in this work and is defined as the product of the degrees of membership of the variables in the rule. It should be noted that in fuzzy theory that an element of a universe of discourse can be the member of more than one fuzzy set and membership function. For the gauss (S shape) membership function used in this work, each value of a variable is typically covered by two membership functions. This situation can be shown at Figure 3. , 4. , 5. , 6.

The processing of the rules starts with calculating the 'if' part. The operator type of the rule block determines which method is used. The characteristic of each operator type is influenced by an additional parameter.
For example:
MIN-MAX, parameter value 0=Minimum Operator (MIN)
MIN-MAX, parameter value 1=Maximum Operator (MAX)
GAMMA, parameter value 0=Product Operator (PROD)

The minimum operator is a generalization of the Boolean 'and'; the maximum operator is a generalization of the Boolean 'or'. The fuzzy composition eventually combines the different rules to one conclusion. If the BSUM method is used all firing rules are evaluated, if the MAX method is used only the dominant rules are evaluated.

Most fuzzy logic based application solutions use production rules to represent the relationship between the linguistic variables and to derive actions from sensor inputs. Production rules consist of a precondition (IF-part) and a consequence (THEN-part). The IF-part can consist of more than one condition linked by linguistic conjunctions like AND and OR. The computation of fuzzy rules is called fuzzy rule inference. The software which we used for this application FuzzyTECH, calculates the inference in two steps: input aggregation and composition with degree of support (DOS). Aggregation uses fuzzy logic operators to calculate the result of the IF part of a production rule when the rule consists of more than one input conditions. One of the linguistic conjunctions, AND or OR, links multiple input conditions. Composition uses the fuzzy logic operator, PROD, to link the input condition to the output condition.

Composition links the validity of the entire condition with the DOS. Thus, composition, the second calculation step of each production rule, uses the validity of the condition to determine the validity of the consequence. In standard MAX-MIN or MAX-PROD inference methods, the consequence of a rule is considered equally as true as the condition.

The defuzzification approach employed in this study is the Tsukamato's model. The mathematical expression of this model is shown in (1), in which crisp value Z is calculated by;

$$Z = \frac{\sum_{i=1}^{n} W_i Z_i}{\sum_{i=1}^{n} W_i} \tag{1}$$

where n is the the number of rules in the rule base. W_i is the degree of membership of the output variable in the i^{th} rule. Z_i is the value of that output variable when the degree of membership is equal to 1 in the i^{th} rule. This method is common and simple.

This method can be shown as identical to Center-of-Area (CoA) using singleton membership functions when each membership function for the linguistic variable to be defuzzified has been defined so that its maximum is 1 and is located at the position of the respective singleton. In these systems, the membership function overlap in the result is ignored. Thus the individual membership function areas can be calculated during compilation time and therefore these areas only need to be multiplied with the inference results at runtime.

To check the performance of fuzzy system lets look an example; the process of rule extraction from a vector with BP, CVP, HR, LR values of 120, 9, 74, 0.544 respectively, is illustrated in Tables 1 and 2. Figure 6 shows this data inputs on the membership functions. The membership functions and corresponding membership degrees for each variable are given Table 5. In this table $A(i), B(i), C(i), D(i), (i = 1,2)$ are the membership functions respectively the fuzzy values of variables BP, CVP, HR and LR. The parameters $W(i), X(i), Y(i), Z(i), (i = 1,2)$ are the membership degrees produced by membership functions of each variable. The 8 rules obtained from the input/output vector are provided Table 6.

Table 6. The corresponding membership degrees of the data vector

Variable	Readings	Covered degrees		
BP	120 mmHg	A(1)=low, W(1)=0,21	A(2)=normal, W(2)=0,79	
CVP	9 mmHg	B(1)=low, W(1)=0,69	B(2)=normal, W(2)=0,31	
HR	74 bpm	C(1)=low, W(1)=0,88	C(2)=normal, W(2)=0,12	
LR	0,544 lt/hr	D(1)=low, W(1)=0,31	D(2)=normal, W(2)=0,21	D

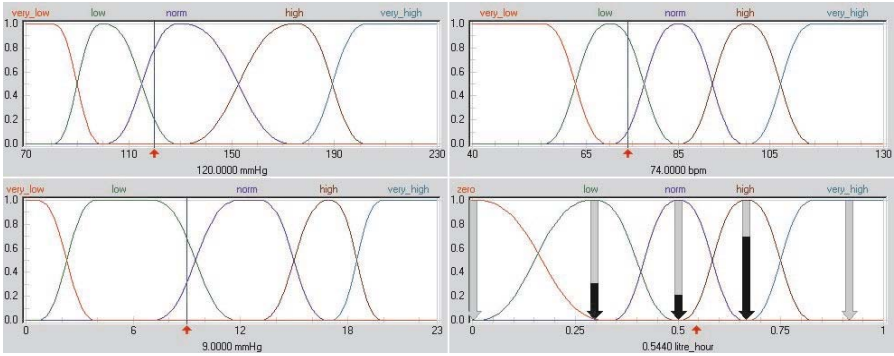


Figure 6. System inputs for a sample example120mmHg BP, 9mmHg CVP and 74 bpm HR, System LR out 0.544 lt/hour

Table 7. Eight rules yielded from one data pair vector

BP	CVP	HR	Inp. Agg.	LR
low	low	low	0,20524	high
low	low	norm	0,11678	high
low	norm	low	0,20524	norm
low	norm	norm	0,11678	norm
norm	low	low	0,69232	high
norm	low	norm	0,11678	high
norm	norm	low	0,30768	low
norm	norm	norm	0,11678	low

Solution is obtained by using Formula 2.

$$Liquid\_rate = \frac{(LowInpAgg \times LLR) + (NormInpAgg \times N.LR) + (HighInpAgg \times H.LR)}{L.LR + N.LR + H.LR}$$

Where:
Liquid_rate: Crisp out value (produced by CoM defuzzification),
Low.Inp.Agg.: Input Aggregation for low linguistic value,
L.LR: Liquid rate for low output linguistic value,
Norm.Inp.Agg.: Input Aggregation for normal linguistic value,
N.LR: Liquid rate for normal output linguistic value,
High.Inp.Agg.: Input Aggregation for high linguistic value,
H.LR: Liquid rate for high output linguistic value,

$$Liquid\ \_\ rate = \frac{(0.30768 \times 0.29762) + (0.20524 \times 0.5) + (0.69232 \times 0.66666)}{0.30768 + 0.20524 + 0.69232} =$$

$$\frac{0.091571 + 0.10262 + 0.461542}{1.20524} = 0.544\, lt\ /\ hour$$

This example is also solved with Matlab 6.01 software for comparison. The comparison is showed each program gives same response under same input conditions.

PIC16C771 microcontroller was used for this system. This powerful CMOS OTP-based 8-bit microcontroller packs Microchip's powerful PIC architecture into an 20-pin package and is upwards compatible with the PIC16C5X and PIC12CXXX devices The PIC16C771 features 6 channels of 12-bit Analog-to-Digital (A/D) converter giving designers the ability to discriminate smaller signal changes and eliminate the need for external circuitry for high precision measurement of analog signals. With 2 additional timers and an enhanced capture/compare/PWM function that make it ideal for the most sophisticated applications requiring higher levels of A/D in medical applications.

7. System simulation and test results

In order to test the system, simulation studies have to be carried out to validate the system and also to test its reliability. Test results are reported for different types of simulations. These results were produced by real recorded data. Figure 7., 8. and 9. shows the compare between the system response and manual response during the operations. Figure 7., shows a sensitive the oldest transplant patient's liquid infusion rate depending the changing in his BP, CVP and HR values that were occurred in every five minutes during his operation. Figure 8.., shows a nominal transplant patient's, who was in mean age (33 yrs) in this study's patients group, operation results. Figure 9., show a resistive the youngest transplant patient's (15 yrs) operation results.

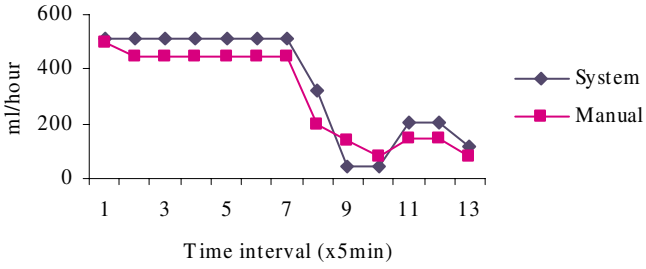


Figure 7. System's and manual LR responses for the oldest transplant patient

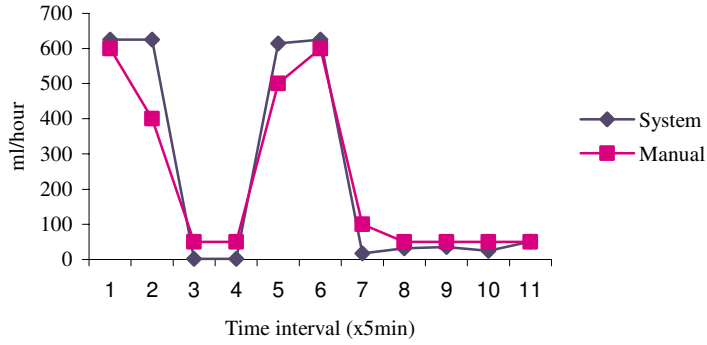


Figure 8. System's and manual LR responses for the nominal transplant patient

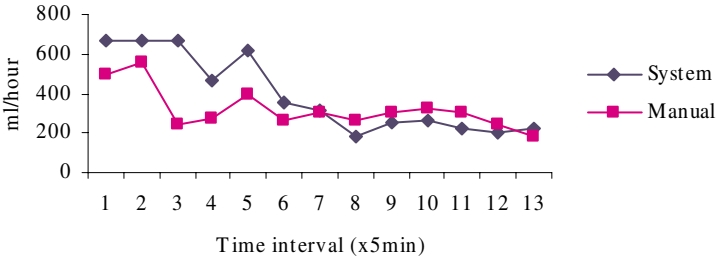


Figure 9. System's and manual LR responses for the youngest transplant patient

8 Conclusion

Resuscitation of intravascular volume and fluid replacement are cornerstones during the RT for transplant patients. Restoration of macrocirculation and microcirculatory perfusion is the primary goal of volume therapy, to prevent deleterious consequences such as organ dysfunction or multiple organ failure. For patients, who undergoing renal transplantation, careful monitoring of renal functions is required. The risk of renal failure can be further reduced by the administration of adequate intravascular volume by crystalloid solutions.

At the same time, careful balancing of intraoperative fluids is necessary against cardiovascular problems frequently encountered in patients with uremia. Close intraoperative monitoring, optimization of intravascular fluid volume status to maximize kidney perfusion, and prompt correction of electrolyte disturbances (especially potassium) are key to short- and long-term success of renal transplants.

The measurement of BP, CVP and HR facilitate safe fluid management in these cases. On the basis of the currently very limited data regarding the specific

situation of kidney transplantation, a fuzzy logic fluid therapy system should be used for management of transplant patient's fluid balance during RT.

Fuzzy logic simplifies the design of a control strategy by providing an easy to understand and intuitive approach to solve this fluid management and control problem. The potential benefits which are aimed at the beginning of the study were achieved. The Fuzzy Logic Fluid Therapy Control System can be used as an equipment which controls the liquid rate during RT. If it doesn't seem to ensure the transplant patient's safety as an equipment which works independently without the anesthesiologist, it can easily be used as a monitor or decision support system to assist the anesthesiologist to estimate the required liquid infusion rate more objectively. The system is release the anesthesiologist so that he or she can devote attention to other tasks that can not yet be adequately automated during RT.

Acknowledge

The authors acknowledge financial support from the Akdeniz University Unit of Scientific Research Projects Managements.

References

1. Alfrey EJ, Dafoe DC, Haddow GR, et al. (1994) Liver/kidney transplantation. *In* Jaffee RA, Samuels SI (eds): Anesthesiologist's Manual of Surgical Procedures. New York, Raven Press., pp 429–450.
2. Anderson RJ, Linas SL, Berns AS, et al. Nonoliguric acute renal failure. N Engl J Med 296:1334–1337
3. Artru AA (1998) Renal effects of sevoflurane during conditions of possible increased risk. J Clin Anesth 10:531–538
4. Belani KG, Palahniuk RJ (1991) Kidney transplantation. Int Anesthesiol Clin 29:17–39
5. Bickford RG (1950) Automatic control of general anesthesia, Electroencephalog. Clin. Neurophysiol., Vol.2, pp.93-96
6. Bickford RG (1951) The use of feedback systems for the control of anesthesia, Elect. Eng., vol.70, pp. 852-855
7. Braun WE, Philips DF, Vidt DG, et al. (1984) Coronary artery disease in 100 diabetics with end-stage renal failure Transplant Proc 16:603–607
8. Brown M, Whitworth J (1992) Hypertension in human renal disease. J Hypertens 10:701–709
9. Carlier M, Squifflet J, Pirson Y (1982) Maximal hydration during anesthesia increases pulmonary arterial pressures and improves early function of human renal transplants. Transplantation, 34:201-4
10. Charters P (1981) Renal transplantation in diabetes mellitus. Anaesthesia 35:199–204, 1981
11. Dawidson I, Berling E, Brynger H, et al. (1987) Intravascular volumes and colloid dynamics in relation to fluid management in living-related kidney donors and recipients. Crit Care Med 15:631–636
12. Dawidson I, Rooth P, Lu C, et al. (1991) Verapamil improves the outcome after cadaver renal transplantation. J Am Soc Nephrol 2:983–990, 1991
13. Dawidson IJ, Sandor ZF, Coorpender L (1992) Intraoperative albumin administration affects the outcome of cadever renal transplantation. Transplantation, 53:774-782

14. Heino A, Orko R, Rosenborg PH (1986) Anaesthesiological complications in renal transplantation: A retrospective study of 500 transplantations. Acta Anaesthesiol Scand 30:574–580
15. Heino A (1988) Operative and postoperative nonsurgical complications in diabetic patients undergoing renal transplantation. Scand J Urol Nephrol 22:53–58
16. Linkens DA, Abbod MF, Mahfouf M (1999) An initial survey of fuzzy logic monitoring and control utilisation in medicine, Journal of Biomedical Engineering , Applications, Basis Communications, Special Issues on Control methods in medicine, Taiwan
17. Najarian JS, Gulyassy PP, Stoney RJ, et al. (1966) Protection of the donor kidney during homotransplantation. Ann Surg 164:398–417
18. Pouttu J (1989) Haemodynamic responses during general anesthesia for renal transplantation in patients with and without hypertensive disease. Acta Anaesthesiol Scand 33:245–249
19. Ragaller MRJ, Thelien H, Koch T (2001) Volume Replacement in Critically ill Ptients with Acute Renal Failure. Journal of American Society of Nephrology, V:12, S:17
20. Sharpe MD, Gelb AW (1999) Anesthesia and Transplantation. Butterworth-Heinemann 241-67, USA
21. Sprung J, Kapural L, Bourke DL, et al. (2000) Anesthesia for kidney transplant surgery. J. Anesthesiology Clinics of North America, V:18, N:4
22. Weinrauch LA, Delia JA, Healy RW, et al. (1978) Asymptomatic coronary artery disease: Angiography in diabetic patients before renal transplantation: Relation of findings to postoperative survival. Ann Intern Med 88:346–348
23. Yardimci A, Onural A (1999) Fuzzy Logic Control of Child Blood Pressure During Anaesthesia, Computational Intelligence Theory and Applications, Lecture Notes in Computer Science V:1625 p:701-704, Springer, Berlin Germany
24. Yardimci A, Ferikoglu A (2000) Analysis of Biological Process with Microcontroller based Fuzzy Logic Systems, Bio.Inf.Eng.Pro. p:261-269 Isik University Istanbul, Turkey
25. Yardimci A, Ferikoglu A, Hadimioglu N (2001) Microcontroller-based Fuzzy Logic Sevofluorane Anesthesia Control System, Lecture Notes in Computer Science Volume:2206, p:137-147, Springer Verlag, Berlin Germany

Interpolative Fuzzy Reasoning in Behaviour-Based Control

Szilveszter Kovács

Department of Information Technology, University of Miskolc,
Miskolc-Egyetemváros, Miskolc, H-3515, Hungary
szkovacs@iit.uni-miskolc.hu
http://www.lit.uni-miskoolc.hu/~szkovacs

Abstract. Some difficulties emerging during the construction of fuzzy behaviour-based control structures are inherited from the type of the applied fuzzy reasoning. The fuzzy rule base requested for many classical reasoning methods needed to be complete. In case of fetching fuzzy rules directly from expert knowledge e.g. for the behaviour coordination module, the way of building a complete rule base is not always straightforward. One simple solution for overcoming the necessity of the complete rule base is the application of interpolation-based fuzzy reasoning methods, since interpolation-based fuzzy reasoning methods can serve usable (interpolated) conclusion even if none of the existing rules is hit by the observation. These methods can save the expert from dealing with derivable rules and help to concentrate on cardinal actions only. For demonstrating the applicability of the interpolation-based fuzzy reasoning methods in behaviour-based control structures a simple interpolation-based fuzzy reasoning method and its adaptation for behaviour-based control will be introduced briefly in this paper.

1 Introduction

In behaviour-based control systems (a good overview can be found in [3]), the actual behaviour of the system is formed as one of the existing behaviours (which fits best the actual situation), or as a kind of fusion of the known behaviours appeared to be the most appropriate to handle the actual situation. Beyond the construction of the behaviours, this structure has two other important tasks. The first is the decision, which behaviour is needed, or in case of behaviour fusion the determination of the necessity levels for each behaviour in solving the actual situation. The second is the way of the behaviour fusion. The first task, the behaviour coordination can be viewed as an actual system state approximation, where the actual system state is the set of the necessities of the known behaviours needed for handling the actual situation. The second is the fusion of the known behaviours based on their necessities. In case of fuzzy behaviour based control structures both tasks are solved by fuzzy logic controllers. If the behaviours are

S. Kovács: *Interpolative Fuzzy Reasoning in Behaviour-based Control*, Advances in Soft Computing **2**, 159–170 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

also implemented on direct fuzzy logic controllers, the behaviours together with the behaviour fusion modules form a hierarchical fuzzy logic controller. Since the classical fuzzy reasoning methods (e.g. compositional rule of inference) are demanding complete rule bases, all these rule bases have to build taking care to fill all the possible rules. In case if there is some rules are missing, there are observations may exist which hit no rule in the rule base and therefore no conclusion is obtained. Having no conclusion at any level of the fuzzy behaviour based control structure is hard to explain. E.g. one solution could be to keep the last real conclusion instead of the missing one, but applying historical data automatically to fill undeliberately missing rules could cause unpredictable side effects. Another solution for the same problem is the application of the interpolation-based fuzzy reasoning methods, where the derivable rules are deliberately missing. Since the rule base of a fuzzy interpolation-based controller, is not necessarily complete, it could contain the most significant fuzzy rules only without risking the chance of having no conclusion for some of the observations. In other words, during the construction of the fuzzy rule base, it is enough to concentrate on the cardinal actions; the "filling" rules (rules could be deduced from the others) can be omitted.

In the followings, first an approximate fuzzy reasoning method based on interpolation in the vague environment of the fuzzy rule base [4], [5], [6] will be introduced. The main benefit of the proposed method is its simplicity, as it could be implemented to be simple and quick enough to be applied in practical direct fuzzy logic control too. Then its adaptation to behaviour-based control structures together with a simple example will be discussed briefly.

2 Interpolation-based Fuzzy Reasoning

One way of interpolative fuzzy reasoning is based on the concept of vague environment [2]. Applying the idea of the vague environment the linguistic terms of the fuzzy partitions can be described by scaling functions [2] and the fuzzy reasoning itself can be simply replaced by classical interpolation. The concept of vague environment is based on the similarity or indistinguishability of the elements. Two values in the vague environment are ε-distinguishable if their distance is grater than ε. The distances in vague environment are weighted distances. The weighting factor or function is called *scaling function (factor)* [2]. Two values in the vague environment X are ε-distinguishable if

$$\varepsilon > \delta_s(x_1, x_2) = \left| \int_{x_2}^{x_1} s(x)dx \right| , \tag{1}$$

where $\delta_s(x_1, x_2)$ is the vague distance of the values x_1, x_2 and $s(x)$ is the scaling function on X. For finding connections between fuzzy sets and a vague environment we can introduce the membership function $\mu_A(x)$ as a level of

similarity **a** to x, as the degree to which x is indistinguishable to **a** [2]. The α-cuts of the fuzzy set $\mu_A(x)$ are the sets which contain the elements that are (1−α)-indistinguishable from **a** (see fig.1.):

$$\delta_s(\mathrm{a},\mathrm{b}) \le 1-\alpha \ , \quad \mu_A(x) = 1 - \min\{\delta_s(\mathrm{a},\mathrm{b}),1\} = 1 - \min\left\{\left|\int_{\mathrm{a}}^{\mathrm{b}} s(x)dx\right|,1\right\} . \tag{2}$$

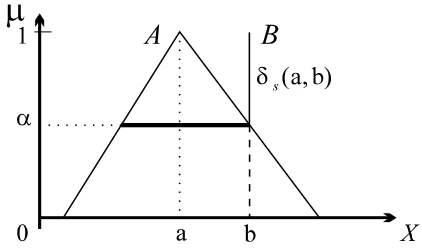


Fig. 1. The α-cuts of $\mu_A(x)$ contains the elements that are (1−α)-indistinguishable from **a**

This case (Fig.1.) the vague distance of points a and b ($\delta_s(\mathrm{a},\mathrm{b})$) is basically the *Disconsistency Measure* (S_D) of the fuzzy sets A and B (where B is a singleton):

$$S_D = 1 - \sup_{x\in X} \mu_{A\cap B}(x) = \delta_s(\mathrm{a},\mathrm{b}) \ \text{ if } \ \delta_s(\mathrm{a},\mathrm{b}) \in [0,1] , \tag{3}$$

where $A \cap B$ is the min t-norm, $\mu_{A\cap B}(x) = \min[\mu_A(x), \mu_B(x)]$ $\forall\, x \in X$.

It means that we can calculate the disconsistency measures between member fuzzy sets of a fuzzy partition and a singleton, as vague distances of points in the vague environment of the fuzzy partition. The main difference between the disconsistency measure and the vague distance is, that the vague distance is a crisp value in range of [0,∞], while the disconsistency measure is limited to [0,1]. Therefore if it is possible to describe all the fuzzy partitions of the primary fuzzy sets (the antecedent and consequent universes) of our fuzzy rule base by vague environments, and the observation is a singleton, we can calculate the "extended" disconsistency measures of the antecedent primary fuzzy sets of the rule base, and the "extended" disconsistency measures of the consequent primary fuzzy sets and the consequence (we are looking for) as vague distances of points in the antecedent and consequent vague environments.

The vague environment is described by its scaling function. For generating a vague environment of a fuzzy partition we have to find an appropriate scaling function, which describes the shapes of all the terms in the fuzzy partition. A fuzzy partition can be characterised by a single vague environment if and only if the membership functions of the terms fulfill the following requirement [2]:

$$s(x) = |\mu'(x)| = \left|\frac{d\mu}{dx}\right| \ \text{ exists iff } \ \min\{\mu_i(x), \mu_j(x)\} > 0 \Rightarrow |\mu'_i(x)| = |\mu'_j(x)| , \tag{4}$$

$\forall i, j \in I$, where $s(x)$ is the vague environment we are looking for.

Generally the above condition is not fulfilling, so the question is how to describe all fuzzy sets of the fuzzy partition with one "universal" scaling function. For this reason we propose to apply the concept of *approximate scaling function*, as an approximation of the scaling functions describes the terms of the fuzzy partition separately [4], [5], [6]. If the vague environment of a fuzzy partition (the scaling function or the approximate scaling function) exists, the member sets of the fuzzy partition can be characterised by points in the vague environment. (In our case the points are characterising the cores of the terms, while the shapes of the membership functions are described by the scaling function itself.) If all the vague environments of the antecedent and consequent universes of the fuzzy rule base are exist, all the primary fuzzy sets (linguistic terms) used in the fuzzy rule base can be characterised by points in their vague environment. So the fuzzy rules (build on the primary fuzzy sets) can be characterised by points in the vague environment of the fuzzy rule base too. This case the approximate fuzzy reasoning can be handled as a classical interpolation task. Applying the concept of vague environment (the distances of points are weighted distances), any interpolation, extrapolation or regression methods can be adapted very simply for approximate fuzzy reasoning [4], [5], [6].

We suggest adapting the *Kóczy-Hirota interpolation* [7]. This method generates the conclusion as a weighted sum of the vague consequent values, where the weighting factors are inversely proportional to the vague distances of the observation and the corresponding rule antecedents:

$$\mathrm{dist}(\mathrm{y}_0,\mathrm{y}) = \frac{\sum_{k=1}^{r} w_k \cdot \mathrm{dist}(\mathrm{y}_0,\mathrm{b}_k)}{\sum_{k=1}^{r} w_k}, \quad w_k = \frac{1}{(\mathrm{dist}(\mathbf{x},\mathbf{a}_k))^p}, \tag{5}$$

where w_k is a weighting factor inversely proportional to the vague distance of the observation and the k^{th} rule antecedent,

$$\mathrm{dist}(\mathbf{a}_k,\mathbf{x}) = \mathrm{dist}(\mathbf{x},\mathbf{a}_k) = \sqrt{\sum_{i=1}^{m}\left(\int_{\mathrm{a}_{k,i}}^{\mathrm{x}_i} s_{\mathrm{X}_i}(\mathrm{x}_i)d\mathrm{x}_i\right)^2}, \quad \mathrm{dist}(\mathrm{y}_0,\mathrm{b}_k) = \int_{\mathrm{y}_0}^{\mathrm{b}_k} s_{\mathrm{Y}}(\mathrm{y})d\mathrm{y}, \tag{6}$$

where s_{X_i} is the i^{th} scaling function of the m dimensional antecedent universe, s_{Y} is the scaling function of the one dimensional consequent universe, $\mathbf{x}$ is the multidimensional crisp observation, $\mathbf{a}_k$ are the cores of the multidimensional fuzzy rule antecedents $\boldsymbol{A}_k$, b_k are the cores of the one dimensional fuzzy rule consequents B_k, $R_\mathbf{i} = \boldsymbol{A}_\mathbf{i} \rightarrow \boldsymbol{B}_\mathbf{i}$ are the fuzzy rules, p is the sensitivity of the weighting factor for distant rules, y_0 is the first element of the one dimensional universe (Y: $\mathrm{y}_0 \le \mathrm{y} \quad \forall\ \mathrm{y} \in \mathrm{Y}$), y is the one dimensional conclusion we are looking for.

A simple one-dimensional example for the approximate scaling function and the Kóczy-Hirota (K-H) interpolation (6) is introduced on Fig. 2 and on Fig. 3.

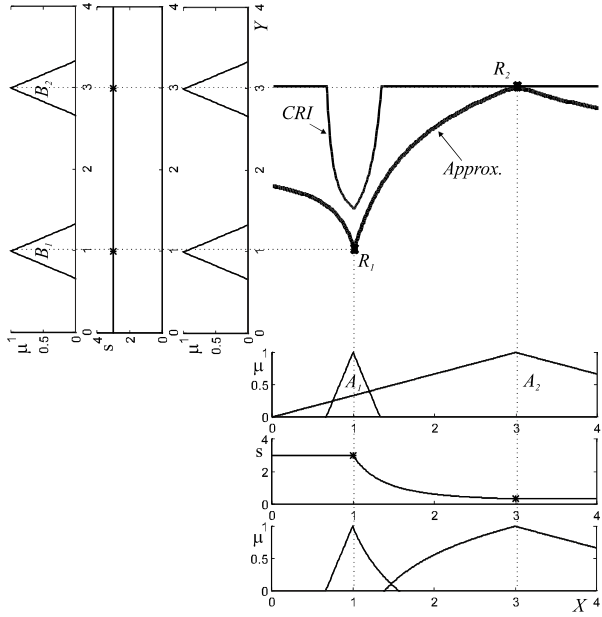


Fig. 2. Interpolation of two fuzzy rules (R_i: $A_i \rightarrow B_i$) (see fig. 3. for notation)

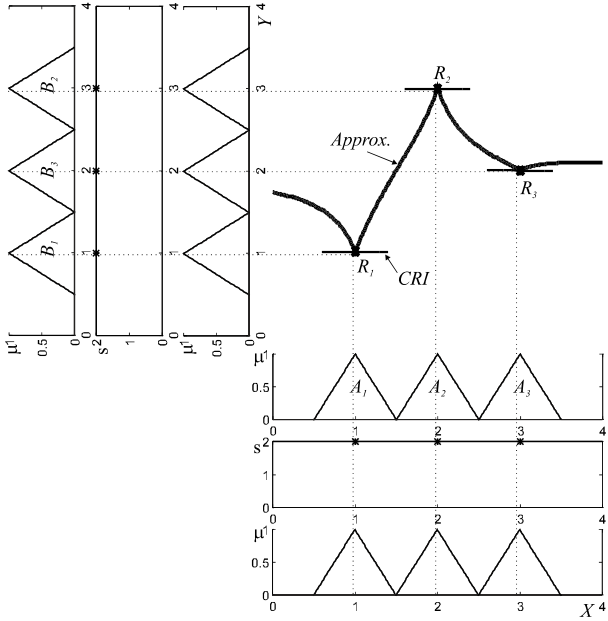


Fig. 3. Interpolation of three fuzzy rules (R_i: $A_i \rightarrow B_i$) in the approximated vague environment of the fuzzy rule base, using the K-H interpolation (p=1) (*Approx.*) and the min-max. CRI with the centre of gravity defuzzification (CRI), where μ is the membership grade, and s is the scaling function

For comparing the crisp conclusions of the K-H interpolation and the classical methods, the conclusions generated by the max-min compositional rule of inference (CRI) and the centre of gravity defuzzification for the same rule base is also demonstrated on the example figures (Fig. 2, Fig. 3). More detailed description of the proposed approximate fuzzy reasoning method can be found in [4], [5], [6].

3 The Applied Fuzzy Behaviour-based Structure

The main benefit of the interpolation-based fuzzy reasoning method, introduced in the previous chapter, is its simplicity. Applying look-up tables for pre-calculating the vague distances, it could be implemented to be simple and quick enough to fit the speed requirements of practical real-time direct fuzzy logic control systems, e.g. the requirements of fuzzy behaviour-based control too. The calculation efforts of many other interpolation-based fuzzy reasoning methods "wasted" for determining the exact membership shape of the interpolated fuzzy conclusion prohibits their practical application in real-time direct fuzzy logic control. The lack of the fuzziness in the conclusion is a disadvantage of the proposed method, but it has no influence in common applications where the next step after the fuzzy reasoning is the defuzzification.

In the followings a pure fuzzy behaviour-based control structure and the adaptation of the proposed interpolation-based fuzzy reasoning method will be discussed more detailed.

In case of pure fuzzy behaviour-based control structures all the main tasks of the behaviour-based control – the behaviour coordination, the behaviour fusion, and the behaviours themselves – are implemented on fuzzy logic controllers. (Such a structure is introduced on Fig.3.) Any of these controllers can apply the proposed interpolation-based approximate fuzzy reasoning method.

For demonstrating the main benefits of the interpolation-based fuzzy reasoning in behaviour-based control, in this paper we concentrate on the many cases most heuristic part of the structure, on the behaviour coordination.

The task of behaviour coordination is to determine the necessities of the known behaviours needed for handling the actual situation. In the suggested behaviour-based control structure, for this task the finite state fuzzy automaton is adapted (Fig.4.) [9]. This solution is based on the heuristic, that the necessities of the known behaviours for handling a given situation can be approximated by their suitability. And the suitability of a given behaviour in an actual situation can be approximated by the similarity of the situation and the prerequisites of the behaviour. (Where the prerequisites of the behaviour is the description of the situations where the behaviour is valid (suitable itself)). This case instead of determining the necessities of the known behaviours, the similarities of the actual situation to the prerequisites of all the known behaviours can be approximated.

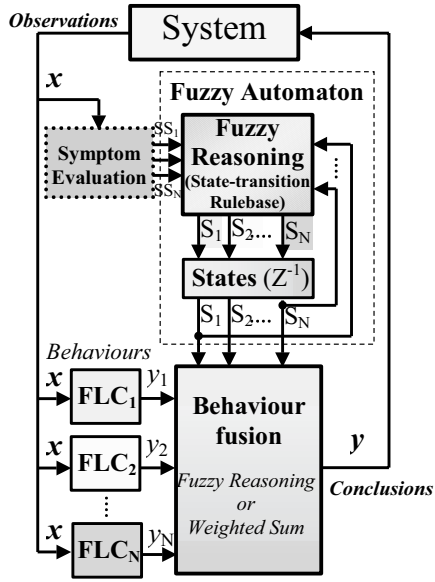


Fig. 4. The suggested fuzzy behaviour-based control structure

Thus the first step of this kind of behaviour coordination is determining the similarities of the actual situation to the prerequisites of all the known behaviours – applying the terminology of fault classification; it is the symptom evaluation (see e.g. Fig.4.). The task of symptom evaluation is basically a series of similarity checking between an actual symptom (observations of the actual situation) and a series of known symptoms (the prerequisites – symptom patterns – of the known behaviours). These symptom patterns are characterising the systems states where the corresponding behaviours are valid. Based on these patterns, the evaluation of the actual symptom is done by calculating the similarity values of the actual symptom (representing the actual situation) to all the known symptoms patterns (the prerequisites of the known behaviours). There are many methods exist for fuzzy logic symptom evaluation. For example fuzzy classification methods e.g. the Fuzzy c-Means fuzzy clustering algorithm [1] can be adopted, where the known symptoms patterns are the cluster centres, and the similarities of the actual symptom to them can be fetched from the fuzzy partition matrix. On the other hand, having a simple situation, the fuzzy logic symptom evaluation could be a fuzzy rule based reasoning system itself.

One of the main difficulties of the system state approximation in behaviour coordination is the fact, that most cases the symptoms of the prerequisites of the known behaviours are strongly dependent on the actual behaviour of the system. Each behaviour has its own symptom structure. In other words for the proper system state approximation, the approximated system state is also needed. A very simple way of solving this difficulty is the adaptation of fuzzy automaton. This case the state vector of the automaton is the approximated system state, and the

state-transitions are driven by fuzzy reasoning (Fuzzy state-transition rule base on Fig.4.), as a decision based on the previous actual state (the previous iteration step of the approximation) and the results of the symptom evaluation.

4 Application Example

For demonstrating the simplicity of defining rule base for interpolation-based fuzzy reasoning, as an example, the state-transition rule base of the previously introduced fuzzy automaton style behaviour coordination module applied for user adaptive information retrieval system in [10] and [11] will be introduced briefly in the followings. In our user adaptive information retrieval system example (introduced in [10] and [11] in more details) the user adaptivity is handled by combination of existing (off-line collected) human opinions (user models) in the function of their approximated similarity to the actual user opinions. As an analogy to the behaviour-based control applications, the different behaviours are the different existing user models, and the actual situation is the similarity of the actual user to the evaluators, originally gave the existing user models. Based on the observations (inputs) – the conclusion of the user feedback (the symptom evaluation about the state-transition to state i, SS_i for all the possible states $\forall i \in [1, N]$) and the previous state values S_i – we have to somehow estimate the new state values, the vector of the suitability of the existing user models. The heuristic we would like to implement in our example is very simple. If we already found a suitable model (S_i) and the user feedback is still supporting it (SS_i), we have to keep it even if the user feedback began to support some other models too. If there were no suitable model, but the user feedback began to support one, we have to pick it at once. In case of interpolation-based fuzzy reasoning, the above heuristic can be simply implemented by the following state-transition rule base [10], [11]. For the i^{th} state variable S_i, $i \in [1, N]$ of the state vector **S**:

If S_i=One **And** SS_i=One **Then** S_i=One **(7.1)**

If S_i=Zero **And** SS_i=Zero **Then** S_i=Zero **(7.2)**

If S_i=One **And** SS_i=Zero
And SS_k=Zero **Then** S_i=One $\forall k \in [1, N], k$ **(7.3)**

If S_i=Zero **And** SS_i=One
And S_k=Zero **And** SS_k=Zero **Then** S_i=One $\forall k \in [1, N], k$ **(7.4)**

If S_i=Zero **And** SS_i=One
And S_k=One **And** SS_k=One **Then** S_i=Zero $\exists k \in [1, N], k$ **(7.5)**

where SS_i is the conclusion of the symptom evaluation about the state-transition to state i, $\forall i \in [1, N]$; N is the number of known behaviours (state variables). The structure of the state-transition rules is similar for all the state

variables. Zero and One are linguistic labels of fuzzy sets (linguistic terms) representing high and low similarity. The interpretations of the Zero and One fuzzy sets can be different in each S_i, SS_i universes.

Please note that rule base (7) is sparse. It contains the main rules for the following straightforward goals only: Rule (7.1) simply keeps the previously chosen state values in the case if the symptom evaluation also agrees. The rule (7.2) has the opposite meaning, if the state values were not chosen, and moreover the symptom evaluation is also disagrees the state value should be suppressed. The rule (7.3) keeps the already selected state values (previous approximation), even if the symptom evaluation disagrees, if it has no better "idea". Rules (7.4) and (7.5) have the task of ensuring the relatively quick convergence of the system to the sometimes unstable (changeable) situations, as new state variables which seem to be fit, can be chosen in one step, if there is no previously chosen state, which is still accepted by the symptom evaluation (7.4). (Rule (7.5) has the task to suppress this selection in the case if exists a still acceptable state which has already chosen.) The goal of this heuristic is to gain a relatively quick convergence for the system to fit the opinions of the actual user, if there is no state value high enough to be previously accepted. This quick convergence could be very important in many application areas e.g. in case of an on-line user adaptive selection system introduced in [10], where the user feed-back information needed for the state changes are very limited.

Some state changes of the fuzzy automaton in the function of the conclusion of the symptom evaluation (SS_1, SS_2) for the two states case (applying the state-transition rule base (7)) are visualised on Fig.5. and Fig.6.

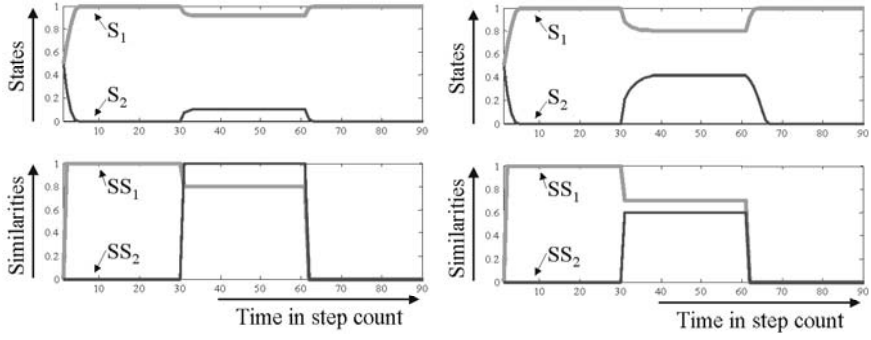


Fig. 5. Do not "pick up" a new state if the previous approximation is still adequate

Counting the rules of the classical (e.g. compositional) fuzzy reasoning for the same strategy we find, that in the two state case the complete rule base needs 16 rules (as we have four observation universes (S_1, SS_1, S_2, SS_2) each with two terms fuzzy partitions (Zero, One) - 2^4 rules), while the sparse rule base (7) contains 5 rules only (see table 1 for the state-transition rule base of state S_1). Taking into account that in the proposed behaviour-based control structure a

separate rule base is needed for each state variables, the behaviour coordination needs 32 rules, while 10 is enough in case of applying the proposed interpolation-based fuzzy reasoning method. Increasing the number of the state variables the situation became even worse. In case of three state variables (S_1, S_2, S_3) the rate become $3 \cdot 2^6$ ($n \cdot 2^{2 \cdot n}$, where n is the number of the states) and $3 \cdot 6$ ($n \cdot (n+3)$) up to the interpolation-based method (see table 2).

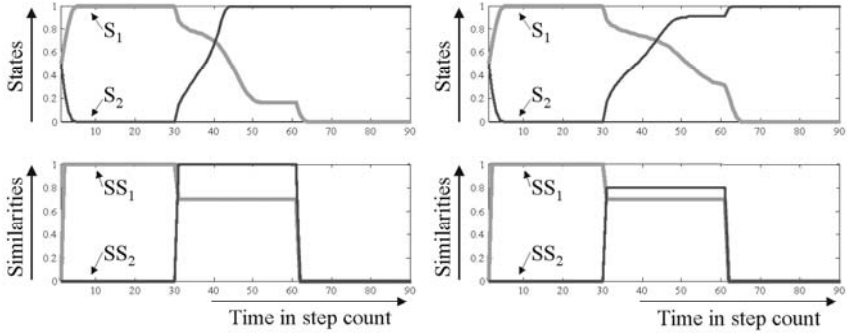


Fig. 6. But "pick it up" if it seems better, or at least as good as the previous was

Table 1. State-transition rule base of state S_1 in case of two state variables (S_1,S_2) according to rule base (7)

R_{S1}:	S_1	SS_1	S_2	SS_2	S_1	
1.,	One	One			One	(according to (7.1))
2.,	Zero	Zero			Zero	(according to (7.2))
3.,	One	Zero		Zero	One	(according to (7.3))
4.,	Zero	One	Zero	Zero	One	(according to (7.4))
5.,	Zero	One	One	One	Zero	(according to (7.5))

Table 2. State-transition rule base of state S_1 in case of three state variables (S_1,S_2,S_3) according to rule base (7)

R_{S1}:	S_1	SS_1	S_2	SS_2	S_3	SS_3	S_1	
1.,	One	One					One	(7.1)
2.,	Zero	Zero					Zero	(7.2)
3.,	One	Zero		Zero		Zero	One	(7.3)
4.,	Zero	One	Zero	Zero	Zero	Zero	One	(7.4)
5.,	Zero	One	One	One			Zero	(7.5)
6.,	Zero	One			One	One	Zero	(7.5)

The exponential rule number "explosion" in case of increasing the number of the input variables makes many heuristic ideas unimplementable and therefore useless. E.g. in the case of the original source of the example application of this paper (introduced in [10]), the behaviour coordination module applied for user adaptive information retrieval system had 4 state variables (one for each emotional models), which makes our simple rule base (7) practically unimplementable as a complete rule base ($4 \cdot 2^8 = 1024$ rules). While our working demonstrational example had only 28 rules thanks to the applied interpolation-based fuzzy reasoning method.

4 Conclusions

The goal of this paper was to introduce an interpolation-based fuzzy reasoning method, which could be implemented to be simple and quick enough to fit the requirements of behaviour-based control structures in real-time direct fuzzy logic control systems. The suggested approximate fuzzy reasoning method based on interpolation in the vague environment of the fuzzy rule base gives an efficient way for designing direct fuzzy logic control applications. The lack of the fuzziness in the conclusion is a disadvantage of the proposed method, but it has no influence in common applications where the next step after the fuzzy reasoning is the defuzzification. For demonstrating the efficiency of the interpolation-based fuzzy reasoning in behaviour-based control, a fuzzy behaviour-based control structure based on fusion of different known behaviours in the function of their actual necessities approximated by fuzzy automaton is introduced briefly in this paper. To give some guidelines for interpolation-based fuzzy reasoning rule base design, some highlights of the behaviour coordination rule base of a user adaptive information retrieval system application ([10], [11]) is also introduced in this paper. The implementation of interpolation-based fuzzy reasoning methods in behaviour-based control structures simplifies the task of fuzzy rule base creation. Since the rule base of a fuzzy interpolation-based controller is not necessarily complete, it could contain the most significant fuzzy rules only without risking the chance of having no conclusion for some of the observations. In other words, during the construction of the fuzzy rule base, it is enough to concentrate on the cardinal actions; the "filling" rules (rules could be deduced from the others) could be deliberately omitted. Thus, compared to the classical fuzzy compositional rule of inference, the number of the fuzzy rules needed to be handled during the design process could be dramatically reduced.

The necessity of the complete rule base in many classical fuzzy reasoning methods (e.g. max-min CRI) and hence the exponential rule number "explosion" in case of increasing the number of the input variables makes numerous rule base ideas unimplementable and therefore useless. The application of interpolation-based fuzzy reasoning methods could provide some implementation chances for many of them (see e.g. our simple example in section 3).

Acknowledgements

This research was supported by the Japan Gifu Prefecture Research Institute of Manufacturing Information Technology and the Intelligent Integrated Systems Japanese Hungarian Laboratory. Szilveszter Kovács is supported by the György Békésy Postdoctoral Scholarship.

References

1. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function, Plenum Press, New York (1981)
2. Klawonn, F.: Fuzzy Sets and Vague Environments, Fuzzy Sets and Systems, 66 (1994) 207-221
3. Pirjanian, P.: Behavior Coordination Mechanisms - State-of-the-art, Tech-report IRIS-99-375, Institute for Robotics and Intelligent Systems, School of Engineering, University of Southern California, http://www-robotics.usc.edu/~paolo/publications/bcm.ps.gz, October (1999)
4. Kovács, Sz.: New Aspects of Interpolative Reasoning, Proceedings of the 6th. International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Granada, Spain (1996) 477-482
5. Kovács, Sz., Kóczy, L.T.: Approximate Fuzzy Reasoning Based on Interpolation in the Vague Environment of the Fuzzy Rule base as a Practical Alternative of the Classical CRI, Proceedings of the 7th International Fuzzy Systems Association World Congress, Prague, Czech Republic, (1997) 144-149
6. Kovács, Sz., Kóczy, L.T.: The use of the concept of vague environment in approximate fuzzy reasoning, Fuzzy Set Theory and Applications, Tatra Mountains Mathematical Publications, Mathematical Institute Slovak Academy of Sciences, Bratislava, Slovak Republic, vol.12. (1997) 169-181
7. Joó, I., Kóczy, L.T., Tikk, D., Várlaki, P.: On a Stable Interpolation Method, Proceedings of the 7th International Fuzzy Systems Association World Congress, Prague, Czech Republic (1997) 133-137
8. Rudas, I. J., Kaynak, M.O., Bitó, J.F., Szeghegyi, Á.: New Possibilities in Fuzzy Controllers Design Using Generalized Operators, Proceedings of the 5th International Conference on Emerging Technologies and Factory Automation, November, Hawaii (1996) 513-517
9. Kovács, Sz.: Interpolative Fuzzy Reasoning and Fuzzy Automaton in Adaptive System Applications, Proceedings of the IIZUKA2000, 6th International Conference on Soft Computing, October 1-4, Iizuka, Fukuoka, Japan (2000) 777-784
10. Kovács, Sz., Kubota, N., Fujii, K., Kóczy, L.T.: Behaviour based techniques in user adaptive Kansei technology, Proceedings of the VSMM2000, 6th International Conference on Virtual Systems and Multimedia, October 3-6, Ogaki, Gifu, Japan (2000) 362-369
11. Kovács, Sz.: Fuzzy Reasoning and Fuzzy Automata in User Adaptive Emotional and Information Retrieval Systems, Proceedings of the 2002 IEEE International Conference on Systems, Man and Cybernetics, October 6-9, Hammamet, Tunisia, 02CH37349C, ISBN: 0-7803-7438-X, WP1N5 (2002) 6

Fuzzy Modeling of Offensive Maneuvers in an Air-to-Air Combat

S. Akabari,[1] M. B. Menhaj,[2] and S. K. Nikravesh[2]

[1] Electrical engineering Department., Kashan University, Kashan, Iran
[2] Department of Electrical Enginnering, Amirkabir University, Hafez Ave. #529, Tehran-Iran,
menhaj@aut.ac.ir

Abstract. In this contribution we propose a new guidance law based on fuzzy logic that can be successfully applied to modeling and generating complicated offensive maneuver in an air combat between two aircraft. Based on human expert's decision-making process, an intelligent based method is proposed to model the maneuvering. Fuzzy "if ... then..." rules are used to represent the pursuer preferences in guiding his/her system. The rules are directly obtained from expert's knowledge. Each rule relates the desired moving directions of the pursuer to the task parameters. The control parameters of the aircraft are computed through a mean square error scheme. A large amount of simulations are used to approve the satisfactory performance of the model. The results show human like maneuvers can be generated by the proposed model.

1 Introduction

The figure of air combat in the future depends heavily on automatic systems that will be integrated into the fighter's cockpit. The complexity and sophistication of flight automation is the result of these automatic systems. While in early days of air combat, pilots had to gather all necessary information and data through their senses, nowadays a large number of electronic and computer systems help the pilots in data acquisition and processing, and the trend is toward more automation of air combats.

S. Akabari et al.: *Fuzzy Modeling of Offensive Maneuvers in an Air-to-Air Combat*, Advances in Soft Computing **2**, 171–184 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

Because of the growing amount of information received by pilots during combat tasks and the capabilities of modern fighters, there will be an emerging need to systems that help pilots in making decisions or evaluate them during an air-to-air engagement [1,2]. The technology improvements will further make possible the realization of unmanned air vehicles (UAV). Thus, complete automation of pilot's actions will be required in near future.

This work might finally help the study for automation of maneuvering decisions in an air-to-air combat. The resulting system can be used in unmanned air fighters as well as flight training simulators.

Development of fuzzy guidance law for planar maneuvers has been also done by the authors using an approach to mimic the human operator's performance [15,16]. This paper is an extension of that research to include a three-dimensional maneuvering task.

The literature on the subject is not so vast. Two approaches to the problem of maneuvering automation could obviously be recognized. The first approach relies heavily on optimization theory especially differential game theory. The research on this line was triggered mainly by the work of Isaacs [3] on differential game theory. The theory -since its main purpose is to deal with conflict tasks- has been widely used for combat modeling. The theory provides optimal strategies for both pursuer and evader in air combat scenarios. The references [4,5,6] used differential game to obtain best maneuvers for pursuer and evader in special occasions of air combat.

The second approach is based on AI and computational intelligence methods. This approach mostly leads to expert system structures for the decision-making model, which in turn would be a part of a very large and complex system. The method usually formalizes the expert's knowledge and experiences in some ways and then builds the model from the formalization. The structures and formalization methods are quite different [7,8,9].

While each of the above approaches has their own limitations, the optimization approach to the problem suffers from some severe drawbacks. The main drawback is that it is very difficult to include realistic combat situations in the formulation. To keep the problem mathematically tractable and solvable as well, some simplifications are necessary; this makes the solution far from what is indeed done in real combat situations by experienced pilots. Furthermore, this approach seldom pays attention to the structure of the performance criterion that models the preferences of a human decision-maker.

The chapter is organized as follows. Section 2 describes the problem under study. The solution method rooted in qualitative knowledge discovery is introduced in section 3. This section presents also the

structure of the proposed model in details. The simulation results are given in Section 4. Finally, Section 5 concludes the paper.

2 The Problem Statement

Here we consider the situation shown in Fig.2.1. Two aircrafts (P and E) are engaged in a pursuer-evader combat task. The initial position is such that one of them (P) is in an offensive and the other (E) is in a defensive position.

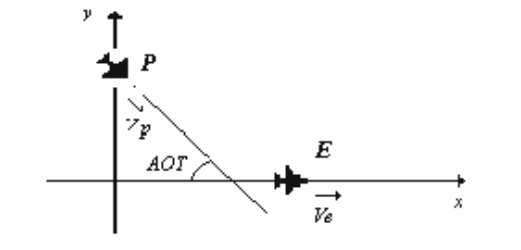


Fig. 2.1. Initial Encounter Situation

The aircraft E has a constant speed and can maneuver in the horizontal plane. P can move in the three-dimensional space and is further able to change its speed. The flight dynamics of both aircrafts are governed by point mass equations. The aircraft P has an energy advantage. This assumption determines the tactics used by human experts.

The problem is to find a suitable maneuver for aircraft P by which it will be placed in a shooting position. By shooting position we mean the position in which the distance between two aircrafts lies in a range of maximum and minimum values and the heading of P is toward E.

In addition to reaching a suitable position, the maneuver should avoid to place the offensive aircraft P in bad positions. A bad position means a relative position of the two aircrafts that allows the offensive aircraft E to gain enough advantage to turn to an offensive or a threatening position. For simulation purposes, we consider the aircrafts to be simplified aircrafts.

The point mass model is used to represent aircraft's flight dynamics. The equations of motion are as follows [10]:

$$\dot{V} = \frac{T - D}{m} - g \sin\gamma, \quad \dot{\chi} = \frac{L \sin\phi}{mV \sin\gamma}, \quad \dot{\gamma} = \frac{g}{V}\left(\frac{L \cos\phi}{mg} - \cos\gamma\right), \tag{2.1a}$$

$$\dot{x} = V\cos\gamma\cos\chi, \quad \dot{y} = V\cos\gamma\sin\chi, \quad \dot{z} = V\sin\gamma \tag{2.1b}$$

In the above, V is the aircraft speed, T, φ, L are thrust, bank angle and lift respectively, D is drag, m is the mass and (x, y, z) are the position vector of the aircraft, γ is the heading angle (the angle between velocity vector and horizontal plane) and χ is the pitch angle.

3 Structure of the Proposed Model

The method used in this research for deriving the rule bases is rooted in qualitative knowledge discovery breifly discussed below.

3.1 Qualitative Knowledge Discovery

The main goal here is to extract the rules governing pilot's decisions (maneuvers) from a set of varying, contradicting, uncertain and vague information obtained through questioning individual fighter pilots about their actions when they faced special situations in a dogfight combat. The approach consists of two steps. In the first step the aforementioned knowledge through a process will be converted to a fuzzy rule base. The main problem in this step is to specify inputs and outputs of the rule-bases as well as the rules relating the inputs to outputs.

In the second step, the membership functions or in other words fuzzy sets that cluster each input and output in the rule base shall be computed. The fuzzy sets are arbitrarily assumed trapezoidal and triangular. Therefore, the problem is reduced to computing the boundary and corners of each membership function. The second step is resolved through extensive simulations described in detail in the following sections.

3.2 The Model Development

Here, we use the approach based on artificial intelligence methods. To relate the suitable maneuver or action to combat situation, the human expert's reaction to the situation is used. The knowledge about these reactions is obtained through some fighter training books and manuals [11,12] as well as through interviews with human experts. Then, the reaction is represented as fuzzy if … then … rules. The block diagram of figure 2 shows the basic structure of our model.

3.2.1 Fuzzy Rule Bases

These are the main parts of the model. As stated above these fuzzy rule bases have been derived from expert's knowledge. The main feature of these rule bases is that their outputs are not control inputs imposed to the fighter (like the regular guidance laws). But, according to the decision process of the expert, the outputs of the rule bases are the desired values of heading angle, pitch angle and velocity in the next time step.

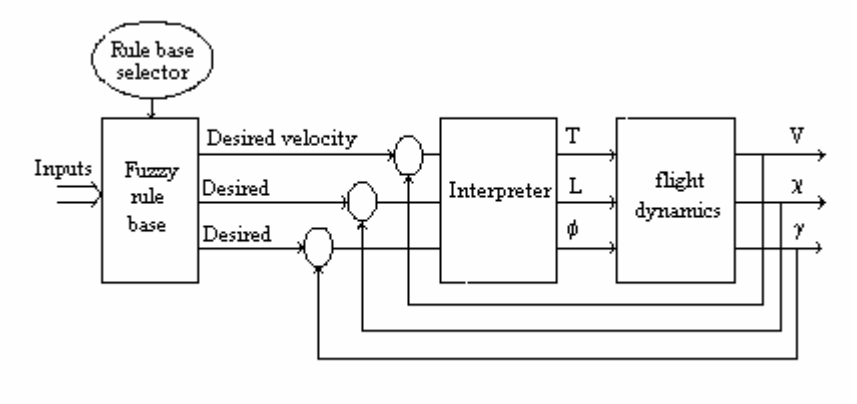


Fig. 3.1 Block Diagram of the Model

To facilitate the derivation of the rule bases and according to the expert's action, it is supposed that the desired heading and pitch angles are determined independently. It means that evaluation of parameters may be done independently for vertical and horizontal components of each maneuver. This results in two separate groups of rule bases. The first group will determine the desired heading while the second will determine the desired pitch angle.

Furthermore, it is supposed that each vertical maneuver shall be divided into two sub-maneuvers: climb and dive. The experienced pilot applies the climb part of the maneuver in order to decrease speed, closure rate and turn radius in the horizontal plane. The dive part is the final phase of the maneuver in which the pilot dives back to the plane of opponent's turn toward a shooting position. Each of these parts has their own rule base: Climb and Dive Rule base.

Inputs to the Climb rule base are distance in the horizontal plane, z-distance, and the pursuer to evader velocity ratio. A sample of the rules is shown below:

if xy distance is vs and altitude-difference is nm and vptovr is one then gama is almost 45.

The Climb rule base includes 75 rules. The membership functions are arbitrarily chosen as triangular and trapezoidal. The membership function parameters have been derived experimentally through a large amount of simulations.

Inputs to the Dive rule base are the x-distance and y-distance. A sample of the rules is shown below:

if altitude-difference/distance is nvb then gama is near vertical.

This rule base has 7 rules. Thus, three rule bases are involved in determining the desired heading and pitch angles for aircraft P in each time step.

The third rule base determines the desired pitch angle in each instant of the maneuver.

3.2.2 The Interpreter

Knowing that the outputs of the rule bases are either desired values or commands, it is necessary to convert these commands to physical inputs of the aircraft's dynamics (e.g. thrust, lift and bank angle). These parameters can be found through the following optimization problem:

$$\min_{T,L,\varphi} \quad (\gamma(t+\Delta t)-\gamma_{des})^2+(\chi(t+\Delta t)-\chi_{des})^2+(V(t+\Delta t)-V_{des})^2 \tag{3.1}$$

To make it simpler, the assumption ***"The (specific) energy of the P remains constant"*** is made; this is natural for an aircraft with an energy advantage. This assumption makes sense because the energy advantage is crucial in a combat and thus the aircraft with a greater energy tries to maintain its energy level. By this assumption the value for thrust will be obtained as follows:

$$E_S = cte \;\Rightarrow\; H+\frac{V^2}{2g}=cte \tag{3.2}$$

Take the derivative of (3.2) to obtain

$$\dot{H}+\frac{2V\dot{V}}{2g}=0 \;\Rightarrow\; \dot{H}=-\frac{V\dot{V}}{g}. \tag{3.3}$$

From point mass equations (Eqs. 2.1), we may write

$$\dot{H} = V \sin\gamma \,. \tag{3.4}$$

This leads to

$$V \sin\gamma = -\frac{V\dot{V}}{g} \;\Rightarrow\; \dot{V} = -g \sin\gamma \,. \tag{3.5}$$

Again from the point mass equations, we may write

$$\dot{V} = \frac{T - D}{M} - g \sin\gamma \,. \tag{3.6}$$

And finally from equations 3.5 and 3.6, we obtain

$$T - D = 0 \;\Rightarrow\; T = D \,. \tag{3.7}$$

Hence, the optimization problem of Eq.3.1 reduces to the following one:

$$\min_{\mathrm{L},\varphi} \quad \left(\gamma(t + \Delta t) - \gamma_{des}\right)^2 + \left(\chi(t + \Delta t) - \chi_{des}\right)^2 \tag{3.8}$$

Due to the fact that point mass equations relate $\dot{\gamma}$ and $\dot{\chi}$ to L and φ and they are able to solve the above minimization problem, it is necessary to rewrite it in terms of the derivatives of γ and χ. It is straightforward to show that the following minimization problem is equivalent to the one given in Eq. 3.8.

$$\min_{\mathrm{L},\varphi} \quad (\dot{\gamma}(t) - \dot{\gamma}_{des})^2 + (\dot{\chi}(t) - \dot{\chi}_{des})^2 \tag{3.9}$$

$\dot{\gamma}_{des}$ and $\dot{\chi}_{des}$ can be approximately computed from γ_{des} and χ_{des} .This optimization problem is subjected to some restrictions on L and φ. The structural limits of the P imposes an upper bound on L, however a more restrictive upper bound is forced by the assumption of the aforementined constant energy . To derive this upper bound, we should note that any aircraft could produce a limited thrust. This and Eq.3.7 will impose an upper bound on the drag, $D_{\max} = T_{\max}$.

Knowing that Lift and drag are related through the following equation [11],

$$D = C_{D0} S \rho \frac{V^2}{2} + \frac{2kL^2}{\rho V^2 S} \tag{3.10}$$

the upper bound of L will be found as:

$$L_{\max} = \sqrt{\left(T_{\max} - C_{D0} s\rho \frac{V^2}{2}\right)\frac{\rho V^2 S}{2k}} \tag{3.11}$$

Hence, The optimization problem will then be of the following form:

$$\begin{aligned} &\min_{L,\varphi} \quad f(L,\phi) \\ &s.t \;\; L^2 = \left(mV\cos\gamma\dot{\theta}\right)^2 + \left(mV\dot{\gamma} + mg\cos\gamma\right)^2 \le L_{\max}^2 \end{aligned} \tag{3.12}$$

where,

$$f(L,\phi) = (\dot{\chi} - \dot{\chi}_{des})^2 + (\dot{\gamma} - \dot{\gamma}_{des})^2 \tag{3.13}$$

Use the point mass equations (Eq.2.1) in (4.13) to obtain

$$f(L,\phi) = \left(\frac{L\sin\phi}{mV\cos\gamma} - \dot{\chi}_{des}\right)^2 + \left(\frac{L\cos\phi}{mV} - \frac{g}{V}\cos\gamma - \dot{\gamma}_{des}\right)^2. \tag{3.14}$$

By solving the minimization problem (Eq.3.12), the values for control parameters (L, φ) will be easily determined.

3.2.3 The Rule Base Selector

As it was stated earlier, each maneuver is divided into two climb and dive phases:. Determination of the active phase in each time step is done in the rule-base-selector block. A Rule base selector is itself a fuzzy rule base obtained from an expert's knowledge. The inputs of this rule base are the closure rate, distance in the horizontal plane, altitude difference and off angle (angle between the velocity vector of P and the Line-Of-Sight).

In the basic model given in sections 3.2.1 to 3.2.2 an implicit assumption in computing the control parameters of the aircraft was made. To elaborate this assumption, we take another look at the minimization problem as follows:

$$\begin{aligned} &\min_{L,\varphi} \quad (\dot{\gamma} - \dot{\gamma}_{des})^2 + (\dot{\chi} - \dot{\chi}_{des})^2 \\ &s.t \;\; L < L_{\max} \;\; , -\pi \le \varphi \le \pi \end{aligned} \tag{3.15}$$

It could be obviously seen that there was no distinction in how to vary χ and γ. In the other words, no matter what the combat situation is, the errors

in heading and in pitch angles have the same weight. But it is not true in a real combat. In fact, there is always a degree of importance between them.

As the dynamics of the aircraft will not allow the pitch and heading angles to change arbitrarily, the pilot in any instant of the maneuver decides which of the two, χ and γ is more important to modify. Therefore, to satisfy this concern in our model, the minimization problem is modified as follows.

$$\begin{aligned} &\min_{L,\varphi} \quad (\dot{\gamma} - \dot{\gamma}_{des})^2 + w(\dot{\chi} - \dot{\chi}_{des})^2 \\ &L < L_{\max}, \quad -\pi \le \varphi \le \pi \end{aligned} \tag{3.16}$$

In the above w is the weighting factor and shows the relative importance of heading and pitch changes. A big w indicates the importance of pitch angle variations while small values of w give more weights on heading angle variations.

The solution to the minimization problem of Eq.3.16 is found by numerical methods.The only remaining point is how to calculate a proper w in each combat situation. Again, the expert knowledge and qualitative knowledge discovery provide us the general insight to develop the proper method to solve the problem. The result of the process is a fuzzy rule base with two inputs- lead angle and distance. The lead angle is the angle between the velocity vector and the line of sight. The rule base has 6 rules relating these inputs to the linguistic variables (fuzzy term sets) of w. It is supposed that w varies between 0.5 and 2.5. A sample rule is as follows:

If lead-angle is about 45 and distance is small then w is small

This means that if we are near the opponent while heading almost toward it, it is better to change our heading angle and moves vertically.

4 Simulation Results

This section is devoted to show and compare the performance of the basic and modified models. To do so, many simulations have been performed with the following flight parameters chosen for P and E.

Tmax=100000N, m=10000kg, Cd0=0.0169, k=0.179, s=26 m^2 ***, ρ=0.8, Vp=200 m/sec, Ve=120 m/sec.***

Simulation results show the capability of the proposed model to produce maneuvers leading to a combat superiority against maneuvering and non-

maneuvering opponents. The resulting maneuvers are shown in Figs. 4.1, and 4.2, respectively for the basic and modified models. In these cases the aircraft E performed in-plane maneuver turning toward P to increase off-angle. This maneuver is a typical one for the aircraft with energy (speed) disadvantage. The control parameters of aircraft P are shown in Fig. 4.3. The control parameters are of Bang-Bang nature that is in line with the experiences from real combats and indicates the near-optimality of the models.

The maneuver of Fig. 4.4 shows a great resemblance to the classic Lag-roll maneuver and indicates the capability of the model to generate maneuvers of real combats.

Fig. 4.5 shows the flight path for the modified model when the measurements of the opponent's position were noisy. The noise level was 10% of the distance between P and E.

To measure the efficiency of the models in different situations, a total of 216 initial positions were tested for the model; 6 test points for the initial position of the pursuer aircraft P and for each of these points, 36 initial heading angles were considered. Without loss of generality, it was supposed that the aircraft E was initially at the location (500,0,1000). For the basic model, the rate of capture when the evader aircraft E doesn't change its flight path is 100% while a capture rate of near 80% obtained for a maneuvering aircraft E. Using the modified model, a small increase in capture rate was observed for E.

6 Conclusion

In this chapter, an AI based model has been presented mainly using fuzzy set theory to model the maneuvering in a human like pursuer-evader combat automation problem. In this regard, the theory of fuzzy set allows us to simply and efficiently employ the expert's knowledge. The simulation results show a good and human-like performance of the proposed model.

7 References

1. Reising J M, Emerson T J (1988) Research in cockpit control and displays. In: Rouse W B (ed) Advances in man-machine system research, v.4, 1988.
2. Rodin E Y, Geist D, Lirov Y (1989) Flight and fire control knowledge representation. 28th IEEE CDC:779-780
3. Issacs R (1965) Differential Games. Kreiger, Huntington, NY, 1965.

4. Hillberg C, Jamark B (1984) Pursuit-evasion between two realistic aircraft. Journal of Guidance, Control and Dynamics 7:690-694
5. Guelman M, Shinar J (1984) Optimal guidance law in the plane. Journal of Guidance, Control and Dynamics 7:471-476
6. Menon P K A, Duke E L (1992) Time-optimal aircraft pursuit evasion with a weapon envelop constraint. Journal of Guidance, Control and Dynamics 15:448-456
7. Hammer J M, Small R L (1996) An intelligent interface in an associate system. In Rouse W B (ed.) Human/technology interactions in complex systems 7
8. Virtanen K, Raivio T, Hamalainen R P (2001) Modeling pilot's sequential maneuvering decisions by multistage influence diagram. Proceeding of AIAA guidance, control and dynamics conference
9. McManus J W, Goodrich K H (1989) Application of artificial intelligence (AI) programming techniques to tactical guidance for fighter aircraft. AIAA paper 89-3525
10. Anderson J D (1989) Introduction to flight, 3rd edition, McGraw-Hill, New York
11 Shaw R L (1988) Fighter combat: Tactics and Maneuvering. United States Naval Inst
12. Crenshaw D. (2000) How to live and die in the virtual sky, http://www.deltahawks.org
13. Hovakiyan N, Melikan A (2000) Optimal pursuit-evasion paths in a game on complete cone. Proceedings of the American Control conference, vol. 6:3865-3869
14. Kachroo P, Shedied S A, Bay J S, Vanlandingham H (2001) Dynamic programming solution for a class of pursuit evasion problems: the herding problem. IEEE Transactions on: Systems, Man and Cybernetics, Part C 31:35-41
15. Akbari S, Menhaj M B, Nikravesh S K Y (2001) A fuzzy guidance law for modeling offensive air-to-air combat maneuvers. Proc. IFSA-NAFIPS
16. Akbari S, Menhaj M B, Nikravesh S K Y (2002) A two-phase fuzzy guidance law for planar offensive air-to-air combat maneuver. Amirkabir Journal of Sci. & Tech. 52:143-154

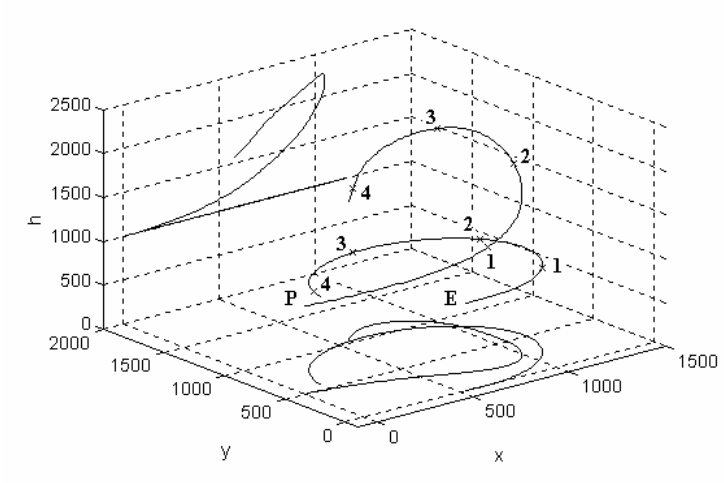


Fig. 4.1. Flight path produced by the basic model against a maneuvering opponent

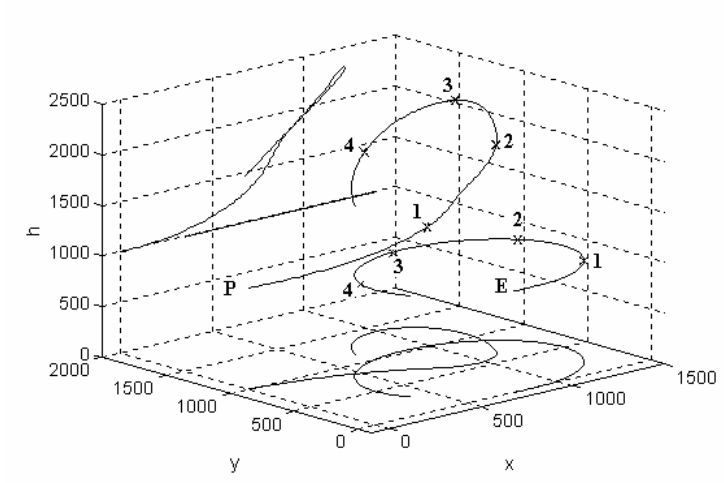


Fig. 4.2. Flight path produced by the modified model against the maneuvering opponent

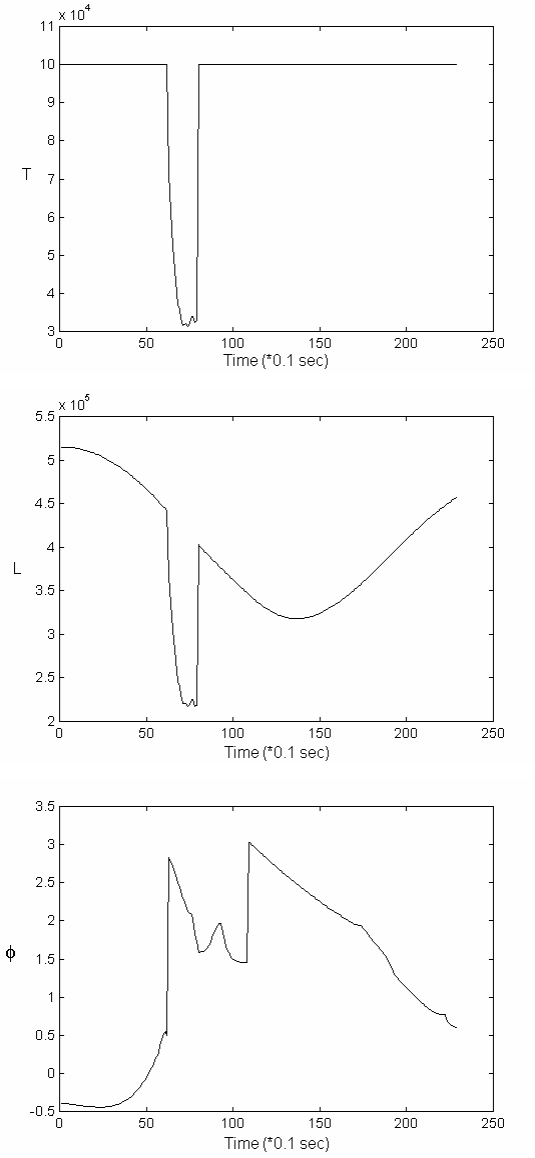


Fig. 4.3. Thrust, Lift and bank angle variation of the aircraft P during a maneuver

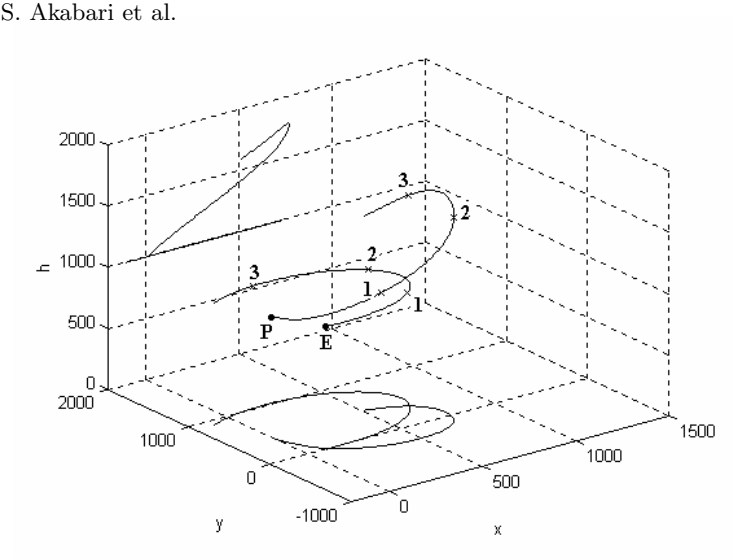


Fig. 4.4. Classic high Yo-Yo maneuver produced by the proposed model

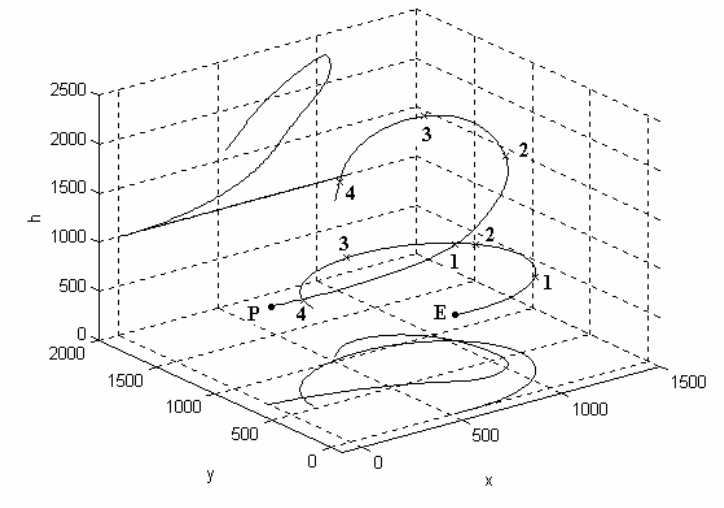


Fig. 4.5. Flight path in a noisy environment

Invited Session

Recent Advances in Theoretical Soft Computing

Session Organiser:
Vilém Novák

Approximation of Fuzzy Functions by Extended Fuzzy Transforms

Martin Štěpnička[1] and Stephan Lehmke[2]

[1] University of Ostrava, Institute for Research and Applications of Fuzzy Modeling, 30.dubna 22, 701 03 Ostrava, Czech Republic
Martin.Stepnicka@osu.cz
[2] University of Dortmund, Computer Science Institute, D-44221 Dortmund, Germany Lehmke@ls1.cs.uni-dortmund.de

Abstract. A fuzzy approximation method called fuzzy transforms for approximation of continuous function is presented in this paper. It is shown how can be fuzzy transforms naturally generalized for functions with more variables. A fuzzy function as an approximated mapping is considered. This leads to an extension of fuzzy transforms for fuzzy function as well as to an extension of generalized fuzzy transforms for fuzzy functions with more variables. It is shown how the proposed method can be used as so called learning to obtain a fuzzy rule base for fuzzy control.

Key words: Fuzzy sets, Approximation, Fuzzy approximation, Fuzzy transforms, Normal forms, Fuzzy control

1 Introduction

Fuzzy transforms (in short F-transforms) have been already several times introduced in a number of publications. Perfilieva I. presented this technique of approximate representation of continuous functions in [4], its application to numeric methods of integrations and solution of ordinary differential equations in [1, 2]. Another application has been published in [6].

The main idea consists in the replacement of an continuous function on a real closed interval by its discrete representation (using the direct F-transform). Afterwards, the discrete representation is transformed back to the space of continuous functions (using the inverse F-transform). The result, obtained by applying both F-transforms is a good simplified approximation of an original function.

In fuzzy control we work with imprecise data and a crisp function describing some proces is described by a fuzzy relation. And any fuzzy relation can be viewed as a fuzzy function. This leads to an idea to extend the method of F-transforms for fuzzy functions to be able to apply it in fuzzy control.

M. Štěpnička and S. Lehmke: *Approximation of Fuzzy Functions by Extended Fuzzy Transforms*, Advances in Soft Computing **2**, 187–195 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

2 Fuzzy Transforms

This section is devoted to Fuzzy transforms - fuzzy approximation method, first published by Perfilieva I. and Chaldeeva E. in [4]. This technique belongs to the area called numerical methods on the basis of fuzzy approximation models. An interval $[a, b]$ of real numbers will be considered as a common domain of all functions in this section.

Definition 1. *Let $x_i = a + h(i-1)$ be nodes on $[a, b]$ where $h = (b-a)(n-1)$, $n \geq 2$ and $i = 1, ..., n$. We say that functions $A_1(x), \ldots A_n(x)$ defined on $[a, b]$ are basic functions if each of them fulfills the following conditions:*

- $A_i : [a, b] \to [0, 1],\ A_i(x_i) = 1,$
- $A_i(x) = 0$ *if* $x \notin (x_{i-1}, x_{i+1})$ *where* $x_0 = a,\ x_{n+1} = b,$
- $A_i(x)$ *is continuous,*
- $A_i(x)$ *strictly increases on* $[x_{i-1}, x_i]$ *and strictly decreases on* $[x_i, x_{i+1}]$,
- $\sum_{i=1}^{n} A_i(x) = 1$, *for all* x,
- $A_i(x_i - x) = A_i(x_i + x)$, *for all* $x \in [0, h],\ i = 2, \ldots, n-1,\ n > 2,$
- $A_{i+1}(x) = A_i(x - h)$, *for all* $x,\ i = 2, \ldots, n-2,\ n > 2.$

We can say that functions $A_i(x)$ determine a fuzzy partition of real interval $[a, b]$.

The technique of fuzzy transforms is based on two transforms - the direct one and the inverse one. The direct fuzzy transform is a mapping which maps continuous functions on $[a, b]$ into the space of real vectors. The inverse F-transform maps a real vector back to the space of continuous functions. We repeat definitions given in [4].

Definition 2. *1. Let $f(x)$ be arbitrary continuous function on $[a, b]$ and $A_1, \ldots, A_n$ basic functions determining a fuzzy partition of $[a, b]$. We say that an n-tuple of real numbers $[F_1, \ldots F_n]$ is the direct F-transform of f with respect to $A_1, \ldots, A_n$ if*

$$F_i = \frac{\int_a^b f(x)A_i(x)\mathrm{d}x}{\int_a^b A_i(x)\mathrm{d}x}. \tag{1}$$

2. Let $f(x)$ be a function known at nodes $x_1, \ldots x_r \in [a, b]$ and $A_1, \ldots, A_n$ basic functions determining a fuzzy partition of $[a, b]$. We say that an n-tuple of real numbers $[F_1, \ldots F_n]$ is the direct F-transform of f with respect to $A_1, \ldots, A_n$ if

$$F_i = \frac{\int_a^b f(x)A_i(x)\mathrm{d}x}{\int_a^b A_i(x)\mathrm{d}x}. \tag{2}$$

Remark: When the basic functions are fixed we denote the direct F-transform of f by $F[f]$ and write $F[f] = [F_1, \dots F_n]$.

Definition 3. *Let $F[f] = [F_1, \dots, F_n]$ be a direct F-transform of a function $f(x)$ with respect to $A_1(x), \dots, A_n(x)$. The function*

$$f_n^F(x) = \sum_{i=1}^{n} F_i A_i(x) \tag{3}$$

will be called an inverse F-transform.

Function f_n^F given by (3) may be considered as an approximation of function $f(x)$. It has been proved (See e.g. [4],[2]) that the sequence of such approximations given by the inverse F-transforms uniformly converges to the original function.

Moreover, it has been shown that components $F_1, ..., F_n$ given by (2), minimize the following piecewise integral least square criterion

$$\Psi(c_1, ..., c_n) = \int_a^b \left(\sum_{i=1}^{n} (f(x) - c_i)^2 A_i(x) \right) \mathrm{d}x \tag{4}$$

and therefore determine the best approximation of $f(x)$ in the following class of approximating functions:

$$\sum_{i=1}^{n} c_i A_i(x), \tag{5}$$

where $c_1, ..., c_n$ are arbitrary real coefficients.

It is worth to mention, that a natural generalization of F-transforms for functions with two and more variables has been done in [7]. This generalization preserves all properties and is based on the following idea.

For simplicity, let us consider a function $f(x, y)$ with two variables which is continuous on $[a, b] \times [c, d]$. At first, we choose basic functions $A_1, ..., A_n$ determining a fuzzy partition of $[a, b]$ and analogously $B_1, ..., B_m$ determining a fuzzy partition of $[c, d]$.Then, formulas for the direct F-transform and the inverse F-transform are as follows:

$$F_{ij} = \frac{\int_c^d \int_a^b f(x, y) A_i(x) B_j(y) \mathrm{d}x \mathrm{d}y}{\int_c^d \int_a^b A_i(x) B_j(y) \mathrm{d}x \mathrm{d}y}, \tag{6}$$

$$f_{n,m}^F(x, y) = \sum_{i=1}^{n} \sum_{j=1}^{m} F_{ij} A_i(x) B_j(y), \tag{7}$$

respectively.

3 Extension for Fuzzy Functions

In fuzzy control, we work with an imprecise data in fuzzy control and a crisp control function $f : X \to Y$ is described by a binary fuzzy relation $\Phi : X \times Y \to [0,1]$ where X, Y are closed real intervals.

Any binary fuzzy relation can be viewed as a mapping (fuzzy function)

$$\Phi : X \to [0,1]^Y, \tag{8}$$

which assigns a fuzzy set $\Phi(x)$ on Y to each node on $x \in X$.

It is a natural idea to extend the technique of F-transforms for approximate representation of control functions to be able to use it in fuzzy control.

At first, let us put some requirements on fuzzy function Φ. We restrict the usage of extended F-transforms only for fuzzy functions with "some" continuity and convexity.

Definition 4. *Let $\Phi : X \times Y \to [0,1]$ be a fuzzy function where X, Y are closed real intervals. Moreover, let $\Phi(x)$ be a convex fuzzy set for all $x \in X$. Then Φ will be called α-continuous if for all $\alpha \in [0,1]$: $\varphi_\alpha^+(x)$ and $\varphi_\alpha^-(x)$ are continuous on X where*

$$\varphi_\alpha^+(x) = \bigvee_{y \in Y} \{y \mid y \in \Phi(x)_\alpha\} \tag{9}$$

$$\varphi_\alpha^-(x) = \bigwedge_{y \in Y} \{y \mid y \in \Phi(x)_\alpha\}. \tag{10}$$

Moreover, if $\Phi(x)$ is a fuzzy set with one-element kernel for all $x \in X$ then $\varphi_1^+(x) = \varphi_1^-(x)$ and we write only $\varphi(x)$.

Applying Zadeh's extensional principal we obtain the following formula for components of extended fuzzy transforms.

Definition 5. *Let $\Phi : X \to [0,1]^Y$ be an α-continuous fuzzy function and $A_1(x), \ldots, A_n(x)$ basic functions forming fuzzy partition of X. We say that n-tuple of fuzzy sets $[\mathcal{F}_1(y), \ldots \mathcal{F}_n(y)]$ on Y is an extended direct F-transform of Φ with respect to $A_1(x), \ldots, A_n(x)$ if*

$$\mathcal{F}_i(y) = \bigvee_{\substack{\xi \in Y^X \\ \frac{\int_X \xi(x) A_i(x) \mathrm{d}x}{\int_X A_i(x) \mathrm{d}x} = y}} \left(\bigwedge_{x \in X} \Phi(x)(\xi(x)) \right). \tag{11}$$

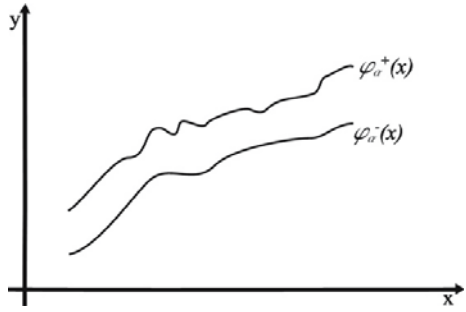


Fig. 1. α-continuity of a fuzzy function

Remark: Due to the fact that in Def 5 we require α-continuity of Φ, it is not necessary to consider all functions $\xi \in Y^X$. It is sufficient to consider functions ξ which are either equal to $\varphi_\alpha^+(x)$ or $\varphi_\alpha^-(x)$ for some $\alpha \in [0,1]$.

This simplifies the formula (11), and we obtain

$$\mathcal{F}_i(y^+) = \alpha \quad \text{where} \quad y^+ = \frac{\int_X \varphi_\alpha^+(x) A_i(x) \mathrm{d}x}{\int_X A_i(x) \mathrm{d}x}, \tag{12}$$

$$\mathcal{F}_i(y^-) = \alpha \quad \text{where} \quad y^- = \frac{\int_X \varphi_\alpha^-(x) A_i(x) \mathrm{d}x}{\int_X A_i(x) \mathrm{d}x}. \tag{13}$$

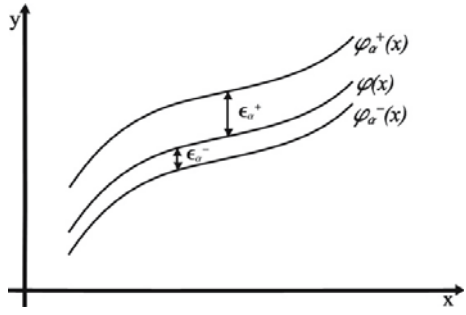


Fig. 2. α-continuity of a fuzzy function with one-element kernel and additional smoothness

Below, we suggest two special simplified cases where we require not only α-continuity, but also some smoothness we additionally require and again one-element kernel of $\Phi(x)$. In the first case, for all $\alpha \in (0,1)$: $\varphi_\alpha^+(x) = \varphi(x) + \varepsilon_\alpha^+$ and $\varphi_\alpha^-(x) = \varphi(x) + \varepsilon_\alpha^-$ where $\varepsilon_\alpha^+, \varepsilon_\alpha^-$ are real numbers.

From which the following implies (See Figure 2)

$$y^+ = \frac{\int_X \varphi_\alpha^+(x) A_i(x) \mathrm{d}x}{\int_X A_i(x) \mathrm{d}x} = \frac{\int_X \varphi(x) A_i(x) \mathrm{d}x}{\int_X A_i(x) \mathrm{d}x} + \frac{\int_X \varepsilon_\alpha^+ A_i(x) \mathrm{d}x}{\int_X A_i(x) \mathrm{d}x} = F_i + \varepsilon_\alpha^+, \tag{14}$$

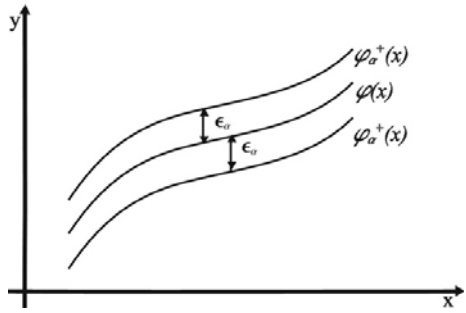


Fig. 3. α-continuity of a fuzzy function with one-element kernel and additional symmetry

$$\text{where} \quad F_i = \frac{\int_X \varphi(x) A_i(x) \mathrm{d}x}{\int_X A_i(x) \mathrm{d}x}.$$

Similarly, $y^- = F_i - \varepsilon_\alpha^-$.

The second case assumes the same conditions as the first one and moreover, the symmetry should hold truth which means $\varepsilon_\alpha^+ = \varepsilon_\alpha^-$ (see Figure 3).

Finally, we have to define an extended inverse F-transform to obtain an approximate representation of the original fuzzy function Φ.

Definition 6. *Let $\mathcal{F}_1, \ldots, \mathcal{F}_n$ be an extended direct F-transform of fuzzy function Φ w.r.t given basic functions $A_1, \ldots, A_n$. Then the fuzzy function*

$$\Phi_n^F(x)(y) = \sum_{i=1}^{n} \mathcal{F}_i(y) A_i(x) \tag{15}$$

is called an extended inverse F-transform.

4 Fuzzy Functions with More Variables

In fuzzy control, we usually meet the situation where control function depends on more than one variable. That is why we have to generalize extended F-transforms for fuzzy functions with more variables.

For the simplicity we introduce the case of fuzzy functions with two variables. Let us be given an α-continuous fuzzy function $\Phi : U \times V \to Y$ where U, V, Y are closed real intervals.

The extension is constructed in the same way as in the case of fuzzy functions with one variable. Having in mind the generalization of F-transform for functions with more variables presented in [7], we construct basic functions $A_1, \ldots, A_n$ and $B_1, \ldots, B_m$ determining fuzzy partitions of U, V, respectively. Then, we obtain the following formula

$$\mathcal{F}_{ij}(y) = \bigvee_{\substack{\xi \in Y^{U\times V} \\ \frac{\int_V \int_U \xi(u,v)A_i(u)B_j(v)\mathrm{d}u\mathrm{d}v}{\int_V \int_U A_i(u)B_j(v)\mathrm{d}u\mathrm{d}v}=y}} \left(\bigwedge_{(u,v)\in U\times V} \Phi(u,v)(\xi(u,v)) \right), \tag{16}$$

for the extended direct F-transform of $\Phi(u, v)$.

For the extended inverse F-transform of $\Phi(u, v)$ we use the following formula

$$\Phi^F_{n,m}(u,v)(y) = \sum_{i=1}^{n}\sum_{j=1}^{m} \mathcal{F}_{ij}(y)A_i(u)B_j(v). \tag{17}$$

5 Data-Based Model

As we have mentioned above and it is published e.g. in [1] it is possible to construct approximate representation of the original function given by the inverse F-transform even if we do not have the full knowledge of the original function. Suppose that f is known at some nodes $x_1, \ldots, x_r$.

This can be viewed as a data-based model with data $(x_k, f(x_k))$ where $k = 1, \ldots, r$. Then we compute the direct F-transform as follows

$$F_i = \frac{\sum_{k=1}^{r} f(x_k)A_i(x_k)}{\sum_{k=1}^{r} A_i(x_k)}. \tag{18}$$

Similarly, we can construct the extended F-transforms based on given data. Let us be given the following sequence of ordered pairs $(x_k, \Phi(x_k))$ where $k = 1, \ldots, r$ and $\Phi(x_k)$ is a fuzzy set on Y. Moreover, let $A_1(x), \ldots, A_n(x)$ be basic functions determining a fuzzy partition. In that case we construct the extended direct F-transform as follows

$$\mathcal{F}_i(y) = \frac{\sum_{k=1}^{r} \Phi(x_k)(y)A_i(x_k)}{\sum_{k=1}^{r} A_i(x_k)}. \tag{19}$$

Of course, the extended inverse F-transform is defined in the same way as formula (15).

Analogously, we construct data-based model for approximation of fuzzy function with more variables. For the case $n = 2$ and data $(u_k, v_l, \Phi(u_k, v_l))$, $k = 1, \ldots, r$ and $l = 1, \ldots s$, we continue as follows

$$\mathcal{F}_{ij}(y) = \frac{\sum_{k=1}^{r}\sum_{l=1}^{s} \Phi(u_k,v_l)(y)A_i(u_k)B_j(v_l)}{\sum_{k=1}^{r}\sum_{l=1}^{s} A_i(u_k)B_j(v_l)}. \tag{20}$$

Let us stress, that data where u_k, v_l are real numbers and $\Phi(u_k, v_l)$ are fuzzy sets are quite natural. These data may be obtained by questioning some expert. For example in fuzzy control of a dynamic robot we know distance E

between the robot and the wall at each moment and we know its derivation dE. We ask some expert about control action. Answer could be either a linguistic expression or a fuzzy number.

Because of this, we obtain the following $n \times m$ fuzzy rules

$$\text{IF } E \text{ is } \mathcal{A}_i \text{ AND } \mathrm{d}E \text{ is } \mathcal{B}_j \text{ THEN } y \text{ is } \mathcal{F}_{ij} \tag{21}$$

comprising a fuzzy rule base for fuzzy control of a dynamic robot. This implies that this method can be used as a learning method.

The model of these IF-THEN rules can be constructed according to formula (17). Due to the choice of basic functions and properties of fuzzy functions, ordinary sums in formula (17) can be replaced by Łukasiewicz sums. Moreover, because basic functions can be represented with fuzzy similarity relation with finite number of fixed, this formula can be taken as an additive normal form with orthogonal condition. It lies between well known conjunctive and disjunctive normal forms (see [3], [5]).

6 Conclusions

We have recalled a numerical method on the basis of fuzzy approximating model called fuzzy transforms. The construction of basic functions (special fuzzy sets determining a fuzzy partition) has been repeated as well. The method of fuzzy transforms has been introduced as a universal approximation method which is possible to use for a (discrete) data-based model.

This method has been successfully extended for fuzzy functions what can be useful in fuzzy control. Furthermore, a generalized F-transforms for functions with more variables have been similarly extended as well. Moreover, a discrete case of extended fuzzy transforms as a data-based model has been introduced. It has been also shown how extended F-transforms can be used as a learning method.

Acknowledgement

This investigation has been partially supported by grant IAA 1187301 of the GA AV ČR and by the Deutsche Forschungsgemeinschaft as part of the Collaborative Research Center "Computational Intelligence" (531).

References

1. Perfilieva I (2004) Fuzzy Transforms. In: Dubois D, Grzymala–Buse J, Inuiguchi M, Polkowski L (eds) Fuzzy Sets and Rough Sets, Lecture Notes in Computer Science. Springer, Heidelberg, to appear
2. Perfilieva I (2003) Fuzzy Approach to Solution of Differential Equations with Imprecise Data: Application to Reef Growth Problem. In: Demicco R V, Klir G J (eds) Fuzzy Logic in Geology. Academic Press, Amsterdam

3. Daňková M (2002) Representation of Logic Fromulas by Normal Forms, Kybernetika 38:717–728
4. Perfilieva I, Chaldeeva E (2001) Proc of the 4th Czech - Japan Seminar on Data Analysis and Decision Making under Uncertainty:116–124
5. Perfilieva I (2004) Normal Forms in BL and ŁΠ Algebras of Functions, Soft Computing 8:291–298
6. Štěpnička M (2003) Fuzzy Transformation and Its Applications in A/D Converter, Journal of Electrical Engineering Vol. 54, NO. 12/s:72–75
7. Štěpnička M, Valášek R (2004) Fuzzy Transforms and Their Application on Wave Equation, Journal of Electrical Engineering, to appear

Fuzzy Control as a General Interpolation Problem

Siegfried Gottwald*

Universität Leipzig, Institut für Logik und Wissenschaftstheorie
Beethovenstr. 15, 04107 Leipzig, Germany
gottwald@uni-leipzig.de

Abstract. The general mathematical problem of fuzzy control is an interpolation problem: a list of fuzzy input-output data, usually provided by a list of linguistic control rules, should be realized as argument-value pairs for a suitably chosen fuzzy function. However, contrary to the usual understanding of interpolation, in the actual approaches this interpolation problem is considered as a global one: one uniformly and globally defined function should realize all the fuzzy input-output data.

In standard classes of functions thus this interpolation problem often becomes unsolvable. Hence it becomes intertwined with an approximation problem which allows that the given fuzzy input-output data are realized only approximately by argument-value pairs.

In this context the paper discusses some quite general sufficient conditions for the true solution of the interpolation problem, as well as similar conditions for suitably modified data, i.e. for a quite controlled approximation.

1.1 Introduction

The standard paradigm of fuzzy control is that one supposes to have given, as an incomplete and fuzzy description of a control function Φ from an input space $\mathbf{X}$ to an output space $\mathbf{Y}$, a family

$$\mathcal{D} = (\langle A_i, B_i \rangle)_{1 \leq i \leq n} \tag{1.1}$$

of (fuzzy) input-output data pairs to characterize this function Φ.

In the usual approaches such a family of input-output data pairs is provided by a finite list

$$\text{IF } x \text{ is } A_i \quad \text{THEN } y \text{ is } B_i, \qquad i = 1, \ldots, n \tag{1.2}$$

of linguistic control rules, also called fuzzy IF-THEN rules.

* The present work has partly been supported by the German-Czech cooperation project CZE 01/022 of the BMBF.

S. Gottwald: *Fuzzy Control as a General Interpolation Problem*, Advances in Soft Computing **2**, 197–204 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

The main mathematical problem of fuzzy control, besides the engineering problem to get a suitable list of linguistic control rules for the actual control problem, is therefore the interpolation problem to find a function $\Phi^* : \mathcal{F}(\mathbf{X}) \to \mathcal{F}(\mathbf{Y})$ which interpolates these data, i.e. which satisfies

$$\Phi^*(A_i) = B_i \quad \text{for each } i = 1, \ldots, n\,, \tag{1.3}$$

and which in this way gives a fuzzy representation for the control function Φ.

Actually the standard approach is to look for *one* single function which should interpolate all these data, and which should be globally defined over $\mathcal{F}(\mathbf{X})$.

This "global" interpolation problem, presented by such a finite family (1.1) of input-output data only, in general has different solutions. However, the main approach toward this global interpolation problem is to search for a solution in a restricted class $\mathcal{IF}$ of functions. And such a restriction of the class of interpolating functions offers also the possibility that within such a class $\mathcal{IF}$ of interpolating functions the interpolation problem becomes unsolvable.

Interpolation in numerical mathematics for functions f over the reals, i.e. over a linearly ordered domain, usually is understood as a *local* representation of the given data via functions of some predetermined type, e.g. by polynomials, which are supposed to realize some "few" neighboring argument-value pairs for f. And this means that some such "few" neighboring argument-value pairs determine the domains of these local representations of the interpolating function. Contrary to this standard usage, in fuzzy control up to now one has not discussed any version of a "localized" interpolation approach.

Instead, the global interpolation problem becomes in a natural way intertwined with an *approximation problem*: one may be interested to look for a function $\Psi^* \in \mathcal{IF}$ which does not really interpolate, but which "realizes" the given fuzzy input-output data "suitably well". Such an approximative approach is completely reasonable if one has in mind that even a true solution Φ^* of the interpolation problem (1.3) only gives a fuzzy representation for the crisp control function Φ.

1.2 Two standard interpolation strategies

More or less the standard theoretical understanding for the design of a fuzzy controller is the reference to the *compositional rule of inference* (CRI) first discussed by Zadeh [10].

A suitable general context for the structure of the corresponding membership degrees, which at the same time are truth degrees of a corresponding many-valued logic, is a lattice ordered abelian monoid enriched with a further operation $\rightarrowtail$, which is connected with the semigroup operation $*$ by the adjointness condition

$$x * z \leq y \quad \text{iff} \quad z \leq (x \rightarrowtail y)\,.$$

The resulting structure often is called a *residuated lattice*. Its corresponding formalized language has besides the (idempotent) conjunction $\wedge$ which is provided by the lattice meet a further (in general not idempotent) "strong" conjunction &, which has the semigroup operation $*$ as its truth degree function.

This formalized language may be further enlarged by a suitable class term notation for fuzzy sets by writing $\{x \parallel H(x)\}$ to denote that one fuzzy set A which has as its membership degree $A(a)$ in the point a of the universe of discourse just the truth degree of the formula $H(a)$.

This context yields for the CRI-based strategy, which was first applied to a control problem by Mamdani/Assilian [7], the following formulation:

From the data (A_i, B_i) one determines a fuzzy relation R in such a way that the approximating function Ψ_R^* for Φ^* becomes "describable" as

$$\Psi_R^*(A) = A \circ R = \{y \parallel \exists x(A(x) \,\&\, R(x,y))\}\,, \tag{1.4}$$

which means, for the membership degrees, the well known definition

$$\Psi_R^*(A)(y) = \sup_{x \in \mathbf{X}} \big(A(x) * R(x,y)\big)\,.$$

Of course, the most preferable situation would be that the function Ψ_R^* really interpolates the given input-output-data.

In general we shall call functions which can, according to (1.4), be represented by a fuzzy relation R simply *CRI-representable*.

A closer look at fuzzy control applications shows that one has, besides this approach via CRI-representable functions and a final application of the CRI to fuzzy input data, also a competing approach: the *method of activation degrees* which first was used by Holmblad/Ostergaard [6] in their fuzzy control algorithm for a cement kiln.

This method of activation degrees changes the previous CRI-based approach in the following way:

For each actual input fuzzy set A and each single input-output data pair (A_k, B_k) one determines a modification B_k^* of the corresponding "local" output B_k, characterized only by the local data (A_k, B_k) and the actual input A, and finally aggregates all these modified "local" outputs into one global output:

$$\Xi^*(A) \;=\; \bigcup_{i=1}^{n} B_i^*\,. \tag{1.5}$$

The particular choice of Holmblad/Ostergaard for B_k^* has been

$$B_k^*(y) = \mathsf{hgt}\,(A \cap A_k) \cdot B_k(y)\,. \tag{1.6}$$

In general terms, this modification of the first mentioned approach does not only offer one particular diverging approach toward the general interpolation problem, it also indicates that besides those both CRI-related approaches other ones with different inference and perhaps also with different aggregation operations could be of interest – as long as they are determined by finite lists of input-output data (A_i, B_i) and realize mappings from $\mathcal{F}(\mathbf{X})$ to $\mathcal{F}(\mathbf{Y})$.

This has not been done up to now in sufficient generality. Further on in this paper we shall present some considerations which point in this direction.

1.3 Some general interpolation strategies

There is the well known distinction between FATI and FITA strategies to evaluate systems of linguistic control rules w.r.t. arbitrary fuzzy inputs from $\mathcal{F}(\mathbf{X})$.

The core idea of a FITA strategy is that it is a strategy which **F**irst **I**nfers (by reference to the single rules) and **T**hen **A**ggregates starting from the actual input information A. Contrary to that, a FATI strategy is a strategy which **F**irst **A**ggregates (the information in all the rules into one fuzzy relation) and **T**hen **I**nfers starting from the actual input information A.

From the two standard interpolation strategies of the last section, obviously (1.4) offers a FATI strategy, and (1.5) provides a FITA strategy.

Both these strategies use the set theoretic union as their aggregation operator. Furthermore, both of them refer to the compositional rule of inference (CRI) as their core tool of inference.

In general, however, the interpolation operators we intend to consider depend more generally upon some inference operator(s) as well as upon some aggregation operator.

By an *inference operator* we mean here simply a mapping from the fuzzy subsets of the input space to the fuzzy subsets of the output space.[2]

And an *aggregation operator* $\mathbf{A}$, as explained e.g. in [1, 2], is a family $(f^n)_{n\in\mathbb{N}}$ of operations, each f^n an n-ary one, over some partially ordered set $\mathbf{M}$ with a bottom element $\mathbf{0}$ and a top element $\mathbf{1}$, such that each operation f^n is non-decreasing, maps the bottom to the bottom: $f^n(\mathbf{0},\ldots,\mathbf{0}) = \mathbf{0}$, and the top to the top: $f^n(\mathbf{1},\ldots,\mathbf{1}) = \mathbf{1}$. Such an aggregation operator $\mathbf{A} = (f^n)_{n\in\mathbb{N}}$ is a *commutative* one iff each operation f^n is commutative. And $\mathbf{A}$ is an *associative* aggregation operator iff e.g. for $n = k + l$ one always has $f^n(a_1,\ldots,a_n) = f^2(f^k(a_1,\ldots,a_k), f^l(a_{k+1},\ldots,a_n))$ and in general

$$f^n(a_1,\ldots,a_n) = f^r(f^{k_1}(a_1,\ldots,a_{k_1}),\ldots,f^{k_r}(a_{m+1},\ldots,a_n))$$

for $n = \sum_{i=1}^{r} k_i$ and $m = \sum_{i=1}^{r-1} k_i$.

[2] This terminology has its historical roots in the fuzzy control community. There is no relationship at all with the logical notion of inference intended here.

Our aggregation operators further on are supposed to be commutative as well as associative ones.[3]

If we now consider interpolation operators Φ of FITA-type and interpolation operators Ψ of FATI-type then they have the abstract forms

$$\Psi_{\mathcal{D}}(A) = \mathbf{A}(\theta_1(A), \ldots, \theta_n(A)), \tag{1.7}$$

$$\Xi_{\mathcal{D}}(A) = \widehat{\mathbf{A}}(\theta_1, \ldots, \theta_n)(A). \tag{1.8}$$

Here we assume that each one of the "local" inference operators θ_i is determined by the single input-output pair $\langle A_i, B_i \rangle$. Therefore we occasionally shall write $\theta_{\langle A_i, B_i \rangle}$ instead of θ_i only. And we have to assume that the aggregation operator $\mathbf{A}$ operates on fuzzy sets, and that the aggregation operator $\widehat{\mathbf{A}}$ operates on inference operators.

With this extended notation the formulas (1.7), (1.8) become

$$\Psi_{\mathcal{D}}(A) = \mathbf{A}(\theta_{\langle A_1, B_1 \rangle}(A), \ldots, \theta_{\langle A_n, B_n \rangle}(A)), \tag{1.9}$$

$$\Xi_{\mathcal{D}}(A) = \widehat{\mathbf{A}}(\theta_{\langle A_1, B_1 \rangle}, \ldots, \theta_{\langle A_n, B_n \rangle})(A). \tag{1.10}$$

Some particular cases of these interpolation procedures have been discussed in [8]. These authors consider four different cases. First they look at the FITA-type interpolation

$$\Psi^1_{\mathcal{D}}(A) = \bigcap_i \big(A \circ (A_i \rhd B_i)\big), \tag{1.11}$$

using as in [4] the notation $A_i \rhd B_i$ to denote the fuzzy relation with membership function

$$(A_i \rhd B_i)(x, y) = A_i(x) \rightarrowtail B_i(y).$$

Their second example discusses a FATI-type approach given by

$$\Xi^2_{\mathcal{D}}(A) = A \circ \bigcap_i \big((A_i \rhd B_i)\big), \tag{1.12}$$

and is thus just the common CRI-based strategy of the S-pseudo-solution, used in this general form already in [3], cf. also [4].

Their third example is again of FITA-type and determined by

$$\Psi^3_{\mathcal{D}}(A) = \bigcap_i \{y \,\|\, \delta(A, A_i) \to B_i(y)\}, \tag{1.13}$$

using besides the previously mentioned class term notation for fuzzy sets the activation degree

$$\delta(A, A_i) = \bigwedge_{x \in \mathbf{X}} (A(x) \to A_i(x)) \tag{1.14}$$

[3] It seems that this is a rather restrictive choice from a theoretical point of view. However, in all the usual cases these restrictions are satisfied.

which is a degree of subsethood of the actual input fuzzy set A w.r.t. the i-th rule input A_i.

And the fourth one is a modification of the third one, determined by

$$\Psi^4_{\mathcal{D}}(A) = \bigcap_{\emptyset \neq J \subseteq N} \{y \,\|\, \delta(A, \bigcup_{j \in J} A_j) \to \bigcup_{j \in J} B_i(y)\}\,, \tag{1.15}$$

using $N = \{1, 2, \ldots, n\}$.

1.4 Stability conditions for the given data

If $\Theta_{\mathcal{D}}$ is a fuzzy inference operator of one of the types (1.9), (1.10), then the interpolation property one likes to have realized is that one has

$$\Theta_{\mathcal{D}}(A_i) = B_i \tag{1.16}$$

for all the data pairs $\langle A_i, B_i \rangle$. In the particular case that the operator $\Theta_{\mathcal{D}}$ is given by (1.4), this is just the problem to solve the system (1.16) of fuzzy relation equations.

Definition 1 *In the present generalized context let us call the property (1.16) the* $\mathcal{D}$-stability *of the fuzzy inference operator* $\Theta_{\mathcal{D}}$.

To find $\mathcal{D}$-stability conditions on this abstract level seems to be rather difficult in general. However, the restriction to fuzzy inference operators of FITA-type makes things easier.

It is necessary to have a closer look at the aggregation operator $\mathbf{A} = (f^n)_{n \in \mathbb{N}}$ involved in (1.7) which operates on $\mathcal{F}(\mathbf{Y})$, of course with inclusion as partial ordering.

Definition 2 *Having* $B, C \in \mathcal{F}(\mathbf{Y})$ *we say that* C *is* $\mathbf{A}$-negligible w.r.t. B *iff* $f^2(B, C) = f^1(B)$ *holds true.*

The core idea here is that in any aggregation by $\mathbf{A}$ the presence of the fuzzy set B among the aggregated fuzzy sets makes any presence of C superfluous.

Proposition 1 *Consider a fuzzy inference operator of FITA-type*

$$\Psi_{\mathcal{D}} = \mathbf{A}(\theta_{\langle A_1, B_1 \rangle}, \ldots, \theta_{\langle A_n, B_n \rangle})\,.$$

It is sufficient for the $\mathcal{D}$*-stability of* $\Psi_{\mathcal{D}}$ *to have*

$$\Phi_{\mathcal{D}}(A_k) = B_k$$

that one has $\theta_{\langle A_k, B_k \rangle}(A_k) = B_k$ *and that for each* $i \neq k$ *the fuzzy set* $\theta_{\langle A_k, B_k \rangle}(A_i)$ *is* $\mathbf{A}$*-negligible w.r.t.* $\theta_{\langle A_k, B_k \rangle}(A_k)$.

The proof follows immediately from the corresponding definitions.

Corollary 2 *It is sufficient for the $\mathcal{D}$-stability of a fuzzy inference operator $\Psi_{\mathcal{D}}$ of FITA-type that one has*

$$\Psi_{\mathcal{D}}(A_i) = B_i \qquad \text{for all } 1 \leq i \leq n$$

and that always $\theta_{\langle A_i, B_i \rangle}(A_j)$ is **A***-negligible w.r.t. $\theta_{\langle A_i, B_i \rangle}(A_i)$.*

1.5 Stability conditions for modified data

The combined approximation and interpolation problem, as previously explained, sheds new light on the standard approaches toward fuzzy control via CRI-representable functions originating from the works of Mamdani/Assilian [7] and Sanchez [9] particularly for the case that neither the Mamdani/Assilian relation R_{MA}, determined by the membership degrees

$$R_{\mathrm{MA}}(x, y) = \bigvee_{i=1}^{n} A_i(x) * B_i(y)\,, \tag{1.17}$$

nor the Sanchez relation $\widehat{R}$, determined by the membership degrees

$$\widehat{R}(x, y) = \bigwedge_{i=1}^{n} (A_i(x) \rightarrowtail B_i(y))\,, \tag{1.18}$$

offer a solution for the system of fuzzy relation equations. In any case both these fuzzy relations determine CRI-representable fuzzy functions which provide approximate solutions for the interpolation problem.

In other words, the consideration of CRI-representable functions determined by (1.17) as well as by (1.18) provides two methods for an approximate solution of the main interpolation problem. As is well known and explained e.g. in [4], the approximating interpolation function CRI-represented by $\widehat{R}$ always gives a lower approximation, and that one CRI-represented by R_{MA} gives an upper approximation for normal input data.

Extending these results, in [5] the iterative combination of these methods has been discussed to get better approximation results. For the iterations there, always the next iteration step consisted in an application of a predetermined one of the two approximation methods to the data family with the original input data and the real, approximating output data which resulted from the application of the former approximation method.

A similar iteration idea was also discussed in [8], however restricted always to the iteration of only one of the approximation methods explained in (1.11), (1.12), (1.13), and (1.15).

Therefore let us now, in the general context given earlier in this paper, discuss the problem of $\mathcal{D}$-stability for a modified operator $\Theta^*_{\mathcal{D}}$ which is determined by the kind of iteration of $\Theta_{\mathcal{D}}$ just explained.

Let us consider the $\Theta_{\mathcal{D}}$-*modified* data set $\mathcal{D}^*$ given as

$$\mathcal{D}^* = (\langle A_i, \Theta_{\mathcal{D}}(A_i)\rangle)_{1\leq i\leq n}\,, \tag{1.19}$$

and define from it the modified fuzzy inference operator $\Theta^*_{\mathcal{D}}$ as

$$\Theta^*_{\mathcal{D}} = \Theta_{\mathcal{D}^*}\,. \tag{1.20}$$

For these modifications, the problem of stability reappears. Of course, the new situation here is only a particular case of the former. And it becomes a simpler one in the sense that the stability criteria now refer only to the input data A_i of the data set $\mathcal{D} = (\langle A_i, B_i\rangle)_{1\leq i\leq n}$. We give only the adaption of Corollary 2.

Proposition 3 *It is sufficient for the $\mathcal{D}^*$-stability of a fuzzy inference operator $\Psi^*_{\mathcal{D}}$ of FITA-type that one has*

$$\Psi^*_{\mathcal{D}}(A_i) = \Psi_{\mathcal{D}^*}(A_i) = \Psi_{\mathcal{D}}(A_i) \qquad \textit{for all } \; 1 \leq i \leq n$$

and that always $\theta_{\langle A_i, \Psi_{\mathcal{D}}(A_i)\rangle}(A_j)$ is **A***-negligible w.r.t. $\theta_{\langle A_i, \Psi_{\mathcal{D}}(A_i)\rangle}(A_i)$.*

References

1. T. Calco, G. Mayor, R. Mesiar (eds.): *Aggregation Operators: New Trends and Applications*, Physica-Verlag: Heidelberg, 2002.
2. D. Dubois, H. Prade: On the use of aggregation operations in information fusion processes, *Fuzzy Sets Systems* **142** (2004), 143–161.
3. S. Gottwald: Characterizations of the solvability of fuzzy equations. *Elektron. Informationsverarb. Kybernet.* **22** (1986) 67–91.
4. S. Gottwald: *Fuzzy Sets and Fuzzy Logic.* The Foundations of Application – From a Mathematical Point of View, Vieweg: Braunschweig/Wiesbaden and Teknea: Toulouse, 1993.
5. S. Gottwald, V. Novák, I. Perfilieva: Fuzzy control and t-norm-based fuzzy logic. Some recent results, in: *Proc. 9th Internat. Conf.* IPMU'2002, ESIA – Université de Savoie, Annecy, 2002, 1087–1094.
6. L.P. Holmblad, J.J. Ostergaard: Control of a cement kiln by fuzzy logic, in: M.M. Gupta/E. Sanchez (eds.), *Fuzzy Information and Decision Processes.* North-Holland Publ. Comp.: Amsterdam, 1982, 389–399.
7. A. Mamdani, S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, *Internat. J. Man-Machine Studies* **7** (1975) 1–13.
8. N.N. Morsi, A.A. Fahmy, On generalized modus ponens with multiple rules and a residuated implication, *Fuzzy Sets Systems*, **129** (2002), 267–274.
9. E. Sanchez, Resolution of composite fuzzy relation equations, *Information and Control*, **30** (1976) 38–48.
10. L.A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. Systems, Man and Cybernet.* **SMC-3** (1973) 28–44.

Galois Connections with Truth Stressers: Foundations for Formal Concept Analysis of Object-Attribute Data with Fuzzy Attributes

Radim Bělohlávek[1],[2], Taťána Funioková[1], and Vilém Vychodil[1]

[1] Dept. Computer Science, Palacky University, Olomouc, , Czech Republic, {radim.belohlavek,tatana.funiokova,vilem.vychodil}@upol.cz
[2] Inst. Research and Applications of Fuzzy Modeling, University of Ostrava, Ostrava, Czech Republic

1 Prologue: (Fuzzy) Galois Connections and Their Applications, and the Need for Further Development

Galois connections appear in several areas of mathematics and computer science, and their applications. A Galois connection between sets X and Y is a pair $\langle^{\uparrow}, {}^{\downarrow}\rangle$ of mappings $^{\uparrow}$ assigning subcollections of Y to subcollections of X, and $^{\downarrow}$ assigning subcollections of X to subcollections of Y. By definition, Galois connections have to satisfy certain conditions. Galois connections can be interpreted in the following manner: For subcollections A and B of X and Y, respectively, $A^{\uparrow}$ is the collection of all elements of Y which are in a certain relationship to all elements from A, and $B^{\downarrow}$ is the collection of all elements of X which are in the relationship to all elements in B. From the very many examples of Galois connections in mathematics, let us recall the following. Let X be the set of all logical formulas of a given language, Y be the set of all structures (interpretations) of the same language. For $A \subseteq X$ and $B \subseteq Y$, let $A^{\uparrow}$ consist of all structures in which each formula from A is true, let $B^{\downarrow}$ denote the set of all formulas which are true in each structure from B. Then, $^{\uparrow}$ and $^{\downarrow}$ is a Galois connection.

As an example of applications of Galois connections, consider the following example which is the main source of inspiration for the present paper. Let X and Y denote a set of objects and attributes, respectively, Let I denote the relationship "to have" between objects and attributes. Then X, Y, and I can be seen as representing an object-attribute data table (for instance, organisms as objects, and their properties as attributes). If, for subcollections A of X and B of Y, $A^{\uparrow}$ denotes the collection of all attributes shared by all objects from A, and $B^{\downarrow}$ denotes the collection of all objects sharing all attributes from B, then $^{\uparrow}$ and $^{\downarrow}$ form a Galois connection. These connections form the core of so-called formal concept analysis (FCA) of object-attribute data, see [18]. Fixed

R. Bělohlávek et al.: *Galois Connections with Truth Stressers: Foundations for Formal Concept Analysis of Object-Attribute Data with Fuzzy Attributes*, Advances in Soft Computing **2**, 205–219 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

points of these connections, i.e. pairs $\langle A, B \rangle$ for which $A^{\uparrow} = B$ and $B^{\downarrow} = A$, are called formal concepts and represent interesting clusters found in the data table (formal concepts in the above-mentioned table with organisms and their properties may be mammals, warm-blooded organisms, etc.). Formal concepts can be partially ordered by subconsept-superconcept hierarchy (a concept can be more general or more particular than a given concept). For instance, the concept "mammal" is more general than "dog". The hierarchically ordered set of all formal concepts, so-called concept lattice, provide us with a derived conceptual information hidden in the data. Formal concept analysis can be thought of as directly formalizing the ideas on what are concepts as developed by so-called Port-Royal logic [1]. FCA has found applications in several areas (software engineering, psychology, text classification, reengineering).

Galois connections have been explicitly introduced in [26]. After some suggestions to use Galois connections for data analysis by Birkhoff, the first systematic paper on data analysis using Galois connections is [3], see also [4]. Probably the most influential paper in FCA is Wille's [29] which started an intensive research on FCA. In basic setting, FCA deals with bivalent attributes, i.e. each object either has (degree 1) or does not have (degree 0) a given attribute. In order to deal with fuzzy (graded) attributes, FCA has been generalized in several papers, see e.g. [5, 16, 27]. A fuzzy attribute can apply to an object to a degree in between 0 and 1 (e.g. 0.3), i.e not only 0 or 1 as in case of bivalent attributes. Galois connections generalized from the point of view of fuzzy approach so that they correspond to FCA of data with fuzzy attributes have been introduced in [6]; fuzzy concept lattices, i.e. fixed points of fuzzy Galois connections, have been studied in [7, 13].

The main motivation of the present paper stems from [14] where the authors showed a way to reduce the number of formal concepts in FCA with fuzzy attributes by considering only so-called crisply generated formal concepts. Crisply generated concepts can be considered as natural concepts with clear interpretation. Moreover, as shown in [14], they can be efficiently generated (without the need to generate all formal concepts and to test whether a particular concept is crisply generated). Now, the question is whether crisply generated formal concepts are fixed points of structures analogous to Galois connections. The present paper gives a positive answer to this question, and elaborates more on the presented topic and some related problems.

2 Preliminaries

We pick complete residuated lattices as the structures of truth values. Complete residuated lattices, being introduced in the 1930s in ring theory, were introduced into the context of fuzzy logic by Goguen [19]. Various logical calculi were investigated using residuated lattices or particular types of residuated lattices. A thorough information about the role of residuated lattices in fuzzy logic can be obtained in [20, 21, 25]. Recall that a (complete) residuated

lattice is an algebra $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \rightarrow, 0, 1\rangle$ such that $\langle L, \wedge, \vee, 0, 1\rangle$ is a (complete) lattice with the least element 0 and the greatest element 1, $\langle L, \otimes, 1\rangle$ is a commutative monoid (i.e. $\otimes$ is a commutative and associative binary operation on L satisfying $a \otimes 1 = a$), and $\otimes$, $\rightarrow$ form an adjoint pair, i.e. $a \otimes b \leq c$ if and only if $a \leq b \rightarrow c$ is valid for each $a, b, c \in L$. In the following, $\mathbf{L}$ denotes an arbitrary complete residuated lattice (with L being the universe set of $\mathbf{L}$). All properties of complete residuated lattices used in the sequel are well-known and can be found e.g. in [11]. Note that particular types of residuated lattices (distinguishable by identities) include Boolean algebras, Heyting algebras, algebras of Girard's linear logic, MV-algebras, Gödel algebras, product algebras, and more generally, BL-algebras (see [21, 23]).

Of particular interest are complete residuated lattices defined on the real unit interval $[0, 1]$ or on some subchain of $[0, 1]$. It can be shown (see e.g. [11]) that $\mathbf{L} = \langle [0, 1], \min, \max, \otimes, \rightarrow, 0, 1\rangle$ is a complete residuated lattice if and only if $\otimes$ is a left-continuous t-norm and $\rightarrow$ is defined by $a \rightarrow b = \max\{c \mid a \otimes c \leq b\}$. A t-norm is a binary operation on $[0, 1]$ which is associative, commutative, monotone, and has 1 as its neutral element, and hence, captures the basic properties of conjunction. A t-norm is called left-continuous if, as a real function, it is left-continuous in both arguments. Most commonly used are continuous t-norms, the basic three of which are Łukasiewicz t-norm (given by $a \otimes b = \max(a + b - 1, 0)$ with the corresponding residuum $a \rightarrow b = \min(1 - a + b, 1)$), minimum (also called Gödel) t-norm ($a \otimes b = \min(a, b)$, $a \rightarrow b = 1$ if $a \leq b$ and $= b$ else), and product t-norm ($a \otimes b = a \cdot b$, $a \rightarrow b = 1$ if $a \leq b$ and $= b/a$ else). It can be shown (see e.g. [24]) that each continuous t-norm is composed out of the three above-mentioned t-norms by a simple construction (ordinal sum). Any finite subchain of $[0, 1]$ containing both 0 and 1, equipped with restrictions of the minimum t-norm and its residuum is a complete residuated lattice. Furthermore, the same holds true for any equidistant finite chain $\{0, \frac{1}{n}, \ldots, \frac{n-1}{n}, 1\}$ equipped with restrictions of Łukasiewicz operations. The only residuated lattice on the two-element chain $\{0, 1\}$ (with $0 < 1$) has the classical conjunction operation as $\otimes$ and classical implication operation as $\rightarrow$. That is, the two-element residuated lattice is the two-element Boolean algebra of classical logic.

A fuzzy set with truth degrees from a complete residuated lattice $\mathbf{L}$ (also simply an $\mathbf{L}$-set) in a universe set X is any mapping $A\colon X \rightarrow L$, $A(x) \in L$ being interpreted as the truth value of "x belongs to A".

Analogously, an n-ary $\mathbf{L}$-relation on a universe set X is an $\mathbf{L}$-set in the universe set X^n, e.g. a binary relation R on X is a mapping $R\colon X \times X \rightarrow L$. A singleton is a fuzzy set $\{ {}^{a}\!/\!_{x}\}$ for which $\{ {}^{a}\!/\!_{x}\}(x) = a$ and $\{ {}^{a}\!/\!_{x}\}(y) = 0$ for $y \neq x$. A fuzzy set A is called normal if $A(x) = 1$ for some $x \in X$. For $a \in L$, the a-cut of a fuzzy set $A \in L^X$ is the ordinary subset ${}^{a}A = \{x \in X \mid A(x) \geq a\}$ of X. For $\mathbf{L}$-sets A and B in X we define $(A \approx B) = \bigwedge_{x \in X}(A(x) \leftrightarrow B(x))$ (degree of equality of A and B) and $S(A, B) = \bigwedge_{x \in X}(A(x) \rightarrow B(x))$ (degree of subsethood of A in B). Note that $\leftrightarrow$ is defined by $a \leftrightarrow b = (a \rightarrow b) \wedge (b \rightarrow a)$.

Clearly, $(A \approx B) = S(A,B) \wedge S(B,A)$. Furthermore, we write $A \subseteq B$ (A is a subset of B) if $S(A,B) = 1$, i.e. for each $x \in X$, $A(x) \leq B(x)$. $A \subset B$ means $A \subseteq B$ and $A \neq B$. The set of all $\mathbf{L}$-sets in X will be denoted by L^X. Note that the operations of $\mathbf{L}$ induce the corresponding operations on L^X. For example, we have intersection $\bigcap$ on L^X induced by the infimum $\bigwedge$ of $\mathbf{L}$ by $(\bigcap_{i\in I} A_i)(x) = \bigwedge_{i\in I} A_i(x)$, etc. A fuzzy set A in X is called crisp if $A(x) = 0$ or $A(x) = 1$ for each $x \in X$. In this case, we write also $A \subseteq X$ since A may be obviously identified with an ordinary subset of X.

3 Galois Connections with Truth Stressers

3.1 Coming to Galois Connections with Truth Stressers

Fuzzy Galois connections and concept lattices

Let X and Y be sets of objects and attributes, respectively, I be a fuzzy relation between X and Y. That is, $I : X \times Y \to L$ assigns to each $x \in X$ and each $y \in Y$ a truth degree $I(x,y) \in L$ to which object x has attribute y (L is a support set of some complete residuated lattice $\mathbf{L}$). The triplet $\langle X, Y, I\rangle$ is called a formal fuzzy context.

For fuzzy sets $A \in L^X$ and $B \in L^Y$, consider fuzzy sets $A^{\uparrow} \in L^Y$ and $B^{\downarrow} \in L^X$ (denoted also $A^{\uparrow_I}$ and $B^{\downarrow_I}$) defined by

$$A^{\uparrow}(y) = \bigwedge_{x\in X} (A(x) \to I(x,y)) \tag{1}$$

and

$$B^{\downarrow}(x) = \bigwedge_{y\in Y} (B(y) \to I(x,y)). \tag{2}$$

Using basic rules of predicate fuzzy logic [11], one can easily see that $A^{\uparrow}(y)$ is the truth degree of the fact "y is shared by all objects from A" and $B^{\downarrow}(x)$ is the truth degree of the fact "x has all attributes from B". Putting

$$\mathcal{B}(X,Y,I) = \{\langle A,B\rangle \mid A^{\uparrow} = B,\ B^{\downarrow} = A\},$$

$\mathcal{B}(X,Y,I)$ is the set of all pairs $\langle A,B\rangle$ such that (a) A is the collection of all objects that have all the attributes of (the intent) B and (b) B is the collection of all attributes that are shared by all the objects of (the extent) A. Elements of $\mathcal{B}(X,Y,I)$ are called formal concepts of $\langle X,Y,I\rangle$; $\mathcal{B}(X,Y,I)$ is called the concept lattice given by $\langle X,Y,I\rangle$. Both the extent A and the intent B of a formal concept $\langle A,B\rangle$ are in general fuzzy sets. This corresponds to the fact that in general, concepts apply to objects and attributes to various intermediate degrees, not only 0 and 1.

Putting

$$\langle A_1,B_1\rangle \leq \langle A_1,B_1\rangle \text{ iff } A_1 \subseteq A_2 (\text{iff } B_1 \supseteq B_2) \tag{3}$$

for $\langle A_1, B_1\rangle, \langle A_2, B_2\rangle \in \mathcal{B}(X, Y, I)$, $\leq$ models the subconcept-superconcept hierarchy in $\mathcal{B}(X, Y, I)$. That is, being more general means to apply to a larger collection of objects and to cover a smaller collection of attributes. Characterization of $\mathcal{B}(X, Y, I)$ is presented in [13], see also [7, 27].

Given $\langle X, Y, I\rangle$, the pair $\langle^{\uparrow}, ^{\downarrow}\rangle$ induced by (1) and (2) satisfies the following natual properties [6]:

$$S(A_1, A_2) \leq S(A_2^{\uparrow}, A_1^{\uparrow}) \tag{4}$$

$$S(B_1, B_2) \leq S(B_2^{\downarrow}, B_1^{\downarrow}) \tag{5}$$

$$A \subseteq A^{\uparrow\downarrow} \tag{6}$$

$$B \subseteq B^{\downarrow\uparrow}, \tag{7}$$

for each $A, A_1, A_2 \in L^X$ and $B, B_1, B_2 \in L^Y$. A pair $\langle^{\uparrow}, ^{\downarrow}\rangle$ satisfying (4)–(7) is called a fuzzy Galois connection. It was proved in [6] that each fuzzy Galois connection is induced by some $\langle X, Y, I\rangle$ by (1) and (2). Note that fuzzy Galois connection obey several further properties which are often used, e.g. $A^{\uparrow} = A^{\uparrow\downarrow\uparrow}$ and $B^{\downarrow} = B^{\downarrow\uparrow\downarrow}$.

Crisply generated formal concepts

An important problem in FCA is a possible large number of formal concepts in $\mathcal{B}(X, Y, I)$. A way to cope with this problem in case of data with fuzzy attributes was proposed in [14]. The following are the basics. A formal concept $\langle A, B\rangle$ consists of a fuzzy set A and a fuzzy set B such that $A^{\uparrow} = B$ and $B^{\downarrow} = A$ which directly captures the verbal definition of a formal concept inspired by Port-Royal logic. However, this definition might actually allow for formal fuzzy concepts which seem not natural. For example, there may exist a formal fuzzy concept $\langle A, B\rangle$ such that for any $x \in X$ and $y \in Y$ we have $A(x) = 1/2$ and $B(y) = 1/2$. A verbal description of such a concept is "a concept to which each attribute belongs to degree 1/2". In general, "a concept to which each attribute belongs to degree 1/2" might be difficult to interpret. This is because people expect concepts to be determined by "some attributes", i.e. by an ordinary set of attributes. This leads to the following definition.

Definition 1. *A formal fuzzy concept $\langle A, B\rangle \in \mathcal{B}(X, Y, I)$ is called* crisply generated *if there is a crisp set $B_c \subseteq Y$ such that $A = B_c^{\downarrow}$ (and thus $B = B_c^{\downarrow\uparrow}$). We say that B_c crisply generates $\langle A, B\rangle$.*

By $\mathcal{B}_c(X, Y, I)$ we denote the collection of all crisply generated formal concepts in $\langle X, Y, I\rangle$, i.e.

$$\mathcal{B}_c(X, Y, I) = \{\langle A, B\rangle \in \mathcal{B}(X, Y, I) \mid \text{ there is } B_c \subseteq Y : A = B_c^{\downarrow}\}.$$

That is, $\mathcal{B}_c(X, Y, I) = \{\langle B_c^{\downarrow}, B_c^{\downarrow\uparrow}\rangle \mid B_c \subseteq Y\}$.

For further information on crisply generated fuzzy concepts, demonstration of the reduction of the number of formal concepts, and an algorithm for listing all crisply generated concepts we refer to [14].

To better understand the structure of crisply generated concepts and their further properties, the following question arises:

> Is there some "Galois-connection-like" structure behind crisply generated concepts which plays the role analogous to the role of Galois connections in FCA?

In the following, we elaborate the basic answer, some partial answers, and outline some open problems and directions.

3.2 Case One: Galois Connections Behind Crisply Generated Formal Concepts

Galois connections with truth stressers

First, we provide another view on crisply generated concepts which turns out to be suitable for our purposes. We will need the notion of a truth stresser. A truth stresser is a unary function * on the set L of truth degrees with the following interpretation: For a truth degree $a \in L$, the value $a^* \in L$ is the degree to which a can be considered as very true. Formally, a truth stresser on a structure $\mathbf{L}$ of truth degrees is a unary function which is required to satisfy some natural properties, e.g. $a^* \leq a$; $a \leq b$ implies $a^* \leq b^*$; $1^* = 1$; $a^* = a^{**}$. Functions with properties of truth stressers were used in [28]. In the context of fuzzy logic, truth stressers go back to [2] and were further elaborated in [21, 22]. For simplicity and because of our motivation by crisply generated concepts, we use only a particular type of a truth stresser in the present paper. Namely, we use what we call a Baaz operator [2] which is a function $^* : L \to L$ defined by

$$a^* = \begin{cases} 1 \text{ for } a = 1 \\ 0 \text{ for } a \neq 1. \end{cases} \tag{8}$$

Throughout the rest of the paper, * denotes the Baaz operator (8).

Consider now the mappings $^{\triangle} : L^X \to L^Y$ and $^{\triangledown} : L^Y \to L^X$ resulting from $\langle X, Y, I \rangle$ by

$$A^{\triangle}(y) = \bigwedge_{x \in X} (A(x) \to I(x, y)) \tag{9}$$

and

$$B^{\triangledown}(x) = \bigwedge_{y \in Y} (B(y)^* \to I(x, y)). \tag{10}$$

Remark 1. (1) Note that we have $A^{\triangle} = A^{\uparrow}$ and $B^{\triangledown} = (B^*)^{\downarrow}$ where $B^*(y) = (B(y))^*$, and $^{\uparrow}$ and $^{\downarrow}$ are defined by (1) and (2).

(2) With regard to the interpretation of a truth stresser *, $B^{\triangledown}(x)$ is the truth degree of "for each y: if it is very true that y belongs to B then x has (attribute) y". The particular meaning depends on the truth stresser *. For Baaz operator (8), this reads "for each y: if y fully belongs to B (i.e., belongs in degree 1) then x has y".

(3) Although we do not consider other truth stressers than (8) in this paper, let us note that another example of a truth stresser is the identity on L, i.e. $a^* = a$. For this choice we clearly have $A^{\triangle} = A^{\uparrow}$ and $B^{\triangledown} = B^{\downarrow}$. Therefore, fuzzy Galois connections result by a particular choice of a truth stresser.

The main points we are going to show in the rest of this section are, first, that crisply generated concepts are exactly fixed points of $\langle {}^{\triangle}, {}^{\triangledown}\rangle$, and, second, that $^{\triangle}$ and $^{\triangledown}$ can be defined axiomatically.

Crisply generated concepts as fixed points of $\langle {}^{\triangle}, {}^{\triangledown}\rangle$

For * defined by (8) denote by $\mathcal{B}(X, Y^*, I)$ the set of all fixed points of $\langle {}^{\triangle}, {}^{\triangledown}\rangle$, i.e.

$$\mathcal{B}(X, Y^*, I) = \{\langle A, B\rangle \in L^X \times L^Y \mid A^{\triangle} = B,\, B^{\triangledown} = A\}.$$

The following theorem shows that $\mathcal{B}(X, Y^*, I)$ are exactly the crisply generated concepts in $\langle X, Y, I\rangle$.

Theorem 1. *For a truth stresser* * *defined by (8),* $\mathcal{B}(X, Y^*, I) = \mathcal{B}_c(X, Y, I)$.

Proof. "$\subseteq$": If $\langle A, B\rangle \in \mathcal{B}(X, Y^*, I)$ then $A^{\triangle} = B$ and $B^{\triangledown} = A$, i.e. $A^{\uparrow} = B$ and $B^{*\downarrow} = A$. From (8) we get that B^* is crisp (i.e. $B^*(y)$ is 0 or 1 for each $y \in Y$). Therefore, $\langle A, B\rangle \in \mathcal{B}_c(X, Y, I)$, by definition.

"$\supseteq$": Let $\langle A, B\rangle \in \mathcal{B}_c(X, Y, I)$, i.e. $A^{\uparrow} = B$, $B^{\downarrow} = A$, and $A = D^{\downarrow}$ for some $D \subseteq Y$. We need to verify $A^{\triangle} = B$ and $B^{\triangledown} = A$, for which it clearly suffices to check $B^{\triangledown} = A$, i.e. $B^{*\downarrow} = A$. Since $A = D^{\downarrow}$ and $B = B^{\downarrow\uparrow}$, we need to check $D^{\downarrow} = D^{\downarrow\uparrow *\downarrow}$. Now observe that we have $D^{\downarrow} = D^{*\downarrow} = D^{*\downarrow\uparrow *\downarrow}$. Indeed, the first equality follows from the fact that D is crisp and thus $D^* = D$. For the second equality, $D^{*\downarrow} \subseteq D^{*\downarrow\uparrow *\downarrow}$ follows from $F \subseteq F^{\uparrow *\downarrow}$ for $F = D^{*\downarrow}$ (easy to verify), while $D^{*\downarrow} \supseteq D^{*\downarrow\uparrow *\downarrow}$ follows from $D = D^*$, from $D^* \subseteq D^{*\downarrow\uparrow}$, and from the fact that if $E \subseteq F$ then $E^{*\downarrow} \supseteq F^{*\downarrow}$ (just put $E = D$ and $F = D^{*\downarrow\uparrow}$). We showed $\langle A, B\rangle \in \mathcal{B}(X, Y^*, I)$, finishing the proof.

Galois connections with truth stressers: $^{\triangle}$ *and* $^{\triangledown}$ *defined axiomatically*

We now turn to the investigation of the properties of $^{\triangle}$ and $^{\triangledown}$ with the aim to provide a simple axiomatic characterization.

Lemma 1. *Let * be defined by (8). Then $^\triangle$ and $^\triangledown$ defined by (9) and (10) satisfy*

$$S(A, B^\triangledown) = S(B^*, A^\triangle) \tag{11}$$

$$(\bigcup_{j\in J} A_j)^\triangle = \bigcap_{j\in J} A_j^\triangle \tag{12}$$

for every $A, A_j \in L^X$ and $B \in L^Y$.

Proof. We have

$$\begin{aligned} S(A, B^\triangledown) &= \bigwedge_{x\in X} A(x) \to (\bigwedge_{y\in Y} B^*(y) \to I(x,y)) = \\ &= \bigwedge_{x\in X}\bigwedge_{y\in Y} A(x) \to (B^*(y) \to I(x,y)) = \\ &= \bigwedge_{y\in Y}\bigwedge_{x\in X} B^*(y) \to (A(x) \to I(x,y)) = \\ &= \bigwedge_{y\in Y} B^*(y) \to (\bigwedge_{x\in X} A(x) \to I(x,y)) = \\ &= S(B^*, A^\triangle), \end{aligned}$$

proving (11). As $^\triangle = {}^\uparrow$, (12) is a consequence of properties of fuzzy Galois connections [6].

Definition 2. *A pair $\langle {}^\triangle, {}^\triangledown\rangle$ of mappings satisfying (11) and (12) is called a fuzzy Galois connection with truth stresser *.*

The following are some consequences of (11).

Lemma 2. *If for * defined by (8) mappings $^\triangle : L^X \to L^Y$ and $^\triangledown : L^Y \to L^X$ satisfy (11) then*

$$(\bigcup_{j\in J} {B_j}^*)^\triangledown = \bigcap_{j\in J} B_j^\triangledown \tag{13}$$

$$B^\triangledown = B^{*\triangledown} \tag{14}$$

$$\{a/x\}^\triangle(y) = a \to \{1/x\}^\triangle(y) \tag{15}$$

$$\{a/y\}^\triangledown(x) = a \to \{1/y\}^\triangledown(x) \tag{16}$$

for any $B, B_j \in L^Y$, $x \in X$, $y \in Y$, $a \in L$.

Proof. We show (13) by showing that $S(A, (\bigcup_i {B^*}_i)^\triangledown) = 1$ iff $S(A, \bigcap_i {B^*}_i^\triangledown) = 1$ for each $A \in L^X$. First note that using (11) we have

$$S(A, (\bigcup_i {B^*}_i)^\triangledown) = S((\bigcup_i {B^*}_i)^*, A^\triangle) = S((\bigcup_i {B^*}_i), A^\triangle).$$

As a results, we have $S(A, (\bigcup_i B^*{}_i)^\triangledown) = 1$ iff $S((\bigcup_i B^*{}_i), A^\triangle) = 1$ iff for each i we have $B^*{}_i \subseteq A^\triangle$ iff for each i we have $S(B^*{}_i, A^\triangle)$ iff for each i we have $S(A, B_i^\triangledown)$ iff $S(A, \bigcap_i B_i^\triangledown)$, showing (13).

(14) follows from (13) for $|J| = 1$.

(15) and (16) follow from $b^* \to \{\, a/x\}^\triangle(y) = a \to \{\, b/y\}^\triangledown(x)$ and $\{\, 1/x\}^\triangle(y) = \{\, 1/y\}^\triangledown(x)$ which we now verify. First,

$$b^* \to \{\, a/x\}^\triangle(y) = S(\{\, b^*/y\}, \{\, a/x\}^\triangle) = S(\{\, b/y\}^*, \{\, a/x\}^\triangle) =$$
$$= S(\{\, a/x\}, \{\, b/y\}^\triangledown) = a \to \{\, b/y\}^\triangledown(x).$$

Second, $\{\, 1/x\}^\triangle(y) = \{\, 1/y\}^\triangledown(x)$ is a consequence of the first equality for $a = b = 1$.

Lemma 3. *Let * be defined by (8). Let $\langle {}^\triangle, {}^\triangledown \rangle$ be a fuzzy Galois connection with *. Then there is a fuzzy relation $I \in L^{X \times Y}$ such that $\langle {}^\triangle, {}^\triangledown \rangle = \langle {}^{\triangle_I}, {}^{\triangledown_I} \rangle$ where ${}^{\triangle_I}$ and ${}^{\triangledown_I}$ are induced by I by (9) and (10).*

Proof. Let I be defined by $I(x, y) = \{\, 1/x\}^\triangle(y) = \{\, 1/y\}^\triangledown(x)$. Then using (15)

$$A^\triangle(y) = (\bigcup_{x \in X} \{\, A(x)/x\})^\triangle(y) =$$
$$= (\bigcap_{x \in X} \{\, A(x)/x\}^\triangle)(y) = \bigwedge_{x \in X} \{\, A(x)/x\}^\triangle(y) =$$
$$= \bigwedge_{x \in X} A(x) \to \{\, 1/x\}^\triangle(y) = \bigwedge_{x \in X} A(x) \to I(x, y) = A^{\triangle_I}(y).$$

Furthermore, using (13) and (14), and (16) we get

$$B^\triangledown(x) = B^{*\triangledown}(x) = (\bigcup_{y \in Y} \{\, B^*(y)/y\})^\triangledown(x) =$$
$$= (\bigcup_{y \in Y} \{\, B(y)/y\}^*)^\triangledown(x) = (\bigcap_{y \in Y} \{\, B(y)/y\}^\triangledown)(x) = \bigwedge_{y \in Y} \{\, B(y)/y\}^\triangledown(x) =$$
$$= \bigwedge_{y \in Y} B^*(y) \to \{\, 1/y\}^\triangledown(x) = \bigwedge_{y \in Y} B^*(y) \to I(x, y) = B^{\triangledown_I}(x).$$

The following theorem shows that there is a one-to-one correspondence between fuzzy Galois connections with * defined by (8).

Theorem 2. *Let $I \in L^{X \times Y}$ be a fuzzy relation, let ${}^{\triangle_I}$ and ${}^{\triangledown_I}$ be defined by (9) and (10). Let $\langle {}^\triangle, {}^\triangledown \rangle$ be a fuzzy Galois connection with * defined by (8). Then*

(1) $\langle {}^{\triangle_I}, {}^{\triangledown_I} \rangle$ satisfy (11) and (12).
(2) $I_{\langle {}^\triangle, {}^\triangledown \rangle}$ defined as in the proof of Lemma 3 is a fuzzy relation and we have

(3) $\langle {}^{\triangle}, {}^{\triangledown}\rangle = \langle {}^{\triangle I_{\langle \triangle, \triangledown\rangle}}, {}^{\triangledown I_{\langle \triangle, \triangledown\rangle}}\rangle$ *and* $I = I_{\langle \triangle_I, \triangledown_I\rangle}$.

Proof. Due to the previous results, it remains to check $I = I_{\langle \triangle_I, \triangledown_I\rangle}$. We have

$$\begin{aligned} I_{\langle \triangle_I, \triangledown_I\rangle}(x, y) &= \{{}^1\!/_x\}^{\triangle_I}(y) = \\ &= \bigwedge_{z \in X} \{{}^1\!/_x\}(z) \to I(z, y) = I(x, y). \end{aligned}$$

Let us consider conditions (4)–(7). These are the defining conditions for fuzzy Galois connections. However, for Galois connections with truth stresser (8), (5) and (7) are not satisfied, as shown by the following example.

Example 1. Take $X = \{x\}, Y = \{y\}$, $I(x, y) = 0.3$, $B_1(y) = 1$, $B_2(y) = 0.8$, take $L = [0, 1]$ equipped with the Łukasiewicz structure, and * defined by (8). (5): We have $0.8 = S(B_1, B_2) \not\leq S(B_2^{\triangledown}, B_1^{\triangledown}) = 0.3$, a counterexample to (5). (7): We have $B(y) = 0.8 \not\leq 0.3 = B^{\triangledown\triangle}(y)$, i.e. $\langle {}^{\triangle}, {}^{\triangledown}\rangle$ does not satisfy (7).

The next lemma shows properties of Galois connections with * which are analogous to (4)–(7).

Lemma 4. *If a pair* $\langle {}^{\triangle}, {}^{\triangledown}\rangle$ *is a fuzzy Galois connection with* * *defined by (8) then*

$$S(A_1, A_2) \leq S(A_2^{\triangle}, A_1^{\triangle}) \tag{17}$$

$$S(B_1{}^*, B_2{}^*) \leq S(B_2^{\triangledown}, B_1^{\triangledown}) \tag{18}$$

$$A \subseteq A^{\triangle\triangledown} \tag{19}$$

$$B^* \subseteq B^{\triangledown\triangle} \tag{20}$$

Proof. The assertion follows from the properties of $\langle {}^{\uparrow}, {}^{\downarrow}\rangle$, Remark 1 (1), and from the fact that $A^{\uparrow} \subseteq A^{\triangle}$ (we omit details).

3.3 Case Two, Three, . . . , and Others

Placement of the truth stresser

The particular case of fuzzy Galois connection with a truth stresser * may be considered just one out of several further possibilities. Namely, the placement of the truth stresser *, inspired by [14] is not the only possible. The aim of this section is to outline some possibilities with some results. However, due to the limited scope of the paper, this section is to be considered only a sketch of a more detailed study which is under preparation and is to be published.

From the epistemic point of view, various placements of truth stressers in formulas which define $^{\triangle}$ and $^{\triangledown}$ lead to various interpretations of concepts. For instance, the concepts determined by the placement introduced in Section 3.2 can be interpreted as "crisply generated by attributes". In much the same

way we can define concepts (i) "crisply generated by objects", (ii) "crisply generated by object and atributes", etc. In general, it seems reasonable to consider $^{\triangle}$ and $^{\triangledown}$ defined by

$$A^{\triangle}(y) = \bigwedge_{x \in X} A^{*_1}(x) \to I^{*_2}(x, y),$$
$$B^{\triangledown}(x) = \bigwedge_{y \in Y} B^{*_3}(y) \to I^{*_4}(x, y),$$

where $^{*_1}, \ldots, ^{*_4}$ are appropriate truth stressers. Taking ^{*_3} to be the Baaz operator (8) and taking the identity for $^{*_1}, ^{*_2}, ^{*_4}$, we obtain Galois connections defined studied in Section 3.2.

A problem that offers itself is to take systems of all formal concepts (fixed points) determined by various placements and study their relationships. The problem is especially interesting for applications of FCA, because some placements can lead to smaller (and yet natural) sets of formal concepts—this might be understood as a purely logical way to reduce the size of the resulting conceptual structure.

Note first that taking other truth stresser than the identity for ^{*_2} and ^{*_4}, we lose the possibility to obtain I back from suitable axiomatic properties of $^{\triangle}$ and $^{\triangledown}$. Suppose for simplicity $^{*_2} = ^{*_4} = ^{*}$. Then the same $^{\triangle}$ and $^{\triangledown}$ are clearly induced by any fuzzy relation J for which $I^* = J^*$. Therefore, if one wants to keep the one-to-one relationship between I's and $\langle {}^{\triangle}, {}^{\triangledown} \rangle$, one has to restrict the attention to fuzzy relations for which $I^* \neq J^*$ for $I \neq J$ (a natural choice in case of idempotent * seems to be the set $\{I^* \mid I \in L^{X \times Y}\}$).

Crisply generated concepts, by attributes, objects, and both

In the following we present some preliminary results for the case when both ^{*_2} and ^{*_4} are identities, and ^{*_1} and ^{*_3} are either the identity or the Baaz operator (8). Given a fuzzy context $\langle X, Y, I \rangle$, we consider the following subsets of $L^X \times L^Y$:

$$\mathcal{B}(X, Y^*, I) = \{\langle A, B \rangle \mid A^{\uparrow} = B,\ B^{*\downarrow} = A\},$$
$$\mathcal{B}(X^*, Y, I) = \{\langle A, B \rangle \mid A^{*\uparrow} = B,\ B^{\downarrow} = A\},$$
$$\mathcal{B}(X^*, Y^*, I) = \{\langle A, B \rangle \mid A^{*\uparrow} = B,\ B^{*\downarrow} = A\}.$$

Clearly, the above definition of $\mathcal{B}(X, Y^*, I)$ coincides with that one presented in Section 3.2. On the verbal level, we can call $\mathcal{B}(X, Y^*, I)$, $\mathcal{B}(X^*, Y, I)$, and $\mathcal{B}(X^*, Y^*, I)$ collections of all formal concepts crisply generated by *attributes, objects,* and *attributes and objects,* respectively.

Theorem 3. *For a truth stresser * defined by (8), we have*

(i) *if* $\langle A, B \rangle \in \mathcal{B}(X, Y^*, I) \cap \mathcal{B}(X^*, Y, I)$ *then* $\langle A, B \rangle \in \mathcal{B}(X^*, Y^*, I)$;
(ii) *if* $\langle A, B \rangle \in \mathcal{B}(X^*, Y^*, I)$ *then*
$\langle A, A^{\uparrow} \rangle \in \mathcal{B}(X, Y^*, I)$ *and* $\langle B^{\downarrow}, B \rangle \in \mathcal{B}(X^*, Y, I)$;

(iii) $|\mathcal{B}(X^*, Y^*, I)| \leq \min(|\mathcal{B}(X, Y^*, I)|, |\mathcal{B}(X^*, Y, I)|)$.

Proof. (i) follows directly by definition.

(ii): Assuming $\langle A, B\rangle \in \mathcal{B}(X^*, Y^*, I)$, we have $A^{*\uparrow} = B$ and $B^{*\downarrow} = A$. We show that $\langle A, A^{\uparrow}\rangle \in \mathcal{B}(X, Y^*, I)$. It suffices to check $A^{\uparrow *\downarrow} = A$ since the rest follows easily. The monotony gives $A \subseteq A^{\uparrow\downarrow} \subseteq A^{\uparrow *\downarrow}$. Conversely, by the idempotency of * we have $A^{\uparrow *\downarrow} = B^{*\downarrow\uparrow *\downarrow} \subseteq B^{*\downarrow} = A$, showing that $\langle A, A^{\uparrow}\rangle \in \mathcal{B}(X, Y^*, I)$. Dually, one can prove $\langle B^{\downarrow}, B\rangle \in \mathcal{B}(X^*, Y, I)$.

(iii) is a consequence of (ii). □

Note that (iii) of Theorem 3 says that the collection of all concepts crisply generated by attributes and objects cannot be strictly greater than the collection of all concepts crisply generated by attributes (objects). If * were not idempotent this property would not hold in general.

Given a fuzzy context $\langle X, Y, I\rangle$ and * defined by (8) the formal concepts contained in $\mathcal{B}(X^*, Y^*, I)$ are in a correspondence with the classical formal concepts [18] of the crisp formal context $\langle X, Y, I^*\rangle$ (recall that crisp **L**-sets can be identitifed with the ordinary sets). Denoting the collection of classical formal concepts of $\langle X, Y, I^*\rangle$ by $\mathcal{B}(X, Y, I^*)$, we have

Theorem 4. *For a truth stresser * defined by (8), we have*

(i) *if* $\langle A, B\rangle \in \mathcal{B}(X^*, Y^*, I)$ *then* $\langle A^*, B^*\rangle \in \mathcal{B}(X, Y, I^*)$;
(ii) *if* $\langle C, D\rangle \in \mathcal{B}(X, Y, I^*)$ *then* $\langle C^{\uparrow *\downarrow}, D^{\downarrow *\uparrow}\rangle \in \mathcal{B}(X^*, Y^*, I)$.

Proof. First, observe that if $A \in L^X$ is crisp then $A^{\uparrow *}(y) = 1$ iff for any $x \in X$: if $A(x) = 1$ then $I^*(x, y) = 1$. Analogously for crisp $B \in L^Y$. Hence, we can identify the ordinary Galois connection induced by $\langle X, Y, I^*\rangle$ with $\langle^{\uparrow *}, ^{\downarrow *}\rangle$.

(i) Let $A = B^{*\downarrow}$ and $B = A^{*\uparrow}$, i.e. $A^* = B^{*\downarrow *}$ and $B^* = A^{*\uparrow *}$ which give $\langle A^*, B^*\rangle \in \mathcal{B}(X, Y, I^*)$.

(ii) For crisp **L**-sets $C \in L^X$ and $D \in L^Y$ with $\langle C, D\rangle \in \mathcal{B}(X, Y, I^*)$ we have $C = D^{\downarrow *}$ and $D = C^{\uparrow *}$. Thus, $C^{\uparrow *\downarrow *\uparrow} = D^{\downarrow *\uparrow}$ and $D^{\downarrow *\uparrow *\downarrow} = C^{\uparrow *\downarrow}$, proving the assertion. □

Theorem 4 yields that the 1-cuts of concepts contained in $\mathcal{B}(X^*, Y^*, I)$ are exactly the classical concepts of $\langle X, Y, I^*\rangle$ (I^* itself is an 1-cut of the original $I \in L^X \times L^Y$). But unlike the concepts in $\mathcal{B}(X, Y, I^*)$, the concepts in $\mathcal{B}(X^*, Y^*, I)$ can contain an additional "fuzzy information" which is lost when considering the crisp context $\langle X, Y, I^*\rangle$.

Automatic generation of statements

The exploration of possible placements of truth stressers bring up the following aspect: we often construct proofs in which we use inequalities of the form $A^{\cdots} \subseteq A^{\cdots}$ (and dually for B), where "..." stand for sequences of $^{\uparrow}$, $^{\downarrow}$, and *. Such inequalities themselves are assertions that should be proven. The proofs of these inequalities are usually purely combinatorial and tedious. On the other

hand a large "database" of such assertions can provide us with an essential insight into the modified Galois connections. For this purpose, we designed a computer program to find the proofs automatically.

For illustration, we present a segment of the assertions, found by the program, which is limited only to inequalities $B^{\cdots} \subseteq B^{\cdots}$ whose left and right side of "$\subseteq$" contain at most four symbols and we also skip trivial formulas like $B^{\downarrow} \subseteq B^{\downarrow}$, $B^{\downarrow\uparrow\downarrow} \subseteq B^{\downarrow}$, etc. Restricted by this limitation, the program generated 210 formulas from which 37 were proven (a table follows); for the remaining 173 ones the engine found a counterexample.

$B \subseteq B^{\downarrow*\uparrow}$	$B^{\downarrow} \subseteq B^{\downarrow\uparrow*\downarrow}$	$B^{*\downarrow\uparrow} \subseteq B^{\downarrow*\uparrow}$	$B^{*\downarrow\uparrow*} \subseteq B^{*\downarrow*\uparrow}$
$B \subseteq B^{\downarrow\uparrow}$	$B^{\downarrow*} \subseteq B^{\downarrow}$	$B^{*\downarrow\uparrow*} \subseteq B^{\downarrow\uparrow}$	$B^{*\downarrow\uparrow*} \subseteq B^{\downarrow*\uparrow*}$
$B^{*} \subseteq B$	$B^{*\downarrow*} \subseteq B^{*\downarrow}$	$B^{\downarrow\uparrow*} \subseteq B^{\downarrow*\uparrow*}$	$B^{\downarrow*\uparrow*} \subseteq B^{\downarrow*\uparrow}$
$B^{*} \subseteq B^{\downarrow\uparrow}$	$B^{\downarrow*} \subseteq B^{*\downarrow*}$	$B^{\downarrow\uparrow*} \subseteq B^{\downarrow\uparrow}$	$B^{\downarrow*\uparrow\downarrow} \subseteq B^{*\downarrow}$
$B^{*} \subseteq B^{*\downarrow\uparrow}$	$B^{*} \subseteq B^{\downarrow*\uparrow}$	$B^{*\downarrow\uparrow*} \subseteq B^{*\downarrow\uparrow}$	$B^{\downarrow*\uparrow\downarrow} \subseteq B^{\downarrow\uparrow*\downarrow}$
$B^{*} \subseteq B^{\downarrow\uparrow*}$	$B^{\downarrow*} \subseteq B^{*\downarrow}$	$B^{*\downarrow*\uparrow} \subseteq B^{\downarrow*\uparrow}$	$B^{\downarrow*\uparrow\downarrow} \subseteq B^{\downarrow}$
$B^{*} \subseteq B^{*\downarrow\uparrow*}$	$B^{\downarrow*} \subseteq B^{\downarrow*\uparrow\downarrow}$	$B^{*\downarrow\uparrow*} \subseteq B^{\downarrow\uparrow*}$	$B^{\downarrow\uparrow*\downarrow} \subseteq B^{*\downarrow}$
$B^{\downarrow} \subseteq B^{*\downarrow}$	$B^{\downarrow*} \subseteq B^{\downarrow\uparrow*\downarrow}$	$B^{*\downarrow\uparrow} \subseteq B^{*\downarrow*\uparrow}$	
$B^{*} \subseteq B^{*\downarrow*\uparrow}$	$B^{*\downarrow\uparrow} \subseteq B^{\downarrow\uparrow}$	$B^{\downarrow\uparrow*} \subseteq B^{\downarrow*\uparrow}$	
$B^{*} \subseteq B^{\downarrow*\uparrow*}$	$B^{\downarrow\uparrow} \subseteq B^{\downarrow*\uparrow}$	$B^{*\downarrow\uparrow*} \subseteq B^{\downarrow*\uparrow}$	

A large database of 375 assertions (with the proofs attached) can be found at `http://vychodil.inf.upol.cz/res/devel/aureas`. The general inference engine is still under construction [17] and will be available soon at the same Internet address.

4 Epilogue: Applications and Further Development

We presented motivations and introductory results on fuzzy Galois connections with truth stressers. We showed that for Baaz truth stresser, Galois connections with truth stressers are exactly the "Galois-like-connections" behind the so-called crisply generated formal concepts obtained from object-attribute data with fuzzy attributes. Let us now demonstrate the reduction of the number of extracted concepts from the object-attribute data $\langle X, Y, I\rangle$—the main effect of considering $\mathcal{B}(X, Y^*, I)$ (crisply generated concepts) instead of $\mathcal{B}(X, Y, I)$ (all formal concepts).

The following experiment is taken from [14]. The experiment demonstrates the factor of reduction, i.e. the ratio $r = |\mathcal{B}(X, Y^*, I)|/|\mathcal{B}(X, Y, I)|$ (the smaller, the larger the reduction). Tab. 1 shows the values of r for 10 experiments (columns) run over randomly generated formal contexts $\langle X, Y, I\rangle$ (rows) with the number of objects equal to the number of attributes (from 5 to 25 objects/attributes) and with $|L| = 11$ (11 truth degrees). Moreover, we show average and dispersion of r. We can see that the dispersion is relatively low and that r decreases with growing size of data.

The future research is needed in the following directions:

	1	2	3	4	5	6	7	8	9	10	Av	Var
5	0,58	0,4	0,38	0,53	0,48	0,38	0,43	0,41	0,48	0,33	0,441	0,0733
6	0,31	0,31	0,38	0,43	0,38	0,32	0,43	0,42	0,36	0,38	0,372	0,0443
7	0,46	0,37	0,31	0,48	0,45	0,27	0,41	0,43	0,4	0,37	0,395	0,0635
8	0,35	0,26	0,32	0,33	0,29	0,3	0,27	0,34	0,3	0,31	0,308	0,0270
...	...	...	...	...	...	...	...	...	...	...	...	...
22	0,1	0,09	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,097	0,0038
23	0,09	0,11	0,1	0,1	0,1	0,1	0,1	0,09	0,1	0,11	0,099	0,0066
24	0,1	0,09	0,08	0,1	0,09	0,1	0,09	0,09	0,09	0,08	0,090	0,0079
25	0,08	0,07	0,09	0,08	0,09	0,09	0,08	0,07	0,09	0,08	0,081	0,0074

Table 1. Behavior of r (its average Av and dispersion Var) in dependence on size of input data table (rows; the numbers $5 \dots 25$ denote the number of objects, this is equal to the number of attributes); columns correspond to experiments.

- more detailed investigation of fuzzy Galois connection with truth stressers with other placement of * than in (9) and (10);
- extension of the results to other truth stressers than (8);
- study of the systems of fixed points of fuzzy Galois connections with truth stressers (i.e. restricted concept lattices) with various placements of *, and their relationship (e.g. show the analogy/generalization of Wille's main theorem of concept lattices [29], see also [13] for fuzzy setting);
- algorithms for generating fixed points, i.e. formal concepts, of Galois connections with truth stressers.

As already demonstrated, better understanding of the presented issues can enhance the applicablity of formal concept analysis of data with fuzzy attributes.

5 Acknowledgement

Supported by grant no. B1137301 of GA AV ČR.

References

1. Arnauld A., Nicole P.: *La logique ou l'art de penser.* 1662. Also in German: *Die Logik oder die Kunst des Denkens.* Darmstadt, 1972.
2. Baaz M.: Infinite-valued Gödel logics with 0-1 projections and relativizations. *GÖDEL' 96 – Logical Foundations of Mathematics, Computer Sciences and Physics,* Lecture Notes in Logic vol. 6, Springer-Verlag 1996, 23–33.
3. Barbut M.: Note sur l'algèbre des techniques d'analyse hiérarchique. Appendice de l'Analyse hiérarchique, M. Matalon, Paris, Gauthier-Villars, 1965, 125–146.
4. Barbut M., Monjardet B.: *L'ordre et la classification, algèbre et combinatoire, tome II.* Paris, Hachette, 1970.

5. Bělohlávek R.: Fuzzy concepts and conceptual structures: induced similarities. In *Proc. Joint Conf. Inf. Sci.'98*, Vol. I, pages 179–182, Durham, NC, 1998.
6. Bělohlávek R.: Fuzzy Galois connections. *Math. Logic Quarterly* **45**,4 (1999), 497–504.
7. Bělohlávek R.: Lattices of fixed points of fuzzy Galois connections. *Math. Logic Quarterly* **47**,1 (2001), 111–116.
8. Bělohlávek R.: Similarity relations in concept lattices. *J. Logic and Computation* Vol. **10** No. **6**(2000), 823–845.
9. Bělohlávek R.: Fuzzy closure operators. *J. Math. Anal. Appl.* **262**(2001), 473-489.
10. Bělohlávek R.: Fuzzy closure operators II. *Soft Computing* **7**(2002) 1, 53-64.
11. Bělohlávek R.: *Fuzzy Relational Systems: Foundations and Principles.* Kluwer, Academic/Plenum Publishers, New York, 2002.
12. Bělohlávek R.: Fuzzy closure operators induced by similarity. *Fundamenta Informaticae* **58**(2)(2003), 79–91.
13. Bělohlávek R.: Concept lattices and order in fuzzy logic. *Ann. Pure Appl. Logic* (to appear, 22 pp.).
14. Bělohlávek R., Sklenář V., Zacpal J.: Crisply generated fuzzy concepts: reducing the number of concepts in formal concept analysis (submitted).
15. Birkhoff G.: *Lattice Theory, 3-rd edition.* AMS Coll. Publ. 25, Providence, R.I., 1967.
16. Burusco A., Fuentes-Gonzales R.: The study of L-fuzzy concept lattice. *Mathware & Soft Computing* **3**(1994), 209–218.
17. Dostál M., Vychodil V.: Evolutionary approach to automated deduction (in preparation).
18. Ganter B., Wille R.: *Formal concept analysis. Mathematical Foundations.* Springer-Verlag, Berlin, 1999.
19. Goguen J. A.: L-fuzzy sets. *J. Math. Anal. Appl.* **18**(1967), 145–174.
20. Gottwald S.: *A Treatise on Many-Valued Logics.* Research Studies Press, Baldock, Hertfordshire, England, 2001.
21. Hájek P.: *Metamathematics of Fuzzy Logic.* Kluwer, Dordrecht, 1998.
22. Hájek P.: On very true. *Fuzzy sets and systems* **124**(2001), 329–333.
23. Höhle U.: On the fundamentals of fuzzy set theory. *J. Math. Anal. Appl.* **201**(1996), 786–826.
24. Klement E. P., Mesiar R., Pap E.: *Triangular Norms.* Kluwer, Dordrecht, 2000.
25. Novák, V., Perfileva I., Močkoř J.: *Mathematical Principles of Fuzzy Logic.* Kluwer, Dordrecht, 1999.
26. Ore O.: Galois connections. *Trans. AMS* **55**(1944), 493–513.
27. Pollandt S.: *Fuzzy Begriffe.* Springer-Verlag, Berlin/Heidelberg, 1997.
28. Takeuti G., Titani S.: Globalization of intuitionistic set theory. *Annal of Pure and Applied Logic* **33**(1987), 195–211.
29. Wille R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival I.: *Ordered Sets.* Reidel, Dordrecht, Boston, 1982, 445—470.
30. Zadeh L. A.: Fuzzy sets. *Inf. Control* **8**(3)(1965), 338–353.

Fuzzy Transforms in Removing Noise*

Irina Perfilieva and Radek Valášek

University of Ostrava
Institute for Research and Applications of Fuzzy Modeling
30. dubna 22, 701 03 Ostrava 1, Czech Republic

{Irina.Perfilieva,radek.valasek}@osu.cz

1 Introduction

The technique of fuzzy transform (F-transform for short) has been introduced in [6, 5]. It consists of two phases: direct and inverse. We have proved that the inverse F-transform has good approximation properties and is very simple to use.

A special interest in this paper is paid to the problem of removing noise which relates to nonlinear signal processing methods. The latter have developed rapidly in recent decades to address various problems characterized by heavy tailed distribution and/or nonstationary statistics. The utilization of nonlinear methods has also been advanced by the increasing complexity of multimedia and communication applications, as well as the still increasing computational power of computers and signal processing hardware. Among nonlinear methods, filters have been widely used due to their robustness to outliers and detail preservation characteristics [1, 2]. In fact, the inverse F-transform can be considered as a special fuzzy identity filter (cf. [3]) which can be utilized, e.g., in image processing to preserve true image edges. In this contribution we characterize an additive noise which can be removed by applying the inverse F-transform to the original function.

2 Fuzzy Partition Of The Universe

We take an interval $[a, b]$ of real numbers as a universe. That is, all (real-valued) functions considered in this chapter have this interval as a common domain. Let us introduce fuzzy sets (given by their membership functions) which are subsets of the universe $[a, b]$ and which form a *fuzzy partition* of the universe.

* This paper has been partially supported by grant IAA1187301 of the GA AV ČR.

I. Perfilieva and R. Valášek: *Fuzzy Transforms in Removing Noise* , Advances in Soft Computing **2**, 221–230 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

Definition 1. *Let $x_1 < \ldots < x_n$ be fixed nodes within $[a, b]$, such that $x_1 = a$, $x_n = b$ and $n \geq 2$. We say that fuzzy sets $A_1, \ldots, A_n$ identified with their membership functions $A_1(x), \ldots, A_n(x)$ defined on $[a, b]$, form a* fuzzy partition *of $[a, b]$ if they fulfil the following conditions for $k = 1, \ldots, n$:*

1. $A_k : [a, b] \longrightarrow [0, 1]$, $A_k(x_k) = 1$,
2. $A_k(x) = 0$ *if* $x \notin (x_{k-1}, x_{k+1})$,
3. $A_k(x)$ *is continuous,*
4. $A_k(x)$ *monotonically increases on* $[x_{k-1}, x_k]$ *and monotonically decreases on* $[x_k, x_{k+1}]$,
5. $\sum_{k=1}^{n} A_k(x) = 1$, *for all* $x \in [a, b]$,

where for the uniformity of denotation, we put $x_0 = a$ and $x_{n+1} = b$. The membership functions $A_1(x), \ldots, A_n(x)$ are called basic functions.

The following formulas give the formal representation of triangular membership functions:

$$A_1(x) = \begin{cases} 1 - \frac{(x - x_1)}{h_1}, & x \in [x_1, x_2], \\ 0, & \text{otherwise}, \end{cases}$$

$$A_k(x) = \begin{cases} \frac{(x - x_{k-1})}{h_{k-1}}, & x \in [x_{k-1}, x_k], \\ 1 - \frac{(x - x_k)}{h_k}, & x \in [x_k, x_{k+1}], \\ 0, & \text{otherwise}, \end{cases}$$

$$A_n(x) = \begin{cases} \frac{(x - x_{n-1})}{h_{n-1}}, & x \in [x_{n-1}, x_n], \\ 0, & \text{otherwise}. \end{cases}$$

where $k = 2, \ldots n - 1$, and $h_k = x_{k+1} - x_k$.

We say that a fuzzy partition $A_1(x), \ldots, A_n(x)$, $n > 2$, is *uniform* if the nodes $x_1, \ldots, x_n$ are equidistant, i.e. $x_k = a + h(k - 1)$, $k = 1, \ldots, n$, where $h = (b - a)/(n - 1)$, and two more properties are fulfilled for $k = 2, \ldots, n - 1$:

6. $A_k(x_k - x) = A_k(x_k + x)$, for all $x \in [0, h]$,
7. $A_k(x) = A_{k-1}(x - h)$, for all $x \in [x_k, x_{k+1}]$ and $A_{k+1}(x) = A_k(x - h)$, for all $x \in [x_k, x_{k+1}]$

In the case of a uniform partition, h is a length of the support of A_1 or A_n while $2h$ is the length of the support of the other basic functions A_k, $k = 2, \ldots, n - 1$. An example of uniform partition by sinusoidal shaped basic functions is given below:

$$A_1(x) = \begin{cases} 0.5(\cos\frac{\pi}{h}(x - x_1) + 1), & x \in [x_1, x_2], \\ 0, & \text{otherwise}, \end{cases}$$

$$A_k(x) = \begin{cases} 0.5(\cos\frac{\pi}{h}(x - x_k) + 1), & x \in [x_{k-1}, x_{k+1}], \\ 0, & \text{otherwise}, \end{cases}$$

where $k = 2, \ldots n-1$, and

$$A_n(x) = \begin{cases} 0.5(\cos\frac{\pi}{h}(x - x_n) + 1), & x \in [x_{n-1}, x_n], \\ 0, & \text{otherwise}. \end{cases}$$

The following lemma shows that in the case of a uniform partition, the definite integral of a basic function does not depend on its concrete shape. This property will be further used to simplify the direct F-transform.

Lemma 1. *Let the uniform partition of* $[a, b]$ *be given by basic functions* $A_1(x), \ldots, A_n(x)$. *Then*

$$\int_{x_1}^{x_2} A_1(x)dx = \int_{x_{n-1}}^{x_n} A_n(x)dx = \frac{h}{2}, \tag{1}$$

and for $k = 2, \ldots, n-1$

$$\int_{x_{k-1}}^{x_{k+1}} A_k(x)dx = h \tag{2}$$

where h *is the length of the support of* A_1.

3 F-transform

In this section, we introduce the technique of two F-transforms: direct and inverse (see also [6, 5]). The direct F-transform takes the original function (which should be at least integrable) and converts it into n-dimensional vector. The inverse F-transform converts the n-dimensional vector into specially represented function which approximates the original one. The advantage of the direct F-transform is that it produces a simple and unique representation of an original function which enables us to use the former instead of the latter in complex computations. After finishing the computations, the result can be brought back to the space of ordinary functions by the inverse F-transform. To be sure that this can be done we need to prove a number of theorems.

3.1 Direct F-transform

The following two definitions introduce the direct F-transform (or fuzzy transform) of a given function.

Definition 2. *Let $f(x)$ be any continuous (real-valued) function on $[a,b]$ and $A_1(x),\ldots,A_n(x)$ be basic functions which form a fuzzy partition of $[a,b]$. We say that the n-tuple of real numbers $[F_1,\ldots,F_n]$ is the direct (integral) F-transform of f w.r.t. $A_1,\ldots,A_n$ if*

$$F_k = \frac{\int_a^b f(x)A_k(x)dx}{\int_a^b A_k(x)dx} \tag{3}$$

Suppose that the basic functions $A_1,\ldots,A_n$ are fixed. Denote the direct F-transform of f with respect to $A_1,\ldots,A_n$ by $\mathbf{F}_n[f]$. Then according to Definition 2, we can write

$$\mathbf{F}_n[f] = [F_1,\ldots,F_n]. \tag{4}$$

The elements $F_1,\ldots,F_n$ are called *components of the F-transform*. Moreover, $F_2,\ldots,F_{n-1}$ are *regular* components and F_1,F_n are *singular* components.

If the partition of $[a,b]$ by $A_1,\ldots,A_n$ is uniform then the expression (2) for components of the direct F-transform may be simplified on the basis of Lemma 1:

$$F_1 = \frac{2}{h}\int_{x_1}^{x_2} f(x)A_1(x)dx, \tag{5}$$

$$F_n = \frac{2}{h}\int_{x_{n-1}}^{x_n} f(x)A_n(x)dx, \tag{6}$$

$$F_k = \frac{1}{h}\int_{x_{k-1}}^{x_{k+1}} f(x)A_k(x)dx, \quad k=2,\ldots,n-1. \tag{7}$$

It is easy to see that if a fuzzy partition (and therefore, basic functions) is fixed then the direct F-transform being a mapping from $C[a,b]$ (the set of all continuous functions on $[a,b]$) to $\mathbb{R}^n$ is linear, so that

$$\mathbf{F}_n[\alpha f + \beta g] = \alpha\,\mathbf{F}_n[f] + \beta\,\mathbf{F}_n[g] \tag{8}$$

for $\alpha,\beta\in\mathbb{R}$ and functions $f,g\in C[a,b]$.

The following theorem shows that components of the direct F-transform are the *weighted mean values* of the given function where the weights are given by the basic functions.

Theorem 1. *Let $f(x)$ be a continuous function on $[a,b]$ and $A_1(x),\ldots,A_n(x)$ be basic functions which form a fuzzy partition of $[a,b]$. Then the k-th component of the direct F-transform gives minimum to the following function:*

$$\Phi(a) = \int_a^b (f(x)-a)^2 A_k(x)dx \tag{9}$$

3.2 Inverse F-transform

A reasonable question is the following: can we reconstruct the function by means of its F-transform? The answer is clear: in general not precisely, because we are losing information when changing to the direct F-transform. However, the function which can be reconstructed (by the inverse F-transform) approximates the original one in such a way that a universal convergence can be established. Moreover, the inverse F-transform fulfils the best approximation criterion which can be called the piecewise integral least square criterion.

Definition 3. *Let* $\mathbf{F}_n[f] = [F_1, \ldots, F_n]$ *be the direct integral F-transform of a function* $f(x)$ *with respect to* $A_1, \ldots, A_n$. *Then the function*

$$f_{F,n}(x) = \sum_{k=1}^{n} F_k A_k(x) \tag{10}$$

is called the inversion formula or the inverse integral F-transform.

The theorem below [6, 5] shows that the inverse integral F-transform $f_{F,n}$ may approximate the original continuous function f with an arbitrary precision.

Theorem 2. *Let* $f(x)$ *be any continuous function on* $[a, b]$. *Then for any* $\varepsilon > 0$ *there exist* n_ε *and a fuzzy partition* $A_1, \ldots, A_{n_\varepsilon}$ *of* $[a, b]$ *such that for all* $x \in [a, b]$

$$|f(x) - f_{F,n_\varepsilon}(x)| < \varepsilon \tag{11}$$

where f_{F,n_ε} *is the inverse integral F-transform of* f *with respect to the fuzzy partition* $A_1, \ldots, A_{n_\varepsilon}$.

4 Removing Noise

In Subsection 3.1, we have claimed that components of the direct F-transform are the weighted mean values of the given function. As a consequence of this fact, we suppose that applying both F-transforms (direct and inverse) to a function, we may remove a certain noise. By this we mean that the inverse F-transform of a noised function is "almost equal" to the inverse F-transform of the original function. In this section we investigate which noise can be removed.

We will consider a noise, represented by a function $s(x)$ and such that $f(x) + s(x)$ is the representation of the noised function f. We will refer to this type of a noise as *additive* noise. On the basis of linearity of the direct F-transform, this noise can be removed if its regular components of the direct F-transform are equal to zero. This simple fact is proved in the following lemma.

Lemma 2. *Let $f(x)$ and $s(x)$ be continuous functions on $[a,b]$ and $A_1(x),\ldots,A_n(x)$, $n>2$, be basic functions which form a fuzzy partition of $[a,b]$. Assume that the regular components of the direct F-transform of s w.r.t. $A_1,\ldots,A_n$ are equal to zero. Then*

$$f_{F,n}(x) = (f+s)_{F,n}(x), \quad x \in [x_2, x_{n-1}]$$

where are the inverse F-transforms of functions f and $f+s$ respectively.

Proof. Let us denote $[S_1,\ldots,S_n]$ the direct integral F-transform of s with respect to $A_1,\ldots,A_n$. By the assumption, $S_2 = \cdots = S_{n-1} = 0$. Then for arbitrary $x \in [x_2, x_{n-1}]$

$$(f+s)_{F,n}(x) = \sum_{k=1}^{n}(F_k+S_k)A_k(x) = \sum_{k=2}^{n-1}(F_k+S_k)A_k(x) = \\ = \sum_{k=2}^{n-1} F_k A_k(x) = f_{F,n}(x).$$

Let us call an additive noise *removable* if it fulfils the assumption of Lemma 2. It is easy to see that a linear combination of removable noises is also a removable noise. This statement is formulated in the lemma given below.

Lemma 3. *Let $A_1,\ldots,A_n$ be a fuzzy partition of interval $[a,b]$ such that $h=(b-a)/(n-1)$ and $n>2$. Let functions $s_1(x), s_2(x)$, defined on $[a,b]$, represent a removable noise, i.e.*

$$\int_{x_{k-1}}^{x_{k+1}} A_k(x)s_i(x)\mathrm{dx} = 0, \qquad \text{for } k=2,\ldots,n-1.$$

where $i=1,2$. Then for arbitrary reals α,β, the function $\alpha s_1(x)+\beta s_2(x)$ represents a removable noise too, i.e.

$$\int_{x_{k-1}}^{x_{k+1}} A_k(x)(\alpha s_1(x)+\beta s_2(x))\mathrm{dx} = 0, \qquad \text{for } k=2,\ldots,n-1$$

Below, we describe properties of an additive noise which guarantee that the noise is removable. Then examples of the removable noise will be given and illustrated in pictures.

Theorem 3. *Let $A_1,\ldots,A_n$ be a uniform fuzzy partition of interval $[a,b]$ such that $h=(b-a)/(n-1)$ and $n>2$. Let moreover, $s(x)$ be a continuous periodical function with the period $2h$ and such that*

$$s(x_k - x) = -s(x_k + x) \qquad \text{on interval } [x_{k-1}, x_{k+1}]$$

where $k = 2, \ldots, n-1$ and $x \leq h$. Then the regular components of the direct F-transform of s w.r.t. $A_1, \ldots, A_n$ are equal to zero, i.e.

$$\int_{x_{k-1}}^{x_{k+1}} A_k(x)s(x)\mathrm{dx} = 0, \qquad \text{for } k = 2, \ldots, n-1.$$

Proof. Let us fix some k, $2 \leq k \leq n-1$. Then

$$\int_{x_{k-1}}^{x_{k+1}} A_k(x)s(x)\mathrm{dx} = \int_{x_{k-1}}^{x_k} A_k(x)s(x)\mathrm{dx} + \int_{x_k}^{x_{k+1}} \underbrace{A_k(x)s(x)}_{x=x_k+y}\mathrm{dx} =$$

$$\int_{x_{k-1}}^{x_k} A_k(x)(x)\mathrm{dx} - \int_0^h \underbrace{A_k(x_k-y)s(x_k-y)}_{A_k(x_k+y)s(x_k+y)=-A_k(x_k-y)s(x_k-y)}\mathrm{dy} =$$

$$\int_{x_{k-1}}^{x_k} A_k(x)s(x)\mathrm{dx} + \int_{x_k}^{x_{k-1}} \underbrace{A_k(z)s(z)}_{z=x_k-y}\mathrm{dz} =$$

$$\int_{x_{k-1}}^{x_k} A_k(x)s(x)\mathrm{dx} - \int_{x_{k-1}}^{x_k} A_k(z)s(z)\mathrm{dz} = 0.$$

Theorem 4. *Let $A_1, \ldots, A_n$ be a uniform fuzzy partition of interval $[a, b]$ such that $h = (b-a)/(n-1)$ and $n > 2$. Let us consider a function $s(x)$ defined on $[a, b]$ and such that*

$$s(x+h) = s(x), \qquad x \in [x_1, x_{n-2}]$$

and

$$\int_{x_1}^{x_2} s(x)\mathrm{dx} = 0$$

Then the regular components of the direct F-transform of s w.r.t. $A_1, \ldots, A_n$ are equal to zero, i.e.

$$\int_{x_{k-1}}^{x_{k+1}} A_k(x)s(x)\mathrm{dx} = 0, \qquad k = 2, \ldots, n-1.$$

Proof. Let us fix some k, $2 \leq k \leq n-1$. Then

$$\int_{x_{k-1}}^{x_{k+1}} A_k(x)s(x)\mathrm{dx} = \int_{x_{k-1}}^{x_k} A_k(x)s(x)\mathrm{dx} + \int_{x_k}^{x_{k+1}} \underbrace{A_k(x)s(x)}_{x=y+h}\mathrm{dx} =$$

$$\int_{x_{k-1}}^{x_k} A_k(x)(x)\mathrm{dx} + \int_{x_{k-1}}^{x_k} \underbrace{A_k(y+h)s(y+h)}_{A_k(y+h)=A_{k-1}(y)=1-A_k(y)} \mathrm{dy} =$$

$$\int_{x_{k-1}}^{x_k} A_k(x)s(x)\mathrm{dx} + \int_{x_{k-1}}^{x_k} s(y)\mathrm{dy} - \int_{x_{k-1}}^{x_k} A_k(x)s(x)\mathrm{dx} = 0$$

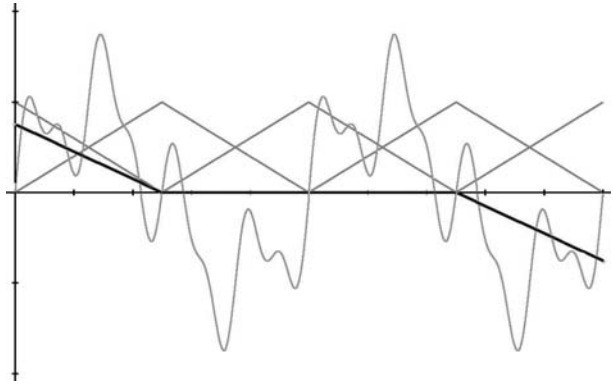


Fig. 1. Fuzzy transform of the removable noise $s(x) = \sin(2x) + 0.6\sin(8x) + 0.3\sin(16x)$ with zero regular components. Bold line is the inverse F-transform of s.

Remark 1. Let us remark that the properties, guaranteeing that the noise is removable (see Theorems 3, 4), are fully determined by a uniform fuzzy partition of interval $[a, b]$. This means that if we are given a partition then we can characterize the removable noise. Vice versa, if we are given a noised function then we can find the partition with respect to which the noise is removable.

5 Conclusion

We have introduced the technique of direct and inverse F-transforms and investigated approximating properties of the inverse F-transform. A special interest has been paid to the problem of removing noise which relates to nonlinear signal processing methods. We showed that the inverse F-transform can be considered as a special fuzzy identity filter which can be utilized, e.g.,

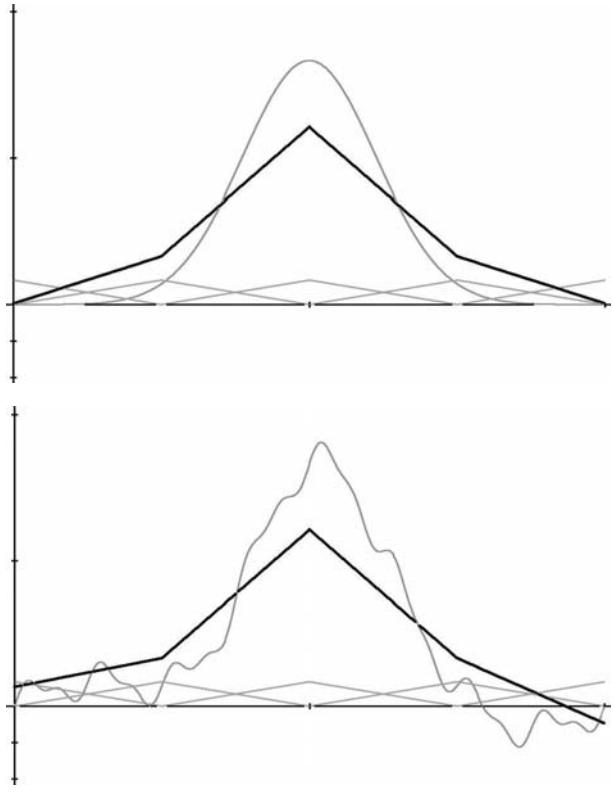


Fig. 2. The upper figure shows the original function $f(x) = 10 * e^{-(x-\pi)^2}$ and its inverse F-transform (bold line). The lower figure shows the noised function $f(x) + s(x)$ and its inverse F-transform (bold line). The noise $s(x)$ is removable and is illustrated on Fig. 1. Both inverse F-transforms of $f(x)$ and $f(x) + s(x)$ are equal up to the first and the last subintervals.

in image processing to preserve true image edges. In the present contribution we characterized an additive noise which can be removed by applying the inverse F-transform to an original function.

References

1. Justusson B J (1981) Median Filtering: Statistical Properties. In: Huang T S (Ed) Two Dimensional Digital Signal Processing II. Berlin, Springer-Verlag
2. Yao Nie and Lee S J (1991) Center Weighted Median Filter and Their Applications to Image Enhancement. IEEE Trans. Circuits Syst. 15: 984-993
3. Ko S J and Barner K E (2004) The Fuzzy Transformation and Its Applications in Image Processing. IEEE Tran. on Image Processing, to appear

4. Novák V, Perfilieva I and Močkoř J (1999) Mathematical Principles of Fuzzy Logic. Kluwer, Boston/Dordrecht
5. Perfilieva I (2001b) Fuzzy Transform: Application to Reef Growth Problem. In: Demicco R V, Klir G J (eds) Fuzzy Logic in Geology, 275–300
6. Perfilieva I and Chaldeeva E (2001) Fuzzy Transformation. Proceedings of IFSA'2001 World Congress, Vancouver, Canada
7. Štepnička M and Valášek R (2003) Fuzzy Transforms for Functions with Two Variables. In: Ramík J, Novák V (eds) Methods for Decision Support in Environment with Uncertainty – Applications in Economics, Business and Engineering, University of Ostrava, IRAFM, 96–102

Safe Modelling of Fuzzy If–Then Rules*

Irina Perfilieva[1] and Stephan Lehmke[2]

[1] University of Ostrava
Institute for Research and Applications of Fuzzy Modeling
30. dubna 22, 701 03 Ostrava 1, Czech Republic
Irina.Perfilieva@osu.cz

[2] University of Dortmund, D-44221 Dortmund, Germany
lehmke@unido.edu

1 Introduction

Nowadays, it is not necessary to advocate in favor of systems of fuzzy IF–THEN rules, because they are widely used in applications of fuzzy set theory such that fuzzy control, identification of dynamic systems, prediction of dynamic systems, decision-making, etc. The reason is in the fact that these systems can be effectively used as an instrument for representation of continuous dependencies. Therefore, the continuity property of a model of fuzzy IF–THEN rules is expected.

In this contribution, we suppose that a system of fuzzy IF–THEN rules is modelled by a fuzzy relation. Each concrete model (relation) is considered as an appropriate formal representation of a dependence which is given partially by a finite set of input-output pairs (joined by IF ... THEN). In order to use this formal representation in further computations with other than given inputs, the compositional rule of inference is applied.

Informally speaking, the continuity of a model of fuzzy IF–THEN rules means that whenever an input fuzzy set is close to a given one (presented on the left-hand side of the rules), the computed output of the model is close to the respective fuzzy set presented on the right-hand side of the rules.

In many research papers dealing with modelling of systems of fuzzy IF–THEN rules, the problem of continuity has been investigated with respect to a concrete (specially constructed) model. In our contribution, we consider this problem in general for an arbitrary model.

We show that under certain assumptions about a model of fuzzy IF–THEN rules, the problem of its continuity is connected with the problem of solvability

* This paper has been partially supported by grants IAA1187301 of the GA AV ČR, 201/04/1033 of the GA ČR, and by the Deutsche Forschungsgemeinschaft as part of the Collaborative Research Center "Computational Intelligence" (531).

I. Perfilieva and S. Lehmke: *Safe Modelling of Fuzzy If–Then Rules*, Advances in Soft Computing **2**, 231–236 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

of the respective system of fuzzy relation equations. Theoretical results on the topic of solvability with necessary and sufficient conditions, and with sufficient conditions only can be found in [1, 5, 6, 7].

2 Fuzzy Relation as a Model of Fuzzy IF–THEN Rules

By a system of fuzzy IF–THEN rules we mean the following set of formal expressions:

$$\begin{array}{c} \mathcal{R}_1 : \ \mathsf{IF}\ X\ \mathsf{is}\ A_1\ \mathsf{THEN}\ Y\ \mathsf{is}\ B_1 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \mathcal{R}_n : \ \mathsf{IF}\ X\ \mathsf{is}\ A_n\ \mathsf{THEN}\ Y\ \mathsf{is}\ B_n \end{array} \tag{1}$$

where $A_i \in \mathcal{F}(\mathbf{X}), B_i \in \mathcal{F}(\mathbf{Y})$ and $\mathcal{F}(\mathbf{X})$, $\mathcal{F}(\mathbf{Y})$ are universes of fuzzy subsets on $\mathbf{X}$, respectively $\mathbf{Y}$.

Let us agree to model this system of fuzzy IF–THEN rules in a class of fuzzy relations on $\mathcal{F}(\mathbf{X} \times \mathbf{Y})$. To be able to express the relationship between system (1) and its model, we need to choose an appropriate algebra of operations over fuzzy subsets. For this purpose, let us choose a complete residuated lattice on $[0, 1]$:

$$\mathcal{L} = \langle [0, 1], \vee, \wedge, *, \rightarrow, \mathbf{0}, \mathbf{1} \rangle \tag{2}$$

with four binary operations and two constants (see [2] for details) extended by the binary operation $\leftrightarrow$ of equivalence:

$$x \leftrightarrow y = (x \rightarrow y) \wedge (y \rightarrow x).$$

Let us remark that in the considered case of a lattice with the support $[0, 1]$, the monoidal operation $*$ is a left continuous t-norm. Moreover,

$$\mathcal{F}(\mathbf{X}) = [0, 1]^{\mathbf{X}} \quad \text{and} \quad \mathcal{F}(\mathbf{Y}) = [0, 1]^{\mathbf{Y}}.$$

Saying that a fuzzy relation $R \in \mathcal{F}(\mathbf{X} \times \mathbf{Y})$ is a model of fuzzy IF–THEN rules (and therefore, of a dependence, partially given by them), we specify how this model can be used in computations.

Definition 1. *We say that a fuzzy set $B \in \mathcal{F}(\mathbf{Y})$ is an output of the (fuzzy) model $R \in \mathcal{F}(\mathbf{X} \times \mathbf{Y})$ given input $A \in \mathcal{F}(\mathbf{X})$ if*

$$B(y) = \bigvee_{x \in \mathbf{X}} (A(x) * R(x, y))$$

(in short, $B = A \circ R$).

Till now, we did not put any restriction on a fuzzy relation which models a set of fuzzy IF–THEN rules. We are going to do it below, considering different connections between the original data containing in IF–THEN rules and their model.

Definition 2. *We say that a model $R \in \mathcal{F}(\mathbf{X} \times \mathbf{Y})$ is a safe model of fuzzy IF–THEN rules (1) if for all $i = 1, \ldots, n$*

$$A_i \circ R = B_i. \tag{3}$$

It is easy to see that a model is safe if and only if it gives a solution to the system of fuzzy relation equations expressed by (3) where the fuzzy sets A_i and B_i are given by (1). In this point we may refer to different criteria of solvability (see Introduction) which tell us when we may expect to have a safe model. We recall the following one to which we will refer later.

Proposition 1. *If system (3) is solvable with respect to an unknown fuzzy relation R then the relation*

$$\hat{R}(x, y) = \bigwedge_{i=1}^{n} (A_i(x) \to B_i(y)) \tag{4}$$

is the greatest solution to (3) (see [1, 7]).

Let us also remark that the notion of a *safe model* has been investigated under the name *correct interpretation* [3].

3 Continuity of Model of Fuzzy IF–THEN Rules

The safeness seems to be a natural property of a model of fuzzy IF–THEN rules. On the other side, it implies a solvability of a corresponding system of fuzzy relation equations which is a rather strong requirement. We may try to weaken the property of safeness requiring a continuity of a model of IF–THEN rules with respect to the given data. This notion is defined in a restrictive meaning which we explain in the following definition.

Definition 3. *A model $R \in \mathcal{F}(\mathbf{X} \times \mathbf{Y})$ of fuzzy IF–THEN rules (1) is continuous with respect to the given data $\{(A_i, B_i)\}, i = 1, \ldots, n$, if for each i and for each input $A \in \mathcal{F}(\mathbf{X})$ the following inequality holds true:*

$$\bigwedge_{y \in \mathbf{Y}} (B_i(y) \leftrightarrow (A \circ R)(y)) \geq \bigwedge_{x \in \mathbf{X}} (A_i(x) \leftrightarrow A(x)). \tag{5}$$

By (5), whenever an input fuzzy set A is close to a fuzzy set A_i, presented on the left-hand side of the rules, the computed output $A \circ R$ of the model R is close to the respective fuzzy set B_i, presented on the right-hand side of the rules. Comparing to the property of safeness, continuity with respect to the given data is weaker because we replace the strict equality in (3) by the fuzzy equivalence in (5). Nevertheless, the safeness and continuity properties of a model of IF–THEN rules turn out to be equal. This is proved in the following theorem and its corollary.

Theorem 1. *Let $R \in \mathbf{F}(\mathbf{X} \times \mathbf{Y})$ be a model of fuzzy IF–THEN rules (1). Then for any $A \in \mathcal{F}(\mathbf{X})$ and all $y \in \mathbf{Y}$ it is true that*

$$B_i(y) \leftrightarrow (A \circ R)(y) \geq \delta_i(y) * \bigwedge_{x \in \mathbf{X}} (A_i(x) \leftrightarrow A(x)) \tag{6}$$

where

$$\delta_i(y) = B_i(y) \leftrightarrow (A_i \circ R)(y)$$

is the degree of solvability (cf. [1]) of the i-th rule of the system (3).

Proof. Let $R \in \mathcal{F}(\mathbf{X} \times \mathbf{Y})$ be a fuzzy relation and $A \in \mathcal{F}(\mathbf{X})$ a fuzzy set. Denote

$$B = A \circ R \tag{7}$$

and observe, that $B \in \mathcal{F}(\mathbf{Y})$. By transitivity of operation $\leftrightarrow$, we obtain

$$B \leftrightarrow B_i \geq (B \leftrightarrow (A \circ \hat{R})) * ((A \circ \hat{R}) \leftrightarrow (A_i \circ \hat{R})) * ((A_i \circ \hat{R}) \leftrightarrow B_i) \tag{8}$$

where $i \in \{1, \ldots, n\}$ and $\hat{R}$ is given in (4). Let us estimate each of three multiplicands in the right-hand side of (8).

By (7), the first one is equal to 1:

$$(B \leftrightarrow (A \circ \hat{R})) \equiv 1.$$

The second one may be estimated from below for arbitrary $y \in \mathbf{Y}$ as follows:

$$(A \circ \hat{R})(y) \leftrightarrow (A_i \circ \hat{R})(y) = (\bigvee_{x \in \mathbf{X}} (A(x) * \hat{R}(x, y))) \leftrightarrow \bigvee_{x \in \mathbf{X}} (A_i(x) * \hat{R}(x, y)) \geq$$
$$\bigwedge_{x \in \mathbf{X}} (A(x) * \hat{R}(x, y) \leftrightarrow A_i(x) * \hat{R}(x, y)) \geq \bigwedge_{x \in \mathbf{X}} (A(x) \leftrightarrow A_i(x)).$$

Here, we used the facts

$$\bigwedge_{i \in I} (a_i \leftrightarrow b_i) \leq (\bigvee_{i \in I} a_i \leftrightarrow \bigvee_{i \in I} b_i)$$

and

$$(a \leftrightarrow b) * (c \leftrightarrow d) \leq (a * c \leftrightarrow b * d)$$

that are valid in any complete residuated lattice and, therefore, in a complete BL-algebra as its particular case.

The third multiplicand

$$(A_i \circ \hat{R})(y) \leftrightarrow B_i(y)$$

is equal to $\delta_i(y)$. Summarizing all three estimations, we come to the conclusion of the theorem.

Theorem 1 estimates a deviation (via equivalence) of a computed output of a model from the known one given on the right-hand side of the rules. The estimation uses the deviation between the respective inputs (again expressed via equivalence) and the so called degree of solvability of the i-th rule of the system (3. Taking into account that the system (3) is solvable if and only if $\delta_i(y) = 1$ for all $i = 1, \ldots, n$ and for all $y \in \mathbf{Y}$, we easily obtain the following corollary.

Corollary 1. *The model $R \in \mathcal{F}(\mathbf{X} \times \mathbf{Y})$ is a safe model of fuzzy IF–THEN rules (1) if and only if it is continuous.*

Proof. $\Rightarrow$ Suppose that $R \in \mathcal{F}(\mathbf{X} \times \mathbf{Y})$ is a safe model of fuzzy IF–THEN rules (1), then R solves the system (3) and therefore, $\delta_i(y) = 1$ for all $i = 1, \ldots, n$ and for all $y \in \mathbf{Y}$. By (6), R is continuous.

$\Leftarrow$ Suppose that $R \in \mathcal{F}(\mathbf{X} \times \mathbf{Y})$ is a continuous model of (1) so that (5) holds for each $i = 1, \ldots, n$ and for each $A \in \mathcal{F}(\mathbf{X})$. Let us take some $i \in \{1, \ldots, n\}$ and apply (5) for $A = A_i$. We obtain

$$\bigwedge_{y \in \mathbf{Y}} (B_i(y) \leftrightarrow (A_i \circ R)(y)) \geq 1$$

which implies that

$$B_i(y) \leftrightarrow (A_i \circ R)(y) = 1$$

for each $y \in \mathbf{Y}$.

Therefore, R solves the system (3) and is safe by the definition.

It is worth noticing that besides the equivalence between safeness and continuity, Corollary 1 establishes the new criterion of solvability of a system of fuzzy relation equations.

4 Conclusion

We have contributed to the problem of modelling of fuzzy IF–THEN rules. Two new notions, namely a safe model of fuzzy IF–THEN rules and a continuous model of fuzzy IF–THEN rules with respect to a given data have been introduced. Both notions characterize a model of fuzzy IF–THEN rules from the point of view of its further utilization in a computation of an output value for some arbitrary input.

We have connected the problem of constructing of a safe model of fuzzy IF–THEN rules with the problem of solvability of the respective system of fuzzy relation equations. We showed that in the case of unsolvable system, its degree of solvability stands as a coefficient in the characteristic inequality of a continuity. Therefore, the decrease of a degree of solvability causes the proportional break of continuity. This means that a model does not match well the given data.

References

1. Gottwald S (1993) Fuzzy Sets and Fuzzy Logic. The Foundations of Application – from a Mathematical Point of View. Vieweg, Braunschweig
2. Höhle U (1995) Commutative residuated l-monoids. In: Höhle U, Klement E P (eds) Non-Classical Logics and Their Applications to Fuzzy Subsets. A Handbook of the Mathematical Foundations of Fuzzy Set Theory. Kluwer, Dordrecht, 53–106
3. Lehmke S, Reusch B, Temme K H, Thiele H (2003) Mathematical Foundations of Fuzzy Inference. In: Schwefel H P, Wegener I, Weinert K (eds) Advances in Computational Intelligence. Springer-Verlag, Berlin
4. Mamdani A, Assilian S (1975) An experiment in linguistic synthesis with a fuzzy logic controller. Int. J. Man-Machine Studies 7:1–13
5. Perfilieva I (2003) Solvability of a System of Fuzzy Relation Equations: Easy to Check Conditions. Neural Network World 13:571–580
6. Perfilieva I, Tonis A (2000) Compatibility of systems of fuzzy relation equations. Internat. J. General Systems 29:511–528
7. Sanchez E (1976) Resolution of composite fuzzy relation equations. Information and Control 30:38–48.

Perception-Based Logical Deduction*

Vilém Novák

University of Ostrava
Institute for Research and Applications of Fuzzy Modeling
30. dubna 22, 701 03 Ostrava 1, Czech Republic

and

Institute of Information and Automation Theory
Academy of Sciences of the Czech Republic
Pod vodárenskou věží 4, 186 02 Praha 8, Czech Republic

Vilem.Novak@osu.cz

1 Introduction

In this paper, we will formalize the way, how people make inferences on the basis of the, so called, linguistic description which is a set of fuzzy IF-THEN rules understood as expressions of natural language. We will explain our idea on the following example.

Let us consider a simple linguistic description consisting of two rules

$$\begin{aligned} \mathcal{R}_1 &:= \ \mathsf{IF}\ X\ \mathsf{is}\ \textit{small}\ \mathsf{THEN}\ Y\ \mathsf{is}\ \textit{very big} \\ \mathcal{R}_2 &:= \ \mathsf{IF}\ X\ \mathsf{is}\ \textit{big}\ \mathsf{THEN}\ Y\ \mathsf{is}\ \textit{very small}. \end{aligned}$$

The variables X, Y formally replace the meaning of some nouns; in technical applications, which are most common in the applications of fuzzy logic, these may be temperature, pressure, angle, size of throttle, electrical current, etc. Such rules are used in a concrete situation (control, decision-making, classification) and so, the expressions "small, big" are used in a certain context — we will call it *linguistic context*. Furthermore, the expressions "small, big" characterize the given situation (what is spoken about) and so, they form a *topic* of the linguistic description.

Let us, for simplicity, consider the linguistic context of both variables X, Y as an interval $[0, 1]$. Then "small" are some values around 0.3 (and smaller) and "big" are some values around 0.7 (and bigger). The linguistic description tells us that small input values should lead to very big output ones and big

* The paper has been supported by grants 201/04/1033 of the GA ČR, A1075301 of the GA AV ČR and Deutsche Forschungsgemeinschaft as part of the Collaborative Research Center "Computational Intelligence" (531).

V. Novák: *Perception-Based Logical Deduction*, Advances in Soft Computing **2**, 237–250 (2005)
www.springerlink.com © Springer-Verlag Berlin Heidelberg 2005

input values to very small output ones. Therefore, given an input, say $X = 0.3$, we expect the output $Y \approx 0.9$ due to the rule $\mathcal{R}_1$ since we evaluate the input value as small, and thus, in this case the output should be very big. Similarly, for $X = 0.75$ we expect the output $Y \approx 0.15$ due to the rule $\mathcal{R}_2$.

After closer look at this way of reasoning, we may see that each rule provides us with a certain knowledge (related to the concrete application) and though vague, we may *distinguish* between both rules. Therefore, the input value of X is evaluated by some linguistic expression from the topic of the linguistic description. The latter expression, which in our case is "small", can be taken as a *perception* of the value $X = 0.3$. Based on this perception and the rule $\mathcal{R}_1$, we conclude that Y should be "very big" which, in the given context, corresponds to the mentioned value $Y \approx 0.9$. Selection of a concrete output value of Y is partly random but it should be in correspondence with the truth degree of the statement "X=0.3 is small". This means that the greater is this degree, the greater must the truth of the statement "the output value of Y is very big". In practice this means that, if the value of X increases then the value of Y decreases because the truth of "X is small" also decreases.

Of course, finding a perception is bound by the way how are the rules specified. For example, if the rule $\mathcal{R}_1$ were replaced by

$$\mathcal{R}'_1 := \ \mathsf{IF}\ X\ \mathsf{is}\ \textit{very big}\ \mathsf{THEN}\ Y\ \mathsf{is}\ \textit{big}$$

then there is no linguistic expression in the topic of the linguistic description which could be taken as a perception of the value 0.3. Therefore, no value of Y would correspond to 0.3.

Our goal is to develop a formal theory which will give results in accordance with the above intuition. We will call the just described way of finding a conclusion on the basis of the linguistic description the *perception-based logical deduction*. Successful solution of this fact depends on the way of interpretation of the linguistic description. In the literature, two principal kinds of this interpretation are described. The first one is called *relational*. Its main idea is to find a good approximation of some function which is known only roughly. Therefore, the function is divided into imprecise "parts" using fuzzy relations constructed from fuzzy sets with continuous membership functions of more or less arbitrary shape. Each such membership function is assigned a name to be able to get better orientation in the rules, but without real linguistic meaning. Formally, linguistic descriptions are assigned one of two kinds of normal forms: the disjunctive or conjunctive normal form (see [21]). The resulting fuzzy relation then depends on the choice of the underlying algebra of truth values. A conclusion is derived on the level of semantics rather than on the level of syntax. Note that the relational interpretation is the most often considered interpretation used in applications.

The second interpretation of the linguistic description is *logical*. This has been presented, e.g. in [8, 13, 20] and elsewhere. The main idea is that the fuzzy IF-THEN rules are genuine linguistic expressions interpreted in a way which mimics human understanding to them and the linguistic description is

a specific text. Components of the fuzzy IF-THEN rules are usually the, so called, *evaluating linguistic expressions*. We argue that logical interpretation is a good basis for the perception-based deduction. In [6, 7, 20], we have tried to use first order predicate fuzzy logic with evaluated syntax for this task. However, the problem goes behind the first order level and therefore, in this paper, we describe the perception-based logical deduction using the means of fuzzy intensional logic. Let us also stress that we have practical experiences with this method, described, e.g. in [5, 18].

2 Fuzzy intensional logic

The main tool for the logical analysis is extension of the fuzzy type theory (FTT) to fuzzy intensional logic (FIL), see [14]. The detailed explanation of FTT can be found in [17, 16] and the classical type theory is in details described in [1].

2.1 Syntax of FIL

FIL is obtained from FTT by adding an additional type ω, which will represent *possible worlds* or *contexts* to the basic elementary types o (truth values) and ϵ (objects). In this section, we will very briefly touch some of the main points of FTT and FIL.

Let ϵ, o, ω be distinct objects. The set of types is the smallest set *Types* satisfying:

(i) $\epsilon, o, \omega \in \mathit{Types}$,
(ii) if $\alpha, \beta \in \mathit{Types}$ then $(\alpha\beta) \in \mathit{Types}$.

The set Form_α is a set of formulas of type $\alpha \in \mathit{Types}$, which is the smallest set satisfying:

(i) $x_\alpha \in \mathit{Form}_\alpha$ and $c_\alpha \in \mathit{Form}_\alpha$,
(ii) if $B \in \mathit{Form}_{\beta\alpha}$ and $A \in \mathit{Form}_\alpha$ then $(BA) \in \mathit{Form}_\beta$,
(iii) if $A \in \mathit{Form}_\beta$ then $\lambda x_\alpha\, A \in \mathit{Form}_{\beta\alpha}$,

If $A \in \mathit{Form}_\alpha$ is a formula of the type $\alpha \in \mathit{Types}$ then we write A_α. To simplify the notation, we will omit brackets wherever it will not lead to misunderstanding. We will also use the "dot" notation: starting with the dot, the rest of the formula is considered to be closed in the brackets.

The structure of truth values is supposed to form a complete IMTL-algebra, i.e. a complete residuated lattice with involutive negation defined by $\neg a = a \to \mathbf{0}$ and fulfilling the prelinearity condition .

We define a special formula which determines a nonzero truth value:

$$\Upsilon_{oo} := \lambda z_o(\neg(\boldsymbol{\Delta}(\neg z_o)).$$

The following properties of FTT will be used in the sequel.

Lemma 1. *(a) Let* $T \vdash (\exists x_\alpha)\boldsymbol{\Delta}B$. *Then* $T \cup B_{x_\alpha}[\mathbf{u}_\alpha]$ *where* $\mathbf{u}_\alpha \notin J(T)$ *is a conservative extension of* T *(rule C).*
(b) Let $T \vdash (\exists x_\alpha)\boldsymbol{\Delta}B_{o\alpha}x_\alpha$. *Then* $T \vdash B_{o\alpha} \cdot \iota_{\alpha(o\alpha)}B_{o\alpha}$.
(c) $\vdash (\exists x)\boldsymbol{\Delta}A_o \Rightarrow (\exists x)A_o$.
(d) Let $T \vdash \Upsilon z_o \,\&\, (z_o \Rightarrow y_o)$. *Then* $T \vdash \Upsilon y_o$.
(e) $\vdash \boldsymbol{\Delta}(x_o \,\&\, y_o) \equiv (\boldsymbol{\Delta}x_o \,\&\, \boldsymbol{\Delta}y_o)$.
(f) $(\exists x_\alpha)(A_o \,\&\, B_o) \Rightarrow (\exists x_\alpha)A_o \,\&\, (\exists x_\alpha)B_o$.

2.2 Semantics of FIL

Let D be a set of objects, L be a set of truth values and W be a set of possible worlds.

A *basic frame* based on D, L, W is a family of sets $(M_\alpha)_{\alpha \in Types}$ where

(i) $M_\epsilon = D$ is a set of objects,
(ii) $M_o = L$ is a set of truth values,
(iii) $M_\omega = W$ is a set of possible worlds,
(iv) for each type $\gamma = \beta\alpha$, the corresponding set $M_\gamma \subseteq M_\beta^{M_\alpha}$.

A *frame* is

$$\mathcal{M} = \langle (M_\alpha, =_\alpha)_{\alpha \in Types}, \mathcal{L}_\Delta \rangle \tag{1}$$

where $(M_\alpha)_{\alpha \in Types}$ is a basic frame such that $\mathcal{L}_\Delta$ is a structure of truth values being a complete, linearly ordered IMTL$_\Delta$ algebra and $=_\alpha$ is a fuzzy equality on M_α for every $\alpha \in Types$. Namely, $=_o$ is the biresiduation $\leftrightarrow$, and if $\beta = \epsilon$ or $\beta = \omega$ then $=_\beta \subsetneq M_\beta \times M_\beta$ is a fuzzy equality on the set M_β. Otherwise, $=_\alpha$ is the fuzzy equality

$$[m =_{\beta\alpha} m'] = \bigwedge_{n_\alpha \in M_\alpha} [m(n)) =_\beta m'(n')] \tag{2}$$

for every $m, m' \in M_{\beta\alpha}$ (the square brackets denote truth value of the fuzzy equality). Recall that each function $F \in M_{\beta\alpha}$ is weakly extensional w.r.t $=_\alpha$ and $=_\beta$, i.e. $[m =_\alpha m'] = \mathbf{1}$ implies that $[F(m) =_\beta F(m')] = \mathbf{1}$ for all $m, m' \in M_\alpha$.

Interpretation of formulas in the frame is defined recursively starting from elementary objects. A *general model* is an interpretation $\mathcal{I}^\mathcal{M}$ such that

$$\mathcal{I}_p^\mathcal{M}(A_\alpha) \in M_\alpha \tag{3}$$

holds true for every formula A_α, $\alpha \in Types$ and every assignment $p \in \mathrm{Asg}(\mathcal{M})$ of elements to variables. This means that the value of each formula is always defined in the general model. As a special case, interpretation $\mathcal{I}_p^\mathcal{M}(\iota_{\alpha(o\alpha)})$ of the description operator $\iota_{\alpha(o\alpha)}$ is some operation assigning to each normal fuzzy set en element from its kernel (cf. [16]). In fuzzy set theory, such an operation is called *defuzzification*. In the case of evaluating linguistic expressions, it is reasonable to use the DEE defuzzification method (see [15]). We also define a special operator

$$\imath z_\alpha A_o := \iota_{\alpha(o\alpha)}(\lambda z_\alpha A_o).$$

A theory T is a set of formulas A_o of type o (i.e. their interpretation is a truth value). Recall that FIL includes predicate IMTL fuzzy logic. Moreover, it is possible to consider other structure of truth values, namely Łukasiewicz MV-algebra with Δ and some other ones.

3 Logical structure of linguistic description

3.1 Fuzzy IF-THEN rules

In this paper, we suppose that the fuzzy IF-THEN rules consist of linguistic expressions and they are themselves taken also as special (conditional) linguistic expressions. In [13, 15, 20], the theory of *evaluating linguistic expressions* has been developed. The fuzzy IF-THEN rules were supposed to consist only of them. In this paper, we present our theory in a more general setting.

A simple linguistic predication is a linguistic expression of the form

$$\langle \text{noun} \rangle \ \mathsf{is}\ \mathcal{A} \tag{4}$$

where $\mathcal{A}$ is a linguistic expression (not predication). A compound linguistic predication is obtained when joining simple linguistic predications by connectives. Namely, the following are compound linguistic predications:

$$\mathcal{R}^A := X \ \mathsf{is}\ \mathcal{A} \ \text{AND}\ Y \ \mathsf{is}\ \mathcal{B}, \tag{5}$$

$$\mathcal{R}^I := \mathsf{IF}\ X \ \mathsf{is}\ \mathcal{A} \ \mathsf{THEN}\ Y \ \mathsf{is}\ \mathcal{B}. \tag{6}$$

Either of (5) or (6) will be call *fuzzy IF-THEN rule* in the sequel.

When speaking about meaning of a linguistic expression, we must distinguish its *intension* and *extension* (cf. [10]). We will formalize these concepts as follows. To simplify notation, we will use special variables (formulas) for specific types only. Namely, $w \in Form_\omega$, $x \in Form_\alpha$, $y \in Form_\beta$ where $\alpha, zbeta$ are some arbitrary, possibly different types.

Let $\mathcal{A}$ be a linguistic expression and $A \in Form_{(o\alpha)\omega}$ be a formula assigned to it. The *intension* and *extension* of $\mathcal{A}$ are the formulas

$$\mathrm{Int}(\mathcal{A}) := \lambda w\, \lambda x \cdot A_{(o\alpha)\omega} w x, \tag{7}$$

$$\mathrm{Ext}_w(\mathcal{A}) := \mathrm{Int}(\mathcal{A}) w \tag{8}$$

respectively. Interpretation of $\mathrm{Int}(\mathcal{A})$ is a function which assigns to each possible world a fuzzy set in the universe of type α.

Using λ-conversion we get

$$\vdash \mathrm{Ext}_w(\mathcal{A}) \equiv \lambda x \cdot A_{(o\alpha)\omega} w x. \tag{9}$$

Thus, interpretation of $\mathrm{Ext}_w(\mathcal{A})$ in a possible world w is the corresponding fuzzy set.

Intension of a fuzzy IF-THEN rule $\mathcal{R}$ is

$$\mathrm{Int}(\mathcal{R}) := \lambda w\, \lambda w' \cdot \lambda x\, \lambda y \cdot A_{(o\alpha)\omega} w x \,\square\, B_{(o\beta)\omega} w' y \tag{10}$$

where $\square$ is either of the connectives **&**, $\wedge$ or $\Rightarrow$.

To simplify the notation, we will introduce the symbol $\varphi := (o\gamma)\omega$ for the type of intension. It represents a function assigning to each possible world a fuzzy set. The type γ can be arbitrary and, therefore, though we write the same φ for different formulas, their types may differ in the position of γ. If necessary, we will explicitly stress that it should be the same. However, since we fixed above the types for x and y, writing $A_\varphi wx$ means, in fact, the formula $A_{(o\alpha)\omega} wx$ and similarly, $B_\varphi wy$ means the formula $B_{(o\beta)\omega} wy$.

We say that the intension $\mathrm{Int}(\mathcal{A})$ is *normal* if

$$T \vdash (\forall w)(\exists x) \boldsymbol{\Delta} A_{(o\alpha)\omega} wx. \tag{11}$$

This means that in each possible world we can find an element that surely has the property named by $\mathcal{A}$. The normality assumption seems to be quite natural. Namely, it says that in each possible world there is an element typical for the given property (it is the prototype of the latter). Indeed, we can always find a typical "red colour", "small distance", "high tree", "good car", etc.

On the basis of (11),

Lemma 2. *Let intensions* $\mathrm{Int}(\mathcal{A})$, $\mathrm{Int}(\mathcal{B})$ *be normal. Then:*

(a) $T \vdash (\forall w)(\forall w')(\forall x)(\exists y) \boldsymbol{\Delta}(A_{(o\alpha)\omega} wx \Rightarrow B_{(o\beta)\omega} w' y)$,
(b) $T \vdash (\forall w)(\forall w')(\exists x)(\exists y) \boldsymbol{\Delta}(A_{(o\alpha)\omega} wx \,\square\, B_{(o\beta)\omega} w' y)$

where $\square$ *is either of the connectives* **&** *or* $\wedge$.

Proof. (a) Let $\mathbf{v} \notin Form_T$, be a new constant of type α. It follows from Lemma 1(a) that $T' = T \cup \{B_{(o\beta)\omega} w' \mathbf{v}\}$ is a conservative extension of T. Then $T' \vdash A_{(o\alpha)\omega} wx \Rightarrow B_{(o\beta)\omega} w' \mathbf{v}$ and so, $T' \vdash \boldsymbol{\Delta}(A_{(o\alpha)\omega} wx \Rightarrow B_{(o\beta)\omega} w' \mathbf{v})$ and finally, $T \vdash (\forall x)(\exists y) \boldsymbol{\Delta}(A_{(o\alpha)\omega} wx \Rightarrow B_{(o\beta)\omega} w' y)$ by properties of FIL and conservativeness of T', which implies (a) using generalization.

(b) Similarly as above, let $\mathbf{u}, \mathbf{b} \notin Form_T$ be new constants of type α. Then $T' = T \cup \{A_{(o\alpha)\omega} w \mathbf{u}, B_{(o\beta)\omega} w' \mathbf{v}\}$ is a conservative extension of T. Using Lemma 1(e) and properties of FIL, we prove $T' \vdash \boldsymbol{\Delta}(A_{(o\alpha)\omega} w \mathbf{u} \,\square\, B_{(o\beta)\omega} w' \mathbf{v})$. Then we obtain (b) by similar arguments as in the case of (a).

The formulas (a) and (b) of this lemma characterize normality of intension of fuzzy IF-THEN rules (depending on their form). By Lemma 1(c), normality of fuzzy IF-THEN rules imply either of

$$T \vdash (\forall w)(\forall w')(\forall x)(\exists y)(A_{(o\alpha)\omega} wx \Rightarrow B_{(o\beta)\omega} w' y),$$
$$T \vdash (\forall w)(\forall w')(\exists x)(\exists y)(A_{(o\alpha)\omega} wx \,\square\, B_{(o\beta)\omega} w' y).$$

3.2 Structure of linguistic description

A system of fuzzy IF-THEN rules

$$\begin{aligned} &\mathsf{IF}\ X\ \mathsf{is}\ \mathcal{A}_1\ \mathsf{THEN}\ Y\ \mathsf{is}\ \mathcal{B}_1,\\ &\mathsf{IF}\ X\ \mathsf{is}\ \mathcal{A}_2\ \mathsf{THEN}\ Y\ \mathsf{is}\ \mathcal{B}_2,\\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ &\mathsf{IF}\ X\ \mathsf{is}\ \mathcal{A}_m\ \mathsf{THEN}\ Y\ \mathsf{is}\ \mathcal{B}_m. \end{aligned} \tag{12}$$

will be called a *linguistic description* and denoted by L . As mentioned, it can be understood as a *special text*. If all $\mathcal{A}, \mathcal{B}$'s are evaluating linguistic expressions then this system can be understood as a description of some strategy (e.g. control or decision-making), behaviour of a system, etc.

On the formal level, the linguistic description is a set of intensions

$$L = \{\mathrm{Int}(\mathcal{R}_1), \ldots, \mathrm{Int}(\mathcal{R}_m)\},$$

each of them defined in (10) and formed within a language $J(T)$ of some theory T. We will write L^I if all rules $\mathcal{R}$ are implications and L^A if they are conjunctions. If all the rules are normal then the linguistic description is called *normal*. On the basis of the normality assumption, the linguistic description L *provides a list of formulas* of the form of Lemma 2(a) or (b) which must be provable in T.

A text written in natural language is a set of sentences. Therefore, we may distinguish topic and focus of each sentence (cf. [9, 22]) which then form topic and focus of the whole text. Namely, the *topic of a linguistic description* is a set of linguistic expressions

$$\{\mathcal{A}_j \mid j = 1, \ldots, m\}. \tag{13}$$

This means that the topic is determined by the set of linguistic predications, each of them occurring in the left part of the respective fuzzy IF-THEN rule. Similarly, its focus is

$$\{\mathcal{B}_j \mid j = 1, \ldots, m\}. \tag{14}$$

Note that the common characteristics both for the topic as well as for the focus of the linguistic description, which ties together all its fuzzy IF-THEN rules, is the variable X or Y, respectively.

On the basis of Lemma 2(a), we get the following theorem.

Theorem 1. *Let T be the theory corresponding to a linguistic description L^I. Let $\mathbf{u}^0 \in Form_\alpha$ be a formula. Then*

$$T \vdash (\forall w)(\forall w') \cdot A_{j,\varphi}\, w\mathbf{u}^0 \Rightarrow \cdot B_{\varphi,j}\, w'(\imath y \cdot A_{\varphi,j}\, w\mathbf{u}^0 \Rightarrow B_{\varphi,j}\, w'y),$$

$j = 1, \ldots, m$.

Proof. For all $j = 1, \ldots, m$ we can write the following proof:

(L.1) $T \vdash (\forall x)(\exists y)\boldsymbol{\Delta}(A_{j,\varphi}\, wx \Rightarrow B_{j,\varphi}\, w'y)$ (assumption in T, substitution)
(L.2) $T \vdash (\exists y)\boldsymbol{\Delta}(A_{j,\varphi}\, w\mathbf{u}^0 \Rightarrow B_{j,\varphi}\, w'y)$ (L.1, substitution)
(L.3) $T' \vdash (\forall w)(\forall w') \cdot A_{j,\varphi}\, w\mathbf{u}^0 \Rightarrow B_{\varphi,j}\, w'y(\imath y \cdot A_{\varphi,j}\, w\mathbf{u}^0 \Rightarrow B_{\varphi,j}\, w'y)$ (L.2, Lemma 1(b), generalization)

It follows from this theorem that replacing the variable $x \in Form_\alpha$ by a formula $\mathbf{u}^0$, each rule from the given linguistic description holds for the latter and the typical element $\imath y \cdot A_{\varphi,i} w\mathbf{u}^0 \Rightarrow B_{\varphi,i} w'y$ which replaces the variable y. Note that $\mathbf{u}^0$ can be arbitrary (substitutable) formula, i.e. also a constant or a variable not occurring in $A_{j,\varphi}$.

Let us put

$$Eval_{o(\varphi\alpha\omega)} := \lambda w\, \lambda x_\alpha\, \lambda z_\varphi\, (\exists z_o)(\Upsilon z_o \,\&\, \cdot z_o \Rightarrow z_\varphi w x_\alpha). \tag{15}$$

Let $\mathrm{Int}(\mathcal{A})$ be intension of some evaluating expression $\mathcal{A}$. We say that an element x in the possible world w is *evaluated* by the expression $\mathcal{A}$ if the following formula is provable:

$$T \vdash Eval_{o(\varphi\alpha\omega)} \cdot wx\, \mathrm{Int}(\mathcal{A}). \tag{16}$$

Recall that Υz_o is a formula stating that z_o is a non-zero truth value. Then (16) means that the truth value of the statement "x in the possible world w is evaluated by $\mathcal{A}$", (i.e. the truth value of the formula $\mathrm{Int}(\mathcal{A})wx$), is at least as big as some *non-zero* truth value z_o. Hence, (16) means that "an element x is evaluated in the possible world w by the linguistic expression $\mathcal{A}$".

3.3 Formal theory of perception-based logical deduction

In this section, we will work with a linguistic description $L^{\ I}$, i.e. it consists of intensions (10) and determines the theory T discussed above.

Theorem 2. *Let $\mathbf{u}^0 \in Form_\alpha$ be a formula representing an object of type α, $\mathbf{w}^0$ be a formula representing a possible world and let $\mathbf{b}_i^0$, $i = 1, \ldots, m$ be formulas of type o such that $T \vdash \mathbf{b}_i^0 \equiv A_{\varphi,i}\mathbf{w}^0\mathbf{u}^0$. Furthermore, let $T \vdash \Upsilon\mathbf{b}_i^0$ and put $\hat{\mathbf{y}} := \imath y \cdot \mathbf{b}_i^0 \Rightarrow B_{\varphi,i} w'y$. Then*

$$T \vdash (\forall w') \cdot Eval_{o(\varphi\alpha\omega)}\, w'\hat{\mathbf{y}} \cdot B_{\varphi,i},$$

and

$$T \vdash (\forall w')\Upsilon(B_{\varphi,i} w'\hat{\mathbf{y}}).$$

Proof. (L.1) $T \vdash \Upsilon\mathbf{b}_i^0 \,\&\, (\mathbf{b}_i^0 \Rightarrow B_{\varphi,i} w'\hat{\mathbf{y}})$ (assumption, Theorem 1)
(L.2) $T \vdash \mathrm{Int}(\mathcal{B}_i) \equiv \lambda w'\, \lambda y\, B_{\varphi,i} w'y$ (definition of intension)
(L.3) $T \vdash (\exists z_o) \cdot \Upsilon z_o \,\&\, (z_o \Rightarrow B_{\varphi,i} w'\hat{\mathbf{y}})$ (L.1, substitution, modus ponens)
(L.4) $T \vdash B_{\varphi,i} w'\hat{\mathbf{y}} \equiv (\lambda \bar{w}'\, \lambda y\, B_{\varphi,i} \bar{w}' y) w'\hat{\mathbf{y}}$ (λ-conversion)

(L.5) $T \vdash (\exists y_o) \cdot \Upsilon z_o \,\&\, (z_o \Rightarrow \mathrm{Int}(\mathcal{B}_i) w' \hat{\mathbf{y}})$
(L.3, L.4, properties of FTT, definition of $\mathrm{Int}(\mathcal{B}_i)$)

(L.6) $T \vdash (\forall w') \cdot Eval_{o(\varphi\alpha\omega)}\, w' \hat{\mathbf{y}} \cdot B_{\varphi,i}$
(L.5, definition of *Eval*, generalization)

The rest follows from Lemma 1(d).

This theorem has the following interpretation. Let us consider a linguistic description $L^{\ I}$ (i.e., the fuzzy IF-THEN rules are implications). If we find a formula $\mathrm{Int}(\mathcal{A}_i)$ of an expression from the topic of $L^{\ I}$ and an element $\mathbf{u}^0$ in the possible world $\mathbf{w}^0$ such that $A_{\varphi,i}\mathbf{w}^0\mathbf{u}^0$ has a non-zero truth degree then we conclude that the element $\hat{\mathbf{y}}$, *typical for the formula* $\mathbf{b}_i^0 \Rightarrow B_{\varphi,i} w' y$, is evaluated by the linguistic expression $\mathcal{B}_i$ in every possible world w', where $\mathrm{Int}(\mathcal{B}_i)$ belongs to the focus of $L^{\ I}$.

In other words, it follows from Theorems 1 and 2 that, when learning that some element $\mathbf{u}^0$ is evaluated by a concrete antecedent $\mathrm{Int}(\mathcal{A}_i)$ of an expression from the topic of $L^{\ I}$, we derive an element $\hat{\mathbf{y}}_i$ about which we know that it *is evaluated* by $\mathcal{B}_i$ from the corresponding focus. This element is the result of our reasoning based on the linguistic description.

Now, let us define a partial ordering of *sharpness* between formulas of type φ:

$$\preceq_{(o\varphi)\varphi} := \lambda z_\varphi\, \lambda z'_\varphi\, (\forall w)(\forall x)(z_\varphi w x \Rightarrow z'_\varphi w x). \tag{17}$$

This ordering means that the meaning of the formula z_φ is *sharper* than that of z'_φ in the following sense: if $z_\varphi wx$ is true in some (nonzero) degree for arbitrary x in arbitrary possible world w than $z'_\varphi wx$ is true at least in the same degree as well. For example, if x is, at least partly, "very big" in all possible worlds then it is also "big" in them.

Furthermore, we define a sharp ordering of sharpness:

$$\prec_{(o\alpha)\alpha} := \lambda z_\alpha\, \lambda z'_\alpha \cdot (z_\alpha \preceq_{(o\alpha)\alpha} z'_\alpha) \,\&\, \neg \boldsymbol{\Delta}(z_\alpha \equiv z'_\alpha) \tag{18}$$

where $\alpha = \varphi$ or $\alpha = o$. In the latter case, $\prec$ is a sharp ordering of truth values.

Lemma 3. *(a) $\preceq$ is a partial ordering on intensions, i.e. $z_\varphi \preceq z_\varphi$, $z_\varphi \preceq z'_\varphi \,\&\, z'_\varphi \preceq z_\varphi \Rightarrow z_\varphi \equiv z_\varphi$ and $z_\varphi \preceq z'_\varphi \,\&\, z'_\varphi \preceq z''_\varphi \Rightarrow z_\varphi \preceq z''_\varphi$.*
(b) $T \vdash \neg(z_\alpha \prec_{(o\alpha)\alpha} z_\alpha)$ where $\alpha = \varphi$ or $\alpha = o$.

Proof. (a) Is a straightforward consequence of the properties of implication $\Rightarrow$, equality $\equiv$, and $\boldsymbol{\Delta}$.

(b)

(L.1) $T \vdash \top \equiv z_\alpha wx \Rightarrow z_\alpha wx$ (properties of FTT)
(L.2) $T \vdash \bot \equiv \neg(z_\alpha wx \Rightarrow z_\alpha wx)$ (L.1., properties of FTT)
(L.3) $T \vdash \neg\neg\boldsymbol{\Delta}(z_\alpha \equiv z_\alpha)$ (properties of FTT)
(L.4) $T \vdash \top \,\nabla\, \bot$ (properties of FTT)
(L.5) $T \vdash \neg\neg\boldsymbol{\Delta}(z_\alpha \equiv z_\alpha)\, \nabla\, \neg(z_\alpha wx \Rightarrow z_\alpha wx)$
(L.2, L.3, L.4, properties of FTT)

(L.6) $T \vdash \neg(z_\alpha \prec_{(o\alpha)\alpha} z_\alpha)$ (L.5, properties of FTT)

The following is a formula of incomparability of intensions:

$$|_{(o\varphi)\varphi} := \lambda z_\varphi \lambda z'_\varphi \cdot \neg(z_\varphi \preceq z'_\varphi) \,\&\, \neg(z'_\varphi \preceq z_\varphi). \tag{19}$$

We will define a formula of type $o\varphi$ characterizing the topic of L :

$$Topic^{LD} := \lambda z_\varphi \bigvee_{i=1}^{m} (z_\varphi \equiv \mathrm{Int}(\mathcal{A}_i)). \tag{20}$$

The following formula expresses that z_φ is a *perception* of $x \in Form_\alpha$ with respect to the given linguistic description:

$$\begin{aligned} &Perc_{(o\varphi)\alpha} := \lambda x \lambda z_\varphi \cdot \boldsymbol{\Delta} Topic^{LD} z_\varphi \,\&\, (\exists w) \boldsymbol{\Delta}(Eval\, wx\, z_\varphi \,\&\, \\ &(\forall z'_\varphi)(\boldsymbol{\Delta}(Eval\, wx\, z'_\varphi \,\&\, Topic^{LD} z'_\varphi) \Rightarrow \cdot z'_\varphi wx \prec z_\varphi wx \vee z_\varphi \prec z'_\varphi \vee z'_\varphi | z_\varphi)) \end{aligned} \tag{21}$$

The meaning of this formula is the following: the intension z_φ is a perception of $x \in Form_\alpha$ with respect to the topic of the given linguistic description if there is a possible world w such that x_α is evaluated by z_α in it, and every other z'_φ which also evaluates x_α in w is either less sharp than z_φ, or it is incomparable with it.

We will also define *local perception* in a possible world w as follows:

$$\begin{aligned} &LPerc_{(o\varphi)\alpha\omega} := \lambda w \lambda x \lambda z_\varphi \cdot \boldsymbol{\Delta} Topic^{LD} z_\varphi \,\&\, \boldsymbol{\Delta}(Eval\, wx\, z_\varphi \,\&\, \\ &(\forall z'_\varphi)(\boldsymbol{\Delta}(Eval\, wx\, z'_\varphi \,\&\, Topic^{LD} z'_\varphi) \Rightarrow \cdot z'_\varphi wx \prec z_\varphi wx \vee z_\varphi \prec z'_\varphi \vee z'_\varphi | z_\varphi)) \end{aligned} \tag{22}$$

Clearly, $T \vdash LPerc \cdot wxz_\varphi$ implies $T \vdash Perc \cdot xz_\varphi$.

Let $\mathbf{u}^0 \in Form_\alpha, \mathbf{w}^0 Form_\omega$ be formulas. We say that the intension $\mathrm{Int}(\mathcal{A})$ is a perception of $\mathbf{u}^0$, or local perception of $\mathbf{u}^0$ in $\mathbf{w}^0$ if

$$T \vdash Perc \cdot \mathbf{u}^0 \,\mathrm{Int}(\mathcal{A}), \tag{23}$$

$$T \vdash LPerc \cdot \mathbf{w}^0 \mathbf{u}^0 \,\mathrm{Int}(\mathcal{A}), \tag{24}$$

respectively. Obviously, $\mathrm{Int}(A)$ is intension of an expression which must belong to the topic of L to be a perception.

Unfortunately, we cannot assure that there is only one perception of the given element $\mathbf{u}^0$. This depends on the choice of linguistic expressions forming L and a general solution within a formal theory does not exist. However, we may confine to a certain class of linguistic expressions, namely the evaluating ones, where the uniqueness of (18) can be assured (unless the very specific cases).

The following is an easy consequence of Theorem 2.

Theorem 3. *Let L^I, T and $\mathbf{u}^0$ be as in Theorem 2. If $T \vdash Perc \cdot \mathbf{u}^0\,\mathrm{Int}(\mathcal{A}_i)$ for some $i \in \{1,\ldots,m\}$ then*

$$T \vdash (\forall w') \cdot Eval_{o(\varphi\alpha\omega)} \cdot w'\,\hat{\mathbf{y}}\,B_{\varphi,i}$$

where $\hat{\mathbf{y}} := \imath y \cdot A_{\varphi,i}\mathbf{w}^0\mathbf{u}^0 \Rightarrow B_{\varphi,i}w'y$ and $\mathbf{w}^0 \notin J(T)$ is a new constant for a possible world.

Proof. Using Lemma 1(a),(e) and (f) we derive from $T \vdash Perc \cdot \mathbf{u}^0\,\mathrm{Int}(\mathcal{A}_i)$ that

$$T \vdash Eval \cdot \mathbf{w}^0\mathbf{u}^0\ \mathrm{Int}(\mathcal{A}_i)$$

from which we conclude that $T \vdash \Upsilon A_{\varphi,j}\mathbf{w}^0\mathbf{u}^0$. Realizing that $T \vdash \mathbf{b}_i^0 \equiv A_{\varphi,i}\,\mathbf{w}^0\mathbf{u}^0$, we may now apply Theorem 2.

This theorem is a formalization of the perception-based logical deduction. According to it, if an element represented by the constant $\mathbf{u}^0$ is given and the linguistic expression $\mathcal{A}_j$ is a perception of it then we derive an element $\hat{\mathbf{y}}$ which is evaluated by the linguistic expression $\mathcal{B}_j$.

This theorem has an easy corollary concerning local perception in some possible world $\mathbf{w}^0$.

Corollary 1. *Let L^I, T, $\mathbf{u}^0$ and $\mathbf{w}^0$ be as in Theorem 2. If $T \vdash LPerc \cdot \mathbf{w}^0\mathbf{u}^0\,\mathrm{Int}(\mathcal{A}_i)$ for some $i \in \{1,\ldots,m\}$ then*

$$T \vdash (\forall w') \cdot Eval_{o(\varphi\alpha\omega)} \cdot w'\,\hat{\mathbf{y}}\,B_{\varphi,i}$$

where $\hat{\mathbf{y}} := \imath y \cdot A_{\varphi,i}\mathbf{w}^0\mathbf{u}^0 \Rightarrow B_{\varphi,i}w'y$.

3.4 Semantics of perception-based logical deduction

In this section, we will characterize the semantics of perception-based logical deduction.

Theorem 4. *Let $\mathcal{M} \models T$ be a model of T and p be an assignment to variables such that $p(w) = w^0 \in M_\omega$ and $p(x) = u^0 \in M_\epsilon$. Let*

$$\mathcal{I}_p^{\mathcal{M}}(LPerc \cdot wx\,\mathrm{Int}(\mathcal{A}_i)) = \mathbf{1}$$

for some $i \in \{1,\ldots,m\}$. Then

(a) the corresponding linguistic expression $\mathcal{A}_i$ is the sharpest among all the expressions from the topic of L (in the sense of $\prec$) with the highest membership degree $\mathcal{I}_p^{\mathcal{M}}(A_{\varphi,i})(w^0,u^0)$, or it is incomparable with them.

(b) $\mathcal{I}_p^{\mathcal{M}}(Eval_{o(\varphi\alpha\omega)} \cdot w'\hat{\mathbf{y}}B_{\varphi,i}) = \mathbf{1}$ holds for all assignments $p(w') \in M_\omega$ where

$$\mathcal{I}_p^{\mathcal{M}}(\hat{\mathbf{y}}) \in \{v \mid \mathcal{I}_p^{\mathcal{M}}(A_{\varphi,i})(w^0,u^0) \leq \mathcal{I}_p^{\mathcal{M}}(B_{\varphi,i})(p(w'),v), v \in M_\epsilon\} \qquad (25)$$

is an element obtained by a defuzzification operation $\mathcal{I}_p^{\mathcal{M}}(\iota_{\alpha(o\alpha)})$.

Proof. The proof is based on the proof of Theorem 3.

(a) Let

$$\mathcal{I}_p^{\mathcal{M}}(LPerc \cdot wx \operatorname{Int}(\mathcal{A}_i)) = \mathbf{1}.$$

Because $\mathcal{I}_p^{\mathcal{M}}(\boldsymbol{\Delta} Topic^{LD} \operatorname{Int}(\mathcal{A}_i)) = \mathbf{1}$, $\mathcal{A}_i$ clearly belongs to the topic of L .

Let $\mathcal{I}_p^{\mathcal{M}}(A_{\varphi,j})(w^0, u^0) > \mathbf{0}$ for some $\mathcal{A}_j$ from the topic of L . This means that u^0 is evaluated in the possible world w^0 by $\mathcal{A}_j$, too. But then at least one of the following relations holds true:

$$\mathcal{I}_p^{\mathcal{M}}(A_{\varphi,j})(w^0, u^0) < \mathcal{I}_p^{\mathcal{M}}(A_{\varphi,i})(w^0, u^0), \tag{26}$$

$$\mathcal{I}_p^{\mathcal{M}}(A_{\varphi,i} \prec A_{\varphi,j}) = \mathbf{1}, \tag{27}$$

$$\mathcal{I}_p^{\mathcal{M}}(A_{\varphi,i} | A_{\varphi,j}) = \mathbf{1}. \tag{28}$$

If (26) holds then the evaluation of u^0 by $\mathcal{A}_i$ is better than that by $\mathcal{A}_j$. If (26) does not hold then either (27) or (28) holds (but not both) — cf. (19). This means that $\mathcal{A}_i$ is either sharper than $\mathcal{A}_j$, or it is incomparable with it.

(b) It follows from the assumption that

$$\mathcal{I}_p^{\mathcal{M}}(A_{\varphi,i}\, wx) = \mathcal{I}_p^{\mathcal{M}}(A_{\varphi,i})(w^0, u^0) > \mathbf{0}.$$

But then

$$\mathcal{I}_p^{\mathcal{M}}(A_{\varphi,i})(w^0, u^0) \to \mathcal{I}_p^{\mathcal{M}}(B_{\varphi,i})(p(w'), v) = \mathcal{I}_p^{\mathcal{M}}(A_{\varphi,i} wx \Rightarrow B_{\varphi,i} w'y) = \mathbf{1}$$

for some $v \in M_\epsilon$ (such v must exist because of the normality assumption on $\operatorname{Int}(\mathcal{B}_i)$) which altogether means that

$$\mathcal{I}_p^{\mathcal{M}}(Eval_{o(\varphi\alpha\omega)} \cdot w'y \operatorname{Int}(\mathcal{B}_i)) = \mathbf{1}$$

for an assignment $p(y) = v$ and all assignments $p(w') \in M_\omega$. Then $\mathcal{I}_p^{\mathcal{M}}(\hat{\mathbf{y}}) = \mathcal{I}_p^{\mathcal{M}}(\imath y \cdot A_{\varphi,i}\, wx \Rightarrow B_{\varphi,i}\, w'y)$ is some element $v^0 \in M_\epsilon$ from the kernel of the fuzzy set

$$\left\{ a/v \;\middle|\; a = \mathcal{I}_p^{\mathcal{M}}(A_{\varphi,i})(w^0, u^0) \to \mathcal{I}_p^{\mathcal{M}}(B_{\varphi,i})(p(w'), v), v \in M_\epsilon \right\}$$

(this kernel is just the set (25)). Clearly, the element v^0 is obtained using the defuzzification operation $\mathcal{I}_p^{\mathcal{M}}(\iota_{\alpha(o\alpha)})$ (interpretation of the description operator).

The practical procedure of the perception-based deduction can now be described as follows. Let a linguistic description L be given and let us observe a concrete value u^0, which can be, e.g. result of some measurement in a given situation (this can be decision-making, control, observing environment, etc.). To relate the situation with the linguistic description, we must first find a model $\mathcal{M} \models T$, $\mathcal{M} = \langle (M_\alpha, =_\alpha)_{\alpha \in Types}, \mathcal{L}_\Delta \rangle$ where T is a theory raised by L . Then the situation corresponds with some possible worlds w^0 for the topic, and $\bar{w}$ for the focus of L . On the basis of Theorem 4, we take an

assignment p to variables such that $p(x) = u^0$, $p(w) = w^0$ and $p(w') = \bar{w}$ and find a local perception $\mathcal{A}_i$ of u^0 in the topic of L (if it exists). Using Theorem 4, we find in the possible world $\bar{w}$ an element $v^0 = \mathcal{I}_p^{\mathcal{M}}(\hat{\mathbf{y}})$. This is the result of the perception-based logical deduction.

Note that in case of, e.g., fuzzy control, the possible worlds w^0 and $\bar{w}$ are constant and only the observation u^0 is changed, say to u^1. Then we must change the assignment p so that $p(x) = u^1$. The rest of the procedure remains the same.

Recall that the described procedure is implemented in the software LFLC 2000. For more details, see [5, 18].

4 Conclusion

In this paper, we have discussed a logical theory of fuzzy IF-THEN rules and linguistic descriptions and formally characterized the concept of perception-based logical deduction. The formal calculus is fuzzy intensional logic which is extension of the fuzzy type theory by a special elementary type for possible worlds.

We have introduced special formulas representing meaning of linguistic expressions and fuzzy IF-THEN rules. Furthermore, we formally defined the concept of linguistic description and a perception of the given value. We proved theorems, using which it is possible to find a conclusion on the basis of the given linguistic description and observation. We have also demonstrated semantics of the perception-based deduction and proposed a practical procedure which can be used in various applications.

In the subsequent paper, we will study the formal theory of perception-based deduction for the special case when the fuzzy IF-THEN rules consist of evaluating linguistic expressions (cf. [15]).

References

1. Andrews P (2002) An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof. Kluwer, Dordrecht
2. Da Ruan and Kerre E E (eds) (2000) Fuzzy If-Then Rules in Computational Intelligence: Theory and Applications. Kluwer Academic Publishers, Boston
3. Dubois D, Prade H (1992) Gradual inference rules in approximate reasoning. Information Sciences, 61(1–2):103–122
4. Dubois D, Prade H (1992) The Semantics of Fuzzy "IF...THEN..." Rules. In: Novák, V., Ramík, J., Mareš, M., Černý, M., Nekola, J. (eds) Fuzzy Approach to Reasoning and Decision–Making. Kluwer, Dordrecht; Academia, Praha
5. Dvořák A, Habiballa H, Novák V, Pavliska V (2003) The concept of LFLC 2000 — its specificity, realization and power for applications. Computers in industry 51:269–280

6. Dvořák A, Novák V (2004) Fuzzy Logic Deduction with Crisp Observations. Soft Computing 8:256–263
7. Dvořák A, Novák V (2004) Formal Theories and Linguistic Descriptions. Fuzzy Sets and Systems 143:169–188
8. Hájek P (1998) Metamathematics of fuzzy logic. Kluwer, Dordrecht
9. Hajičová, E., Partee, B. J., Sgall, P.: Topic-Focus Articulation, Tripartite Structures, and Semantic Content. Kluwer, Dordrecht 1998.
10. Gallin D (1975) Intensional and Higher-Order Modal Logic (With Applications to Montague Semantics) North-Holland, Amsterdam
11. Materna P (1998) Concepts and Objects. Acta Philosophica Fennica 63, Helsinki
12. Novák V (2001) Antonyms and Linguistic Quantifiers in Fuzzy Logic. Fuzzy Sets and Systems 124:335–351
13. Novák V (2003) Approximation Abilities of Perception-based Logical Deduction. Proc. Third Conf. EUSFLAT 2003, University of Applied Sciences at Zittau/Goerlitz, Germany, 630–635
14. Novák, V.: From Fuzzy Type Theory to Fuzzy Intensional Logic. Proc. Third Conf. EUSFLAT 2003, University of Applied Sciences at Zittau/Goerlitz, Germany 2003, 619–623.
15. Novák V (2003) Fuzzy logic deduction with words applied to ancient sea level estimation. In: Demicco R, Klir G J (eds) Fuzzy logic in geology. Academic Press, Amsterdam, 301–336
16. Novák V (2003) Descriptions in Full Fuzzy Type Theory. Neural Network World 13:559–569
17. Novák V (to appear) On fuzzy type theory. Fuzzy Sets and Systems.
18. Novák V, Kovář J (2000) Linguistic IF-THEN Rules in Large Scale Application of Fuzzy Control. In: Da Ruan and E E. Kerre (eds) Fuzzy If-Then Rules in Computational Intelligence: Theory and Applications. Kluwer Academic Publishers, Boston, 223–241
19. Novák V, Perfilieva I (eds) (2000) Discovering World With Fuzzy Logic. Springer-Verlag, Heidelberg
20. Novák V, Perfilieva I, Močkoř J (1999) Mathematical Principles of Fuzzy Logic. Kluwer, Boston/Dordrecht
21. Perfilieva, I., Normal Forms for Fuzzy Logic Functions and Their Approximation Ability, Fuzzy Sets and Systems 124(2001), No 3, pp. 371-384..
22. Sgall P, Hajičová E, Panevová J (1986) The Meaning of the Sentence in Its Syntactic and Pragmatic Aspects. D. Reidel, Dordrecht
23. Zadeh L A (2000) Toward a Logic of Perceptions Based on Fuzzy Logic. In: [19], 4–28.
24. Zadeh L A (1973) Quantitative Fuzzy Semantics. Inf.Sci. 3:159–176

Invited Session

Towards Intelligent Decision Support Systems via Soft Computing

Session Organiser:
Janusz Kacprzyk

Fuzzy Object-Oriented Modelling with Metadata Attributes in C♯

Fernando Berzal, Juan Carlos Cubero, Nicolás Marín**, and Olga Pons

IDBIS - Intelligent DataBases and Information Systems Research Group
Department of Computer Science and Artificial Intelligence
University of Granada
Granada - 18071 - Spain
{fberzal,JC.Cubero,nicm,opc}@decsai.ugr.es

Abstract. The development of many soft computing applications asks for mechanisms which allow the representation and management of fuzziness in the object-oriented data model. This paper proposal allows the management of imprecision in the description of objects. Moreover, it responds to programmers' needs by providing an easy-to-use transparent mechanism that can be used to develop applications which deal with fuzzy information. We present a framework which enables the development of object classes with imprecise attributes in modern programming languages such as C♯. Our framework includes generic code for the comparison of fuzzily described objects and its implementation makes use of advanced features of the .NET platform, reflection and metadata attributes, in particular. Our framework makes the implementation of fuzzy user-defined classes easier and it can be used without interfering with the usual object-oriented software development tasks.

1 Introduction

Object-oriented programming techniques have progressively taken over purely procedural programming techniques. Most computer applications currently under development are expressed in terms of classes and objects. Since the appearance of the object-oriented programming paradigm, new programming languages and development platforms have emerged to assist programmers in building complex applications.

An important group of software applications can use soft computing techniques to manage real world problems where information is affected by imprecision or uncertainty. Soft computing techniques have proved to be a suitable solution for the management of imperfect information. More and more efforts are being directed towards what researchers call "computing with words". And, soft computing craves for new tools to expand its influence within object-oriented programming platforms.

** Corresponding author: Nicolás Marín, nicm@decsai.ugr.es

F. Berzal et al.: *Fuzzy Object-oriented Modelling with Metadata Attributes in C♯*, Advances in Soft Computing **2**, 253–262 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

During the last few years, some researchers have focused their efforts in this direction. Zadeh's Fuzzy Subsets Theory [1] has been used to extend data models in order to allow the representation of imperfect data. Proposals can be found for both the relational [2] and the object-oriented [3, 4] database models. Fuzzy database management tools allow the storage and retrieval of data with imprecision and uncertainty. Nevertheless, programmers have no assistance when they have to develop applications which deal with imperfect data. Therefore, an easy-to-use transparent mechanism to cope with this kind of applications is highly desirable.

In this paper, we present a framework which allows programmers to handle imprecision in the description of objects, responding to the aforementioned need. Our framework provides a whole hierarchy of built-in classes which can be transparently used to develop user-defined classes with imprecise attributes in any .NET-enabled programming language (such as C♯) and it also includes the capability of comparing fuzzily-described objects. Our framework implementation takes advantage of advanced features included in the .NET platform in order to facilitate its use. In particular, it makes extensive use of reflection and metadata attributes.

The paper is organized as follows: Section 2 is devoted to the kind of programming platforms where our proposal is applicable, describing the advanced features that allow our framework usage to be non-intrusive. Section 3 presents some notes on the use of Fuzzy Subsets theory to describe imprecise objects. Section 4 describes the framework we have developed for the .NET platform, while Section 5 presents a case study in *C♯* which illustrates our framework usage. Finally, some conclusions and guidelines for future work close our paper.

2 Object-orientation and modern programming platforms

Due to the popularity the object-oriented data paradigm enjoys, many programming platforms have appeared to develop object-oriented code. Sun's Java Platform[1] and Microsoft's .NET Platform[2] are leaders in this area. Both share some important characteristics:

- The use of an compiled intermediate language which is interpreted or compiled into machine language at runtime. Since a stack-based virtual machine is used, the development of truly portable applications is allowed. In the case of Java, you can run an application anywhere provided that you have a Java Virtual Machine for your operating system. The .NET Framework, initially targeted at Windows systems, also provides an standard common language runtime. This enables the use of .NET in non-Windows platforms and some projects are under way in this direction
- Both the Java and the .NET platforms provide an unified and extensive set of class libraries. The .NET framework allows the use of different programming languages for the development of different parts of a system, making

[1] http://java.sun.com
[2] http://www.microsoft.com/net/

software interoperability a breeze within .NET and allowing cross-language inheritance, error handling, and debugging.
– Both Java and .NET provide standardized web application programming models as part of their class libraries: JSP and servlets on the Java side, ASP.NET on the .NET side.

These two programming platforms currently lead the current software development market. In some sense, the .NET framework (with its C♯ flagship programming language [5]) has been Microsoft's response to the rise of Sun Microsystem's Java [6].

Our research is mainly focused on the development of tools which can be transparently used within current software development platforms in order to build soft computing applications.

To accomplish this task using C♯ in the .NET Platform, we have made use of the following advanced programming features:

– **Metadata definition through attributes:** Here, metadata is information about the programming structures that conform a software system (i.e. classes, methods, and fields).. Attributes are the mechanism provided by the .NET platform for adding user-defined metadata to user-defined classes.
– **Reflection:** This feature allows a program to introspect or seek information from its own structure. Software can use this introspection to inform its users or, what is more interesting, to modify its own behavior.

3 Fuzziness and object-orientation

Fuzzy Subsets Theory has been used to enhance object-orientation at many levels, allowing the representation of fuzzily-described objects and providing fuzzy extensions of conventional object-oriented features like instantiation and inheritance [3, 4]. In this paper, we focus on the use of fuzziness to represent and manage imprecise objects.

In order to deal with fuzzily-described objects in our programs, we have to accomplish the following tasks:

– Providing support for the representation of special domains which allow fuzzy attribute values with different semantics.
– Providing support for the comparison of complex objects which can be described using values of the these special domains.

For instance, consider that we want to develop some application code to deal with classrooms and students. We have two main classes, `Room` and `Student`. Every room is characterized by its *quality*, its *extension*, the *floor* where it is, and a set of *students*. Every student is, in turn, characterized by the set of attributes {*name,age,height*}. In our context, the description of the objects belonging to these classes may be imprecise due to the following reasons:

– The *quality* attribute is expressed by an imprecise label (e.g. high, regular).

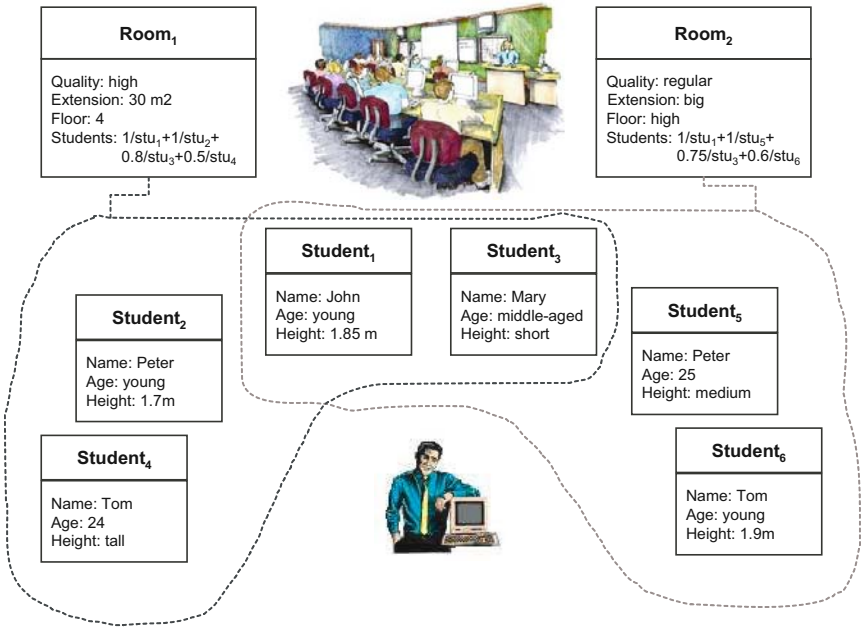


Fig. 1. Fuzzily-described rooms and students

- The *floor,extension, age,* and *height* attributes can be expressed using a numerical value or an imprecise label (e.g. high, big, young, and tall, respectively).
- The collection of *students* can also be fuzzy. The membership degree of a student in a given room can be set according to the time the student spends taking lessons in that room.

Figure 1 depicts two rooms and six students which are fuzzily described according to the previous guidelines.

With respect to the first task we mentioned at the beginning of this section, we must provide a suitable set of domains for the attributes corresponding to rooms and students. These attributes can take imprecise values. Their domains, therefore, must be defined according to the different semantic interpretations those imprecise values may have.

With respect to the second task, the complex object comparison capability, we must provide a technique for comparing fuzzily-described rooms and students. This comparison involves an aggregation problem:

- First, we have to measure the resemblance between attributes values. This step comprises the following cases:
  - Comparison of basic *crisp* attribute values.

  - Comparison of imprecise attribute values.
  - Comparison of (crisp or fuzzy) collections of objects.
  - Comparison of objects, in case the attribute values are themselves full-fledged objects.
- Then, we need to aggregate the particular resemblance values we have collected from comparing particular attributes in order to compute a global resemblance opinion for the whole objects. An importance degree can be used to give different weights to different attributes in the aggregation process. For instance, in the previous example, we can use the weights {0.5, 0.8, 1.0, 1.0} to set the relative importance of the room attributes {*quality, extension, set of students, floor*}, while the weights {1.0, 0.75, 0,75} might be sensible for student attributes {*name, age, height*}.

The implementation of this comparison process is complex and needs a recursive computation which might involve cycle resolution (i.e. for comparing rooms, student resemblance must be computed, which itself might involve the comparison of the rooms the students are associated to). A complete theoretical formulation of this object comparison problem can be found in [7], including suitable operators for the comparison situations mentioned above and the aggregation of resemblance opinions.

We provide programmers with a framework which enables the application of the ideas described above in real applications. Our fuzzy objects library simplifies the creation of special domains to represent fuzzy attribute values, as well as the implementation of user-defined classes which use those values. It also provides the infrastructure which allows the comparison of fuzzily-described objects by making use of well-known object-oriented design principles. Our framework includes all these novel capabilities and the programmer can use them without writing sophisticated code nor creating complex structures, as will be described in the following section.

4 A supporting framework in C♯

Our framework incorporates a predefined class hierarchy that supports the representation of fuzzily-described objects, as part of a more general fuzzy object-oriented model [8].

The root in this hierarchy is a generic `FuzzyObject` class which serves as a basis for any class requiring fuzzy comparison capabilities. This class implements a generic `FuzzyEquals` method which performs the fuzzy object comparison described in the previous sections. This method can be used to compare objects belonging to any subclass of `FuzzyObject`. Since the comparison it performs requires access to the particular fields of the objects being compared and, in order to be reusable, it must not be fitted to any particular class structure, the method implementation uses reflection to perform the object comparison. As we mentioned in the previous section, the comparison algorithm is recursive in order to manage complex objects and is designed so that it seamlessly deals with the cyclic structures which are common in object graphs.

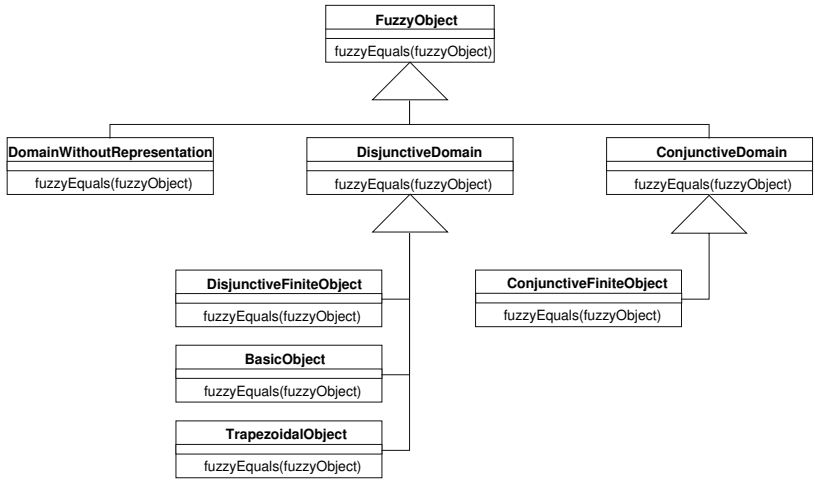


Fig. 2. Our framework class hierarchy

Below the `FuzzyObject` class in our framework class hierarchy, shown in Figure 2, other predefined classes represent common kinds of domains for imprecision, such as linguistic labels without an underlying representation (`DomainWithoutRepresentation`), domains where labels are possibility distributions over an underlying basic domain (`DisjunctiveDomain`), and fuzzy collections of fuzzy objects (`ConjunctiveDomain`). Disjunctive domains include subclasses to represent labels with finite support-set, basic domain values, and functional representations of labels with infinite support-set (e.g. trapezoidal). All these classes define their own `FuzzyEquals` method, whose implementation uses the suitable operators depending on the semantics of the domains they represent.

5 An example

A soft computing application dealing with rooms and students can be easily developed in C♯ using the framework we introduced in the previous section.

```
public class Age : AbstractDisjunctiveDomain
{
  // Redefined constructors
  public Age (int basicValue): base(basicValue) {}
  public Age (Label label, ArrayList semantic): base (label, semantic, typeof(int)) {}
  public Age (Label label, float a, float b, float c, float d): base (label, a, b, c, d) {}
}
```

Fig. 3. C♯ definition of the *Age* domain

```
public class Student : FuzzyObject
{
    // Instance variables

    private string name;
    private Age    age;
    private Height height;

    // Constructor

    public Student (String name, Age age, Height height) {
        this.name   = name;
        this.age    = age;
        this.height = height;
    }

    // Properties

    [FuzzyImportance(1.0f)]
    public string Name {
        get { return name; }
        set { name = value; }
    }

    [FuzzyImportance(0.75f)]
    public Age StudentAge {
        get { return age; }
        set { age = value; }
    }

    [FuzzyImportance(0.75f)]
    public Height StudentHeight {
        get { return height; }
        set { height = value; }
    }
}
```

Fig. 4. C♯ definition of the `Student` class

`Room` and `Student` imprecise attributes can be easily implemented just by extending the classes provided by our framework without having to worry about the `FuzzyEquals` implementation.

For instance, Figure 3 shows the code needed the `Age` domain for students. The implementation just extends the predefined `DisjunctiveDomain` class. The `Height` domain can be similarly defined. Once these domains are defined, the `Student` class can be easily implemented using `FuzzyObject` as its base class, as shown in Figure 4. As you can see, the programmer does not have to write any code for object comparison. That feature is automatically provided by the `FuzzyObject` base class.

Our framework defines a metadata attribute, `FuzzyImportance`, which can be used in accessor methods to indicate the weight of a property in the comparison process.

Once the `Student` class is defined, the programmer can create a class to represent rooms with students. The `StudentCollection` domain can be created just by extending our framework `ConjunctiveDomain`. The `Quality` attribute is represented by a class inheriting from `DomainWithoutRepresentation`, while

```
public class Room : FuzzyObject
{
    // Instance variables

    private Quality           quality;
    private Extension         extension;
    private Floor             floor;
    private StudentCollection students;

    // Constructor

    public Room (Quality quality, Extension extension,
                 Floor floor, StudentCollection students) {
        this.quality   = quality;
        this.extension = extension;
        this.floor     = floor;
        this.students  = students;
    }

    // Properties

    [FuzzyImportance(0.5f)]
    public Quality RoomQuality {
        get { return quality; }
        set { quality = value; }
    }

    [FuzzyImportance(0.8f)]
    public Extension RoomExtension {
        get { return extension; }
        set { extension = value; }
    }

    [FuzzyImportance(1.0f)]
    public Floor RoomFloor {
        get { return floor; }
        set { floor = value; }
    }

    [FuzzyImportance(1.0f)]
    public StudentCollection Students {
        get { return students; }
        set { students = value; }
    }
}
```

Fig. 5. C♯ definition of the `Room` class

```
// Students

Age young  = new Age ( new Label("young"),        0,  0, 23, 33 );
Age middle = new Age ( new Label("middle-aged"), 23, 33, 44, 48 );

Height shortHeight  = new Height ( new Label("short"),    0,   0, 150, 160);
Height mediumHeight = new Height ( new Label("medium"), 150, 160, 170, 180);
Height tall         = new Height ( new Label("tall"),   170, 180, 300, 300);

Student student1 = new Student ( "John",  young,       new Height(185) );
Student student2 = new Student ( "Peter", young,       new Height(170) );
Student student3 = new Student ( "Mary",  middle,      shortHeight );
Student student4 = new Student ( "Tom",   new Age(24), tall );
Student student5 = new Student ( "Peter", new Age(25), mediumHeight );
Student student6 = new Student ( "Tom",   young,       new Height(190) );

// Rooms

ArrayList  highSemantics = new ArrayList();
Quality highQuality   = new Quality ( new Label("high"), false, highSemantics);

ArrayList mediumSemantics  = new ArrayList();
mediumSemantics.Add ( 0.8f );
Quality mediumQuality = new Quality ( new Label("medium"), false, mediumSemantics);
Extension big  = new Extension ( new Label("big"),   21,  31, 100, 100);
ArrayList  highFloorSemantics = new ArrayList();

highFloorSemantics.Add ( new MembershipDegree (0.0f, 1 ) );
highFloorSemantics.Add ( new MembershipDegree (0.0f, 2 ) );
highFloorSemantics.Add ( new MembershipDegree (0.7f, 3 ) );
highFloorSemantics.Add ( new MembershipDegree (1.0f, 4 ) );

Floor highFloor = new Floor ( new Label("high"), highFloorSemantics );
ArrayList vector1 = new ArrayList();
vector1.Add ( new MembershipDegree (1.0f, student1 ) );
vector1.Add ( new MembershipDegree (1.0f, student2 ) );
vector1.Add ( new MembershipDegree (0.8f, student3 ) );
vector1.Add ( new MembershipDegree (0.5f, student4 ) );
StudentCollection set1 = new StudentCollection ( vector1 );

ArrayList vector2 = new ArrayList();
vector2.Add ( new MembershipDegree (1.0f, student1 ) );
vector2.Add ( new MembershipDegree (1.0f, student5 ) );
vector2.Add ( new MembershipDegree (0.75f, student3 ) );
vector2.Add ( new MembershipDegree (0.6f, student6 ) );
StudentCollection set2 = new StudentCollection ( vector2 );

Room room1 = new Room ( highQuality,   new Extension(30), new Floor(4), set1 );
Room room2 = new Room ( mediumQuality, big,               highFloor,    set2 );
```

Fig. 6. Room and student object instantiations.

`Extension` and `Floor` are `DisjunctiveDomains`. The resulting code for the `Room` class appears in Figure 5. Once more, it should be noted that the programmer does not have to write any particular code to enable the fuzzy object comparison provided by our framework. She just defines the application classes as she would do when building standard applications.

Figure 6 shows how the rooms and students depicted in Figure 1 can be created using standard C♯ code. Our framework reflective implementation automatically enables object comparison by means of the `FuzzyEquals` method. In order to compare two rooms, the user just needs to invoke this method:

```
Console.WriteLine ( "room1 fvs. room2 = " + room1.FuzzyEquals(room2) );
Console.WriteLine ( "room2 fvs. room1 = " + room2.FuzzyEquals(room1) );
```

6 Conclusions

This paper has presented a framework which allows the implementation of soft computing applications dealing with fuzzy objects in C♯. In order to maximize our framework user-friendliness, we have taken advantage of advanced capabilities provided by the .NET framework, namely reflection and metadata attributes.

Our framework fuzzy object comparison capability is based on previous theoretical work [7] and is currently being generalized to allow the use of different operators in the comparison logic, let them be predefined or even user-defined. This allowing ad hoc comparison policies at run time, depending on the execution context. Finally, we are also extending our framework with more flexible ways to express the type of a class.

References

1. Zadeh, L.A.: Fuzzy sets. Information and Control **8** (1965) 338–353
2. O. Pons, A.V., Kacprzyk, J., eds.: Knowledge management in fuzzy databases. Volume 39 of Studies in Fuzziness and Soft Computing. Physica-Verlag (2000)
3. Caluwe, R.D.: Fuzzy and Uncertain Object-Oriented Databases: Concepts and Models. Volume 13 of Advances in Fuzzy Systems-Applications and Theory. World Scientific (1997)
4. Lee, J., Kuo, J.Y., Xue, N.L.: A note on current approaches to extent fuzzy logic to object oriented modeling. International Journal of Intelligent Systems **16** (2001) 807–820
5. Heljsberg, A., Wiltamuth, S., Golde, P.: The C♯ programming language. Addison-Wesley (2003)
6. Spell, B.: Professional Java Programming. Wrox Press (2000)
7. Marín, N., Medina, J., Pons, O., Sánchez, D., Vila, M.A.: Complex object comparison in a fuzzy context. Information and Software Technology **45** (2003.) 431–444
8. Berzal, F., Marín, N., Pons, O., Vila, M.A.: Using classical object-oriented features to build a foodbs. In Lee, J., ed.: Software Engineering with Computational Intelligence. Springer Verlag (2003)

Strategies for Decision Making in the Conditions of Intuitionistic Fuzziness

Vassia Atanassova

Student in Faculty of Mathematics and Informatics, Sofia University
15 Elemag Str, Sofia 1113, Bulgaria
vassia@fmi.unu-sofia.bg

Abstract. The present article aims to explore various situations when the three coefficients in the intuitionistic fuzzy triple *(μ, ν, π)* form different majorities. Besides the detailed geometrical representation, the article presents four new extended modal operators $V_{\alpha,\beta}$, $V'_{\alpha,\beta}$, $W_{\alpha,\beta}$ and $W'_{\alpha,\beta}$ for adjusting the intuitionistic fuzzy estimations.

Keywords. Intuitionistic fuzziness, Decision making

1 Ideas of decision making in the conditions of intuitionistic fuzziness

Whenever a choice between two or more options has to be made, the subjective intention vested with pursued strategy is the major factor for decision making. We shall also take into consideration the options' nature and tools for their evaluation, since these are the objective factors that conduct the process of selection.

If we are only given the instrumentation of standard Boolean logic, we will select only these of the objects, which satisfy our criterion; the rest will fail. Correspondence to the imposed criterion is expressed by the truth-value 1 of the predicate articulating the criterion.

On a higher level of mathematical experience, where we are able to handle fuzzy objects, our choice strategy involves setting some thresholds, which represent the requirements stated in our criteria. It can be one threshold for the degree of membership only, or there can be two thresholds, respectively for the degrees of membership and non-membership. Objects here are associated with fuzzy pairs *(μ, ν)*, where $\mu + \nu = 1$, which means that μ is the degree of

V. Atanassova: *Strategies for Decision Making in the Conditions of Intuitionistic Fuzziness*, Advances in Soft Computing **2**, 263–269 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

the object's relevance to the criterion, and ν is the degree of its irrelevance. A possible strategy is to have a preference for options with their μ being higher than the first threshold, and their ν being lower than the second one. However, without restrictions, we can conclude that the essence of this sole choice strategy here is: "the higher option's degree of membership, the higher chance to choose this option".

The fact that we are handling fuzzy objects still does not change the picture at all: there are two parameters, depending on each other, whose behaviour is hence predictable.

But how to react when options are associated with intuitionistic fuzzy estimations? In this case $\mu + \nu \leq 1$, but the interesting case is $\mu + \nu < 1$, which allows a third parameter $\pi = 1 - \mu - \nu$ to appear on the scene. In the literature this parameter is called intuitionistic fuzzy index, expressing our lack of knowledge in either membership or non-membership (sort of a degree of ignorance), on the contrary to both μ and ν, which are the positive and negative coefficients of our knowledge.

At this place decision making gets more complicated. We have to consider these three factors for each option. It is obvious that sticking to the higher values of μ or to the higher values of ν is reciprocal, hence it constitutes one and the same choice strategy. Our confidence in this statement is based on the standard negation in the theory of intuitionistic fuzzy sets: $\neg(\mu, \nu) = (\nu, \mu)$. It is equivalent to taking the negation of the imposed criterion, and selecting the options with the highest degrees of non-membership, because we only deal with the degrees of our knowledge. And here comes the more interesting question: Are there any occasions when considering the higher degrees of π proves better abidance to the imposed criterion?

Let us articulate the three possible choice strategies when handling intuitionistic fuzzy options:

1. First, choose the items with largest degrees of membership. Then sift out these with the highest degrees of non-membership. The case is: $\mu > \nu > \pi$. (It is equivalent to considering the negation: $\nu > \mu > \pi$).
2. First, choose the items with largest degrees of membership. Then sift out these with the highest degrees of ignorance. The case is: $\mu > \pi > \nu$. (It is equivalent to considering the negation: $\nu > \pi > \mu$).
3. First, choose the items with largest degrees of ignorance. Then sift out these with the highest degrees of membership. The case is: $\pi > \mu > \nu$. (It is equivalent to considering the negation: $\pi > \nu > \mu$).

The visual distribution of these strategies over the intuitionistic fuzzy triangle, [1], is more than beautiful: each of the strategies corresponds to a sub-triangle formed by the sides of the intuitionistic fuzzy triangle and the three medians (see Fig. 1).

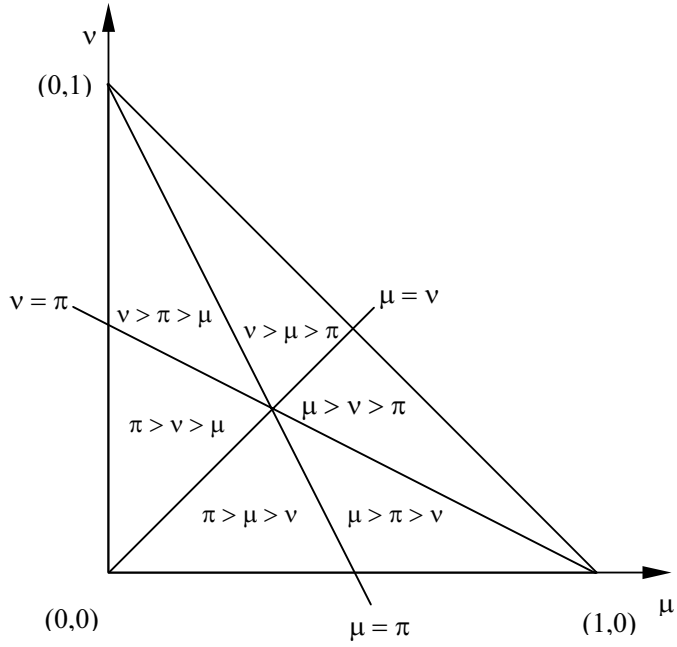


Fig. 1. Distribution of the choice strategies over the intuitionistic fuzzy triangle

2 Examples of the proposed strategies

The first of the proposed strategies is applicable when we insist on the smallest possible degree of ignorance π, which means that the sum of the two degrees describing sure, either positive or negative aspects of our knowledge, expressed via membership μ and non-membership ν shall be as large as possible. It holds in situations when preciseness is decisive.

For instance, a lecturer examines his students, assessing them with intuitionistic fuzzy scores: μ corresponds to the number of correct answers, ν – to the incorrect ones, and π corresponds to the number of hesitant or ill-formulated answers. The lecturer's strategy is determining the exact level of student's knowledge. He is not biased towards or against the student, hence

both strategies *(*$\mu > \nu > \pi$*)* and *(*$\nu > \mu > \pi$*)* hold, depending on the student's good or bad performance at the examination.

The second strategy supports high membership degree versus low non-membership degree. It is admissible that the chosen options are characterized by an average π, as long as $\mu > \pi > \nu$.

This peculiar connivance may be observed when we endeavour to retrieve as much as possible relevant information. Imagine a search engine, which retrieves Internet documents, using intuitionistic fuzzy coefficients for each document with respect to the keywords of the inquiry. Having retrieved some amount of documents, the engine shall undertake the so important task of ranking them, naturally utilizing their intuitionistic fuzzy coefficients of relevance, irrelevance and ignorance. It is not necessary to be a narrow specialist, in order to evaluate that the problem of ranking documents by relevance to some key words is fundamental in the area of information retrieval. Internet consumers often use web search engines but rarely survey all retrieved documents. Search engines are expected to to expel the fully irrelevant documents, and present the best relevant documents on the top of the retrieved list. As far as there are no irrelevant documents in the top hits, users are prone to admit documents whose intuitionistic fuzzy estimation vacillates between the degree of relevance and the degree of lack of knowledge.

The last strategy is the most peculiar one. Intuitionistic fuzzy sets have been created as a tool to account the lack of knowledge, but yet all attempts in this direction have been focused in reducing its percent, since its presence prevents from taking the most informative decisions. In defiance of these attempts, there are still many situations in real life when choosing the least informative option proves to yield the best results.

Let us consider some intricate case of medical diagnostics. A patient indicates symptoms that may correspond to various maladies requiring mutually incompatible treatments. Before all medical investigations and analyses have given their results, the doctors shall purposefully avoid any hurried diagnoses and abide to prognoses with predominant degrees of uncertainty: $\pi > \mu > \nu$.

3 Extended modal operators adjusting the IF estimations

In [1] there have been formulated the following operators, which extend the well-known modal logic operators "necessity", denoted in the theory of intuitionistic fuzzy sets by $\Box$, and "possibility", denoted by $\Diamond$:

$$\Box A = \{\langle x, \mu_A(x), 1-\mu_A(x)\rangle \mid x \in E\},$$
$$\Diamond A = \{\langle x, 1-\nu_A(x), \nu_A(x)\rangle \mid x \in E\}.$$

The extended modal operators have respectively the following formula representation:

$$F_{\alpha,\beta}(A) = \{\langle x, \mu_A(x) + \alpha.\pi_A(x), \nu_A(x) + \beta.\pi_A(x)\rangle \mid x \in E\},$$
$$G_{\alpha,\beta}(A) = \{\langle x, \alpha.\mu_A(x), \beta.\nu_A(x)\rangle \mid x \in E\},$$
$$H_{\alpha,\beta}(A) = \{\langle x, \alpha.\mu_A(x), \nu_A(x) + \beta.\pi_A(x)\rangle \mid x \in E\},$$
$$H^*_{\alpha,\beta}(A) = \{\langle x, \alpha.\mu_A(x), \nu_A(x) + \beta.(1-\alpha.\mu_A(x) - \nu_A(x))\rangle \mid x \in E\},$$
$$J_{\alpha,\beta}(A) = \{\langle x, \mu_A(x) + \alpha.\pi_A(x), \beta.\nu_A(x)\rangle \mid x \in E\},$$
$$J^*_{\alpha,\beta}(A) = \{\langle x, \mu_A(x) + \alpha.(1-\mu_A(x) - \beta.\nu_A(x)), \beta.\nu_A(x)\rangle \mid x \in E\},$$

where A is a given intuitionistic fuzzy set, E is the universe, $\alpha, \beta \in [0,1]$ and only in $F_{\alpha,\beta}$, $\alpha + \beta \le 1$. Operators $F_{\alpha,\beta}$, $G_{\alpha,\beta}$, $H_{\alpha,\beta}$, $H^*_{\alpha,\beta}$, $J_{\alpha,\beta}$ and $J^*_{\alpha,\beta}$ have been illustrated in Figs. 2a and 2b.

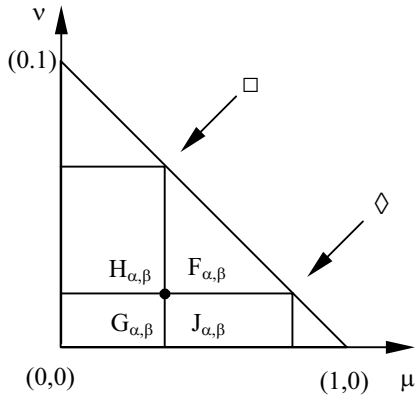


Fig. 2a. Operators $F_{\alpha,\beta}$, $G_{\alpha,\beta}$, $H_{\alpha,\beta}$, $J_{\alpha,\beta}$

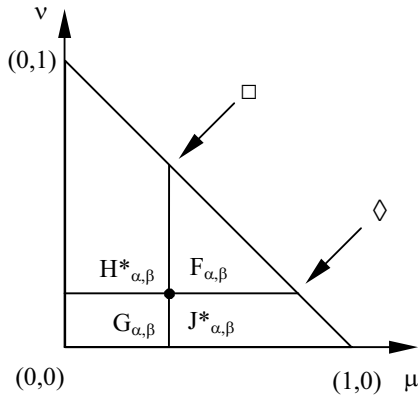


Fig. 2b. Operators $F_{\alpha,\beta}$, $G_{\alpha,\beta}$, $H^*_{\alpha,\beta}$, $J^*_{\alpha,\beta}$

The appearance of these extended modal operators in 1988 coincided in time with the occurrence of the geometrical interpretation of intuitionistic fuzzy sets as a triangle, which contributed to the operators' better understanding and gave rise to the invention of other extended modal operators and related properties.

The modal operators presented above can be used for external intervention, somewhat adjustment of the objects' intuitionistic fuzzy estimations, due to the subjective intention expressed in the pursued strategy. But these are not the

only operators that realize these functionalities. The rest of the present article will be devoted to introduction of four brand new extended modal operators that also enable the movement of the point inside the intuitionistic fuzzy triangle, and explain the motives for this adjustment.

First of all, let us observe that there is one more possible splitting of the intuitionistic fuzzy triangle. When the idea for this new splitting appeared, it was initiated by the possible strategies that shift the point into the triangle. Depending on which of the intuitionistic fuzzy degrees weighs down (or up), there are six figures that describe the regions of action of the new modal operators. They correspond to the three mirroring choice strategies, articulated in the first section of the present article, and are illustrated in Fig. 3.

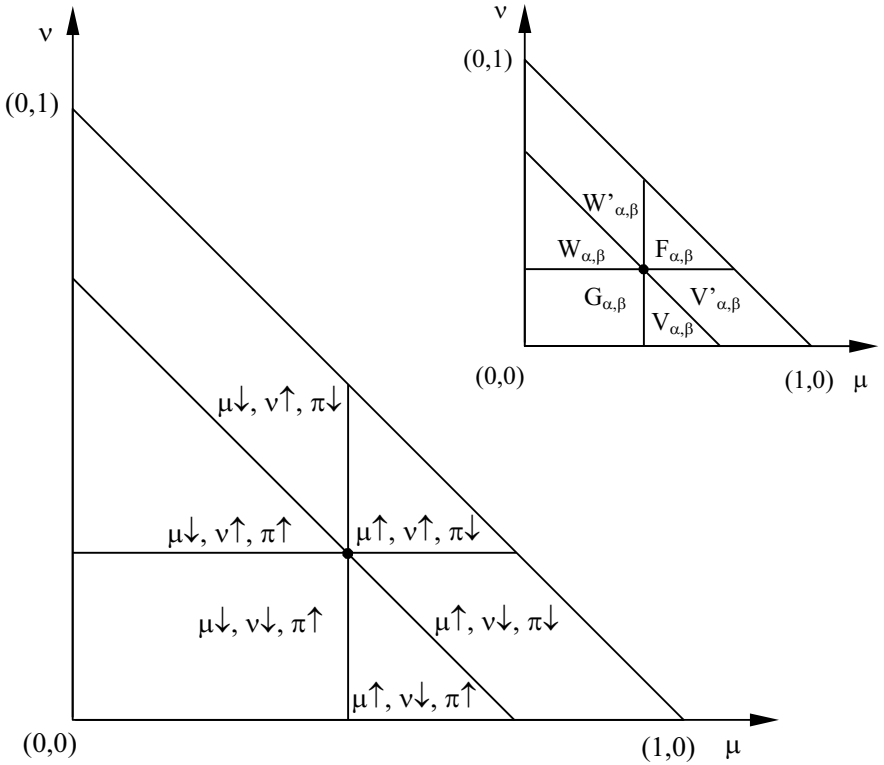


Fig. 3. Distribution of the new extended modal operators $V_{\alpha,\beta}$, $V'_{\alpha,\beta}$, $W_{\alpha,\beta}$ and $W'_{\alpha,\beta}$ over the intuitionistic fuzzy triangle

The formulae that correspond to each of the four modal operators are given below (operators $F_{\alpha,\beta}$,$G_{\alpha,\beta}$ have already been formulated):

$$V_{\alpha,\beta}(A) = \{\langle x, \mu_A(x) + \alpha.\nu_A(x), \beta.\nu_A(x)\rangle \mid x \in E\},$$
$$V'_{\alpha,\beta}(A) = \{\langle x, \mu_A(x) + \alpha.\pi_A(x) + (1-\beta).\nu_A(x), \beta.\nu_A(x)\rangle \mid x \in E\},$$
$$W_{\alpha,\beta}(A) = \{\langle x, \alpha.\mu_A(x), \nu_A(x) + \beta.\mu_A(x)\rangle \mid x \in E\},$$
$$W'_{\alpha,\beta}(A) = \{\langle x, \alpha.\mu_A(x), \nu_A(x) + \beta.\pi_A(x) + (1-\alpha).\mu_A(x)\rangle \mid x \in E\}.$$

It is easy to prove that Morgan's Law holds for these operators:

$$\neg V_{\alpha,\beta}(\neg A) = W_{\beta,\alpha}(A), \qquad \neg V'_{\alpha,\beta}(\neg A) = W'_{\beta,\alpha}(A),$$
$$\neg W_{\alpha,\beta}(\neg A) = V_{\beta,\alpha}(A), \qquad \neg W'_{\alpha,\beta}(\neg A) = V'_{\beta,\alpha}(A).$$

The following property, relating all of the four new operators, is also valid:

$$W'_{\alpha,\beta}(A) \subset W_{\alpha,\beta}(A) \subset A \subset V_{\alpha,\beta}(A) \subset V'_{\alpha,\beta}(A).$$

In a future research, there shall be studied the interrelations between operators the old operators $H_{\alpha,\beta}$, $H^*_{\alpha,\beta}$, $J_{\alpha,\beta}$ and $J^*_{\alpha,\beta}$, and the new ones $V_{\alpha,\beta}$, $V'_{\alpha,\beta}$, $W_{\alpha,\beta}$ and $W'_{\alpha,\beta}$, as far as their graphic representation over the intuitionistic fuzzy triangle reports that their regions of action overlay.

It is easy to prove that Morgan's Law holds for these operators:

$$\neg V_{\alpha,\beta}(\neg A) = W_{\beta,\alpha}(A), \qquad \neg V'_{\alpha,\beta}(\neg A) = W'_{\beta,\alpha}(A),$$
$$\neg W_{\alpha,\beta}(\neg A) = V_{\beta,\alpha}(A), \qquad \neg W'_{\alpha,\beta}(\neg A) = V'_{\beta,\alpha}(A).$$

The following property, relating all of the four new operators, is also valid:

$$W'_{\alpha,\beta}(A) \subset W_{\alpha,\beta}(A) \subset A \subset V_{\alpha,\beta}(A) \subset V'_{\alpha,\beta}(A).$$

In a future research, there shall be studied the interrelations between operators the 'old' operators $H_{\alpha,\beta}$, $H^*_{\alpha,\beta}$, $J_{\alpha,\beta}$ and $J^*_{\alpha,\beta}$, and the 'new' ones $V_{\alpha,\beta}$, $V'_{\alpha,\beta}$, $W_{\alpha,\beta}$ and $W'_{\alpha,\beta}$, as far as their graphic representation over the intuitionistic fuzzy triangle reports that their regions of action overlay.

References

[1] Atanassov K (1999) Intuitionistic fuzzy sets. Physica-Verlag Heidelberg

Fuzzy Linguistic Summaries in Text Categorization for Human-Consistent Document-Driven Decision Support Systems

Janusz Kacprzyk[*,**] and Sławomir Zadrożny[*]

[*]Systems Research Institute, Polish Academy of Sciences
ul. Newelska 6, 01-447 Warsaw, Poland
[**]Warsaw School of Information Technology (WSISiZ)
ul. Newelska 6, 01-447 Warsaw, Poland
{kacprzyk, zadrozny}@ibspan.waw.pl

Abstract. The paper concerns one of relevant issues related to the handling of textual information, that is the dominant form of information in many real world problems, for providing support for decision making. We discuss the issue of text document categorization that is a prerequisite for further analyses. We indicate how the use of fuzzy linguistic summaries for text categorization may help the decision maker to have documents classified in a human consistent way into categories, which in turn should greatly help him or her extract relevant information and knowledge from textual documents available, and then use them to arrive at a better decision in a more effective and efficient way. We indicate that the solutions proposed can be of use for enhancing the power of so-called document driven decision support systems.

1. Introduction

Decision making is now becoming more and more sophisticated, time consuming and difficult for human beings who require some decision support, notably in the form of a computer based decision support system (DSSs). DSSs are heavily based on data, information and knowledge. Their importance is growing in recent times in particular in view of a transition to knowledge centered economy, that is a dominant tendency in virtually all developed countries, and a need of a better knowledge management.

We can distinguish the following basic types of DSSs:

- Data driven,
- Communication driven and group DSSs,

[1] Research supported by KBN Grant 4 T11F 012 25.

J. Kacprzyk and S. Zadrożny: *Fuzzy Linguistic Summaries in Text Categorization for Human-Consistent Document-Driven Decision Support Systems*, Advances in Soft Computing **2**, 271–279 (2005)
www.springerlink.com © Springer-Verlag Berlin Heidelberg 2005

- Document driven,
- Model driven,
- Knowledge driven,
- Web based and interorganizational.

Roughly speaking:

- Data Driven DSSs emphasize access to and manipulation of internal company data and sometimes external data, and may be based –from the low to high level – first on simple file systems with query and retrieval tools, then data warehouses, and finally with On-line Analytical Processing (OLAP) or data mining tools.
- Communications Driven DSSs use network and communications technologies to facilitate collaboration and communication.
- Group GDSSs are interactive, computer-based systems that facilitate solution of unstructured problems by a set of decision-makers working together as a group.
- Document Driven DSSs integrate a variety of storage and processing technologies for a complete document retrieval and analysis; documents may contain numbers, text, multimedia.
- Model Driven DSSs emphasize access to and manipulation of a model, e.g., statistical, financial, optimization and/or simulation; use data and parameters, but are not usually data intensive.
- Knowledge Driven DSSs are interactive systems with specialized problem-solving expertise consisting of knowledge about a particular domain, understanding of problems within that domain, and "skill" at solving some of these problems.
- Web based DSSs are computerized system that deliver decision support related information and/or tools to a manager/analyst using a "thin-client" Web browser (Explorer); TCP/IP protocol, etc.

and one should bear in mind that this classification should not be considered as a chronology of development of DSSs but as a classification with respect to what a particular system is meant for, as well as which aspect of decision support it focuses on.

In this paper we concentrate on the document driven DSSs, and in particular on-how to effectively and efficiently retrieve, from various documents available in the company or organization, information and knowledge that may be of use to the decision maker while making decisions. We concentrate on the problem of how to categorize (classify) text documents, and to attain a higher human consistency. We propose the use of fuzzy linguistic summaries (cf. Yager, 1982, Kacprzyk and Yager, 2001, Kacprzyk, Yager and Zadrozny, 2000, 2001) dealt with by fuzzy logic with linguistic quantifiers.

2. Linguistic Summarization

Text categorization is one of the tasks considered within information retrieval (IR). It may be meant in a more general context - not necessarily limited to a pure classification problem. For example, a concise description of a collection of text documents may be sought within the same framework. On a more sophisticated, semantic level this is related to information extraction and text summarization. As soon as we assume some numerical representation of documents, e.g., based on the widely accepted vector space model, we may try to apply quantitative techniques of data mining. In respect to a concise description of a set of objects (here: documents) such a relatively new and promising technique is the linguistic summarization. This paper discusses some possibilities of application of this new technique for the purposes of information retrieval.

The main idea of linguistic summarization is to provide means for an intuitive, human-consistent description of a group of objects. A linguistic summary may be exemplified in the context of a corporate database by:

"*Many* orders have low commisssion" (1)

"*Most* of the young employees have high salary" (2)

The concept has been introduced by Yager (cf., e.g., [14]) and further developed by many authors including Kacprzyk and Zadrożny [8,9]. The elements emphasized in the examples (1) and (2) are linguistic terms typically used by humans to assess values of some features ("low", "young", "high") and quantify some cardinalities ("many", "most"). Thus, formally, such linguistic summaries may be conveniently expressed using the language of the linguistically quantified propositions due to Zadeh [16], as follows:

$$Qy\text{'s are } S \tag{3}$$

for (1) and the following for (2)

$$QRy\text{'s are } S \tag{4}$$

In Yager's approach the following notation related to linguistic summaries is assumed. $Y=\{y_1,\ldots,y_n\}$ is a set of objects to be summarized, e.g. the set of workers. $A=\{A_1,\ldots, A_m\}$ is a set of attributes characterizing the objects from Y, e.g. salary, age etc. $A_j(y_i)$ denotes value of attribute A_j for object y_i. A linguistic summary of data set Y consists of:

- a quantity in agreement Q, i.e., a linguistic quantifier (e.g. "many"),
- a summarizer S, i.e., an attribute together with a linguistic term defined on domain of an attribute A_j (e.g. "low commission" for attribute "commission"),

- optionally, a qualifier *R*, i.e., another attribute together with a linguistic term defined on domain of an attribute A_k determining a (fuzzy) subset of *Y* (e.g. "young" for attribute "age")
- truth (validity) *T* of the summary, i.e., a number from the interval [0,1] assessing truth (validity) of the summary (e.g. 0.7),

and may be, therefore, represented as a quadruple (*Q, S, R, T*).

Using Zadeh's [16] fuzzy-logic-based calculus of linguistically quantified propositions, a (proportional, nondecreasing) linguistic quantifier *Q* is assumed to be a fuzzy set in the interval [0,1] as, e.g.

$$\mu_Q(x) = \begin{cases} 1 & \text{for } x \geq 0.8 \\ 2x - 0.6 & \text{for } 0.3 < x < 0.8 \\ 0 & \text{for } x \leq 0.3 \end{cases} \tag{5}$$

Then, the truth (validity) of (3) and (4) are calculated, respectively, as

$$\text{truth}(Qy\text{'}s \text{ are } S) = \mu_Q[\tfrac{1}{n}\textstyle\sum_{i=1}^{n} \mu_S(y_i)]$$

$$\text{truth}(QRy\text{'}s \text{ are } S) = \mu_Q[\textstyle\sum_{i=1}^{n} (\mu_R(y_i) \wedge \mu_S(y_i)) / \sum_{i=1}^{n} \mu_R(y_i)]$$

Both the summarizer (*S*) and qualifier (*R*) are assumed above as referring to just one attribute. They can be extended to cover more sophisticated summaries involving some confluence of various attribute values as, e.g, "young and well paid".

Kacprzyk and Zadrożny [10] review a number of approaches proposed in the literature for the derivation of linguistic summaries in a more or less general form. They include the use of efficient algorithms for association rules mining. It seems to be quite a promising direction. It makes it possible to derive linguistic summaries in a slightly simplified form (basically, both qualifiers and summarizers have to be conjunctions of atomic conditions), however it offers efficient algorithms elaborated in the area of traditional association rule mining.

Briefly stating, in this approach an interpretation of the following generalized and fuzzified form of an association rule:

$$A_1 \text{ IS } f_1 \wedge ... \wedge A_n \text{ IS } f_n \rightarrow A_{n+1} \text{ IS } f_{n+1} \wedge ... \wedge A_{n+m} \text{ IS } f_{n+m} \tag{6}$$

as a linguistic summary of type (4) represented by the following quadruple:

$$(Q, A_{n+1} \text{ IS } f_{n+1} \wedge ... \wedge A_{n+m} \text{ IS } f_{n+m}, A_1 \text{ IS } f_1 \wedge ... \wedge A_n \text{ IS } f_n, T)$$

is employed. In the above, f_i's denote linguistic terms defined in the domain of attributes A_i, while *Q* and *T* are determined by a so-called confidence measure of the association rule (6). For more details on this approach, see [10] and references therein.

The number of derived linguistic summaries, may be huge, and hence some prunning scheme should be used to eliminate unnecessary summaries. This leads to a substantial, lossless reduction of the number of rules.

3. Text categorization

The task of text categorization is a special case of the general problem of classification. Usually, the following or similar notation is assumed: $D = \{d_i\}_{i=1,N}$ - a set of text documents, $C = \{c_i\}_{i=1,S}$ - a set of categories, Ξ: $D \times C \rightarrow \{0,1\}$ - assignment of categories to documents, $T = \{t_j\}_{j=1,M}$ - a set of terms. Additionally, a set of training documents is considered, i.e., such a set $D_1 \subset D$ that $\Xi(d,c)$ is known for $d \in D_1$ and any $c \in C$. Then the documents are usually represented via a function:

$$F: D \times T \rightarrow [0, 1] \tag{7}$$

i.e. a document is represented as a vector: $d_i \rightarrow [w_1,...,w_M]$, $w_j = F(d_i,t_j)$ $d_i \in [0,1]^M$, where each dimension corresponds to a term and the value of w_j (weight) determines to what extent a term $t_j \in T$ is important for the description of the document. A popular version of function F used for the automatic processing of documents is a *tf×idf* function $F(d_i,t_j)=f_{ij} * \log(N/n_j)$ where f_{ij} is the frequency of a term t_j in a document d_i and n_j is a number of documents containing term t_j. As a starting point, in this work we assume a normalized version of *tf×idf*.

Having a numerical representation of documents (7) any algorithm of classifier construction may be applied, including rule-based systems, decision trees, artificial neural networks, etc, cf., e.g., [11]. The common feature of such algorithms is a need to aggregate partial results obtained when particular terms are taken into account separately. This provides an opportunity to apply linguistic quantification, especially in cases when transparency, easy interpretability of obtained rules is required. For example, in [19] we have employed the classical Rocchio algorithm developed in the area of IR. The learning phase consists in computing a centroid vector for each category of documents. Then, in the classification phase, a document is classified to a category whose centroid is most similar to this document. The similarity may be meant in several ways – in the original Rocchio's approach it corresponds to the Euclidean distance.

As the categories (their centroids) represent many documents, then one should not expect a match between a centroid and a document along all dimensions of their representation. More reasonable is to formulate a requirement that along most of these dimensions there is a match. This may be formalized using the following linguistically quantified proposition:

> "A document belongs to a category if most of the important terms present in the document are also present in the centroid of the category".

Another aspect of text categorization pertinent to the application of linguistic quantifiers is the following. Usually the classifiers considered in a given context compute for each document and category a matching degree that yields an ordered list of categories for each document. If just one category is to be assigned to a document a reasonable way is to choose one with the highest rank.

However, in case of multilabel categorization the classifier has to decide how many of top ranked categories should be assigned to a document. This is referred to as a thresholding strategy problem (cf.., e.g., [15]). The following strategies may be adopted: choosing a fixed number of top ranked categories for each document; assigning such a number of documents to each category so as to preserve a proportion of the cardinalities of particular categories in the training set; or assigning a category only if its matching score is higher than a fixed threshold. In [19,18] we proposed a number of approaches based on linguistic quantification. One consists in choosing such a threshold r that

> "most of the important categories had a number of sibling categories similar to r in the training data set"

and by the sibling category for a category c_i we mean a category that is assigned to the same document as category c_i.

Another approach may be expressed as follows: select such a threshold r that:

> "most of the important categories are selected and most of the selected categories are important"

A threshold r is selected for each document d separately.

4. Linguistic summaries for text categorization

It is easy to see that the above problem formulations correspond to linguistic summaries that should be mined. Linguistic summaries were originally meant in the context of databases. Databases usually feature strictly determined structure with clearly identified attributes with their domains. Thus, summaries are well-defined knowing the schema with additional metadata in the form of a dictionary of relevant linguistic terms (cf., e.g., [8,9]). Text documents usually lack such a strict structure and thus are much less suitable for standard mining techniques, including linguistic summaries. However, as we focus here on the text documents available in the Internet, the situation is more promising. This is due to the fact, that most of the Internet based text documents reveal some structure. The most popular format of the Web is HTML which secures a certain degree of structure of compliant documents (especially in case of XHTML). More and more documents available and exchanged via Internet follow XML specification, which supports quite a rich and well defined structure. These makes it possible to distinguish dif-

ferent parts of documents that may be important for the purposes of text categorization.

In our work we assume the model of Internet-based documents as proposed by Bordogna and Pasi (cf. e.g., [4]). Basically, a document is assumed to be divided into sections (parts). Depending on the application area one may assume a more or less rich structure. For example, for a set of XML documents based on the same schema a very rich structure may be taken into account. For typical HTML documents at least TITLE and BODY sections are usually present. Let $P=\{p_k\}_{k\in[1,K]}$ a set of sections (parts) distinguished in documents. Then, a document is represented as the vector $d_i=(d_{i11},...,d_{iMN})$, where d_{ikj} denotes the weight of term t_j in section k of document d_i. Thus, the weights d_{ikj} are computed by a function (instead of (7)):

$$F: D \times P \times T \rightarrow [0, 1] \tag{8}$$

Thus, we convert a set of the original textual documents into the vectors of terms weights belonging to the interval [0,1]. These may be accompanied by the information on category (class) belonginess of given document, if available. In order to reduce the term set we perform typical operations of stopwords elimination and stemming. What we obtain is, formally, a counterpart of a set of numerical data typically dealt with using data mining techniques, including linguistic summarization.

Depending on the goal of the linguistic summarization we derive a set of linguistic summaries for a set of documents representing one category or various categories. The former may be useful in case of simple filtering of information, while the latter for a regular categorization of documents. Linguistic summaries derived in the former scenario may be also useful per se – as a description of given set of documents, not necessarily assigned to a specific category. Due to their human consistency and readability, linguistic summaries may be easier manipulated and handled by a human expert leading possibly to even better categorization rules.

In this paper, we discuss an application based on vector space model whereas documents are represented as vectors of keywords. However, even more attractive from the point of view of manual manipulations of description of categories is a representation referring to the concepts dealt with in a document. In such a case, linguistic summaries may take into account hierarchies of concepts.

Numerical experiments on some well-known text corpora are encouraging, and will be presented in next papers along with a more detailed analysis of algorithmic and programming solutions adopted in the study.

4. Concluding remarks

The paper concerned one of relevant issues related to the handling of textual information, that is the dominant form of information in many real world problems, for providing support for decision making. We discussed the issue of text document categorization that is a prerequisite for further analyses. This provides for an effective and efficient management of knowledge that is present in textual documents. Such documents are a primary form of knowledge sources in many practical situations.

We indicated how the use of fuzzy linguistic summaries for text categorization may help the decision maker to have documents classified in a human consistent way into categories, which in turn should greatly help him or her extract relevant information and knowledge from textual documents available, and then use them to arrive at a better decision in a more effective and efficient way.

References

[1] M.-L. Antonie, O. R. Zaïane (2002). Text Document Categorization by Term Association, in Proc. of the IEEE 2002 International Conference on Data Mining (ICDM'2002), pp 19-26, Maebashi City, Japan, December 9 - 12, 2002

[2] Boley D. et al. (1999). Document categorization and query generation on the World Wide Web using WebACE. AI Review, vol. 13, N. 5-6, pp. 365-391.

[3] G. Bordogna and G. Pasi (2000). Flexible representation and querying of heterogeneous structured documents. Kybernetika, 36(6): 617-633.

[4] G. Bordogna and G. Pasi (2003). Flexible representation and retrieval of Web documents. In P.S. Szczepaniak, J. Segovia, J. Kacprzyk and L.A. Zadeh, eds. Intelligent Exploration of the Web, pp. 38-53, Springer-Varlag.

[5] G. Chen, Q. Wei and E. Kerre (2000). Fuzzy data mining: discovery of fuzzy generalized association rules. In G. Bordogna and G. Pasi, editors Recent Issues on Fuzzy Databases, pages 45—66. Springer-Verlag.

[6] R. Feldman and H. Hirsh (1996). Mining associations in text in the presence of background knowledge. In Proc. Of the Second International Conference on Knowledge Discovery from Databases.

[7] Y.-Ch. Hu, R.-Sh. Chen and G.-H. Tzeng. (2002). Mining fuzzy association rules for classification problems. Computers & Industrial Engineering, 43:735–750.

[8] J. Kacprzyk and R.R. Yager (2001). Linguistic summaries of data using fuzzy logic. International Journal of General Systems, 30: 33-154.

[9] J. Kacprzyk, R.R. Yager R. and S. Zadrożny (2000). A fuzzy logic based approach to linguistic summaries of databases. International Journal of Applied Mathematics and Computer Science, 10, 4: 813-834.

[10]J. Kacprzyk and S. Zadrożny (2004). Fuzzy linguistic data summaries as a human consistent, user adaptable solution to data mining. In B. Gabrys, K. Leiviska and J. Strackeljan, editors, Do smart adaptive systems exist? – Best practice for selection and combination of intelligent methods, Springer, Heidelberg and New York, to appear.
[11]Kuncheva, L.I. (2000). Fuzzy Classifier Design. Physica-Verlag, Heidelberg New York.
[12]B. Liu, W. Hsu and Y.M. Ma (1998). Integrating classification and association rule mining. In Proc. of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98), pp. 80-86, New York, USA.
[13]F. Sebastiani (1999) A tutorial on automated text categorisation In A. Amandi, A. Zunino (Eds.) Proceedings of ASAI-99, 1st Argentinian Symposium on Artificial Intelligence, Buenos Aires, AR, 7-35.
[14]R.R. Yager (1996). Database discovery using fuzzy sets. International Journal of Intelligent Systems, 691-712.
[15]Yang Y. (1999). An Evaluation of Statistical Approaches to Text Categorization. Information Retrieval, vol. 1, No. 1 / 2, pp. 69-90.
[16]L.A. Zadeh (1983). A computational approach to fuzzy quantifiers in natural languages. Computers and Maths with Appls. 9: 149—184.
[17]S. Zadrożny and J. Kacprzyk (2003). On the application of linguistic quantifiers for text categorization. In Proceedings of International Conference on Fuzzy Information Processing, volume 1, 435-440, Beijing.
[18]S. Zadrożny and J. Kacprzyk (2003). Linguistically quantified thresholding strategies for text categorization. In Proc. of the Third International Conference in Fuzzy Logic and Technology (EUSFLAT'2003), pages 38-42, Zittau, Germany, September 10-12, 2003.
[19]S. Zadrożny and J. Kacprzyk (2003) Computing with words for text processing: an approach to the text categorization. Information Sciences, to appear.

An Application of Intuitionistic Fuzzy Relational Databases in Football Match Result Predictions

Boyan Kolev[1], Panagiotis Chountas[2], Ilias Petrounias[3], Vassilis Kodogiannis[2]

[1] Centre for Biomedical Engineering- Bulgarian Academy of Sciences, Acad.G.Bonchev Str., Bl.105, Sofia-1113, BULGARIA
bik@clbme.bas.bg
[2] Health Care Computing Group, School of Computer Science, University of Westminster Watford Road, Northwick Park, London, HA1 3TP, UK
{P.I.Chountas,V.Kodogiannis}@westminster.ac.uk
[3] Department of Computation, UMIST PO Box 88, Manchester M60 1QD, UK
ilias@co.umist.ac.uk

Abstract. This paper presents a model for prediction of football league matches. We use intuitionistic fuzzy SQL (IFSQL) and intuitionistic fuzzy relational databases (IFRDB) to store and manage data about football matches and predictions. We take advantage of intuitionistic fuzzy sets by relating the degree of indefiniteness to an uncertainty about the estimation for a team's capabilities. The uncertainty is produced by insufficient statistical data.

1. Introduction

In [1, 3] we presented an IFRDB model and described the IFSQL used for querying in intuitionistic fuzzy databases. The IFRDB model extends the classical and fuzzy relational database models [7, 8] by introducing the usage of intuitionistic fuzzy sets. In this paper we present an application of this model, which develops a system for prediction of football matches, which we introduced in [2]. The system computes and stores ratings for the teams, which are similar to the Elo ratings [4]. The system takes advantage of the theory of the intuitionistic fuzzy sets [5, 6] in the following way: At a certain moment each team has a rating which is the answer of the

B. Kolev et al.: *An Application of Intuitionistic Fuzzy Relational Databases in Football Match Result Predictions*, Advances in Soft Computing **2**, 281–289 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

question “Is the team good?” The rating is an intuitionistic fuzzy value with degree of indefiniteness, which corresponds to the uncertainty about our estimation for the team’s capabilities. This uncertainty decreases as we get more statistical data for the team. Initially the indefiniteness is equal to 1 as we don’t know anything about the team. The system adjusts a team’s rating after each match following these steps:

- computes the win expectancy using the ratings of the two teams
- evaluates the match result with a number in the interval [0, 1]
- computes the new ratings taking into account the difference between the expectancy and the result. The degrees of truth and falsity of the new rating are modified considering the a.m. difference but in a way that decreases the degree of indefiniteness.

The paper emphasizes on the advantage of using the intuitionistic fuzzy sets theory in game prediction models and proposes a concrete model for prediction of football league matches, described in details in the next chapter.

In Chapter 3 we describe the structure and functionality of the IFRDB, which implements the football prediction model. The database uses IFSQL statements to satisfy the needs of the prediction model.

2. Description of the prediction model

Let us define an alternative form of an intuitionistic fuzzy value, which represents the value with a couple of numbers <σ, τ> in the interval [0, 1], where σ represents the truth degree (do not confuse it with μ) and τ represents the definiteness. In other words, σ is the degree of truth in the terms of classical fuzzy sets (falsity is 1-σ) and τ answers the question “at what degree our estimation for σ is correct”. The formulae for conversion between <μ, ν> and <σ, τ> representations are the following:

<μ, ν> to <σ, τ>	<σ, τ> to <μ, ν>
$\sigma = \begin{cases} \dfrac{\mu}{\mu+\nu}, if\ \mu+\nu>0 \\ 0.5, if\ \mu+\nu=0 \end{cases}$ $\tau = \mu+\nu$	$\mu = \sigma\tau$ $\nu = (1-\sigma)\tau$

2.1. Prerequisites

R_0^h is the current rating of the home team. R_0^a is the current rating of the away team.

ΔR (see Eq. 1) is the difference between ratings of the home and away teams. The coefficient C_H is used for adjustment of the strength of the home team rating in the computation of win expectancy. It artificially increases the rating of the home team. It may depend on the championship or even on the certain match and is usually greater than 1.

$$\Delta R = \sigma(R_0^h)^{\frac{1}{C_H}} - \sigma(R_0^a) \tag{1}$$

2.2. Computing the win expectancy

The win expectancy w_e is an intuitionistic fuzzy value, which answers the question "Do we expect the home team to win?". It is computed by the following formulae:

$$\sigma(w_e) = 0.5 + \frac{1}{2} sg(\Delta R) \mid \Delta R \mid^{\frac{1}{C_E}}$$

$$\tau(w_e) = \frac{\tau(R_0^h) + \tau(R_0^a)}{2},$$

where sg (x) is a sign function (see Eq. 2) and C_E is a coefficient for adjustment of the strength of the rating difference in the computation of w_e.

$$sg(x) = \begin{cases} 1, if\ x \geq 0 \\ -1, if\ x < 0 \end{cases} \tag{2}$$

The win expectancy can be used for prediction, i.e. the system claims that the match result (see below) will be most probably in the interval $[\mu(w_e), 1-\nu(w_e)]$.

2.3. Evaluating the match result

This consists of computing a number *w* in the interval [0, 1], which represents the result of the match. Consider the scores of the home and away teams, respectively S_H and S_A.

Let $\Delta S = S_H - S_A$ be the goal difference. Then:

$$w = \begin{cases} \dfrac{1}{-\Delta S + 2}, \text{if } \Delta S < 0 \\ 0.5, \Delta S = 0 \\ 1 - \dfrac{1}{\Delta S + 2}, \text{if } \Delta S > 0 \end{cases}$$

Here are some examples for the computation of this parameter:

ΔS	w
-3	0.2
-2	0.25
-1	0.333
0	0.5
1	0.667
2	0.75
3	0.8

2.4. Computing the new ratings

For the new ratings we will take into account the difference between the result and the expectancy:

$$\Delta w = w - \sigma(w_e)$$

Now we will compute the modification parameters $\Delta\mu$ and $\Delta\nu$, which will be added respectively to the degrees of truth and falsity of the home team rating. They will also be used for modification of the away team rating but in reverse order, i.e. $\Delta\nu$ will be added to the degree of truth and $\Delta\mu$ - to the degree of falsity. We must make sure that the following condition is observed:

$$\Delta\mu + \Delta\nu = \Delta\pi,$$

where $\Delta\pi$ is a constant representing the decrease of the degree of indefiniteness of the ratings after each match. It depends on the certain championship. For example, if the league consists of 20 teams, each team must play 38 matches, so $\Delta\pi$ must not be greater than 1/38 otherwise the sum of truth and falsity of each team after the last match will be greater than 1 (assuming that all the teams begin the championship with full indefiniteness degree).

We use the following formulae for computation of $\Delta\mu$ and $\Delta\nu$:

$$\Delta\mu = \Delta\pi(0.5 + \frac{1}{2}\mathrm{sg}(\Delta w).|\Delta w|^{\frac{1}{C_R}})$$

$$\Delta\nu = \Delta\pi(0.5 - \frac{1}{2}\mathrm{sg}(\Delta w).|\Delta w|^{\frac{1}{C_R}}),$$

where C_R is a coefficient for adjustment of the strength of Δw in the computation of the new ratings

Considering the rating modification parameters, the new ratings of the home and away teams (R_1^h and R_1^a respectively) are computed the following way:

$$\mu(R_1^h) = \mu(R_0^h) + \Delta\mu$$
$$\nu(R_1^h) = \nu(R_0^h) + \Delta\nu$$
$$\mu(R_1^a) = \mu(R_0^a) + \Delta\nu$$
$$\nu(R_1^a) = \nu(R_0^a) + \Delta\mu$$

Repeating the above steps with each round of the championship we adjust the ratings of the teams taking into account the results from the matches they have played. Thus, at the end of the championship each team has a rating with minimal degree of indefiniteness. We can use this rating as a base for the initial rating of the team for the next championship season. Of course, we should increase the degree of indefiniteness (however reserving the ratio between the truth and falsity degrees), because of many reasons, e.g. a team may have changed its management, or some of the players, etc. Doing so, we will have to select a proper value for $\Delta\pi$ for the next season, because this time the ratings won't be initialized with full indefiniteness.

3. IFRDB representation of the model

The structure of the model is organized in the following tables:

- table *team (id, name)* contains a list of the teams in the championship
- table *match (id, date, home_team_id, away_team_id, home_score, away_score)* contains the matches in the championship; *home_team_id* and *away_team_id* are references respectively to the records in team for the home and away teams; *home_score* and

away_score contain the result of the match, initially before the match they are NULL
- table *good_teams (team_id, truth, falsity)* is an intuitionistic fuzzy set of the good teams (contains the current ratings of the teams)
- table *rating (team_id, match_id, truth, falsity)* contains a history of the teams' ratings

Assume that intuitionistic fuzzy predicate is a function, which returns a couple of numbers in the interval [0, 1] representing the truth and falsity degrees of the predicate value. The following stored functions are defined in the database model:

- an intuitionistic fuzzy predicate *is_team_good (team_id)*, which makes a lookup in the table *good_teams* to obtain the current rating of a team
- an intuitionistic fuzzy predicate *is_team_good (team_id, match_id)*, which makes a lookup in the table *rating* to obtain the rating of a team at a certain match
- an intuitionistic fuzzy predicate *is_team_good (team_id, date)*, which makes a lookup in the table *rating* to obtain the rating of a team at a certain date
- explanatory database with translation rules for the above three predicates
- an intuitionistic fuzzy predicate *win_expectancy (home_rating, away_rating)*, which computes the expectancy for the home team to win
- a function *match_result (home_score, away_score)*, which evaluates the result of the match with a number in the interval [0, 1]
- a function *delta_mu (win_expectancy, match_result)*, which computes $\Delta\mu$
- a function *delta_nu (win_expectancy, match_result)*, which computes $\Delta\nu$
- an intuitionistic fuzzy modifier *ADD (delta_mu, delta_nu)*, which adds *delta_mu* and *delta_nu* respectively to the degrees of truth and falsity
- a translation rule in the explanatory database for the modifier

Now we will describe the steps, which the model follows, with the terms of intuitionistic fuzzy SQL:

3.1. Compute the win expectancy for a set of matches

The following SELECT returns an intuitionistic fuzzy relation, which answers the query *"Find matches, for which the home team is expected to win"*:

```
SELECT m.id, h.name AS home_team, a.name AS
away_team
FROM match m
   JOIN team h ON m.home_team_id = h.id
   JOIN team a ON m.away_team_id = a.id
WHERE win_expectancy (is_team_good (home_team_id),
is_team_good (away_team_id));
```

For the translation of the intuitionistic fuzzy SQL statement to a standard SQL we use the explanatory database to find the translation rule for the predicate *is_team_good (team_id)*. The translator adds two more joins in the FROM clause in order to obtain the values of the predicates *is_team_good (home_team_id)* and *is_team_good (away_team_id)*. The translated statement is the following:

```
SELECT m.id, h.name AS home_team, a.name AS
away_team
   , win_expectancy (if_value (hr.truth,
hr.falsity), if_value (ar.truth, ar.falsity)).truth
AS _mship
   , win_expectancy (if_value (hr.truth,
hr.falsity), if_value (ar.truth,
ar.falsity)).falsity AS _nmship
FROM match m
   JOIN team h ON m.home_team_id = h.id
   JOIN team a ON m.away_team_id = a.id
   JOIN good_teams hr ON m.home_team_id = hr.team_id
   JOIN good_teams ar ON m.away_team_id =
   ar.team_id;
```

Here *if_value (truth, falsity)* is a function, which encapsulates the two number parameters into a composite object with two fields.

3.2. Compute the new ratings

For each match we perform the following steps to adjust the new ratings of the teams, as we already know the result of the match:

- a variable *we* stores the winning expectancy for the home team
- a variable *mr* stores *match_result (home_score, away_score)*

- variables *home_team_id* and *away_team_id* store the corresponding columns from the *match* table

```
UPDATE good_teams MODIF ADD [delta_mu (we, mr),
delta_nu (we, mr)]
   WHERE team_id = home_team_id;
UPDATE good_teams MODIF ADD [delta_nu (we, mr),
delta_mu (we, mr)]
   WHERE team_id = away_team_id;
```

According to the translation rule for the intuitionistic fuzzy operator *ADD* these statements are translated to the following standard SQL statements:

```
UPDATE good_teams SET truth = truth + delta_mu(we,
mr), falsity = falsity + delta_nu(we, mr)
   WHERE team_id = home_team_id;
UPDATE good_teams SET truth = truth + delta_nu (we,
mr), falsity = falsity + delta_mu (we, mr)
   WHERE team_id = away_team_id;
```

The new ratings are added to the *rating* table.

4. Conclusions

We presented a model for prediction of football match results based on the theory of intuitionistic fuzzy sets. The benefits of this model are the following:

- it takes into account the uncertainty in the computation of the rating of a given team
- when the match wins a team, which is expected to lose, its rating is increased significantly and so is decreased the rating of the other team
- the model provides an easily configurable system using the parameters described below

The values of the parameters C_H, C_E and C_R may be specific for each championship or even for a certain match. For example, in some championships we can take into account the fact that the audience gives more support to the home team than usual, so for these ones the coefficient C_H should be greater. In some championships, however, there are no home teams in fact (international leagues), so the audience support is not significant and C_H should be equal to 1. An exception makes only the host of the championship. The value of the parameter C_R may vary depending on the rating of the championship. For example, the results from the

champion's league matches are more significant than those from the domestic leagues, so the value of C_R for the first ones should be greater. These parameters may be adjusted experimentally for a certain championship in order to achieve a maximal prediction success. We made some experiments with the results from the English premiership and concluded that the systems gives good prediction results with values for the parameters C_H, C_E and C_R respectively 3, 1 and 10.

The IFRDB implementation of the model gives the opportunity to easily store and manage data for football matches results and retrieve intuitionistic fuzzy query results with simple IFSQL statements. We can use this model for prediction of football matches as well as for making intuitionistic fuzzy queries using the history about teams' performance. Examples for such queries can be: "*How often* Manchester Utd. was *approximately as good as* Liverpool as a home team *lately*?", "Find teams, which *seldom* were *stronger* than *comparatively good* teams?", etc.

References

1. Kolev B, "Intuitionistic Fuzzy Relational Databases and Translation of the Intuitionistic Fuzzy SQL", in press
2. Kolev B, "An Intuitionistic Fuzzy Approach in Football Match Result Predictions", in press
3. Kolev B, Chountas P (2003) "Intuitionistic Fuzzy Relational Databases", In: - Proceedings of the Seventh International Conference on Intuitionistic Fuzzy Sets, 23-24 August 2003, Sofia, Bulgaria, 109-113
4. Sauceda E., Sauceda Rating System Description, http://www.masseyratings.com/theory/sauceda.htm
5. Atanassov K (1999) Intuitionistic Fuzzy Sets, Springer-Verlag, Heidelberg.
6. Atanassov K (1998) Generalized Nets in Artificial Intelligence, Vol. 1: Generalized Nets and Expert Systems, "Prof. M. Drinov" Publ. House, Sofia
7. Connoly T, Begg C, Strachan A (1998) Database Systems: A Practical Approach to Design, Implementation and management, Addison-Wesley, Harlow, England
8. Raju, K, Mahumdar A K (1988) "Fuzzy Functional Dependencies and Lossless Join Decomposition of Fuzzy Relational Database Systems", ACM Transactions on Database Systems, Vol. 13, No. 2, June 1988, p. 129-166

Generalized Net Model for Adaptive Electronic Assessment, Using Intuitionistic Fuzzy Estimations

Anthony Shannon[1], Daniela Langova-Orozova[2], Evdokia Sotirova[3], Krassimir Atanassov[4], Pedro Melo-Pinto[5], Taekyun Kim[6]

[1] KvB Institute of Technology, North Sydney, 2060, Australia, tony@kvb.edu.au
[2] Free University of Bourgas, Bourgas 8000, Bulgaria, orozova@btu.bg
[3] "Prof. Asen Zlatarov" University, Bourgas 8000, Bulgaria, esotirova@btu.bg
[4] CLBME –Bulgarian Academy of Sciences, Acad. G. Bonchev Str., Bl. 105, Sofia-1113, Bulgaria, krat@argo.bas.bg
[5] CETAV - University of Tras-os-Montes e Alto Douro, Ap. 1014, 5001-911 Vila Real, Portugal, pmelo@utad.pt
[6] Institute of Science Education, Kongju National University, Kongju 314-701, South Korea, tkim@kongju.ac.kr

Abstract. A generalized net model with intuitionistic fuzzy estimations has been constructed in order to simulate electronic adaptive assessment of students. The final mark is determined on the basis of a set of such assessment units as the problem, the test, or the examination. Each assessment unit has been associated with weight coefficients, represented by intuitionistic fuzzy estimations, determining the unit's importance.

Keywords. e-Learning, Generalized nets, Intuitionistic Fuzzy Sets, Knowledge estimations

1 Introduction

With the present article we aim to model the process of adaptive dynamic assessment during the process of e-learning. We have chosen generalized nets for our modelling tool.

The training materials can be classified as:

- information units (for acquiring of knowledge and skills);
- assessment units – problem, test, examination

Trainees may also utilize the library resources.

In this paper we shall use the apparatus of the Intuitionistic Fuzzy Sets (see [1]) and Generalized Nets (GNs; see [2]).

A. Shannon et al: *Generalized Net Model for Adaptive Electronic Assessment, Using Intuitionistic Fuzzy Estimations*, Advances in Soft Computing **2**, 291–297 (2005)
www.springerlink.com © Springer-Verlag Berlin Heidelberg 2005

The estimations, rendering account of the students' knowledge, are formed on the basis of a set of intuitionistic fuzzy estimations <μ, ν>, related to the respective assessment units. These intuitionistic fuzzy estimations reflect the degree of student's good performance μ or bad performance ν at each assessment unit.

We introduce fuzzy coefficients <δ, ε>, setting weights of each assessment unit that participates in forming the final mark. Coefficient δ is shaped on the basis of the number of the successive assessment units, and coefficient ε is shaped on the basis of the number of the preceding assessment units. For instance, a trainee shall sit for eight assessment units, divided in three levels of difficulty (easy, middle, difficult). Let there be three assessment units from the first level, three assessment units from the second level, and two assessment units from the third level. Then the weight coefficients will be distributed as follows: from first level: $<\frac{5}{8},0>$, from second level: $<\frac{2}{8},\frac{3}{8}>$ and from third level: $<0,\frac{6}{8}>$. In this way, (k+1)-st estimation $<\mu_{k+1},\nu_{k+1}>$, for k ≥ 0, is calculated on the basis of the preceding estimations $<\mu_k,\nu_k>$ due to the formula:

$$<\mu_{k+1},\nu_{k+1}> = <\frac{\mu_k.k+\delta_i.m+\varepsilon_i.n}{k+1},\frac{\nu_k.k+\delta_i.n+\varepsilon_i.m}{k+1}>,$$

where <m, n> is the estimation of the current assessment unit, m,n ∈ [0, 1] and m + n ≤ 1, and <δ_i, ε_i> is the weight coefficients of the i^{-th} assessment unit, for $\delta_i + \varepsilon_i \leq 1$.

2 Short Remarks on Generalized Nets

The way of defining the GNs is principally different from the ways of defining the other types of Petri nets. The first basic difference between GNs and the ordinary Petri nets is the "place – transition" relation. Here, the transitions are objects of a more complex nature.

Formally, every transition (see Fig. 1) is described by a seven-tuple, but here we shall use only five of them:

$$Z = \langle L', L'', r, M, \Box \rangle,$$

where:

(a) L' and L'' are finite, non-empty sets of places (the transition's input and output places, respectively). For the transition in Fig. 1 these are

$$L' = \{ l'_1, l'_2, \ldots, l'_m\} \text{ and } L'' = \{ l''_1, l''_2, \ldots, l''_n\};$$

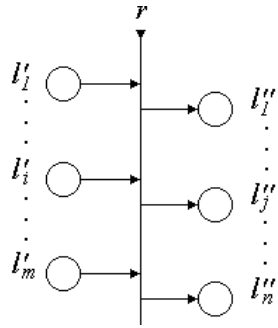


Fig. 1. GN-transition

(b) r is the transition's *condition* determining which tokens will pass (or *transfer*) from the transition's inputs to its outputs; it has the form of an Index Matrix (see [2]):

$$R = \begin{array}{c|ccc} & l''_1 & l''_j & l''_n \\ \hline l'_1 & & & \\ & & r_{i,j} & \\ l'_i & & (r_{i,j} - predicate) & \\ & & (1 \le i \le m,\ 1 \le j \le n) & \\ l'_m & & & \end{array}$$

$r_{i,j}$ is the predicate which corresponds to the i-th input and j-th output places. When its truth value is "*true*", a token from the i-th input place can be transferred to the j-th output place; otherwise, this is not possible;

(c) M is an index matrix of the capacities of transition's arcs:

$$M = \begin{array}{c|ccc} & l''_1 & l''_j & l''_n \\ \hline l'_1 & & & \\ & & m_{i,j} & \\ l'_i & & (m_{i,j} \ge 0 - natural\ \ number) & \\ & & (1 \le i \le m,\ 1 \le j \le n) & \\ l'_m & & & \end{array}$$

(d) □ is an object having a form similar to a Boolean expression. It may contain as variables the symbols which serve as labels for transition's input places, and is an expression built up from variables and the Boolean connectives $\wedge$ and $\vee$ whose semantics is defined as follows:

$\wedge(l_{i_1}, l_{i_2}, ..., l_{i_u})$ – every place $l_{i_1}, l_{i_2}, ..., l_{i_u}$ must contain at least one token,

$\vee(l_{i_1}, l_{i_2}, ..., l_{i_u})$ – there must be at least one token in all places $l_{i_1}, l_{i_2}, ..., l_{i_u}$,

where $\{l_{i_1}, l_{i_2}, ..., l_{i_u}\} \subset L'$.

When the value of a type (calculated as a Boolean expression) is "*true*", the transition can become active, otherwise it cannot.

The ordered four-tuple (here also use a reduced GN-form)

$$E = \langle\langle A, \mathrm{c}\rangle, K, X, \Phi\rangle\rangle$$

is called a reduced *Generalized Net* (GN) if:

- A is a set of transitions;
- c is a function giving the capacities of the places, i.e., $c : L \to \mathrm{N}$;
- K is the set of the GN's tokens;
- X is the set of all initial characteristics the tokens can receive when they enter the net;
- Φ is a characteristic function which assigns new characteristics to every token when it makes a transfer from an input to an output place of a given transition.

It is also convenient to assume that the functions Φ have other forms. For our needs we will use the above form only of function Φ as follows:

$$\Phi = \bigcup_{i=1}^{|L-Q^I|} \Phi_i ,$$

where Φ_i calculates the characteristics which the tokens will receive in the i-th GN place and $|X|$ is the cardinality of the set X.

3 Generalized Net model

The so-developed generalized net model of e-learning assessment module has been represented in Fig. 2.

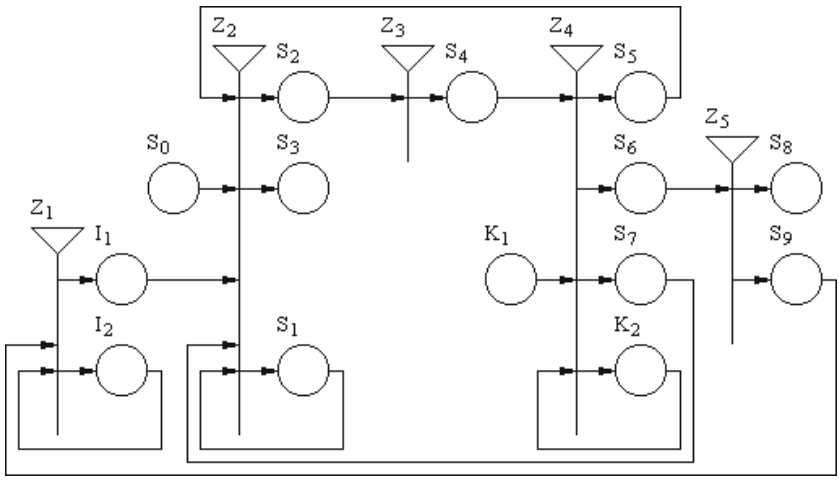


Fig. 2. Generalized net model of e-learning assessment module

The places in the generalized net fall in three categories: S – identifiers of the appraised subjects (students), I – assessment units, and K – criteria for assessment. The generalized net contains the following types of tokens: α-tokens, representing the subjects of appraisal, β-tokens, interpreting the separate assessment units, and γ-tokens, related to the criteria for evaluation. The set contains the following set of transitions

$$A = <Z_1, Z_2, Z_3, Z_4, Z_5>,$$

where the transitions stand for the following processes:

- Choice of an assessment unit and a student for evaluation – transitions Z_1 and Z_2;
- Forming the student's answers at the chosen assessment unit – transition Z_3;
- Evaluation of the student's performance at the current assessment unit – transition Z_4;
- Forming the final mark– transition Z_5.

Initially, place I_1 contains q in number β_t-tokens with the characteristic "assessment unit t", where $t = 1, 2,\ldots q$. In a certain moment of time, each of these tokens may split in several new tokens. The original tokens will remain in place I_1, and the new ones will pass via transition Z_1.

The appraised students (interpreted by α_v-tokens, $v = 1, 2, \ldots, p$), enter the net via place S_1. The evaluation criteria (interpreted by γ_w tokens, $w = 1, 2, \ldots, r$) enter the net via place K_1.

The transitions have the respective representation, given below. Let everywhere t be the number of assessment units (for $t = 1, 2,\ldots q$), and v be the number of evaluated students (for $v = 1, 2, \ldots, p$).

$$Z_1 = <\{I_1, S_9\}, \{I_1, I_2\}, R_1, M_1, \vee (L_1, S_9)>,$$

$$R_1 = \begin{array}{c|cc} & I_1 & I_2 \\ \hline I_1 & \text{false} & W_1 \\ S_9 & W_{9,1} & \text{false} \end{array}; \quad M_1 = \begin{array}{c|cc} & I_1 & I_2 \\ \hline I_1 & 0 & q \\ S_9 & p & 0 \end{array},$$

where
W_1 = "An assessment unit has been chosen",
$W_{9,1}$ = "The successive assessment unit shall be assigned".

The tokens, entering place I_2, obtain the characteristic:

"assessment unit t, mark $< \mu_k, \nu_k >$".

$$Z_2 = <\{S_0, I_2, S_1, S_5\}, \{S_1, S_2, S_3\}, R_2, M_2, \vee (\wedge(I_2, S_0), S_1, S_5)>,$$

$$R_2 = \begin{array}{c|ccc} & S_1 & S_2 & S_3 \\ \hline S_0 & W_{0,1} & \text{false} & \text{false} \\ I_2 & W_{2,1} & \text{false} & \text{false}; \\ S_1 & \text{false} & W_{1,2} & W_{1,3} \\ S_5 & W_{5,1} & \text{false} & \text{false} \end{array} \quad M_2 = \begin{array}{c|ccc} & S_1 & S_2 & S_3 \\ \hline S_0 & p & 0 & 0 \\ I_2 & q & 0 & 0 \\ S_1 & 0 & p & p \\ S_5 & p & 0 & 0 \end{array},$$

where
$W_{0,1}$ = “There is a student to be appraised”,
$W_{2,1}$ = “An assessment unit has been chosen”,
$W_{1,2}$ = “An assessment unit for the student has been chosen”,
$W_{1,3}$ = “There is a student who has given up examination”,
$W_{5,1}$ = “The successive assessment unit shall be assigned”.

The α_t-tokens originate from place S_0 and when the respective predicate $W_{0,1}$ values “true”, the tokens enter place S_1 with the characteristic

“student *v*, estimation $< \mu_k, \nu_k >$”.

Place S_1 may contain β_v-tokens from place I_2 with the characteristic “assessment unit *t*”, in case the respective predicate $W_{2,1}$ values “true”. Initially, when the student has not been submitted to any assessment unit, his estimation is $< 0,0 >$. The tokens entering places S_2 and S_3 obtain respective characteristics: “student *v*, assessment unit *t*, mark $< \mu_k, \nu_k >$” and “student *v*, mark $< \mu_k, \nu_k >$”.

$$Z_3 = <\{ S_2, S_7\}, \{S_4\}, R_3, M_3, \wedge (S_2)> , \quad R_3 = \begin{array}{c|c} & S_4 \\ \hline S_2 & W_{2,4}; \\ S_7 & W_{7,4} \end{array} \quad M_3 = \begin{array}{c|c} & S_4 \\ \hline S_2 & p \\ S_7 & p \end{array}$$

where
$W_{2,4}$ = “The student has formulated an answer to the assessment unit”,
$W_{7,4}$ = “The answer to the assessment unit shall be formulated again”.

The tokens entering place S_5 obtain characteristic:

“student *v*, assessment unit *t*, answer, mark $< \mu_k, \nu_k >$”.

$$Z_4 = <\{ S_4, K_1, K_2\}, \{S_5, S_6, S_7, K_2\}, R_4, M_4, \vee (\wedge(K_2, S_4), K_1)> ,$$

$$R_4 = \begin{array}{c|cccc} & S_5 & S_6 & S_7 & K_2 \\ \hline S_4 & \text{false} & \text{false} & \text{false} & W_{4,2} \\ K_1 & \text{false} & \text{false} & \text{false} & W_2 \\ K_2 & W_{2,5} & W_{2,6} & W_{2,7} & \text{false} \end{array}, \quad M_4 = \begin{array}{c|cccc} & S_5 & S_6 & S_7 & K_2 \\ \hline S_4 & 0 & 0 & 0 & p \\ K_1 & 0 & 0 & 0 & r \\ K_2 & p & p & p & 0 \end{array},$$

where
$W_{4,2} = W_{2,4}$,
W_2 = "The criteria for answers' evaluation are available",
$W_{2,5}$ = "The formulated answer is incorrect",
$W_{2,6}$ = "The formulated answer is correct",
$W_{2,7}$ = "The formulated answer is ill defined, or there is a technical error".

The tokens entering places S_5, S_6 and S_7 obtain the characteristic:

"student *v*, assessment unit *t*, answer, mark $< \mu_k, \nu_k >$".

$$Z_5 = <\{ S_6 \}, \{S_8, S_9 \}, R_5, M_5, \wedge(S_6)> ,$$

$$R_5 = \begin{array}{c|cc} & S_8 & S_9 \\ \hline S_6 & W_{6,8} & W_{6,9} \end{array}, \qquad M_5 = \begin{array}{c|cc} & S_8 & S_9 \\ \hline S_6 & p & p \end{array},$$

where
$W_{6,8}$ = "The final mark has been formed",
$W_{6,9}$ = "The successive assessment unit shall be retrieved".

The tokens entering place S_9 obtain characteristic:

"student *v*, mark $< \mu_k, \nu_k >$".

The tokens entering place S_8 obtain characteristic "student ν, final mark $<\mu, \nu>$", where μ and ν are the estimations due to the levels of the students' ability or disability to assimilate the module, which is calculated on the basis of the number of tokens in places S_6 and S_5. In the cases when the answer has been ill defined or a technical error has been committed, the degree of indeterminacy depends on the number of tokens in place S_7.

Conclusion

The so-formed final mark obtained by the proposed generalized net model of e-learning assessment module, using intuitionistic fuzzy estimations, can be utilized for analysis of students' knowledge and performance at examinations.

References

[1] Atanassov K (1999) Intuitionistic Fuzzy Sets, Springer Physica-Verlag, Heidelberg.
[2] Atanassov K (1991) Generalized Nets, World Scientific, Singapore.

Fuzzy Logic in Decision Support

Analytic Hierarchy Process Based on Fuzzy Analysis

Tomoe Entani[1], Kazutomi Sugihara[2] and Hideo Tanaka[3]

[1] Kochi University, 2-5-1 Akebono, Kochi 780-8520, JAPAN
entani@cc.kochi-u.ac.jp
[2] Fukui University of Technology, 3-6-1 Gakuen, Fukui 910-8505, JAPAN
sugihara@ccmails.fukui-ut.ac.jp
[3] Hiroshima International University, Gakuendai 555-36, Kurose, Hiroshima 724-0695, JAPAN
h-tanaka@he.hirokoku-u.ac.jp

Abstract. Analytic Hierarchical Process(AHP) is proposed to give the priority weight with respect to many items. The priority weights are obtained from the pairwise comparison matrix whose elements are given by a decision maker as crisp values. We extend the crisp pairwise comparisons to fuzzy ones based on uncertainty of human judgement. To give uncertain information as a fuzzy value is more rational than as a crisp value. We assume that the item's weight is a fuzzy value, since the comparisons are based on human intuition so that they must be inconsistent each other. We propose a new AHP, where the item's weight is given as a fuzzy value, in order to deal with inconsistency in the given matrix. The purpose is to obtain fuzzy weights so as to include all the given fuzzy pairwise comparisons, in the similar way to the upper approximation in interval regression analysis.

1 Introduction

AHP (Analytic Hierarchical Process) is a useful method in multi-criteria decision making problems. The priority weights of the items are obtained from a pairwise comparison matrix by eigenvector method [1]. The elements of the matrix called pairwise comparisons are relative measurements given by a decision maker. Therefore, the weights obtained from the given matrix can reflect his/her attitude in actual decision problem. The weights obtained by the conventional AHP lead to a linear order of items. Uncertainty of an order of items in AHP is discussed in [2].

We extend crisp pairwise comparisons to fuzzy ones. Fuzzy pairwise comparisons are more rational to represent decision maker's uncertain judgements than crisp ones. The width of each fuzzy comparison represents uncertainty

T. Entani et al.: *Analytic Hierarchy Process Based on Fuzzy Analysis*, Advances in Soft Computing **2**, 301–309 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

of the judgement. In this paper, it is assumed that the esitmated weights are fuzzy values in order to reflect inconsistency in pairwise comparisons. We deal with fuzzy weights as intervals by considering their h-level sets. The fuzzy weights are determined so as to include the given fuzzy pairwise comparisons at h-level. This concept is similar to the upper approximation in interval regression analysis [5]. Instead of solving the eigenvector problem in the conventional AHP, we can obtain the weights by solving one QP problem. Then, inconsistency in the given matrix is shared with all items to some extent.

2 Crisp priority weights by conventional AHP

AHP is a method to deal with the weights with respect to many items and proposed to determine the weights of each item [1]. When there are n items, a decision maker compares a pair of items for all possible pairs then we can obtain a comparison matrix A as follows.

$$A = [a_{ij}] = \begin{pmatrix} 1 & \cdots & a_{1n} \\ \vdots & a_{ij} & \vdots \\ a_{n1} & \cdots & 1 \end{pmatrix}$$

where a_{ij} shows the priority weight of item i comparing to j. The diagonal elements are all 1, that is $a_{ii} = 1$ and other elements are reciprocal, that is $a_{ij} = 1/a_{ji}$.

From the given comparison matrix by eigenvector method, the priority weights w_i^* are obtained. The eigenvector problem is formulated as follows.

$$A\boldsymbol{w} = \lambda \boldsymbol{w} \tag{1}$$

where the eigenvalue λ and the eigenvector $\boldsymbol{w}$ are the decision variables.

Solving (1), the eigenvector $(w_1^*, \ldots, w_n^*)$ corresponding to the principal eigenvalue λ_{max} is obtained as the weight of each item. The sum of the obtained weights w_i^* is normalized to be 1.

Using the obtained weights w_i^* the estimated pairwise comparison a_{ij}^* can be denoted as follows.

$$a_{ij}^* = \frac{w_i^*}{w_j^*}$$

If the pairwise comparison matrix is perfectly consistent, $a_{ij} = a_{ij}^*$ hold for all pairs of (i, j). However, a decision maker gives pairwise comparisons intuitively so that they usually are inconsistent, that is $a_{ij} \neq a_{ij}^*$. The larger the number of compared items is, the more difficult it becomes to give consistent pairwise comparisons, since a decision maker compares only two items at one time. Therefore, it seems more pragmatic to give items interval, instead of crisp, weights which reflect inconsistency in the given comparison matrix.

In this paper, each pairwise comparison is given as a fuzzy value to reflect uncertainty of a decision maker's judgement. Then, considering inconsisntency of a human judgements, the items' weights are obtained as fuzzy values from the given fuzzy pairwise comparisons. We use a QP problem instead of eigenvector problem so that the weights of all items are obtained as uncertain to some extent.

3 Fuzzy weights with fuzzy matrix by AHP

3.1 Fuzzy pairwise comparison matrix

Since decision maker's judgements are uncertain, it is easier for him/her to give a pairwise comparison as a fuzzy value than a crisp value. When a decision maker compares a pair of items for all possible pairs with n items, we can obtain a fuzzy comparison matrix $\tilde{A}$ as follows.

$$\tilde{A} = [\tilde{a}_{ij}] = \begin{pmatrix} 1 & \cdots & \tilde{a}_{1n} \\ \vdots & \tilde{a}_{ij} & \vdots \\ \tilde{a}_{n1} & \cdots & 1 \end{pmatrix}$$

where $\tilde{a}_{ij}$ is a fuzzy value and shows the priority weight of item i comparing to item j. The diagonal elements are all 1, that is $\tilde{a}_{ii} = 1$.

We denote a fuzzy value $\tilde{a}_{ij}$ as parametrically (ac_{ij}, ad_{ij}) where ac_{ij} and ad_{ij} are the center and width respectively as follows.

$$\mu_{\tilde{a}_{ij}}(x) = \begin{cases} 0 \ (x \leq ac_{ij} - ad_{ij},\ ac_{ij} + ad_{ij} \leq x) \\ \dfrac{x - (ac_{ij} - ad_{ij})}{ad_{ij}} \ (ac_{ij} - ad_{ij} \leq x \leq ac_{ij}) \\ \dfrac{x - (ac_{ij} + ad_{ij})}{ad_{ij}} \ (ac_{ij} \leq x \leq ac_{ij} + ad_{ij}) \end{cases}$$

where $\mu_{\tilde{a}_{ij}}(x)$ is the membership function of a fuzzy value $\tilde{a}_{ij}$, and shown in Figure 1.

We deal with a fuzzy value considering its h-level sets [4]. A fuzzy value can be represented by intervals as

$$B = \cup_h [B]_h, \quad 0 \leq h \leq 1$$

where B is a fuzzy value and h is a real value. h-level set of B, $[B]_h$, is defined as follows.

$$[B]_h = \{x | \mu_B(x) \geq h\}$$

where $\mu_B(x)$ is the membership function of B. Then $[B]_h$ is considered as an interval.

In the same way, h-level set of the fuzzy pairwise comparison is considered as follows.

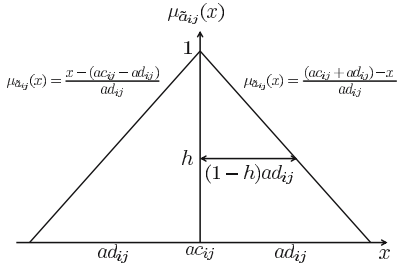


Fig. 1. Membership function of fuzzy pairwise comparison $\tilde{a}_{ij}$

$$[\tilde{a}_{ij}]_h = [\underline{a}_{ij}, \overline{a}_{ij}] = [ac_{ij} - (1-h)ad_{ij}, ac_{ij} + (1-h)ad_{ij}]$$

The above interval pairwise comparison satisfies the following reciprocal property.

$$ac_{ij} + (1-h)ad_{ij} = \frac{1}{ac_{ji} - (1-h)ad_{ji}}$$

3.2 Fuzzy priority weights

We estimate the priority weights assumed as a triangular fuzzy values, $\tilde{w}_i = (c_i, d_i)\ \forall i$, which are defined as the following membership function.

$$\mu_{\tilde{w}_i}(x) = \begin{cases} 0\ (x \le c_i - d_i, c_i + d_i \le x) \\ \dfrac{x-(c_i-d_i)}{d_i}\ (c_i - d_i \le x \le c_i) \\ \dfrac{x-(c_i+d_i)}{d_i}\ (c_i \le x \le c_i + d_i) \end{cases}$$

where c_i and d_i are the center and width, respectively.

Considering that a fuzzy value is represented by intervals [4], a fuzzy weight can be dealt with its h-level set denoted as follows.

$$[\tilde{w}_i]_h = [\underline{w}_i, \overline{w}_i] = [c_i - (1-h)d_i, c_i + (1-h)d_i]$$

Then, the pairwise comparison with respect to items i and j is approximated as an interval ratio as follows.

$$a_{ij} \approx \frac{[\tilde{w}_i]_h}{[\tilde{w}_j]_h} = \left[\frac{\underline{w}_i}{\overline{w}_j}, \frac{\overline{w}_i}{\underline{w}_j}\right]$$

where $[\tilde{w}_i]_h$ and $[\tilde{w}_j]_h$ are h-level sets of the estimated fuzzy weights and the interval ratio $[\tilde{w}_i]_h / [\tilde{w}_j]_h$ is defined as the maximum range.

Based on the upper approximation in interval regression analysis, the fuzzy weights are determined to include the given fuzzy pairwise comparisons.

$$\begin{aligned}\tilde{a}_{ij} \in \frac{\tilde{w}_i}{\tilde{w}_j} &\leftrightarrow [\tilde{a}_{ij}]_h \in \frac{[\tilde{w}_i]_h}{[\tilde{w}_j]_h} \\ &\leftrightarrow \frac{\underline{w}_i}{\overline{w}_j} \leq \underline{a}_{ij},\ \ \overline{a}_{ij} \leq \frac{\overline{w}_i}{\underline{w}_j}\end{aligned} \tag{2}$$

3.3 Formulation as QP problem

We note that the sum of weights obtained by the conventional AHP is normalized to be one. We consider interval probability functions proposed in [6] so as to normalize the interval weights. Their conditions are defined as follows.

Definition 1. *h-level sets of fuzzy priority weights* $([\tilde{w}_1]_h, ..., [\tilde{w}_n]_h)$ *where* $[\tilde{w}_i]_h$ *is denoted as an interval* $[\underline{w}_i, \overline{w}_i]$ *is called interval probability if and only if*

$$\begin{aligned}&\textstyle\sum_{i\neq j} \overline{w}_i + \underline{w}_j \geq 1 \ \ \forall j \\ &\textstyle\sum_{i\neq j} \underline{w}_i + \overline{w}_j \leq 1 \ \ \forall j.\end{aligned}$$

It can be said that the conventional normalization is extended to the interval normalization in the above definition. This is effective to reduce redundancy under the condition that the sum of crisp weights within the interval weights is equal to one.

The problem to obtain h-level sets of fuzzy weights, $[\tilde{w}_i]_h = [\underline{w}_i, \overline{w}_i]\ \forall i$, are formulated as the following QP problem.

$$\begin{aligned}&\min \quad \textstyle\sum_{i=1}^{n} (\overline{w}_i - \underline{w}_i)^2 \\ &\text{s.t. } \frac{\underline{w}_i}{\overline{w}_j} \leq \underline{a}_{ij} \ \ \forall (i,j) \\ &\quad \overline{a}_{ij} \leq \frac{\overline{w}_i}{\underline{w}_j} \ \ \forall (i,j) \\ &\quad \textstyle\sum_{i\neq j} \overline{w}_i + \underline{w}_j \geq 1 \ \ \forall j \\ &\quad \textstyle\sum_{i\neq j} \underline{w}_i + \overline{w}_j \leq 1 \ \ \forall j \\ &\quad \overline{w}_i \geq \underline{w}_i \ \ \forall i \\ &\quad \underline{w}_i \geq 0 \ \ \forall i\end{aligned} \tag{3}$$

where the first and second constraint conditions are the inclusion relation (2) and the third and forth ones are the interval normalization in Definition 1. In order to obtain the least upper approximation, the sum of squared widths of interval weights is minimized in the similar way of least square method.

The widths of fuzzy weights reflect some inconsistency in the given fuzzy pairwise comparison matrix. In other words, the obtained weights can be regarded as the possible ranges estimated from the given pairwise comparisons. The estimated fuzzy weights are obtained so as to include all the given fuzzy pairwise comparisons.

The problem to obtain the priority weights has also been formulated as the following LP problem.

$$\begin{aligned}&\min \quad \textstyle\sum_{i=1}^{n} (\overline{w}_i - \underline{w}_i) \\ &\text{s.t. constraint conditions in (3)}\end{aligned} \tag{4}$$

where the quadratic objective function in (4) is transformed to the linear one.

The width of the obtained weight illustrates the degree of uncertainty of the item's weight and reflects inconsisntency of corresponding pairwise comparisons. The degree of uncertainty of each item can be considered to some extent by the first and second constraint conditions in both formulations (3) and (4). The diffrence of the objective functions is explained as follows. In the case of LP problem (4) such that the sum of widths is minimized, it happens that the widths of some items are quite large while others are small. The variance of widths tends to become large. Such a result indicates that inconsistency contained in the given comparison matrix is caused only by the items whose widths are obtained as large. In the case of the QP problem (3) such that the sum of squared widths is minimized, not only the special items but also all items have certain widths. Then, inconsistency in the given matrix is shared by all items. Since it is not easy to find the exact items which cause inconsistency in the given comparisons, QP prolbem is more suitable to real situation than LP problem.

We introduce accuracy measure, in order to compare the results from QP and LP problems. The accuracy measure can be defined as the average coverage rate of the given pairwise comparisons to the approximated ones as follows.

$$Accuracy\ measure = \frac{1}{m} \sum_{i=1}^{n} \sum_{j \neq i, j>i}^{n} \frac{\overline{a}_{ij} - \underline{a}_{ij}}{\overline{a}^*_{ij} - \underline{a}^*_{ij}} \tag{5}$$

where m is the number of the given pairwise comparisons and the approximated comparison consists of the obtained weights, $\underline{w}^*_i$ and $\overline{w}^*_i$, by (3) and (4) denoted as follows.

$$\overline{a}^*_{ij} = \frac{\overline{w}^*_i}{\underline{w}^*_j}, \quad \underline{a}^*_{ij} = \frac{\underline{w}^*_i}{\overline{w}^*_j}$$

If all the given and approximated comparisons are the same, the accuracy measure becomes one. When the more consistent matrix is given, the larger the accuracy measure becomes. Accuracy measure represents the deviation of the approximated and given pairwise comparisons, therefore, it depends on inconsisntency in the given comparisons.

4 Numerical example

There are 5 items in this numerical example and a decision maker compares all pairs of items to give their priority weights. The given fuzzy pairwise comparison matrix $\tilde{A} = [\tilde{a}_{ij}]$ where $\tilde{a}_{ij} = (ac_{ij}, ad_{ij})$ is as follows.

$$\tilde{A} = \begin{pmatrix} 1 & (1/2, 1/6) & (1/3, 1/6) & (1/4, 1/6) & (1/5, 1/6) \\ & 1 & (2/3, 1/2) & (1/2, 1/6) & (2/5, 1/6) \\ & & 1 & (3/4, 1/6) & (3/5, 1/6) \\ & & & 1 & (4/5, 1/6) \\ & & & & 1 \end{pmatrix}$$

The 0.5-level sets of the given fuzzy matrix $[\tilde{A}]_{0.5} = [[\tilde{a}_{ij}]_{0.5}]$ where $[\tilde{a}_{ij}]_{0.5} = [\underline{a}_{ij}, \overline{a}_{ij}]$ is as follows.

$$[\tilde{A}]_{0.5} = \begin{pmatrix} 1 & [0.417, 0.583] & [0.250, 0.417] & [0.167, 0.333] & [0.117, 0.283] \\ & 1 & [0.417, 0.917] & [0.417, 0.583] & [0.317, 0.483] \\ & & 1 & [0.667, 0.833] & [0.517, 0.683] \\ & & & 1 & [0.717, 0.883] \\ & & & & 1 \end{pmatrix}$$

The fuzzy weights obtained from the given fuzzy comparison matrix at 0.5-level by QP problem (3) and LP problem (4) are shown in Table 1 and Figure 3. Figure 2 illustrates the widths of the obtained fuzzy weights by the QP and LP problems. A decision maker gives each pairwise comparison as a fuzzy value and each item's weight are also obtained as a fuzzy value.

	QP		LP	
	$[\tilde{w}_i]_{0.5}$	$\tilde{w}_i$	$[\tilde{w}_i]_{0.5}$	$\tilde{w}_i$
1	[0.041,0.069]	(0.055,0.029)	[0.041,0.086]	(0.064,0.046)
2	[0.090,0.152]	(0.121,0.063)	[0.087,0.165]	(0.126,0.078)
3	[0.166,0.215]	(0.191,0.049)	[0.180,0.208]	(0.194,0.029)
4	[0.249,0.278]	(0.264,0.029)	[0.250,0.269]	(0.260,0.019)
5	[0.315,0.348]	(0.331,0.033)	[0.305,0.349]	(0.327,0.044)
Accuracy measure	0.836		0.769	

Table 1. Fuzzy weights and their 0.5-level sets

The width of the fuzzy comparison ad_{ij} represents uncertainty of a decision maker's judgement and it might be reflected to uncertainty of the compared items i and j, denoted as their widths d_i and d_j. In this example, the width of $\tilde{a}_{23}$, which is the comparison with respect to items 2 and 3, is larger than others. Therefore, the widths of $\tilde{w}_2$ and $\tilde{w}_3$ are expected to be large. From Figure 2, in the case of the LP problem (4), the width of $\tilde{w}_2$ is the maximum of all weights, while that of $\tilde{w}_3$ is small. This result indicates that the given matrix is inconsistent almost because of item 2. Using the LP problem, the special items tend to have large widths comparing to others. On the other hand, in the case the QP problem (3), both the widths of $\tilde{w}_2$ and $\tilde{w}_3$ are larger than others. The deviation of the maximam and minimum widths by the QP problem is 0.033 which is smaller than that by the LP problem 0.058. Then, the variance of the widths obtained by the QP problem are smaller than that by the LP problem. It is difficult to find the exact items which cause inconsistency in the fuzzy pairwise comparisons. Formulating as a QP problem is suitable to a real situation, since all items share the inconsistency in the given matrix to some extent and the degree of uncertainty of each

pairwise comparison can be reflected to the widths of the comapared two items' weights.

The accuracy measures obtained by (5) are shown at the bottom row of Table 1. The accuracy measure by the QP problem is larger than that by the LP problem. It indicates that the approximated comparisons calculated with the obtained fuzzy weights by the QP problem are similarer to the the given ones than by the LP problem. Therefore, formulating as a QP problem is suitable to this example in view of approximating the given comparisons.

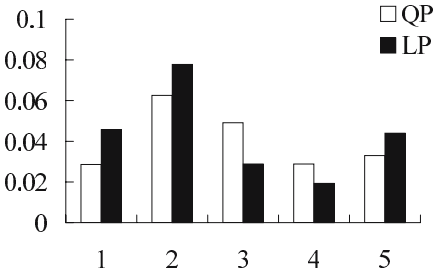


Fig. 2. Widths of 0.5-level sets of fuzzy weights

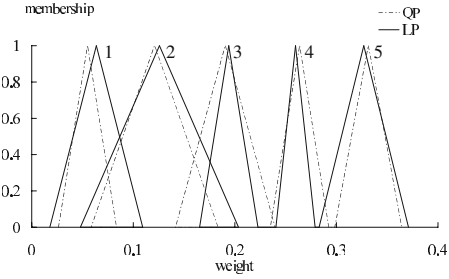


Fig. 3. Fuzzy weights by QP and LP problems

5 Concluding remarks

We extended crisp pairwise comparisons to fuzzy ones so that a decision maker's uncertain judgements can be represented by fuzzy pairwise comparisons. Then, we proposed the model to obtain fuzzy weights which reflect inconsistency in the given pairwise comparisons. Since human judgements are

usually inconsistent to each other, it is rational to give items priority weights as fuzzy values. We dealt with fuzzy pairwise comparisons and fuzzy weights as intervals by considering their h-level sets. The fuzzy weights are obtained so as to include the given fuzzy pairwise comparisons and its concept is similar to the upper approximation in interval regression analysis. Using a QP problem, inconsisntecy in the given comparison matrix is shared by all items. It is difficult to find exact items which cause inconsistency so that QP problem is suitable to real situation.

References

1. T.L.Saaty (1980) The Analytic Hierarchy Process. McGraw-Hill
2. T.L.Saaty, L.G.Vergas(1987), Uncertainty and Rank Order in the Analytic Hiearchy Process. European Journal of Operations Research 32:107-117
3. K.Sugihara, Y.Maeda, H.Tanaka(1999) Interval Evaluation by AHP with Rough Set Concept. New Directions in Rough Sets, Data Mining, and Granular-Soft Computing, Lecture Note in Artificial Intelligence 1711, Springer, 375–381,
4. L.A.Zadeh(1971) Similarity relation and fuzzy orderings. Information Sciences 3:177-200
5. H.Tanaka, P.Guo(1999) Possibilistic Data Analysis for Operation Research. Physica-Verlag, A Springer Verlag Company
6. H.Tanaka, K.Sugihara, Y.Maeda(Article in Press), Non-additive Measures by Interval Probability Functions. Information Sciences

A Fuzzy-Ga Hybrid Technique for Optimization of Teaching Sequences Presented in ITSs

David Pritchard and Mircea Gh. Negoita
School of Information Technology, Wellington Institute of Technology Private Bag 39089, Wellington, Buick Street, Petone, New Zealand
david.pritchard@weltec.ac.nz
mircea.negoita@weltec.ac.nz

Abstract. Many highly motivated students fail because of a lack of resources to help them overcome learning obstacles. A possible solution is to produce more intelligent tutoring systems to support the tutors work in the classroom. In our paper we first give an overview of our previous work on ITS systems and follow with a summary of the complete ITS system that we are developing. We relate how the student model developed in previous work is now used with an optimiser agent to "fine-tune" the linguistic variables of a fuzzy rule decision structure that is used by a tutor model to decide on "**what**" should be presented next to a student and "how" it should be presented. There is a detailed description of how our concept of an "*optimiser*" works followed by concluding remarks.

1. Introduction

The entire motivation of our research is the fact that many highly motivated students, with learning difficulties, are "falling through the cracks" in our education system. There are simply not enough resources to give them the "help" they need. What is required is a support system for tutors - an intelligent learning system able to adapt to the unique needs of each student, helping them learn in the quickest possible time with the best possible results. (Negoita and Pritchard, 2003).

The main problem that we are dealing with in our work is the vital requirement that "learning systems to be able to change to fit the student needs. (Negoita and Pritchard, 2003a). It's important to deliver "the right content to the right user in the right form at the right time". (Smyth, 2003). The system needs to always take into account the psychological learning characteristics of the student. We wish to take, for example, the work done by Chen in using multi-agent technology, providing advice to students and instructions, one step further. (Chen, 2003). We though our system would generate the advice and then give it to another agent in the system to help its work. Ongoing study of the work being done with adaptive ITSs is continuing to influence our work and provide the "spark" for new innovative ideas. (Park, Kim, 2003), (Kosba, Dimitrova, Pope, 2003). A great amount of this work is done with static models – the parameters are set at the beginning and stay fixed. We are looking at a more dynamic model. Interesting work has been done by comparing dynamic versus static student models using Bayesian Networks. (Millan, Pertez-de-la-Cruz, Garcia, 2003). Fuzzy logic and fuzzy reasoning on the knowledge structures in the student model was given in (Nkambou, 1999). This gave one suggestion for developing an adaptable student model. Such adaptive hypermedia systems also provided good input. (Lascio, Fischetti, Gisolfi, 1999). An interesting development (Papanikolaou, Grigoriadou, Kornilakis, Masgoulas, 2003) refers "to make a shift towards a more 'learning-focused' paradigm of instruction".

D. Pritchard and M.G. Negoita: *A Fuzzy-Ga Hybrid Technique for Optimization of Teaching Sequences Presented in ITSs*, Advances in Soft Computing **2**, 311–316 (2005)
www.springerlink.com © Springer-Verlag Berlin Heidelberg 2005

2. An Overview of our previous work on ITS

In one previous paper (Negoita and Pritchard, 2003a), we worked on the beginnings of an important component of the system – the student model – an intelligent agent that reflects the key learning characteristics of a human student. At this point we made the deliberate decision to keep certain components of the system (e.g. student and knowledge models) very simple so that we could concentrate on the complexities and dynamics of the Fuzzy-GA components. However, the student model we produced can learn and forget just like its human counter-part, producing a typical learning curve when asked by the system to learn a knowledge item. Our previous work clearly showed that we could create different classifications of student, each having its own unique learning curve. (Negoita and Pritchard, 2003b).

3. A summary of the work in the paper – the Optimiser agent in an ITS.

In our current paper we place our student model into the typical ITS system configuration (McTaggart, 2001), and add an agent we call the "*Optimiser"*. The system, as a whole, works in the following way.

- The *human student* interacts with the system, which presents a sequence of learning activities that result in the student learning the knowledge structure. The key to this learning sequence is the fuzzy rule decision structure used by the *tutor agent* to decide on "**what**" to present and "**how**" to present it. However the "shape" of the linguistic variables held in these fuzzy rules doesn't guarantee optimal learning. Each student learns differently so what we need is the system to find the best "shape" for the linguistic variables, for that particular student. Hence our motivation for an "*Optimiser*" agent.

- When the student first logs onto the system, the *Optimiser* agent makes its own copy of the main agents (*student*, *tutor* and *knowledge*), including the current fuzzy rule structure. While the student is working with the main system, the *Optimiser* works in the background, trying different "shapes" to the linguistic variables in an effort to improve the current fuzzy rule structure. This would result in the tutor deciding on a quicker, higher quality learning sequence.

The *Optimiser* sends its "shape refined" linguistic variables back to the main tutor, replacing its fuzzy rule structure with a more efficient one. The *Optimiser* then starts over, taking the latest information about how the student is performing, and works to find an even better "shape" to the linguistic variables.

4. The Optimiser Agent

The Optimiser works in the following way:

Step 1 – The systems creates the *Optimiser*, which in turn, creates copies of the *student, tutor* and *knowledge* agents from the main system. (see **key 1** in **Fig 1**). It also copies the current fuzzy rule decision structure. (see **key 2** in **Fig 1**). It then creates an initial GA population of 20 chromosomes. Each chromosome represents changes that *can be made* – (*possibility be made*), to the shape of the basic linguistic variables (see **key 3** in **Fig 1**).

Step 2 – Each chromosome can have up to 10 genes in it. Each activated gene represents a change that *can be made* (*possibly be made*) to the shape of any of the linguistic variables. (see **key 4** in **Fig 1**).

Step 3 – A chromosome is passed to the tutor and used to modify the "shapes" of the linguistic variables resulting in a different variation of the current fuzzy rule structure. (see **key 5** in **Fig 1**).

Step 4 – The *tutor* uses this modified version of its fuzzy rules to (a) take learning activities from the knowledge agent and (b) present them to the student (see **key 6a** and **key 6b** in **Fig 1**).

Step 5 - **Keys 6b** and **6b** are iteratively repeated until the student has learnt the whole knowledge structure. (see **key 7** in **Fig 1**).

Step 6 – The student is tested to see how well the knowledge has been learnt. The chromosome is evaluated using a Fitness Function of the following form. (see **key 8** in **Fig 1**):

$$f(t,q) = t \times q$$

Where:

f(t,q) = fitness value of the chromosome.
t = number of steps taken to learn content.
Q = number of errors student makes in the test.

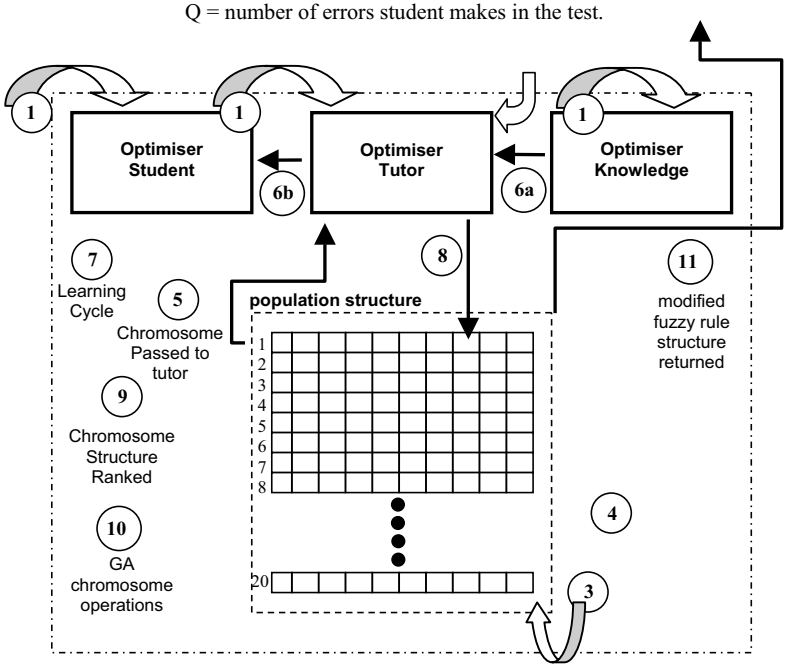


Figure 1 – overview of the Optimiser agent

The population of chromosomes is ranked in descending order by fitness value, the top 20 individuals are kept and the rest discarded. (see **key 9** in **Fig 1**). Each new generation is operated on using GA operators of selection, cross-over and mutation, to arrive at approximately 40 chromosomes. The whole process then goes back to **step 5**. (see **key 10** in **Fig 1**). A number of generations are produced until the stop condition has been meet. The best chromosome is passed

back to the main *tutor* agent where it is used to modify the "shapes" of the linguistic variables. This gives a more efficient fuzzy rule decision structure for this particular student. (see **key 11** in **Fig 1**).

Selection is done using the "roulette wheel" method and the parents retained, meaning that both parents and children must "compete" to stay in the population. Ten breeding couples are selected and the same chromosome may be selected more than once. Breeding is done using a two-point cross-over point. (See **Fig 2**). These two cross-over points produce an inner zone (shaded area in **Fig 2**) and an external zone (unshaded area). Each breeding couple produce two children. An example is shown in **Fig 3**. The first child inherits the outer zone from parent A and the inner zone from parent B. With the second child the process is reversed.

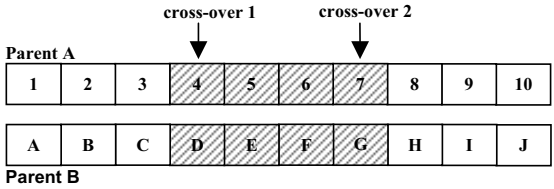
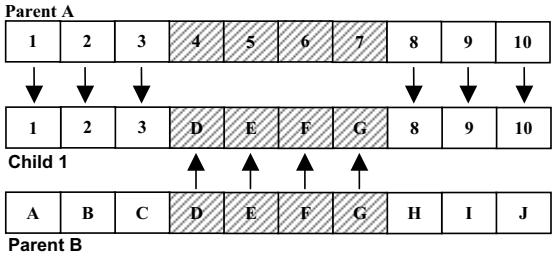


Figure 2 – selection of cross-over zones



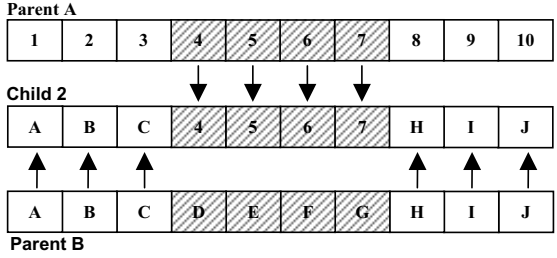


Figure 3 – example of breeding child

Once the breeding process is finished the GA undergoes mutation. Whereas the unit for operating the GA selection is the whole gene, in mutation it is the internal structure of the gene, which is being modified. Two kinds of mutation occur. The first is used to try and allow for slight changes within a narrow volume in the solution space. It is an effort to encourage convergence onto a possible solution. Each gene within the chromosome has a slight probability to be chosen for mutation and only one node of the 4 nodes within the gene will possibility be mutated. This *type-A*

mutation is only carried out on the parents involved in breeding. In the second type of mutation increasing the probability of a gene being selected for mutation encourages a wide expansion of volume in the solution space and if this happens, all nodes of the linguistic variable may be changed. This *type-B mutation* is carried out on those chromosomes not selected as parents. No mutation is carried out on the children.

To allow for the fact volumes of the solution space that have not been investigated might hold the optimal solution – five new chromosomes are created at the beginning of each new generation. If they turn out to be bad creations they will simply be eliminated during evaluation and sorting of the chromosome structure. – If however they show promise they will be allowed to "complete" with the other possibilities. Of course, more details regarding observations on the experiment will be presented in the final paper.

5. Concluding Remarks

This paper offers a new line of thinking in developing intelligent tutor systems or learning environments that can dynamically adapt its scheduling of teaching to result in quicker, more efficient tutoring of an individual student. The full results of the experiment will be published in the final paper. However there is much work still to do in the future before we can come close to a real-life intelligent tutoring system. More study must be done on analysing the cost of the system in terms of processing resources. How well will the system perform when confronted with a real-life situation - real students and a more complex knowledge structure? We have already started to develop a student model that represents more accurately any student who works with the system. These improved knowledge structures and student models will, for sure, add greater efficiency to the system. While we are, at the moment, concentrating on a Fuzzy-GA hybrid method we still need to look at the possibility of other hybrid intelligent techniques to be used in making our ITS more efficient.

References

1. Chen W (2003) Supporting Collaborative Knowledge Building with Intelligent Agents. In: Palade V, Howlett J R, Jain L (eds) Knowledge-Based Intelligent Information and Engineering Systems. Springer-Verlag, Berlin Heidelberg New York, Part II, pp 238-244
2. Kosba, E., Dimitrova, V., Boyle, R.; *Using Fuzzy Techniques to Model Students in Web-Based Learning Envirionments*. 7th international Conference, KES 2003, Oxford, UK, Proceedings Part II, p222.
3. Lascio, L.; Fischetti, E., Gisolfi, A.; *A Fuzzy-Based Approach to Stereotype Selection in Hypermedia.* User Modeling and User-Adapted Interaction, volume 9 Issue 4, April 1999.
4. Millan, E., Perez-de-la-Cruz, J. L., Garcia, F.; *Dynamic versus Static Student Models Based on Bayesian Networks: An Empirical Study*. 7th International Conference, KES 2003, Oxford, UK, Proceedings Part II, p11337.
5. McTaggart, J. *Intelligent Tutoring Systems and Education for the Future*. CI 512X Literature Review, April 30, pp2., 2001.
6. Nkambou, R. *Managing Inference Process in Student Modeling for Intelligent Tutoring Systems.* Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligent, November 1999.

6. Negoita, M., Pritchard, D. *Some Test Problems Regarding Intelligent Tutoring Systems*. 7th International Conference, KES 2003a, Oxford, UK. Proceedings Part II, p986.
7. Negoita, M. Pritchard, D. *Testing Intelligent Tutoring Systems by Virtual Students*. International Conference on Machine Learning and Applications, ICMLA 2003b, Proceedings, p98.
8. Papanikolaou, K.A., Grigoriadou, M., Kornilakis, H., Magoulas, G.D., *Personalizing the Interactioin in a Web-based Educational Hypermedia System: the case of INSPIRE.* User Modeling and User-Adapted Interaction, Volume 13, Issue 3, August 2003.
9. Park, C., Kim, M, *Development of a Level-Based Instruction Model in Web-Based Education*. 7th international Conference, KES 2003, Oxford, UK, Proceedings Part II, p215.
10. Smyth B, (2003) Intelligent Navigation on the Mobile Internet . In: Palade V, Howlett J R, Jain L (eds) Knowledge-Based Intelligent Information and Engineering Systems. Springer-Verlag, Berlin Heidelberg New York, Part I, pp 17-19

Consistency Conditions for Fuzzy Choice Functions

Irina Georgescu

Turku Centre for Computer Science
Åbo Akademi University
Institute for Advanced Management Systems Research
Lemminkäisenkatu 14, FIN-20520 Turku, Finland
irina.georgescu@abo.fi

Abstract. We introduce the consistency conditions $F\alpha$, $F\beta$, $F\delta$ as fuzzy forms of Sen's properties α, β and δ. One first result shows that a fuzzy choice function satisfies $F\alpha$, $F\beta$ if and only if the congruence axiom $WFCA$ holds. The second one shows that if h is a normal fuzzy choice function then $F\delta$ holds if and only if the associated preference relation R is quasi-transitive.

1 Introduction

The revealed preference theory for a competitive consumer was developed by Samuelson [15] and Houthakker [11] and extended to an axiomatic framework by Uzawa [21], Arrow [1], Sen [16, 17, 18] and many others.

If X is a universal set of alternatives then the rationality of a choice function C on X is expressed in terms of a preference relation R on X. C has the meaning of selecting the R-best alternatives from a conceivable set of alternatives. Consistency and rationality of a consumer are considered to have distinct meaning [6]: "Consistency is concerned with what happens to choices when the set of available alternatives expands or contracts. Rationality is concerned with how the choices are related to a binary relation on the set of alternatives."

We distinguish contraction consistency conditions and expansion consistency conditions. The first ones give "information on what elements are chosen from subsets from information of what are chosen from supersets" while the second ones give "information on what is chosen from supersets from information on what is chosen from subsets" ([12], pp. 30-31).

Banerjee [3] studies choice functions with a fuzzy behaviour. The domain of a Banerjee choice function consists of all non-empty finite subsets of X and the range consists of non-zero fuzzy subsets of X.

I. Georgescu: *Consistency Conditions for Fuzzy Choice Functions*, Advances in Soft Computing **2**, 317–324 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

In this paper we consider fuzzy choice functions defined on a family of non-zero fuzzy subsets of X with the range consisting of non-zero fuzzy subsets of X. Banerjee fuzzifies only the range of a choice function; in our approach both the domain and the range of a choice function are fuzzified – in this case the results on fuzzy choice functions get a much deeper meaning.

This paper contains generalizations to fuzzy choice functions of results in [16, 17, 18]. Our investigation is focused on the consistency conditions concerning the behaviour of fuzzy choice functions.

Section 2 contains some generalities on fuzzy relations. In Sect. 3 we recall the fuzzy revealed preference axioms $WAFRP$, $SAFRP$, the fuzzy congruence axioms $WFCA$, $SFCA$ and we discuss the framework of this paper under two natural hypotheses on fuzzy choice functions. Section 4 deals with consistency conditions $F\alpha$, $F\beta$, the fuzzy forms of Sen's properties α, β [16]. The formulation of $F\alpha$ uses the subsethood function $I(.,.)$ defined in Sect. 2 and $F\beta$ is stated in terms of $\wedge$ and $\rightarrow$. We prove that a fuzzy choice function satisfies $F\alpha$, $F\beta$ if and only if $WFCA$ holds for h. Section 5 is concerned with condition $F\delta$, a fuzzy version of Sen's condition δ. The main result shows that for a normal fuzzy choice function, the associated fuzzy preference relation R is quasi-transitive if and only if condition $F\delta$ holds.

2 Basic Facts on Fuzzy Relations

The results of this paper will be formulated and proved in the framework of the fuzzy set theory based on the Gödel t-norm. Let us consider the residuated lattice $([0,1], \vee, \wedge, \rightarrow, 0, 1)$ where $a \vee b = \max(a, b)$, $a \wedge b = \min(a, b)$, $a \rightarrow b = \bigvee\{c \in [0,1] | a \wedge c \leq b\}$. We also define the negation $\neg$ by putting $\neg a = a \rightarrow 0$. The biresiduum $\leftrightarrow$ is a binary operation on $[0,1]$ defined by $a \leftrightarrow b = (a \rightarrow b) \wedge (b \rightarrow a)$.

Let X be a non-empty set . A fuzzy subset of X is a function $A : X \rightarrow [0,1]$. We denote by $\mathcal{P}(X)$ the crisp subsets of X and by $\mathcal{F}(X)$ the fuzzy subsets of X. For any $A, B \in \mathcal{F}(X)$ we write $A \subseteq B$ if $A(x) \leq B(x)$ for all $x \in X$. A fuzzy subset A of X is non-zero if $A(x) \neq 0$ for some $x \in X$; a fuzzy subset A of X is normal if $A(x) = 1$ for some $x \in X$.

The support of $A \in \mathcal{F}(X)$ is defined by $suppA = \{x \in X | A(x) > 0\}$. If $x_1, \ldots, x_n \in X$ then $[x_1, \ldots, x_n]$ denotes the characteristic function of the set $\{x_1, \ldots, x_n\}$.

A fuzzy relation R on X is a fuzzy subset of X^2, i.e. a function $R : X^2 \rightarrow [0,1]$. Let R be a fuzzy relation on X. R will be called

- reflexive if $R(x,x) = 1$ for any $x \in X$;
- transitive if $R(x,y) \wedge R(y,z) \leq R(x,z)$ for all $x, y, z \in X$;
- total if $R(x,y) > 0$ or $R(y,x) > 0$ for all distinct $x, y \in X$;
- strongly total if $R(x,y) = 1$ or $R(y,x) = 1$ for all distinct $x, y \in X$.

A fuzzy preference relation on X is a binary fuzzy relation R on X. A regular preference on X is a fuzzy preference relation on X which is reflexive, transitive and strongly total.

The transitive closure of a fuzzy relation R is the intersection $T(R)$ of all transitive fuzzy relations on X including R. Of course R is transitive if and only if $T(R) = R$.

For any $A, B \in \mathcal{F}(X)$ we shall denote $I(A,B) = \bigwedge_{x \in X} (A(x) \to B(x))$. The number $I(A,B) \in [0,1]$ is called the subsethood degree of A in B and expresses the truth value of the statement "each element of A is an element of B." (see [5]). Note that for any $A, B \in \mathcal{F}(X)$, $I(A,B) = 1$ if and only if $A \subseteq B$.

Besides $I(.,.)$, there exist plenty of indicators expressing the inclusion of one fuzzy set into another. A large class of such indicators was axiomatically developed by Sinha and Dougherty [7, 19].

3 Fuzzy Choice Functions

A rich literature is dedicated to fuzzy preference relations [2]. Most authors consider that the social choice is governed by fuzzy preferences but the act of choice is exact [4]. They study crisp choice functions associated with a fuzzy preference relation.

In [3] Banerjee admits the vagueness of the act of choice and studies choice functions with a fuzzy behavior. The domain of a Banerjee choice function is made of finite sets of alternatives and the range is made of fuzzy sets.

In [8, 9] we have considered a larger class of fuzzy choice functions: the domain and the range of a fuzzy choice function is made of fuzzy sets of alternatives.

A fuzzy choice space is a pair $< X, \mathcal{B} >$ where X is a non-empty set and $\mathcal{B}$ is a non-empty family of non-zero fuzzy subsets of X. A fuzzy choice function on a fuzzy choice space $< X, \mathcal{B} >$ is a function $h : \mathcal{B} \to \mathcal{F}(X)$ such that for each $B \in \mathcal{B}$, $h(B)$ is non-zero and $h(B) \subseteq B$.

Although a part of fuzzy choice functions theory can be developed in this general setting, in order to obtain deeper results we need some natural hypotheses. In this paper we work under the following hypotheses:

(H_1) Every $B \in \mathcal{B}$ and $h(B)$ are normal fuzzy subsets of X;

(H_2) $\mathcal{B}$ includes all functions $[x_1, \ldots, x_n]$, $n \geq 1$, $x_1, \ldots, x_n \in X$.

For the crisp case ($\mathcal{B} \subseteq \mathcal{P}(X)$) the hypothesis (H_1) asserts that any $B \in \mathcal{B}$ and $h(B)$ are non-empty, hence (H_1) is automatically fulfilled in accordance with the definition of a choice function; for the same case, (H_2) asserts that $\mathcal{B}$ includes all finite subsets of X.

We remark that for crisp choice functions, the results in [1, 16, 17] are proved in the hypothesis that $\mathcal{B} \subseteq \mathcal{P}(X)$ contains all non-empty finite subsets

of X. Thus (H_2) appears as a very natural extension of this hypothesis in the crisp case.

Let $< X, \mathcal{B} >$ be a choice space and Q a fuzzy preference relation on X. We define a function $h_Q : \mathcal{B} \to \mathcal{F}(X)$ by $h_Q(S)(x) = S(x) \wedge \bigwedge_{y \in X} (S(x) \to Q(x, y))$ for any $S \in \mathcal{B}$ and $x \in X$. In general h_Q is not a fuzzy choice function. A fuzzy choice function h is rational if $h = h_Q$ for some fuzzy preference relation Q on X; in this case Q is called the rationalization of h.

We recall the definitions of the following fuzzy relations on X [8].

Let $h : \mathcal{B} \to \mathcal{F}(X)$ be a fuzzy choice function on $< X, \mathcal{B} >$.

Definition 1. *(i)* $R(x, y) = \bigvee_{S \in \mathcal{B}} (h(S)(x) \wedge S(y))$;
(ii) $P(x, y) = R(x, y) \wedge \neg R(y, x)$;
(iii) $I(x, y) = R(x, y) \wedge R(y, x)$
for any $x, y \in X$.

Definition 2. *(i)* $\bar{R}(x, y) = h([x, y])(x)$;
(ii) $\bar{P}(x, y) = \bar{R}(x, y) \wedge \neg \bar{R}(y, x)$;
(iii) $\bar{I}(x, y) = \bar{R}(x, y) \wedge \bar{R}(y, x)$
for any $x, y \in X$.

Definition 3. *(i)* $\tilde{P}(x, y) = \bigvee_{S \in \mathcal{B}} (h(S)(x) \wedge S(y) \wedge \neg h(S)(y))$;
(ii) $\tilde{R}(x, y) = \neg \tilde{P}(y, x)$;
(iii) $\tilde{I}(x, y) = \tilde{R}(x, y) \wedge \tilde{R}(y, x)$
for any $x, y \in X$.

We denote by W the transitive closure of R and by P^* the transitive closure of $\tilde{P}$.

Remark 1. For any fuzzy choice function $h : \mathcal{B} \to \mathcal{F}(X)$ let us consider its "image" $\hat{h} = h_R$. $h(S) \subseteq \hat{h}(S)$ for each $S \in \mathcal{B}$ (see [8], Proposition 4.5); thus $\hat{h}$ is always a fuzzy choice function.

A fuzzy choice function h is normal if $h = \hat{h}$.
Now we shall consider the following axioms of fuzzy revealed preference:
$WAFRP$ (Weak Axiom of Fuzzy Revealed Preference)
$\tilde{P}(x, y) \leq \neg R(y, x)$ for all $x, y \in X$;
$SAFRP$ (Strong Axiom of Fuzzy Revealed Preference)
$P^*(x, y) \leq \neg R(y, x)$ for all $x, y \in X$.
and the axioms of congruence for fuzzy choice functions:
$WFCA$ (Weak Fuzzy Congruence Axiom)
For any $S \in \mathcal{B}$ and $x, y \in X$: $R(x, y) \wedge h(S)(y) \wedge S(x) \leq h(S)(x)$.
$SFCA$ (Strong Fuzzy Congruence Axiom)
For any $S \in \mathcal{B}$ and $x, y \in X$: $W(x, y) \wedge h(S)(y) \wedge S(x) \leq h(S)(x)$.

Remark 2. Axioms $WAFRP$, $SAFRP$, $WFCA$, $SFCA$ are fuzzy versions of axioms $WARP$, $SARP$, WCA, SCA in classical consumer theory.

Remark 3. Since $\tilde{P}(x,y) \leq P^*(x,y)$ and $R(x,y) \leq W(x,y)$ for any $x, y \in X$ the following implications hold true for any fuzzy choice function h: $SAFRP \Rightarrow WAFRP$; $SFCA \Rightarrow WFCA$.

If $\tilde{P}$ (resp. R) is transitive then $\tilde{P} = P^*$ (resp. $R = W$), therefore in this case $SAFRP \Leftrightarrow WAFRP$ (resp. $SFCA \Leftrightarrow WFCA$).

Theorem 1. *For a fuzzy choice function h the following assertions are equivalent:*

(i) R is a regular preference and h is normal;
(ii) $\bar{R}$ is a regular preference and h is normal;
(iii) h verifies $WFCA$;
(iv) h verifies $SFCA$.

The previous theorem is a part of a more complete result of [8] that generalizes to fuzzy choice functions an important theorem of Sen [16].

4 Conditions α and β

Conditions α and β were introduced by Sen [16] for crisp choice functions. In this section we will consider fuzzy versions $F\alpha$ and $F\beta$ of these conditions and we will prove, for the Gödel t-norm $\wedge$, that a fuzzy choice function h satisfies $F\alpha$ and $F\beta$ if and only if h satisfies $WFCA$. We consider a class of fuzzy choice function satisfying a new hypothesis (U). In the crisp case (U) expresses that $h(S)$ is a singleton for each S. Among results under hypothesis (U) there is the equivalence between $F\alpha$ and $WFCA$.

We recall the (crisp) conditions α and β. Let $h : \mathcal{B} \to \mathcal{P}(X)$ be a crisp choice function.

Condition α. For any $S, T \in \mathcal{B}$ and for any $x \in X$, we have the implication $x \in S, x \in h(S)$ and $S \subseteq T \Rightarrow x \in h(S)$.

Condition β . For any $S, T \in \mathcal{B}$ and for any $x, y \in X$, we have the implication $x, y \in h(S)$ and $S \subseteq T \Rightarrow x \in h(T)$ if and only if $y \in h(T)$.

These two conditions can be extended to the fuzzy case.

Let $h : \mathcal{B} \to \mathcal{F}(X)$ be a fuzzy choice function on a choice space $< X, \mathcal{B} >$.

Condition $F\alpha$. For any $S, T \in \mathcal{B}$ and $x \in X$, $I(S,T) \wedge S(x) \wedge h(T)(x) \leq h(S)(x)$.

Condition Fβ. For any $S, T \in \mathcal{B}$ and $x, y \in X$, $I(S,T) \wedge h(S)(x) \wedge h(S)(y) \leq h(T)(x) \leftrightarrow h(T)(y)$ where $\leftrightarrow$ is the biresiduum of the minimum t-norm.

Obviously conditions $F\alpha$, $F\beta$ generalize α, β.

Proposition 1. *If h is a normal fuzzy choice function then $F\alpha$ is verified.*

Theorem 2. *For a fuzzy choice function h the following are equivalent:*

(1) h verifies conditions $F\alpha$, $F\beta$;
(2) $WFCA$ holds for h.

Remark 4. The previous theorem generalizes to fuzzy choice functions a result of Sen (see [16], (T8)).

In classical consumer theory a special case are choice functions $h : \mathcal{B} \to \mathcal{P}(X)$ with the property that $h(S)$ is a singleton for any $S \in \mathcal{B}$. We generalize this case considering fuzzy choice functions $h : \mathcal{B} \to \mathcal{F}(X)$ that verify (U):

For any $S \in \mathcal{B}$, $h(B) = [x]$, for some $x \in X$.

For fuzzy choice functions h that verify U there is a unique $x \in X$ such that

$$h(S)(y) = \begin{cases} 1 \text{ if } y = x \\ 0 \text{ if } y \neq x. \end{cases}$$

Proposition 2. *If a normal fuzzy choice function h verifies (U) then R is a regular preference.*

Proposition 3. *Let h be a fuzzy choice function satisfying (U). The following assertions are equivalent:*

(1) $WFCA$ holds for h;
(2) h satisfies condition $F\alpha$.

5 Quasi-transitivity and Condition $F\delta$

This section deals with a fuzzy form $F\delta$ of Sen's condition δ. For a normal fuzzy choice function h we prove that the associated fuzzy relation R is quasi-transitive if and only if condition $F\delta$ holds.

Let Q be a fuzzy relation on X and P_Q be the fuzzy relation on X defined by $P_Q(x, y) = Q(x, y) \wedge \neg Q(y, x)$ for any $x, y \in X$. If R is the fuzzy relation associated with a fuzzy choice function h then $P_R = P$.

Recall that a fuzzy relation Q on X is quasi-transitive if $P_Q(x, y) \wedge P_Q(y, z) \leq P_Q(x, z)$ for any $x, y, z \in X$.

Proposition 4. *Let Q be a reflexive and strongly total fuzzy relation on X. If Q is transitive then Q is quasi-transitive.*

Definition 4. *We say that the fuzzy choice function h satisfies the condition $F\delta$ if for any $S = [a_1, \ldots, a_n]$, $T = [b_1, \ldots, b_m]$ in $\mathcal{B}$ and for any $x, y \in X$, $I(S, T) \leq (h(S)(x) \wedge h(S)(y)) \to \neg(h(T)(x) \wedge \bigwedge_{t \neq x} \neg h(T)(t))$.*

In case of crisp choice functions condition $F\delta$ is exactly condition δ.

Theorem 3. *If h is a normal fuzzy choice function, then R is quasi-transitive if and only if condition $F\delta$ is verified.*

Remark 5. Theorem 3 is the generalization for fuzzy choice functions of a Sen result. (see [16], $(T10)$)

6 Concluding Remarks

In the real world most of the preferences are fuzzy, consequently they are more adequately modelled by binary fuzzy relations. Banerjee's thesis in [3] is that "If preferences are permitted to be fuzzy it seems natural to permit choice functions to be fuzzy as well. This also tallies with the experience".

In an abstract setting this leads to a mathematical treatment of fuzzy choice functions. Then we must define what we mean by a rational behavior of a fuzzy choice function. The rationality of a fuzzy choice function was defined in [8, 9]. There this concept was investigated by developing a revealed preference theory for fuzzy choice functions.

This paper completes the results of [8, 9]. Our main contribution is to study consistency conditions $F\alpha$, $F\beta$ and $F\gamma$, fuzzy forms of Sen's conditions α, β, γ. [16, 17, 18]. We generalize in a fuzzy framework some important theorems of Sen from [17, 18].

Consistency conditions for fuzzy choice functions assure a knowledge complementary to the one realized by properties of congruence and revealed preference. They show to what extent the act of fuzzy choice is perturbed when varying between fuzzy subsets and supersets. Diverse in their content, consistency conditions sometimes offer equivalences for properties of rationality. For example, Weak Fuzzy Congruence Axiom ($WFCA$) is equivalent to the conjunction of $F\alpha$ and $F\beta$. In other cases, a consistency condition establishes the nature of the preference relation that defines the rationality (see Theorem 3).

Consistency properties $F\alpha$, $F\beta$ and $F\delta$ are formulated in terms of residual structure of the interval $[0, 1]$. The use of the inclusion operator $I(.,.)$ is more adequate to the fuzzy context. This is certified by the way the proofs of the classical theory of choice functions are converted in their fuzzy versions.

Open problem 1 Consistency conditions $F\alpha$, $F\beta$, $F\delta$ can also be formulated in terms of Sinha–Dougherty indicators. Which of the results proved in this paper are still valid in this axiomatic context?

Open problem 2 The definitions and the properties of this paper can be easily stated in a fuzzy set theory based on a continuous t-norm. Which results might be proved for an arbitrary continuous t-norm?

Open problem 3 In the treatment of consistency conditions $F\alpha$, $F\beta$, $F\gamma$, is it possible to eliminate or to relax hypotheses (H_1) and (H_2)?

Open problem 4 Investigate the fuzzy version of Sen's consistency condition γ [16].

References

1. Arrow K J (1959) Economica, 26:121–127
2. De Baets B, Van de Walle B, Kerre E (1995) J Fuzzy Math 3:373–381
3. Banerjee A (1995) Fuzzy Sets Syst 70:31–43

4. Barret C R, Pattanaik P K, Salles M (1990) Fuzzy Sets Syst 34:197–212
5. Bělohlávek R (2002) Fuzzy relational systems. Foundations and principles. Kluwer
6. Bordes G (1976) Rev Ec Studies 43:451–457
7. Cornelis C, Van der Donck C, Kerre E (2003) Fuzzy Sets Syst 134:283–295
8. Georgescu I (2004) J Syst Sci Syst Eng, forthcoming
9. Georgescu I (2004) Rationality and congruence axioms for fuzzy choice functions. In Proceedings of ESTLYF 2004, Jaén, Spain, forthcoming
10. Hájek P (1998) Metamathematics of fuzzy logic. Kluwer
11. Houthakker H S (1950) Economica 17:159–174
12. Kelly J S (1978) Arrow impossibility theorems. Academic Press, New York
13. Klement E P, Mesiar R, Pap E (2000) Triangular norms. Kluwer
14. Richter M (1966) Econometrica 34:635–645
15. Samuelson P A (1938) Economica 5:61–71
16. Sen A K (1971) Rev Ec Studies 38:307–312
17. Sen A K (1969) Rev Ec Studies 36:381–393
18. Sen A K (1977) Econometrica 45:53–89
19. Sinha D, Dougherty E R (1993) Fuzzy Sets Syst 55:15–42
20. Turunen E (1999) Mathematics behind fuzzy logic. Physica-Verlag
21. Uzawa H (1959) Preference and rational choice in the theory of consumption. In: Arrow K J, Karlin S, Suppes P (eds) Mathematical methods in social sciences. Stanford, Stanford University Press
22. Zimmermann H J (1984) Fuzzy set theory and its applications. Kluwer

Applications of Fuzzy Systems

A Fuzzy Logic Application to Environment Management System: A Case Study for Goksu Streams Water Quality Assesment

A. Yardimci[1], A. Muhammetoglu[2], H. Oguz[2]

[1] Department of Instrumentation&Process Control Technology, TBMYO, Akdeniz University, Antalya, Turkey
yardimci@akdeniz.edu.tr

[2] Departments of Environmental Pollution & Control, TBMYO, Akdeniz University, Antalya, Turkey
muhammetoglu@usa.net
hknoguz@akdeniz.edu.tr

Abstract. Water quality management is an important issue of relevance in the context of present times. Water quality indices are computed for classification of water wherein the integration of parametric information on water quality data and the expert's knowledgebase on their importance & weights are considered. Considerable uncertainties are involved in the process of defining water quality for specific usage. Antalya City, located along the coasts of Mediterranean Sea in Turkey, is famous worldwide due to its tourism potential. Antalya City has a beautiful landscape composed of mountains, forests, beautiful beaches and the sea. In order to apply sustainable tourism principles in Antalya, the protection of valuable environmental resources gains a particular importance. A land survey study was carried out to determine the pollution loads of Bogacay Stream, an important land-based pollution source of Antalya City, for one year duration. According to the water quality classifications obtained from Fuzzy Logic, water quality changes temporally in Bogacay Stream and an occasional critical level of water quality was determined in July which coincides with the peak use of the beach for recreational activities. Goksu Stream is the main source of Bogacay Stream. It always carries main part of water to Bogacay. So, on a large scale Goksu Stream determines the Bogacay Stream's water quality.

In this study, both field measurements and lab analyses have been realized and pollution loads of the polluting parameters have been calculated for Goksu Stream. A recent study has been realized to assess the water quality of Goksu Stream using Fuzzy Logic approach.

A. Yardimci et al.: *A Fuzzy Logic Application to Environment Management System: A Case Study for Goksu Streams Water Quality Assesment*, Advances in Soft Computing **2**, 327–340 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

1 Introduction

Antalya province has more than 600-km coastal line on the Turkish Mediterranean Sea. The cleanest coasts in Turkey are found in Antalya Bay. Antalya is one of the most popular tourism places in Turkey for both the local and foreign tourists. The city has two important beaches namely Lara Beach in the east and Konyaaltı Beach in the west of the city. There are a few streams discharging their water to Konyaaltı Beach and affect the water quality for recreational purposes along the beach. Among these streams, Boğaçay is the most important one in terms of the flow rate and pollution load. Goksu Stream is the main source of Boğaçay. In summer, many beach visitors prefer to swim at the outlet of Boğaçay Stream to the Mediterranean Sea because of its relatively cooler water.

A recent research study has been carried out to assess the water quality of Goksu Stream. An extensive monthly field measurement and lab analyses have been realized for one year to determine the physical, chemical and biological characteristics of Goksu Stream (Oğuz, 2001). Many parameters such as flow rate, salinity, conductivity, Biochemical Oxygen Demand (BOD), Chemical Oxygen Demand (COD), total suspended solids, nitrogen and phosphorous compounds, total and fecal coliforms have been determined. The results show temporal variations with somewhat high total and fecal coliform content. The results of measurements and analyses have been used to assess the water quality level according to water quality classifications used for inland waters in Turkey.

In this study, the Fuzzy Logic, has been utilized for the assessment of Goksu Stream water quality. Fuzzy theory emerged in the second half of this century by challenging basic assumptions of three classical theories: the assumption of sharp boundaries in classical set theory; the assumption of classical (Aristotelian) logic that each proposition must either be true or false; and the assumption of additivity in classical measure theory and, in particular, probability theory. The Fuzzy Logic has been recently applied to assess the water quality of rivers and other water systems.

2 Fuzzy Logic Application to Bogacay Stream

In this study, a fuzzy logic control was used to evaluate to the water quality of Goksu Stream before the discharge point to the Bogacay Stream. Dissolved oxygen saturation percent (DO), fecal coliform (FCol), chemical oxygen demand (COD) and total phosphorus (TP) parameters have been selected to define the water quality level. The adopted water quality evaluation system is presented in Figure 1. All of the input membership functions have three linguistic terms (good, normal, bad). The water quality output membership function has four linguistic terms (I, II, III, IV) being identical to inland water quality classifications used in Turkey (Ref.).

Water quality experts are identified and relevant field data. Rule base was prepared by the collaboration of water quality experts. The membership functions (MBF) of the selected water quality parameters; DO, FCol, COD, TP an WQI are shown in Figure 2.a, 2.b, 2.c, 2.d and 2.e.

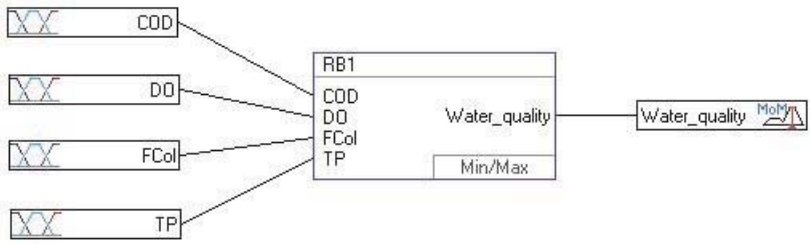


Figure 1. The adopted water quality evaluation system

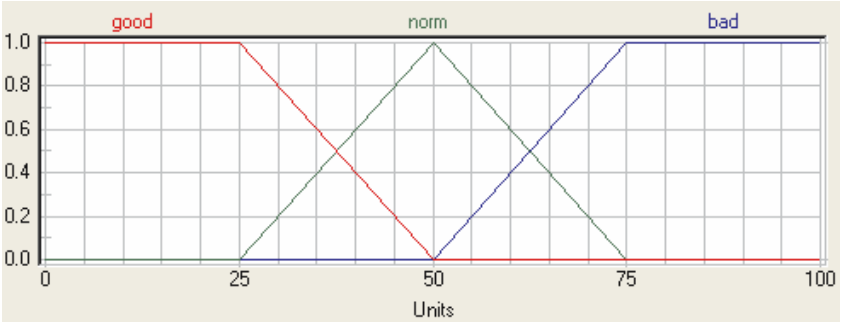


Figure 2.a. MBF of "COD"

Figure 2.b. MBF of "DO"

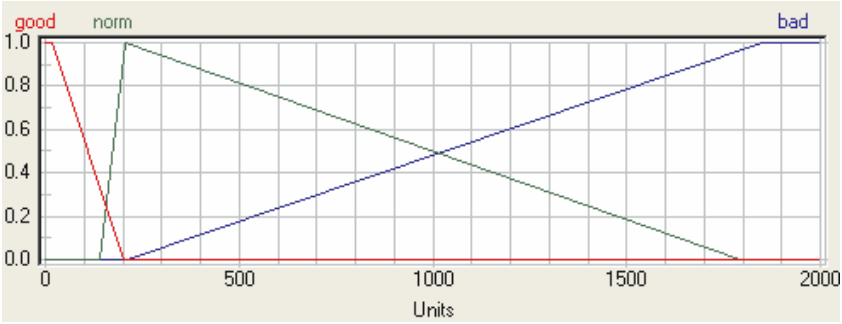


Figure 2.c. MBF of "FCol"

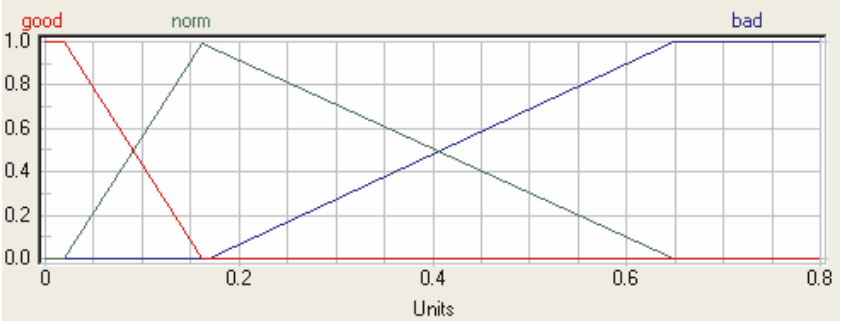


Figure 2.d. MBF of "TP"

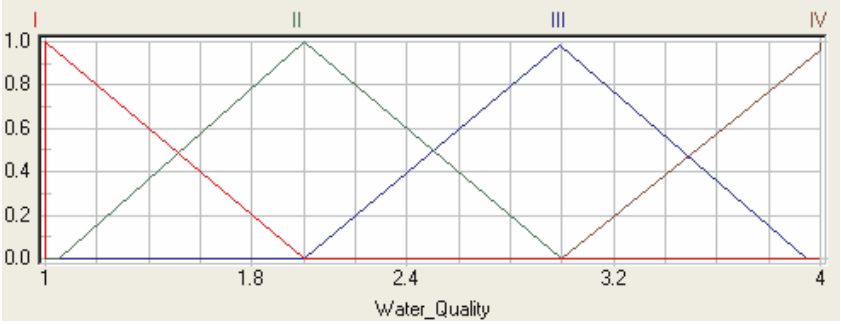


Figure 2.e. MBF of "Water Quality"

The computation of fuzzy rules is called fuzzy rule inference. The inference is a calculus consisting of the steps: aggregation, composition and result aggregation. The first step of the fuzzy inference aggregation, determines the degree to which

the complete part of the rule is fulfilled. The minimum operator has been utilized for computation of the fuzzy rule.

$$\text{AND: } \mu_{IF} = \min_i (\mu_i) \quad \text{.....(2)}$$

The result produced from the evaluation of fuzzy rules is still fuzzy. Out membership function is used to retranslate the fuzzy output into a crisp value. This translation is known as defuzzification and can be performed using several methods. To obtain clear evaluation of water quality classified as Class I, II, II and IV, Mean of Maximum (MoM) defuzzification method was preferred. The Mean of Maximum method, computes a system output only for the term with the highest resulting degree of support. If the maximum is not unique, the mean of maximizing interval is computed. The MoM procedure evaluates the most significant of the different terms j of a linguistic output variable then obtain Y by using maximum Y_j of the each term j.

$$Y = Y_j \, (\mu_{Result,\, Term,\, Max}) \quad \text{.....(3)}$$

Hypothetical example

Before applying the Fuzzy logic to the measured concentrations of Goksu Stream, an explanatory hypothetical example is described below to explain clearly how the water quality evaluation system works. In this example, the selected water quality parameters are assigned the following values;

- DO (Dissolved oxygen saturation percent, %): 85
- FCol (Fecal coliform number, No/100 ml): 600
- COD (Chemical oxygen demand concentration, mg/l): 20
- TP (Total phosphorus concentration, mg/l as PO_4-P): 0.0525

The water quality classification of the given example will be done using two different approaches, firstly the classical one and secondly the Fuzzy Logic.

Classical approach

First of all, a general water quality classification will be done using the assigned values of the four water quality parameters, according to the inland water quality classification levels used in Turkey (TÇV, 2002). The inland water quality classifications are presented in Table 1 and definitions of the water quality classes used in Table 1 are described below. According to the classification described in Table 1, the classification of the pre-assigned water quality levels is given in Table 2.

- **Class I** : *very clean water* (requires only disinfection to be used for water supply, suitable for recreational activities, trout production, animal husbandry and farms)

- **Class II**: *slightly polluted water* (requires an adequate treatment to be used for water supply, suitable for recreational activities, not suitable for trout production, may be used as irrigation water if the related standards are met)
- **Class III**: *polluted water* (requires an adequate treatment to be used for industrial supply except food and textile industries)
- **Class IV**: *very polluted water* (describes a poorer water quality class being worse than the above mentioned classes)

Table 1. Inland water quality classification levels

Water Quality Parameters	Water Quality Classes			
	I	**II**	**III**	**IV**
DO (Sat. %)	90	70	40	<40
FCol (No/ 100 ml)	10	200	2000	>2000
COD (mg/l)	25	50	70	>70
TP (mg/l)	0.02	0.16	0.65	>0.65

Table 2. Classification of the pre-assigned water quality levels

Water Quality Parameters	Value	Water Quality Class
DO (Sat. %)	85	**I**
FCol (No/ 100 ml)	600	**II**
COD (mg/l)	20	**I**
TP (mg/l)	0.0525	**I**

As can be observed from the above example, the values used as limits for different water quality classes do not present a range and most of the time, the values of the parameters lie in between two water quality classes. As an example, when we want to classify a water quality class for 85% dissolved oxygen saturation, the real water quality class is between I and II but to give a definite class, we define it as Class I. The same situation is valid for Fecal Coliform as well. Additionally, if we want to define overall water quality class from the results of four parameters, there is again a difficulty because three parameters are classified as Class I, while the fourth parameter is classified as Class II. Again using our judgment, we conclude that, the water quality can be described as first class (Class I) according to the assigned values of the four parameters. As a result of this classical water quality classification practice, we observe that there is an ambiguity in the definition of the different classes and we need to use our judgment to reach an overall evaluation.

Fuzzy Logic approach

In the second approach, the Fuzzy Logic is used to describe the water quality for the given example. Figure 4 shows the selected input values and the calculated output related to these inputs. As a result of the analysis, the water quality is calculated as Class I (the first class) according to inland water quality classifications in Turkey. The output degree was obtained by *MoM method*. As mentioned before, this result has been produced from the evaluation of fuzzy rules and passed *defuzzication stage*, using *out membership function*, to retranslate the fuzzy output into a crisp value. Now, the question is, what are the mathematical operations being performed by this process. To find the answer, we should look at the rule base to see which rules are valid under this circumstance. Table 3 shows the running rules on the system for the defined input variables as mentioned above. Degree of support for all the running rules is given in Table 4.

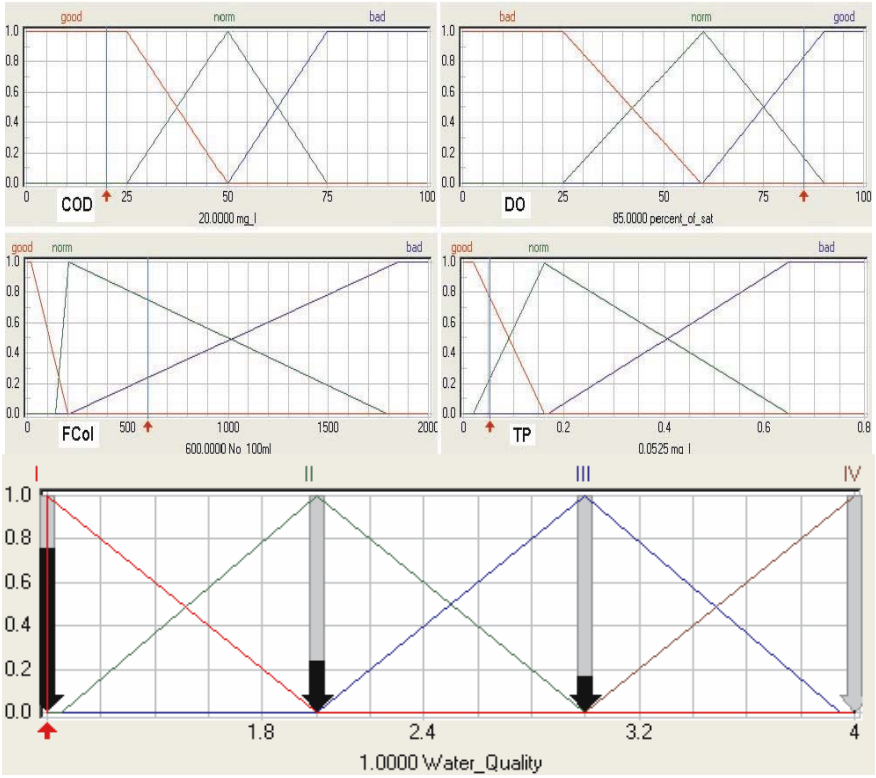


Figure 4. The evaluation of water quality for the assigned input values

Table 3. Valid rules for the solution of the given example

DO	FCol	COD	TP	WQ
good	norm	good	good	I
good	norm	good	norm	II
good	bad	good	good	II
good	bad	good	norm	II
norm	bad	good	good	II
norm	bad	good	norm	III
norm	norm	good	good	II
norm	norm	good	norm	II

Table 4. Degree of support (DoS) for all the running rules

DO	FCol	COD	TP	DoS	Water Quality Classes
Good 0.83	Norm 0.75	Good 1	Good 0.77	**0.75**	**I**
Good 0.83	Norm 0.75	Good 1	Norm 0.23	0.23	II
Good 0.83	Bad 0.25	Good 1	Good 0.77	0.25	II
Good 0.83	Bad 0.25	Good 1	Norm 0.23	0.23	II
Norm 0.17	Bad 0.25	Good 1	Good 0.77	0.17	II
Norm 0.17	Bad 0.25	Good 1	Norm 0.23	0.17	III
Norm 0.17	Norm 0.75	Good 1	Good 0.77	0.17	II
Norm 0.17	Norm 0.75	Good 1	Norm 0.23	0.17	II

As can be observed from Table 3, there are eight different cases to define the valid rules of the given example. These running rules end up with different values of degree of support as presented in Table 4. The most interesting point in this example is that, according to the values of degree of support, water quality is defined as Class II in six cases out of eight, and only one case defines the water quality as either Class I or Class III. However, according to the rules of defuzzification method, the highest value of degree of support becomes dominant and governs the decision. As a result, the water quality is obtained as Class I.

Application to Bogacay Stream

The Fuzzy Logic has been applied for the classification of water quality of Boğaçay Stream using the water quality measurement results of four parameters as used in the previous example. The selection of the number of parameters included in the fuzzy logic approach is mostly dependent on the type of application. When the number of selected parameters is four, as used in this application, the total number of rule bases reaches up to 81, which is a manageable combination. However, when the number of selected parameters is chosen as five, the total number of rule bases rises up to 243 and this selection creates an ambiguity in the decision making process which may lead to confusion. The results of the water quality measurements and the analysis for some selected parameters are given in Table 5 (Oğuz, 2001). The water quality classes (WQ) of Boğaçay Stream are shown in Figure 5 for each month, for one year duration of the measurement and sampling period. In this figure, the linguistic terms, I, II, III and IV are showed as different water quality classes. The first monthly record belongs to the first measurement session, which was in March 2000, and the last monthly record belongs to the measurement session in February 2001. Figure 6 depicts the fuzzy results of water quality (FRWQ) and the measured parameters along the months of the year.

Table 5. The results of water quality measurements and analysis for Goksu Stream

Month	*DO Sat. (%)*	*Turb. (NTU)*	*TS (mg/l)*	*TN (mg/l)*	*TP (mg/l)*	*COD (mg/l)*	*Fecal Coliform (No/100 ml)*
March	98.5	35.3	480	0.90	0.13	8	110
April	90.2	32.2	460	1.00	0.10	28	100
May	91.7	15.1	460	3.00	0.10	56	180
June	70.1	4.1	400	1.20	0.80	24	260
July	70.6	19.2	480	1.40	0.30	20	400
Aug.	76.1	3.3	420	2.00	0.35	8	600
Sept.	67.5	6.1	420	1.00	0.10	20	300
Oct.	99.4	40.2	420	1.00	0.10	16	480
Nov.	99.3	29.3	380	0.70	0.02	24	320
Dec	100.3	40.2	490	1.90	0.31	14	410
Jan.	100.6	42.5	520	1.20	0.35	40	310
Feb.	100.4	36.5	500	1.11	0.14	12	120

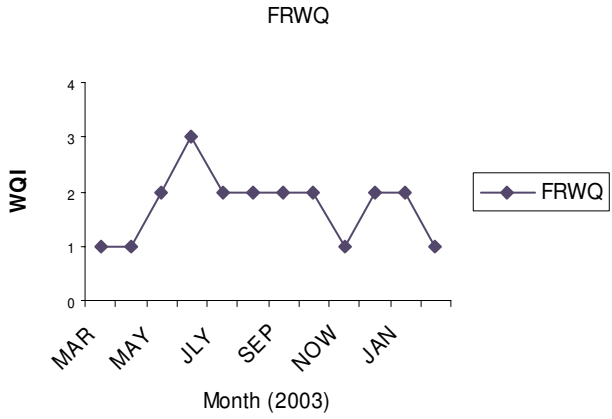
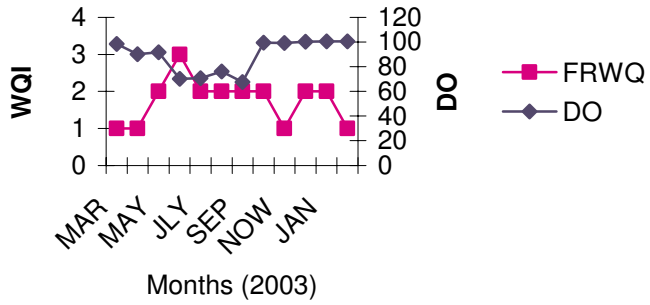
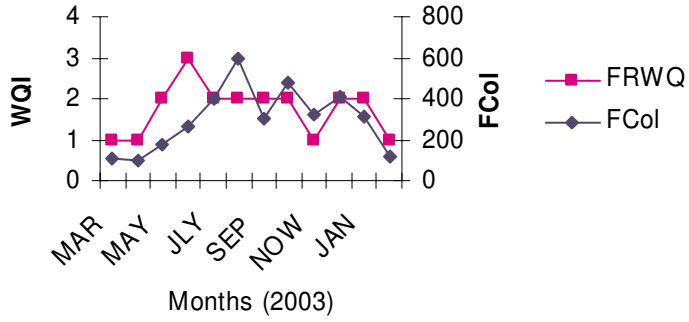


Figure 5. Changes of fuzzy results of water quality (FRWQ) versus the months of the year for Boğaçay Stream

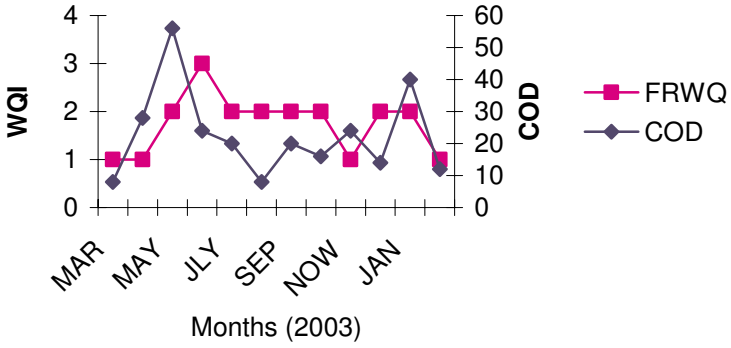
When the fuzzy results of water quality classes are observed carefully, it can be easily noticed that the 5th month, corresponding to July, has the worst water quality classification being Class III. This is a significant result of Fuzzy Logic approach as July represents nearly the most important month within the year, when Boğaçay stream water is highly used for swimming. Apart from this month, the water quality class changes between Class I and Class II for the other months. Additionally, it is observed from Figure 6 that, there is a good agreement between the water quality classes and the values of water quality parameters. In other words, when the water quality is classified as polluted, the measured water quality parameters also indicate water quality pollution. In this respect, the water quality classification of Boğaçay Stream using Fuzzy Logic approach can be considered as a successful case study.



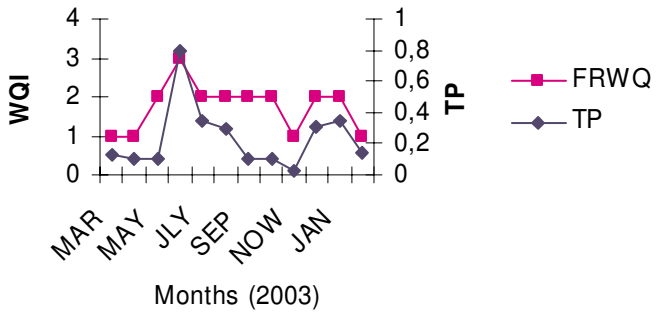
(a)



(b)



(c)



(d)

Figure 6.a.b.c.d Monthly changes of fuzzy results of water quality (FRWQ) and the parameter values of DO, FCol, COD and TP for Boğaçay Stream

3 Conclusion

Boğaçay Stream is an important land-based pollution source of Antalya City discharging its water to Antalya Bay through Konyaaltı Beach, which is a famous Blue Flag awarded beach. In order to assess the water quality of Boğaçay Stream and its pollution impacts on the sea environment, a land survey study has been carried out. During this one year study, monthly field measurements and lab analyses have been realized and the pollution loads of the polluting parameters have been calculated. Due to the existence of many factors influencing water quality, it was rather difficult to determine the water quality status of the stream. The Fuzzy Logic is a new approach which can analyze these factors by defining fuzzy expressions of degree of importance. In this study, the Fuzzy Logic approach has been applied to evaluate the water quality of Boğaçay Stream which exhibits high temporal variations. The values of dissolved oxygen saturation percent, chemical oxygen demand, fecal coliform and total phosphorus were included in the Fuzzy Logic approach. The number of selected parameters was restricted to four in order to avoid confusion in the decison making process. The output degree was obtained by *MoM method.* The fuzzy results have been produced from the evaluation of fuzzy rules and passed *defuzzication stage*, using *out membership function*, to retranslate the fuzzy output into a crisp value. The monthly changes of fuzzy results of Boğaçay Stream water quality have been obtained according to the inland water quality classifications used in Turkey. There was a good agreement between the fuzzy results and the values of the selected water quality parameters. The fuzzy results indicated a poor water quality for Boğaçay Stream for the month of July, which coincides with the intensive use of the stream water for recreational activities.

Acknowledgement

The authors acknowledge financial support from the Akdeniz University Unit of Scientific Research Projects Managements.

References

Bezdek J.C., (1993) Fuzzy models-what are they and why? *IEEE Transactions on Fuzzy Systems*; 1: 1-6.

Oğuz, H. (2001). Determination of Pollution Loads Carried to Antalya Bay From the Activities in Boğaçay Catchment Area and Recommendations for Solution, Ms. Thesis, Akdeniz University, Environmental Sciences, February 2001, 110 pp. (In Turkish)

Ross J.T, (1995) "Fuzzy Logic With Engineering Applications", *McGraw-Hill, USA.*

TÇV, (2002) Environmental Foundation of Turkey (Türkiye Çevre Vakfı), Environmental Law (Çevre Mevzuatı). (In Turkish)

Zadeh L.A., (1965) Fuzzysets. *Information and Control*; 8:338-352.

Zadeh L.A., (1978) “Fuzzy sets as a basis for a theory of possibilities” *Fuzzy Sets and Systems*; 1: 5-28.

Combination Rule of Normal Degrees on Automated Medical Diagnosis System (AMDS)

Yutaka Hata[1], Osamu Ishikawa[2], Syoji Kobashi[1] and Katsuya Kondo[1]

[1] Division of Computer Engineering, Graduate School of Engineering, University of Hyogo, 2167, Shosha, Himeji, 671-2201, Japan
[2] Ishikawa Hospital, 784, Bessho, Himeji, 671-0221, Japan

Abstract. This paper describes combination rule of normal degrees in human body in automated medical diagnosis system. The normal degree is defined in a framework of fuzzy logic. Physician usually examines whether a patient is either normal or abnormal for a disease. The normal degree is calculated in automated medical diagnosis system. The practical examples of medical images and blood test are described. In it, it is shown that union or inter-section operators are introduced for calculating normal degrees on MR meniscal tear images and blood test for diabetes.

1. Introduction

Currently, many computer assisted medical systems have been developed [1, 2]. Especially, computer assisted surgery systems, computer assisted diagnosis systems have been received much considerable attention. In it, imaging systems with Magnetic Resonance Imaging (MRI) and X-ray Computed Tomography Images have used to disclose the inside information of human body [2]. For these systems, the primary goal is supporting for medical diagnosis. In our previous works [3, 4], we defined a normal degree in the human body by employing theory of hierarchical definability (THD) [5, 6] by Zadeh. We represented a tree structure of the fuzzy information granularity of the normal degree, and we then proposed a calculation method of the normal degree. We then described an automated medical diagnosis system (AMDS), which automatically determines the normal degree of a medical sign for a patient. AMDS requires database of rules

Y. Hata et al.: *Combination Rule of Normal Degrees on Automated Medical Diagnosis System (AMDS)*, Advances in Soft Computing **2**, 339–347 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

and medical data for every user since the normal degree should be independently calculated by the personal rules of a user. That is, the system determines different degrees for different patients even if the examined medical signs are of the same. However, in these, there is no discussion on fundamental combination rule of the normal degree.

This paper describes two fundamental properties of combination rule for calculating the normal degree. In medical images of MR meniscus, we can examine meniscal tears by observing these three-dimensional shapes [7, 8]. This combination rule of the normal degrees is determined as fuzzy max or min operators. Secondly, the combination rule of two normal degrees of Glycemia and Hb-A1c are considered for a patient of diabetes. This rule is determined as fuzzy min operator. Finally, we conclude our technical results.

2. Normal Degree and AMDS

The normal degree is defined as follows [3].

Definition: Let X be a space of objects (points), with a generic element of X denoted by x. Thus X={x}. The normal degree, N(x), in X is characterized by a membership function N(x) which associates with each point in X a real number in the interval [0, 1]. The 0 stands for the no normality, i.e., abnormal, and 1 stands for the full normality.

We can calculate the normal degree with respect to a granule of medical data as follows [3]. For a granule, G;

Step 1) Determine sets in G with degree 1, i.e., full normal sets in G by medical experts.

Step 2) Extract enough features for the full normal sets to identify the full normality.

Step 3) Calculate the normal degrees for the remaining sets by comparing the features of full normal sets with those of remaining sets.

Next we describe automated medical diagnosis system (AMDS) [4]. AMDS consists of two parts: one is construction of AMDS and the other is determination of normal degrees.

2.1 Construction of AMDS

Suppose a granule, G, which is one of medical signs, clinical data, medical images and so on. For G, first, we collect the clinical/medical data, and do statistical analysis. Then we can obtain statistical results for

G. From these results and personal data for a user, we extract features enough to determine fuzzy if-rules for calculating a normal degree for G on a user. We construct fuzzy inference system using these fuzzy if-then rules. The system specification depends on the user. That is, a normal degree for G is different from user to user. Then, the fuzzy if-then rules and the features are stored in a database. That is, the database consists of these features and fuzzy if-then rules.

2.2 Determination of Normal Degree

Daily personal data for G is provided to AMDS. AMDS extracts this feature, and it calculates a normal degree by the fuzzy inference system for a user. Then, the provided data is also stored in the database.

3. Combination Rule of Normal Degrees

3.1 Representation of Normal degree

First, we consider normal degree for an attribute, such as items in blood test. For a granule, X, let xi (i=1, 2, …, n) be an attribute, where n is the number of these attributes xi. Then N(X) can be represented by,

$$N(X)=N(x1)/x1 + N(x2)/x2 +... +N(xn)/xn. \quad (1)$$

For example, consider diabetes in blood test. Glycemia and Hb-A1c are the signs. Then, set X= diabetes, and

$$N(\text{diabetes})=N(\text{Glycemia})/\text{ Glycemia} + N(\text{Hb-A1c})/\text{ Hb-A1c}. \quad (2)$$

Second, we have many medical scanner modalities, such as MRI, X-ray CT, Ultrasound, MEG and so on. Especially MRI can produce exactly matched images compared with human internal organs and lesions [9]. Figure 1 shows the illustrations of MR meniscus images. Consider the examination of the meniscal tear. Figure 1(a) shows a typical tear state in natural objects. Anyone can understand the tear. Figure 1(b) shows the examination region, i.e., almost always meniscal tear happens inside region of meniscus shown in the circle. The MR three-dimensional images segmented by our proposed method in Refs. [7, 8] are shown in the upper figures of Figures 1 (c), (d) and (e). These contours are shown in the lower figures. Then, we can distinguish the normal case (Cases 2 and 3) with injured case (Case 1) by observing these figures.

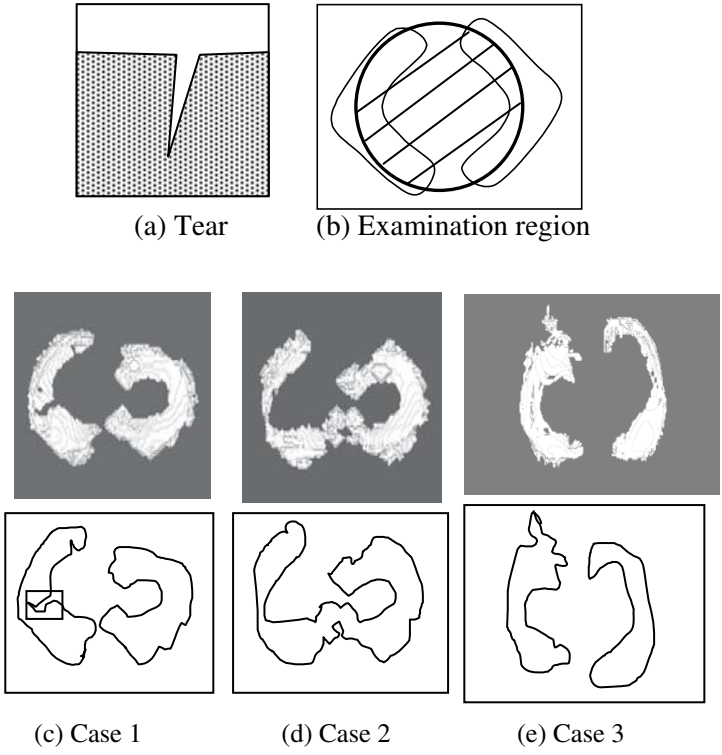


Fig. 1 Examination of meniscal tears.

Let X be the meniscal tear on the 3D MR image and let xi (i=1, 2, …, n) be exact meniscal state on location xi in the 3D image, where n is the location number of the 3D MR image. Then N(x) can be represented by,

$$N(X) = N(x1)/x1 + N(x2)/x2 + ... + N(xn)/xn. \quad (3)$$

Here, we determine the normal degrees in Figure 1. Figure 1(c) has a tear in the location of inside center in the right meniscus, whose location is denoted by xcr. Then,

$$N(\text{Case 1}) = 0.2/xcr + 1/\text{other place} \quad (4)$$

For Figures 1(d) and (e), we can determine them by

$$N(\text{Cases 2 and 3}) = 1/xcr + 1/\text{other place}. \quad (5)$$

3.2 Combination Rule

First, we consider normal degree for diabetes in blood test. Many studies on diabetes are done. The following facts are known in Japanese on blood test.

<Statistical Result>

The standard values in Japanese are as follows.

Glycemia: 60-110mg/dl under fasting glycemia

Hb-A1c: 4.3-5.8%

For diabetes: when both glycemia is larger than 140mg/dl under fasting glycemia and Hb-A1c is larger than 7%, there is high probability of a patient being retinopathy and nephropathy.

Here, we consider a patient (74 years old, female, ten years passed after diabetes). The clinical data of glycemia and Hb-A1c of the patient is shown in Figures 3(a) and (b). For the patient, we make two membership functions, Gmf, and Hmf for her glycemia and Hb-A1c, respectively, as shown in Figure 2. Let G(x) and H(x) be the normal degrees for her glycemia and Hb-A1c to diagnose her diabetes. For a single value a in a domain x, a fuzzy singleton function s(x) is defined as s(x)=1; if x=a, s(x)=0; otherwise. Then, G(x) and H(x) are calculated by,

$$G(x) = \text{minimum}\,(Gmf, s(x)). \tag{6}$$

$$H(x) = \text{minimum}\,(Hmf, s(x)). \tag{7}$$

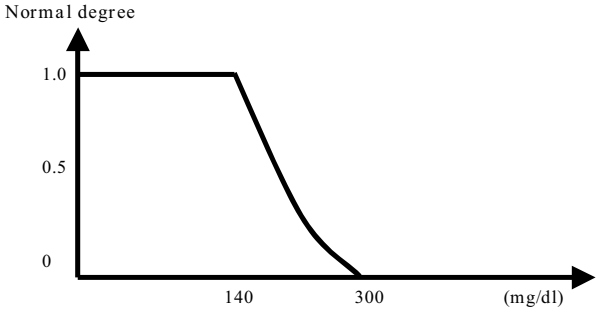
These calculated normal degrees are shown in Figures 3 (c) and (d). Here, we consider a combination method between two normal degrees. Figure 3(e) shows the three graphs by using maximum(MAX), minimum(MIN) and product(Product) combination rules. We can agree that the intersection rule such as minimum or product is appropriate for combination rule in this case, i.e., let X be diabetes of the human. Then we can agree that

$$N(X) = \text{minimum}(G(x), H(x)) \text{ or } \text{product}(G(x), H(x)). \tag{8}$$

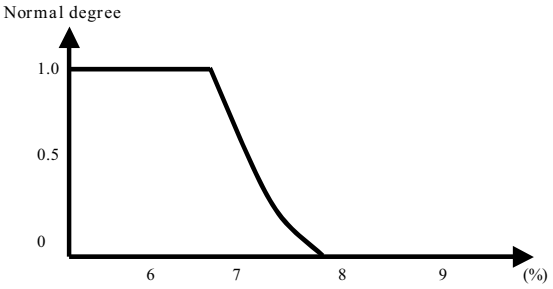
Second, we consider the meniscal images in Section 3.1. Usually, meniscal tear is cut and removed by the endoscope operation. Figure 1(c) is the image of meniscal tear for a patient. Figure 1(d) is the image removed by the operation for the same patient. We have already assigned these normal degrees by Equations (4) and (5). We can combine the normal degrees for the meniscal tear for the patient by

N(tear)= maximum(0.2, 1)/xcr +maximum (1, 1)/other place. (9)

= 1/xcr + 1/ other place



(a) Fuzzy membership function Gmf.



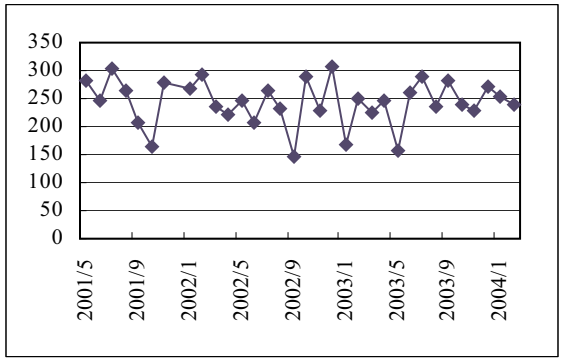
(b) Fuzzy Membership function Hmf.

Fig. 2 Fuzzy Membership functions.

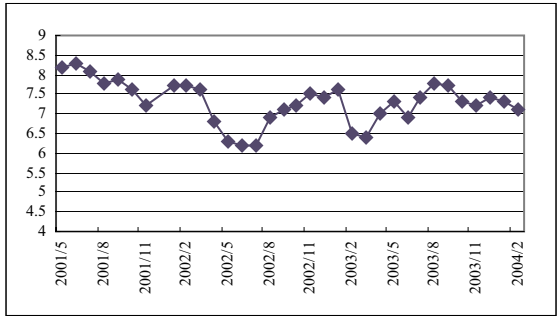
Thus, we can agree that a maximum combination rule is appropriate for the case. Considering this type of cases, we can derive the following rules. Let Ai be the normal degree at a time t on domain X, and let Ai' be the normal degree at the time t+d on the same domain, then

$$\begin{aligned} N(X) &= \text{maximum } (N(Ai), N(Ai')) \text{ if } Ai \leq Ai' \\ &= \text{minimum } (N(Ai), N(Ai')) \text{ if } Ai > Ai' \end{aligned} \quad (10)$$

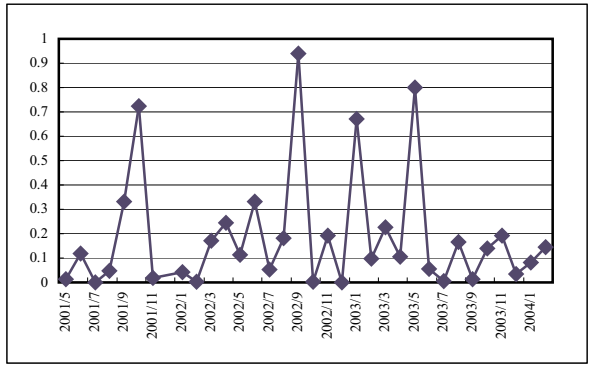
That is, generally, N(X) = N(Ai') holds.



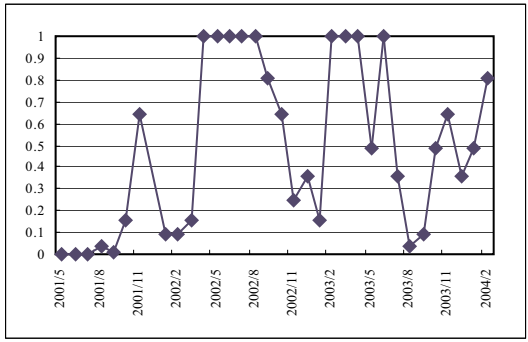
(a) Glycemia (mg/dl).



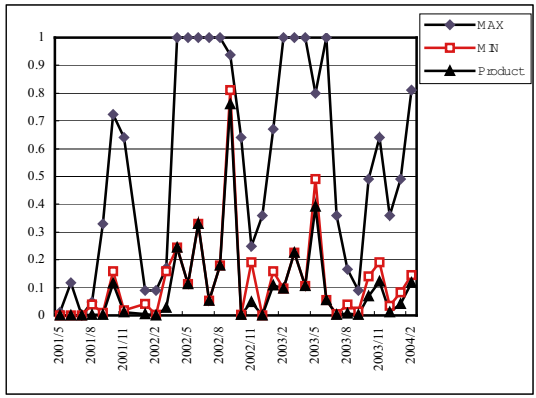
(b) Hb-A1c (%).



(c) Normal degree of Glycemia, G(x).



(d) Normal degree of Hb-A1c, H(x).



(e) MIN, MAX and product.

Fig. 3 Blood test and the normal degrees.

4. Conclusions

This paper has described the combination rules of normal degree for the case of MR image of meniscal tear and blood test on diabetes. For the combination rules;

Let X be a disease (granule) and let Ai(i=1, 2, …, n) be a state of same sign at different time, where n is the number, Then

N(X) = UNION or INTERSECTION of N(A1) and N(A2) and …, and N(An).

Let X be a disease (granule) and let Ai(i=1, 2, …, n) be a state of different signs, where n is the number, Then

N(X) = INTERSECTION of N(A1) and N(A2) and …, and N(An).

Theses rules are fundamental combination rules for fusion of normal degree. Practically, a problem appears in the case of a complex disease. A disease examination consists of two or more different signs whose conditions are not independent. We will then introduce the conditioned normal degree. Moreover, when we consider the progress of the disease, the algebra with no commutativity or no associativity and so on takes into account. It thus remains as future studies to investigate these complex combination rules.

References

1. Giarratano J, Riley G (1998) Expert Systems: Principles and Programming", 3rd Edition, , PWS Publishing Co pp. 509-556
2. For example: Computer Assisted Surgery and Medical Image Topics in Trans. of IEEE EMBS Society
3. Hata Y, Ishikawa O, Kobashi S, Kondo K, (2003) Degree of Normality Based on Fuzzy Logic for a Diagnostic Analysis of Signs Observed in a Human Body, Proc. 20th Annual Meeting of the North American Fuzzy Information Processing Society - NAFIPS, pp.155-160
4. Hata Y, Ishikawa O, Kobashi S, Kondo K, (2004) Automated Medical Diagnosis System (AMDS) with Normal Degree Based on Fuzzy Logic, Proc. 2nd IASTED Int. Conf. on Biomedical Engineering, pp.590-593, Austria
5. Zadeh LA (2003) Fuzzy Logic as a Basis for a Theory of Hierarchical Definability, Proc. of Int. Sympo. on Multiple-Valued Logic, pp.3-4
6. Zadeh LA (2003). A Perception-Based Approach to Decision Analysis and Causality" Proc. 20th Annual Meeting of the North American Fuzzy Information Processing Society - NAFIPS, p.1
7. Hata Y, Kobashi S, Tokimoto Y, Ishikawa M (2001) Computer Aided Diagnosis System of Meniscal Tears with T1 and T2 weighted MR Images Based on Fuzzy Inference, 7th FUZZY DAYS Int. Conf. on Computational Intelligence, pp. 55-58
8. Sasaki T, Hata Y, Ando Y, Ishikawa M (1999) Fuzzy rule-based approach to segment the menisci regions from MR images, Proc. SPIE Medical Imaging 1999, vol. 3661, pp. 258-263
9. Hata Y, Kobashi S, Hirano S, Kitagaki H, Mori E (2000) Automated Segmentation of Human Brain MR Images Aided by Fuzzy Information Granulation and Fuzzy Inference, IEEE Trans. Syst., Man, Cybern. C, 30:3: 381-395

Generation of Representative Symptoms Based on Fuzzy Concept Lattices

Tatiana Kiseliova[1,3] and Stanislav Krajči[2]

[1] University of Dortmund, Department of Computer Science I,
D-44221 Dortmund, Germany
Tatiana.Kiseliova@udo.edu
[2] Institute of Computer Science, UPJŠ Košice, Slovakia
krajci@science.upjs.sk
[3] Ivane Javakhishvili Tbilisi State University
Chavchavadze, av. 1., 380028 Tbilisi, Republic of Georgia

Abstract. It is very challenging and interesting task in medicine to find a set of representative symptoms for the disease, i.e., the set of symptoms that best characterizes a disease. In this paper we propose a method for constructing such representative symptoms for a particular disease, based on the data taken from the patient-records. The method uses a closure operator on the base of which a (one-sided) fuzzy concept lattice is defined.

1 Introduction

Consider the following so called incidence matrix $R_{SP} = \{r_{ij}\}$, $i = 1 \dots 4$, $j = 1 \dots 5$:

$$\begin{array}{c|ccccc} & p_1 & p_2 & p_3 & p_4 & p_5 \\ s_1 & 1 & 0 & 1 & 0 & 0 \\ s_2 & 1 & 1 & 0 & 1 & 0 \\ s_3 & 1 & 1 & 1 & 1 & 1 \\ s_4 & 0 & 1 & 0 & 1 & 1 \end{array} \tag{1}$$

which defines the relation of connectivity between **a set of people** $P = \{p_1, p_2, p_3, p_4, p_5\}$ and **a set of symptoms** $S = \{s_1, s_2, s_3, s_4\}$, where $r_{ij} = 1$ denotes that person p_j has the symptom s_i and $r_{ik} = 0$ denotes that person p_k does not have it. Notice that this information concerns one diagnosis and can be taken from the patient records with the already proven diagnosis [5],[3], [4].

The **level of connection** among patients may be defined as the number of common symptoms they exhibit minus one. Similarly, the level of connection among symptoms is the number of common patients affected by those symptoms minus one. For instance, persons p_3 and p_4 have in common only s_3 and hence they are connected at level 0, meanwhile patients p_1, p_2 and p_4 have in common symptoms s_2 and s_3, i.e., they are connected at level 1. Similarly for the connectivity of symptoms.

T. Kiseliova and S. Krajči: *Generation of Representative Symptoms Based on Fuzzy Concept Lattices*, Advances in Soft Computing **2**, 349–354 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

q-connectivity	Q_p-chains	q-connectivity	Q_s-chains
$q=2$	$\{p_1\},\{p_2,p_4\}$	$q=4$	$\{s_3\}$
$q=1$	$\{p_1,p_2,p_3,p_4\}$	$q=3$	$\{s_3\}$
$q=0$	$\{p_1,p_2,p_3,p_4\}$	$q=2$	$\{s_2,s_3,s_4\}$
		$q=1$	$\{s_1,s_2,s_3,s_4\}$
		$q=0$	$\{s_1,s_2,s_3,s_4\}$

Table 1. Groups of representative symptoms and patients

The connectivity between people Q_p and symptoms Q_s are given by the following formulas:

$$Q_p = R^{\mathsf{T}}R - \Omega \tag{2}$$

$$Q_s = RR^{\mathsf{T}} - \Omega \tag{3}$$

where Ω is simply the matrix with unit elements. Connections between elements in (2) and (3) can be presented as in the Table 1. The information, contained in this table can be interpreted. For instance, symptoms $\{s_2, s_3, s_4\}$ are more representative for the considered disease, because they are connected at the level $q = 2$ (one-element set is not interesting from the medical point of view, although the level of connection is higher).

So far so good, but if we consider the following incidence matrix

$$\begin{array}{c|cccc} & p_1 & p_2 & p_3 & p_4 \\ s_1 & 1 & 1 & 1 & 1 \\ s_2 & 0 & 0 & 1 & 1 \\ s_3 & 1 & 1 & 1 & 1 \end{array}$$

then, due to (2) p_1 and p_3 are connected (via s_1 and s_2) at the same level ($q = 1$) as p_1 and p_2 (via s_1 and s_2). But obviously, p_1 is more like p_2 than p_3. Moreover, the problem of measuring the "likeness" of the patients becomes more interesting and complicated if the connection of symptoms and patients are described as fuzzy sets.

2 Fuzzy concept lattices for symptoms clustering

We present the approach using so-called (one-sided) fuzzy concept lattices [1]: Consider a matrix $R_{SP} = \{r_{ij}\}, i = 1 \dots n,\ j = 1 \dots m$ with values r_{ij} between 0 and 1 (how these values are introduced, we say later) with rows $s_i,\ i = 1 \dots n$ from the set S and columns $p_j,\ j = 1 \dots m$, from the set P, i.e., $R_{SP} : S \times P \to [0, 1]$. For example,

$$\begin{array}{c|ccccc} & p_1 & p_2 & p_3 & p_4 & p_5 \\ s_1 & 1 & 0.8 & 0.2 & 0.3 & 0.5 \\ s_2 & 0.8 & 1 & 0.2 & 0.6 & 0.9 \\ s_3 & 0.2 & 0.3 & 0.2 & 0.3 & 0.4 \\ s_4 & 0.4 & 0.7 & 0.1 & 0.2 & 0.3 \\ s_5 & 1 & 0.9 & 0.3 & 0.2 & 0.4 \end{array} \tag{4}$$

where s_1, s_2, s_3, s_4, s_5 can be symptoms such *cough*, *fever*, *ruddy checks*, *low blood pressure*, *pain*, *walking dyspnoea* accordingly.

Denote the family of all crisp subsets of the set S by $\mathcal{P}(S)$, and the family of all fuzzy subsets of the set P (i.e., all functions from P to $[0,1]$) by $\mathcal{F}(P)$. Then values in the row i can be understood as a function in $\mathcal{F}(P)$, which says about p_j belonging to s_i.

The interpretation of belonging can be different. For example, these values can show the severity, strength of expressiveness, the importance of the symptom s_i for the patients for the considered disease.

Define the pair of mappings $\tau_{R_{SP}} : \mathcal{P}(S) \to \mathcal{F}(P)$ and $\sigma_{R_{SP}} : \mathcal{F}(P) \to \mathcal{P}(S)$ in the following way:

If $X \in \mathcal{P}(S)$ then $\tau_{R_{SP}}(X)$ will be the function which value p_j in P is defined by

$$\tau_{R_{SP}}(X)(p_j) = \min\{R_{SP}(s_i, p_j) : s_i \in X\},\ i = 1 \dots n, j = 1 \dots m$$

(i.e. $\tau_{R_{SP}}(X)$ is the greatest lower bound of above mentioned belonging-functions of rows from X). The medical explanation is that taking the group of symptoms X for the patient p_j the function $\tau_{R_{SP}}$ chooses the value $\tau_{R_{SP}}(X)(p_j)$ that defines how strong the symptoms in the set X (together) are present in the person p_j.

If $g \in \mathcal{F}(P)$ then $\sigma_{R_{SP}}(g)$ will be the set defined by

$$\sigma_{R_{SP}}(g) = \{s_i \in S : (\forall p_j \in P)\ R_{SP}(s_i, p_j) \geq g(p_j)\},\ i = 1 \dots n, j = 1 \dots m$$

(i.e., $\sigma_{R_{SP}}(g)$ is a set of all rows, above mentioned belonging-functions of which dominates over g). It means that taking into account a kind of threshold $g(p_j)$ one can consider the symptoms that are more important for the particular disease than the others.

These two mappings make a Galois connection (i.e., the properties $X_1 \subseteq X_2 \to \tau_{R_{SP}}(X_1) \geq \tau_{R_{SP}}(X_2)$, $f_1 \leq f_2 \to \tau_{R_{SP}}(f_1) \supseteq \tau_{R_{SP}}(f_2)$, $X \subseteq \tau_{R_{SP}}(\tau_{R_{SP}}(X))$ and $f \leq \tau_{R_{SP}}(\tau_{R_{SP}}(f))$ are fulfilled). We will use their composition

$$\mathrm{cl}_{R_{SP}} : \mathcal{P}(S) \to \mathcal{P}(S)$$

that is a *closure operator*, i.e., the following properties are fulfilled:

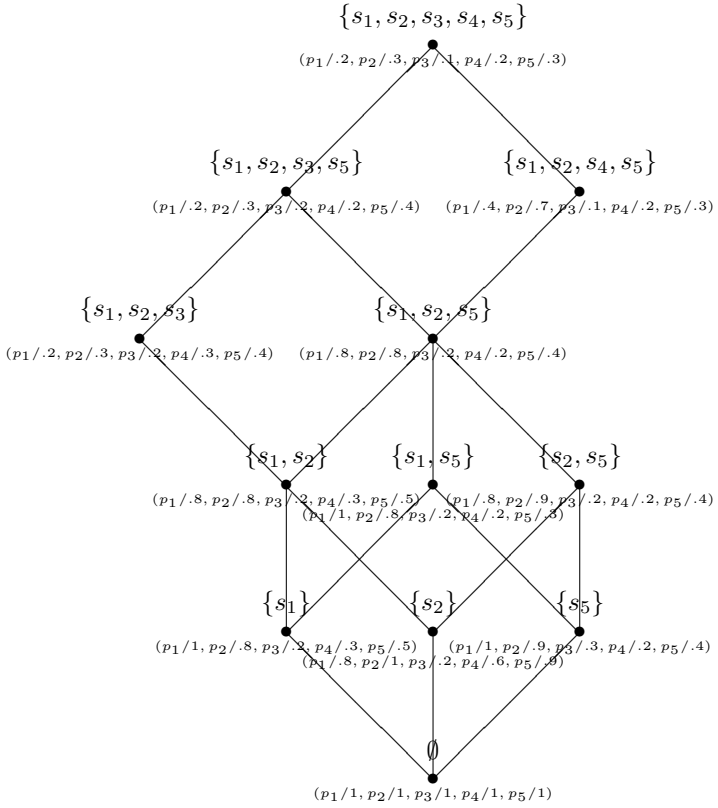
1. $X \subseteq \mathrm{cl}_{R_{SP}}$
2. if $X_1 \subseteq X_2$ then $\mathrm{cl}_{R_{SP}}(X_1) \subseteq \mathrm{cl}_{R_{SP}}(X_2)$
3. $\mathrm{cl}_{R_{SP}}(\mathrm{cl}_{R_{SP}}(X)) = \mathrm{cl}_{R_{SP}}(X)$

If X is a set of symptoms, all patients with symptoms from X may have some other symptoms from set S (with some grade). It probably means that these "other" symptoms are related to symptoms from X. The closure of the set X tries to express this idea. So we can consider only these sets which are the same as their closure because they have no other related symptoms, i.e. they are complete in such sense and therefore they are good candidates for clusters. Hence define

$$L_{R_{SP}} = \{X \in \mathcal{P}(S) : \mathrm{cl}_{R_{SP}}(X) = X\}$$

This set is a *(one-sided) fuzzy concept* lattice and its ordering is the ordinary inclusion. Elements of this lattice are called (fuzzy) concepts.

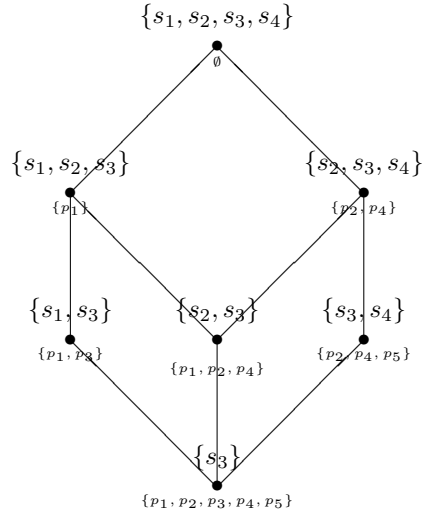
In our case (see example (4)) $L_{R_{SP}}$ contains these 12 concepts: $\emptyset$, $\{s_1\}$, $\{s_2\}$, $\{s_5\}$, $\{s_1, s_2\}$, $\{s_1, s_5\}$, $\{s_2, s_5\}$, $\{s_1, s_2, s_5\}$, $\{s_1, s_2, s_4, s_5\}$, $\{s_1, s_2, s_3\}$, $\{s_1, s_2, s_3, s_5\}$, and $\{s_1, s_2, s_3, s_4, s_5\}$ (all symptoms). They can be depicted in the following hierarchical way (by a so-called Hasse diagram). The first line in every vertex is a concept and the second (tiny) line is its $\tau_{R_{SP}}$ (a corresponding belonging function):



This Hasse diagram can be considered as a good structure for navigation when looking for the best characterization of the disease, i.e., for the set of representative symptoms for this disease.The heuristic procedure such as *starting from the high levels of the Hasse diagram, that concept is chosen, which has the maximum number of incoming edges and contains not less than 2 elements. From more than one chosen concepts at the different levels, those are taken, which have maximum outcoming edges.* For the Hasse diagram shown above, $\{s_1, s_2, s_5\}$ can be taken as a set of representative symptoms.

The original, crisp version of this approach is described in [2] and it is a special case of ours if sets and their characteristic functions are identified. It

can be used for our crisp example (1) and it produces the 7 concepts: $\{s_3\}$, $\{s_1, s_3\}$, $\{s_2, s_3\}$, $\{s_3, s_4\}$, $\{s_1, s_2, s_3\}$, $\{s_2, s_3, s_4\}$, and $\{s_1, s_2, s_3, s_4\}$. They can be depicted by a Hasse diagram too. The first line in every vertex is a concept again and the second (tiny) line is its set of common p_j.



Using the heuristic procedure described above, $\{s_2, s_3\}$ can be taken as a set of representative symptoms.

3 Conclusions

In this paper we present a concept lattice as another possible method for clustering symptoms of diseases. It is based on common presence of symptoms in patients and it expresses the idea that a common presence of symptoms in patients may means their common cause. We use the classical crisp version and a rather new fuzzy version of a concept lattice in a case that presence of a symptom in a patient is expressed by number from the interval $[0, 1]$. The new one-side fuzzy concept lattice introduced in this paper is demonstrated to be very useful in medicine, in particular, in establishing the representative symptoms. "Crisp" and "fuzzy" examples illustrate this observation.

References

1. S. Krajči. Cluster based efficient generation of fuzzy concepts. *Neural Network World*, 13(5):521–530, 2003.
2. B. Ganter and R. Wille. *Formal Concept Analysis, Mathematical Foundation*. ISBN 3-540-62771-5. Springer Verlag, 1999.

3. T. Kiseliova and H. Peters. Decision making in diagnosis of oral mucosa lesions. In B. Reusch, editor, *Computational Intelligence. Theory and Applications: International Conference on Computational Intelligence, 7. Fuzzy Days in Dortmund*, pages 33–39, Dortmund, Germany, October, 1-3, 2001. Springer-Verlag Heidelberg.
4. T. Kiseliova and H. Peters. Representation of the different approaches to computer-assisted diagnosis of oral mucosa lesions based on a general scheme. In *Proceedings of the 10th International Fuzzy System Association World Congress IFSA2003*, pages 103–106, Turkey, Istanbul, June 29-July 2, 2003.
5. T. Kiseliova and B. Reusch. On fuzzy decision support method for medical diagnoses. In R. J. Patton and J. Chen, editors, *Proceedings of IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS'97*, volume 2, pages 938–943, The University of Hull, Kingston Upon Hull, United Kingdom, August, 26-29, 1997.

Connectives

On the Direct Decomposability of Fuzzy Connectives, Negations and Implications Based on T-Norms and T-Conorms on Product Lattices

Funda Karaçal

Department of Mathematics, Karadeniz Technical University, 61080 Trabzon, Turkey
fkaracal@yahoo.com

Abstract. In this paper, we study and discuss the direct products of fuzzy connectives, negations and implications based on t-norms and t-conorms and give a necessary and sufficient condition for their direct decomposability.

1 Introduction

In fuzzy logic, connectives AND ,OR and NOT are usually modelled by t-norms , t-conorms and strong negations on $[0,1]$, respectively [7, 14]. Based on these logical operators on $[0,1]$, three fundamental classes of fuzzy implications on $[0,1]$ (i.e. S-,R-,and QL-implication on $[0,1]$) were defined and studied [3, 5, 6, 8, 9, 11, 14, 15]. Ma and Wu [9] introduced the notion of t-norms on a complete lattice L and discussed the relation between the t-norms and the implications on L.

In [13], Wang and Yu established a unitive algebraic theory of logical operators based on a complete Brouwerian lattice L, introduced the concept of pseudo-t-norms on L and, discussed the relation between the set of all infinitely $\vee$-distributive pseudo-t-norms and the set of all infinitely $\wedge$-distributive implications on L in detail.

F. Karaçal: *On the Direct Decomposability of Fuzzy Connectives, Negations and Implications Based on t-norms and t-conorms on Product Lattices*, Advances in Soft Computing **2**, 357–367 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

The concept of the direct product of t-norms on product lattice (external direct product) was introduced by De Baets and Mesiar [2].They characterized $\vee$-distributive t-norms on a product lattice that are a direct product of t-norms.

In [13], Wang and Yu studied the pseudo-t-norms and implication operators on a complete Brouwerian lattice L , and discussed their direct products and direct product decompositions.

In this paper, we give a necessary and sufficient condition to direct decomposability for pseudo- t-norms and implication operators on product lattices. We study the direct products of t-conorms, strong negations, implication operators on product lattice. Furthermore we discuss the direct decomposability of S-implications and R-implications. Finally, we formulate our conjecture about the direct decomposability of OL-implication operators.

Let $(L, \leq)$ be a complete lattice with top and bottom elements 1,0 respectively.

2 Preliminaries

In this section, we shall briefly recall some definitions about t-conorms, strong negations, pseudo- t-norms and implication operators on L , and a property of strong negations on L.

Definition 1 ([7, 9]) *A triangular conorm (t-conorm, for short) is a commutative, associative,non-decreasing function* $S : L \times L \to L$ *such that* $S(x, 0) = x \forall x \in L$.
Moreover, if S satisfies the conditions commutative, associative,non-decreasing and $S(x, 1) = x$ *for all* $x \in L$, *then S is called a triangular norm (t-norm).*

Definition 2 *A t-conorm S on L is said to be* $\wedge$*-distributive if*

$$x, y, z \in L \Rightarrow S(x, y \wedge z) = S(x, y) \wedge S(x, z).$$

Definition 3 ([12, 13]) *An implication I on L is a binary operation on L that satisfies the following two conditions:*
$(I.1) I(1, y) = y$ *and* $I(0, y) = 1 \forall y \in L$;
$(I.2) x, y, z \in L, y \leq z \Rightarrow I(x, y) \leq I(x, z)$

Definition 4 ([12]) *A binary operation T on L is called a pseudo-t-norm if it satisfies the following conditions:*
$(T.1) T(1,y) = y$ *and* $T(0,y) = 0 \forall y \in L$;
$(I.2) x, y, z \in L, y \leq z \Rightarrow T(x,y) \leq T(x,z)$

Definition 5 *An implication operator I on a lattice L is called $\wedge$-distributive if*
$I(a, b_1 \wedge b_2) = I(a,b_1) \wedge I(a,b_2) \forall a, b_1, b_2 \in L.$

Definition 6 *A pseudo-t-norm T on a lattice L is called $\vee$-distributive if*
$T(a, b_1 \vee b_2) = T(a,b_1) \vee T(a,b_2) \forall a, b_1, b_2 \in L.$

Definition 7 ([9]) *A mapping $n : L \to L$ is called a strong negation if*
$(N.1) n(n(x)) = x \forall x \in L$
$(N.2) x \leq y, x, y \in L \Rightarrow n(y) \leq n(x).$
Obviously,if n is a strong negation on L, then n(0)=1 and n(1)=0.

Theorem 1 ([12]) *Let $x_j \in L, j \in J$. If n is a strong negation on L, then*
$(1) n(\bigvee_{j \in J} x_j) = \bigwedge_{j \in J} n(x_j).$
$(2) n(\bigwedge_{j \in J} x_j) = \bigvee_{j \in J} n(x_j).$
i.e. , De Morgan laws hold.

Definition 8 ([3, 6]) *S-implications are based on the classical idea of implication ($P \to Q$ is defined as $\neg P \vee Q$) and are given by*

$$I_S(x,y) = S(n(x), y) \forall x, y \in L,$$

where S is a t-conorm and n is a strong negation on L.

Definition 9 ([3, 6, 7]) *R-implications are based on a residuation concept in lattice structures equipped with a semigroup operation T that may stand for a conjunction, namely*

$$I_R(x,y) = \bigvee \{u \in L : T(x,u) \leq y\} \forall x, y \in L.$$

where T is an t-norm on L and R stands for "residuated" .

Definition 10 ([3, 6]) *QL-implications have the form used in quantum logic and are based on an "if ...then...else..." view of a fuzzy rule, and are defined by*

$$I_{QL}(x,y) = S(n(x), T(x,y)) \forall x, y \in L,$$

where S and T are, respectively t-conorm on L and t-norm on L, and n is a strong negation on L.

3 The direct decomposability of strong negations on the product lattices

In this section, we shall examine the direct products and direct decompositions of strong negations.

Definition 11 *Let L_1 and L_2 be lattices, n_1 a strong negation on L_1 and n_2 a strong negation on L_2 . By the direct product of n_1 and n_2 , denoted by $n_1 \times n_2$, we mean the mapping , defined by $(n_1 \times n_2)(x,y) = (n_1(x), n_2(y))$.*

Proposition 1 *The direct product $n_1 \times n_2$ of n_1 and n_2 is a strong negation on the product lattice $L_1 \times L_2$.*

Now we give the following proposition to define the internal direct product of strong negations.

Proposition 2 *Let L_1 ,L_2 be complete sublattices of L, and n_1 a strong negation on L_1 and n_2 a strong negation on L_2. Suppose that for each $x \in L$ there exist $x_1 \in L_1$ and $x_2 \in L_2$ uniquely defined such that $x{=}x_1 \wedge x_2$. Then the mapping $n : L \to L$, defined by*

$$n(x) := n_1(x_1) \wedge n_2(x_2), \text{ where } x = x_1 \wedge x_2,$$

is a strong negation on L.

Definition 12 *The strong negation constructed in Proposition 2 will be called the internal direct product of n_1 and n_2, and it will be denoted by $n_1 \bigotimes n_2$.*

Definition 13 *Strong negations n on a lattice L and n^* on a lattice M will be called isomorphic if there exists a lattice isomorphism $H : L \to M$ such that*

$$H(n(a)) = n^*(H(a)) \forall a \in L.$$

Proposition 3 *Let L_1, L_2 be complete lattices, $L = L_1 \times L_2$ and n a strong negation on L. Then*

(i) $n(0,1) = (1,0)$ *if and only if* $n(1,0) = (0,1)$.

(ii) $n(0,1) = (0,1)$ *if and only if* $n(1,0) = (1,0)$.

Theorem 2 *Let L_1 and L_2 be two complete lattices, and their direct product lattice $L = L_1 \times L_2$, and n be a strong negation on $L = L_1 \times L_2$. The following conditions are equaivalent:*

(i) n(0,1)=(1,0) (or n(1,0)=(0,1));

(ii) n is the direct product of a strong negation n_1 on L_1 and n_2 on L_2;

(iii) n is the internal direct product of a strong negation n_1^* on $L_1 \times \{1\}$ and a strong negation n_2^* on $\{1\} \times L_2$.

4 The direct decomposability of t-conorms and implication operators on product lattices

Proposition 4 *Consider a t-conorm S_1 on a complete lattice L_1 and a t-conorm S_2 on a complete lattice L_2 , then the direct product $S_1 \times S_2$ of S_1 and S_2 , defined by*

$$(S_1 \times S_2)((x_1, x_2), (y_1, y_2)) = (S_1(x_1, y_1), S_2(x_2, y_2))$$

is a t-conorm on the product lattice $L_1 \times L_2$.

Remark 1 *The concept of the direct product of implication operators was introduced in [13].*

Now we give the following proposition to define the internal direct product of t-conorms(implication operators).

Proposition 5 *Consider two complete sublattices L_1 and L_2 of L, and a t-conorm (implication operator) S_1 on L_1 and a t-conorm (implication operator) S_2 on L_2. Suppose that for each $x \in L$ there exist $x_1 \in L_1$ and $x_2 \in L_2$ uniquely defined such that $x = x_1 \wedge x_2$. Then the function $S : L \times L \to L$, defined by*

$S(x, y) := S_1(x_1, y_1) \wedge S_2(x_2, y_2)$, *where* $x = x_1 \wedge x_2$, $y = y_1 \wedge y_2$, *is a t-conorm (implication operator) on L.*

A t-norm S(implication operator) constructed as in Proposition 5, will be called the internal direct product of S_1 and S_2, and will be denoted by $S_1 \bigotimes S_2$.

Let L be a complete lattice, S be a t-conorm(implication operator) on L and $L_1 \subseteq L$. The notation $S \downarrow L_1$ stands for the restriction of S to L_1.

Definition 14 *Triangular conorms (implication operators) S on a lattice L and P on a lattice M will be called isomorphic if there exists a lattice isomorphism $H : L \to M$ such that*

$$H(S(x, y)) = P(H(x), H(y)) \forall x, y \in L.$$

The following proposition gives the connection between external and internal direct products of t-conorms.

Theorem 3 *Let $L = L_1 \times L_2$ and $S = S_1 \times S_2$, where S_i is a t-conorm on the lattice L_i for $i \in \{1, 2\}$. Then $S \downarrow (L_1 \times \{1\})$ and $S \downarrow (\{1\} \times L_2)$ are t-conorms on $L_1 \times \{1\}$ and $\{1\} \times L_2$ respectively, and, they are also isomorphic to S_1 and S_2, respectively. Furthermore $S = S \downarrow (L_1 \times \{1\}) \bigotimes S \downarrow (\{1\} \times L_2)$ Conversely, let $S = S \downarrow (L_1 \times \{1\}) \bigotimes S \downarrow (\{1\} \times L_2)$. Then $S = S_1 \times S_2$, where S_1 on L_1 and S_2 on L_2 are t-conorms such that S_1 and S_2 are isomorphic to t-conorms $S \downarrow (L_1 \times \{1\})$ and $S \downarrow (\{1\} \times L_2)$ on $L_1 \times \{1\}$ and $\{1\} \times L_2$, respectively.*

A similar theorem to Theorem 3 can be given for t-norms dually.

Corollary 1 *A t-conorm S on a product lattice $L = L_1 \times L_2$ is the direct product of a t-conorm on L_1 and a t-conorm on L_2 if and only if for $(x, y), (z, t) \in L$*

$$S((x, y), (z, t)) = S((x, 1), (z, 1)) \wedge S((1, y), (1, t)) (*)$$

Remark 2 *When working with t-conorms on product lattices, we have the freedom to use t-conorms that do not act coordinatewisely. But if we want the direct decomposability of a t-conorm S on product lattice, S must act coordinatewisely.*

Remark 3 *If a t-conorm S is a $\wedge$-distributive t-conorm on a product lattice $L = L_1 \times L_2$, then S satisfies the equality $(*)$. The converse of this statement is not true: It is not necessary for a t-conorm satisfying the equality (*) to be $\wedge$-distributive t-conorm. For example,let L_1 and L_2 be non-distributive lattices. Let $L = L_1 \times L_2$ and, let $S = \vee$ on L. Then S satisfies the equality $(*)$, but it is not a $\wedge$-distributive t-conorm.*

Theorem 4 *Let $L = L_1 \times L_2$ and $I = I_1 \times I_2$, where I_i is an implication operator on the lattice L_i for $i \in \{1, 2\}$. Then $I \downarrow (L_1 \times \{1\})$ and $I \downarrow (\{1\} \times L_2)$ are implication operators on $L_1 \times \{1\}$ and $\{1\} \times L_2$, respectively, and, they are also isomorphic to I_1 and I_2, respectively. Furthermore $I = I \downarrow (L_1 \times \{1\}) \bigotimes I \downarrow (\{1\} \times L_2)$.*
Conversely, let $I = I \downarrow (L_1 \times \{1\}) \bigotimes I \downarrow (\{1\} \times L_2)$. Then $I = I_1 \times I_2$, where I_1 on L_1 and I_2 on L_2 are implication operators such that I_1 and I_2 are isomorphic to implication operators $I \downarrow (L_1 \times \{1\})$ and $I \downarrow (\{1\} \times L_2)$ on $L_1 \times \{1\}$ and $\{1\} \times L_2$, respectively.

A similar theorem to Theorem 4 can be given for pseudo-t-norms.

Corollary 2 *(i)An implication operator I on a product lattice $L = L_1 \times L_2$ is the direct product of an implication operator on L_1 and an implication operator on L_2 if and only if for all $(x, y), (z, t) \in L$*

$$I((x, y), (z, t)) = I((x, 1), (z, 1)) \wedge I((1, y), (1, t))(**)$$

and, $L_1 \times \{1\}$ and $\{1\} \times L_2$ are closed under I on $L_1 \times L_2$.

(ii) A pseudo t-norm T on a product lattice $L = L_1 \times L_2$ is the direct product of an implication operator on L_1 and an implication operator on L_2 if and only if for all $(x, y), (z, t) \in L$

$$T((x, y), (z, t)) = T((x, 0), (z, 0)) \vee T((0, y), (0, t))$$

and $L_1 \times \{0\}$ and $\{0\} \times L_2$ are closed under T on $L_1 \times L_2$.

Example 1 *Consider the lattices* $L_1 = \{0, x, 1\}$, $0 < x < 1$ *and* $L_2 = \{0, 1\}$, $0 < 1$. *We define* I *on* $L_1 \times L_2$ *as follows:*

$$I((x,y),(z,t)) = \begin{cases} (1,1), & (x,y) = (0,0) \\ (1,t), & (x,y) = (0,1) \\ (z,1), & (x,y) = (1,0) \\ (0,0), & \text{Otherwise} \end{cases}$$

It can be shown that I *is an implication operator on* $L_1 \times L_2$ *and satisfies the equality (**). I* $\downarrow (L_1 \times \{1\})$ *is not implication operator on* $L_1 \times \{1\}$, *since* $L_1 \times \{1\}$ *is not closed under I. Thus by Theorem 4, I is not be a direct product of an implication operator on* L_1 *and an implication operator on* L_2.

Theorem 5 *Let I be a* $\wedge$*-distributive implication operator on a product lattice* $L = L_1 \times L_2$. *If I satisfies the following conditions:*

i) $I((x,y),(z,t) \vee (k,l)) = I((x,y),(z,t)) \wedge I((z,t),(k,l))$
$\forall (x,y),(z,t),(k,l) \in L$;

ii) $I((0,1),(0,0)) = (1,0), I((1,0),(0,0)) = (0,1)$,

then there exist an implication operator I_1 *on* L_1 *and an implication operator* I_2 *on* L_2 *such that* $I = I_1 \times I_2$.

Theorem 6 *Let T be a* $\vee$*-distributive pseudo-t-norm on a product lattice* $L = L_1 \times L_2$. *If T satisfies the following conditions:*

i) $T((x,y),(z,t) \wedge (k,l)) = T((x,y),(z,t)) \wedge T((z,t),(k,l))$,
$\forall (x,y),(z,t),(k,l) \in L$;

ii) $T((0,1),(1,0)) = T((1,0),(0,1)) = (0,0)$,

then there exist a pseudo-t-norm T_1 *on* L_1 *and a peudo-t-norm* T_2 *on* L_2 *such that* $T = T_1 \times T_2$.

Remark 4 *The first parts of Theorem 3.4 and Theorem 3.5 in [13] can be obtained as corollaries of Theorem 5 and Theorem 6, respectively.*

5 The direct decomposability of R-implications and S-implications on product lattices

Theorem 7 *An R-implication operator* I_R *on a product lattice* $L = L_1 \times L_2$ *is the direct product of an R-implication operator on* L_1 *and an R-implication operator on* L_2 *if and only if for all* $(x,y),(z,t) \in L$

$$I_R((x,y),(z,t)) = I_R((x,1),(z,1)) \wedge I_R((1,y),(1,t)).$$

Let $I_S((x,y),(z,t)) = S(n(x,y),(z,t)), \forall (x,y),(z,t) \in L = L_1 \times L_2$, be an S-implication operator on L, where S is a t-conorm and n is a strong negation on L.

Lemma 1 *Let I_S be an S-implication operator satisfying the equality $(**)$ on a product lattice $L_1 \times L_2$. Then $n(0,1) = (1,0)$.*

Lemma 2 *Let I_S be an S-implication operator satisfying the equality $(**)$ on a product lattice $L_1 \times L_2$. Then*

$$S(n(x,1),(z,1)) = S(n(x,0),(z,1)) \forall x, z \in L_1$$

and

$$S(n(1,y),(1,t)) = S(n(0,y),(1,t)) \forall y, t \in L_2.$$

Using Lemma 1 and Lemma 2, we obtain the following theorem.

Theorem 8 *Consider two complete lattices L_1 and L_2, and an S-implication operator $I_S((x,y),(z,t)) = S((n(x,y),(z,t))$ on the product lattice $L_1 \times L_2$. Then the following conditions are equivalent:*

(i) n is the direct product of a strong negation n_1 on L_1 and a strong negation n_2 on L_2, and S is the direct product of a t-conorm S_1 on L_1 and a t-conorm S_2 on L_2;

(ii) I_S is the direct product of an S_1-implication operator on L_1 and an S_2-implication operator on L_2;

(iii) I_S is the internal direct product of an $S \downarrow (L_1 \times \{1\})$-implication operator $I_S \downarrow (L_1 \times \{1\})$ on $L_1 \times \{1\}$ and an $S \downarrow (\{1\} \times L_2$-implication operator $I_S \downarrow (\{1\} \times L_2)$ on $\{1\} \times L_2$.

Corollary 3 *An implication operator I_S on a product lattice $L = L_1 \times L_2$ is the direct product of an S-implication operator on L_1 and an S-implication operator on L_2 if and only if for all $(x,y),(z,t) \in L$*

$$I_S((x,y),(z,t)) = I_S((x,1),(z,1)) \wedge I_S((1,y),(1,t))$$

Remark 5 *A similar theorem to Theorem 8 is not true for QL-implication operators. For example: Let L_1 be the lattice $\{0,1\}$, where $0 < 1$ and $L = L_1 \times L_1$. We consider the implication operator*

$$I_{QL}((x,y),(z,t)) = S_M(n(x,y), T((x,y),(z,t)))$$

on L, where

$$S_M((x,y),(z,t)) = \begin{cases} (z,t)\,, & (x,y) = (0,0) \\ (x,y)\,, & (z,t) = (0,0)\,, \\ (1,1)\,, & \text{Otherwise} \end{cases}$$

is the t-conorm, $T = \wedge$ is the t-norm, n is the strong negation defined by

$$n(0,1) = (1,0), n(1,0) = (0,1), n(0,0) = (1,1), n(1,1) = (0,0)$$

on L. I_{QL} is a direct product of two QL-implication operators on L_1. But $S_M \downarrow (L_1 \times \{1\})$ is not a t-conorm on $L_1 \times \{1\}$.

6 Conjecture

We formulate the following conjecture about the direct decomposability of QL-implication operators.

Conjecture.

An QL-implication operator I_{QL} on a product lattice $L = L_1 \times L_2$ is the direct product of an QL-implication operator on L_1 and an QL-implication operator on L_2 if and only if for all $(x,y),(z,t) \in L$

$$I_{QL}((x,y),(z,t)) = I_{QL}((x,1),(z,1)) \wedge I_{QL}((1,y),(1,t))$$

References

[1] G. Birkhoff, Lattice Theory, 3 rd edition, Providence, 1967.

[2] B. De Baets, R. Mesiar, Triangular norms on product lattices, Fuzzy Sets and Systems, 104(1999) 61-75.

[3] D. Dubois, H. Prade, Fuzzy Sets in approximate reasoning, Part 1: inference with possibility distributions, Fuzzy Sets and Systems 42(1991)143-202.

[4] B. De Baets, R. Mesiar, Triangular norms on product lattices, Fuzzy Sets and Systems, 104(1999) 61-75.

[5] J.C. Fodor, Contrapositive symmetry of fuzzy implications , Fuzzy Sets and Systems 69(1995) 141-156.

[6] J.C. Fodor, T. Keresztfalvi, A characterizian of the Hamacher family of t-norms, Fuzzy Sets and Systems 65(1994) 51-58.

[7] E.P. Klement, R. Mesiar, E. Pap,Trends in Logics, Studia Logica Library, vol.8, Kluwer Academic Publishers, Dordrecht,2000.

[8] S. Kundu, J.H. Chen, Fuzzy Logic or Lukasiewich logic: a clarification, Fuzzy Sets and Systems 95(1998)369-379. 69(1995) 141-156.

[9] Z. Ma, W-M. Wu, Logical operators on complete lattices, Inform. Sci. 55(1991) 77-97.

[10] S. Jenei, B. De Baets, On the direct decomposability of t-norms on product lattices, 139(2003)699-707.

[11] E. Turunen, Algebraic structures in fuzzy logic, Fuzzy Sets and Systems 52(1992) 181-188.

[12] Z. Wang, Y.Yu, Pseudo-t-norms and implication operators on a complete Brouwerian lattice, Fuzzy Sets and Systems, 132(2002) 113-124.

[13] Z. Wang, Y.Yu, Pseudo-t-norms and implication operators: direct products and direct product decompositions, Fuzzy Sets and Systems, 139(2003) 673-683.

[14] S. Weber, A general concept of fuzzy connectives,negations and implications based on t-norms and t-conorms, Fuzzy Sets and Systems 11(1983) 115-134.

[15] W.-M. Wu, A multivalued logic system with respect to t-norms, in: M.M. Gupta, T. Yamakawa (Eds.), Fuzzy Computing, North Holland, Amsterdam, 1988, pp. 101-118.

The Cancellation Law for Addition of Fuzzy Intervals

Andrea Stupňanová

Dept. of Mathematics, Slovak University of Technology
Radlinského 11, Sk-813 68 Bratislava, Slovakia
andy@math.sk

Abstract. The cancellativity $A \oplus_T B = A \oplus_T C \Rightarrow B = C$ means that the equation $A \oplus_T X = D$ has unique solution. The cancellation law for sum of fuzzy quantities based on the strongest t-norm T_M holds for arbitrary fuzzy interval A, see e.g. [2, 7, 8]. For the weakest t-norm T_D the cancellation law holds only for very special fuzzy intervals. Based on our results from [1] and Zagrodny's results [10] we will present conditions for validity of the cancellation law for addition based on a continuous Archimedean t-norm.

1 Introduction

The addition of fuzzy quantities A and B is based on the generalized version of Zadeh's extension principle

$$A \oplus_T B(x) = \sup_{y+z=x} T(A(y), B(z)),\ z \in \mathbb{R}$$

where T is a t-norm.

Definition 1. *A fuzzy quantity A is called a fuzzy interval if it is continuous and its u-cuts*

$$A^{(u)} = \{x, A(x) \geq u\}\ ,$$

$u \in]0,1]$, are convex closed subsets of $\mathbb{R}$, i.e., $A^{(u)} = [a_1^{(u)}, a_2^{(u)}]$.

A special subclass of fuzzy intervals is formed by the LR-fuzzy intervals. In fact, LR-fuzzy intervals are fuzzy intervals with a bounded support.

Definition 2. *Consider two strictly decreasing, continuous transformations L and R of $[0,1]$ such that $L(0) = R(0) = 1$ and $L(1) = R(1) = 0$, and $(l, r, \alpha, \beta) \in \mathbb{R}^4$ such that $l \leq r$, $\alpha > 0$ and $\beta > 0$. The fuzzy quantity A defined by*

A. Stupňanová.: *The Cancellation Law for Addition of Fuzzy Intervals*, Advances in Soft Computing **2**, 369–375 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

$$A(x) = \begin{cases} 0 & , \text{ if } x \le l - \alpha \\ L\left(\frac{l-x}{\alpha}\right) & , \text{ if } l - \alpha \le x \le l \\ 1 & , \text{ if } l \le x \le r \\ R\left(\frac{x-r}{\beta}\right) & , \text{ if } r \le x \le r + \beta \\ 0 & , \text{ if } r + \beta \le x \end{cases}$$

is called an LR-fuzzy interval, and is denoted by $A \equiv (l, r, \alpha, \beta)_{LR}$.

Consider an *LR*-fuzzy interval $(l, r, \alpha, \beta)_{LR}$. l and r are called the left and right peak, α and β are called the left and right spread, and L and R are called the left and right shape function, respectively. A special type of *LR*-fuzzy intervals, called trapezoidal (or linear) fuzzy intervals $A = (l, r, \alpha, \beta)$ is obtained when L and R are linear: $L(x) = R(x) = 1 - x$.

The unimodal fuzzy interval A^a (i.e. $ker(A) = A^{(1)} = \{a\}$) is called *a fuzzy number*.

As far as for arbitrary fuzzy intervals A, B, C and t-norm T holds the following property

$$A \oplus_T B = A \oplus_T C \quad \Rightarrow ker(B) = ker(C),$$

when discussing the cancellation law it's enough to investigate the fuzzy numbers with zero's peak, (i.e., A^0).

2 The cancellation law for sum based on the strongest and the weakest t-norm

Theorem 1. *Consider the strongest t-norm* T_M. *Let* A, B, C *be fuzzy intervals. Then the cancellation law holds, i.e.*

$$A \oplus_{T_M} B = A \oplus_{T_M} C \quad \Rightarrow B = C\,.$$

Proof. By formula for addition of fuzzy quantities based on the min

$$(A \oplus_{T_M} B)^{(u)} = A^{(u)} + B^{(u)} \text{for any } u \in [0,1]$$

we can easily verify the validity of the cancellation law.
Let $A \oplus_{T_M} B = A \oplus_{T_M} C$. Then

$$[a_1^{(u)}, a_2^{(u)}] + [b_1^{(u)}, b_2^{(u)}] = [a_1^{(u)}, a_2^{(u)}] + [c_1^{(u)}, c_2^{(u)}] \text{ for all } u \in]0,1]. \quad \text{Thus}$$
$$[a_1^{(u)} + b_1^{(u)}, a_2^{(u)} + b_2^{(u)}] = [a_1^{(u)} + c_1^{(u)}, a_2^{(u)} + c_2^{(u)}] \Rightarrow b_1^{(u)} = c_1^{(u)} \text{ and } b_2^{(u)} = c_2^{(u)}$$
$$\Rightarrow B^{(u)} = C^{(u)} \text{ for all } u \in]0,1] \quad \Rightarrow B = C,$$

i.e., the cancellation law holds. □

Remark 1. The cancellation law with respect to T_M fails if A is fuzzy quantity with $\sup A < 1$ or $\inf A > 0$. See [8].

On the other hand, consider the weakest t-norm T_D. Then the cancellation law holds only in special cases.

Theorem 2. *Consider the addition based on the T_D. Let A^a, B^b, C^c be fuzzy numbers. Then the cancellation law is valid if and only if*

$$(i)\ A^a(x-b) \le B^b(x-a)$$
$$(ii)\ A^a(x-c) \le C^c(x-a)\ .$$

Proof. Using the formula for addition based on the drastic product from [7]

$$A^a \oplus_{T_D} B^b(x) = \max\left(A^a(x-b), B^b(x-a)\right)$$

we can easily get conditions for cancellativity. □

3 The cancellation law for sum based on an Archimedean continuous t-norm

In this section at first we describe some Zagrodny's results [10]. Further we will apply them for investigation of validity of cancellation law for addition of fuzzy numbers based on an Archimedean continuous t-norm.

3.1 The cancellation law for inf-convolution - Zagrodny's results

Definition 3. *Let $g,\ h : \mathbb{R} \to \mathbb{R} \cup \{\infty\}$. The inf-convolution of g and h at $x \in \mathbb{R}$ is defined by*

$$g \square h(x) := \inf_{y+z=x} (g(y) + h(z))\ .$$

Definition 4. *Let $h : \mathbb{R} \to \mathbb{R} \cup \{\infty\}$. The function h is said uniformly convex if for $\forall \varepsilon \ge 0\ \exists\, \delta > 0$*

$$|a-b| \ge \varepsilon\ \Rightarrow h\left(\frac{a+b}{2}\right) \le \frac{h(a)+h(b)-\delta|a-b|}{2},\ \forall\, a, b \in dom\ h\ .$$

Note that the domain of functions g, h can be restricted to some intervals. Zagrodny in [10] deal with more general functions in Banach space.

Theorem 3. *Let X be a reflexive Banach space. If $q, g, h : X \to \mathbb{R} \cup \{\infty\}$ are proper lower semicontinuous convex functions such that h is strictly convex and $\lim\limits_{\|x\| \to \infty} \frac{h(x)}{\|x\|} = \infty$ then $q \square h = g \square h$ implies $q = g$.*

Theorem 4. *Let X be a Banach space and $q, g, h : X \to \mathbb{R} \cup \{\infty\}$ be proper lower semicontinuous convex functions. Moreover, suppose h is uniformly convex. Then $q \square h = g \square h$ implies $q = g$.*

3.2 The cancellation law for sum based on a strict t-norm

Recall that addition of fuzzy quantities based on some Archimedean continuous t-norm T can be expressed by

$$A \oplus_T B(x) = f^{[-1]}(\inf_{y+z=x} (f(A(y)) + f(B(z))) = f^{[-1]} f \circ A \,\square\, f \circ B\,(x)),\ x \in \mathbb{R},$$

where f is additive generator of t-norm T, i.e., $f : [0,1] \to [0,\infty]$ is continuous strictly decreasing mapping verifying $f(1) = 0$, and pseudo-inverse $f^{[-1]} : [0,\infty] \to [0,1]$ of f is defined by

$$f^{[-1]}(x) = f^{-1}\left(\min\left(f(0), x\right)\right)\ .$$

Archimedean continuous t-norms can be divided into two classes: strict and nilpotent. An additive generator of a strict t-norm is unbounded, and then

$$f^{[-1]} = f^{-1}\ .$$

Theorem 5. *Consider a strict t-norm T with an additive generator f. Let A^a, B^b and C^c be fuzzy numbers, such that $f \circ B^b$ and $f \circ C^c$ are convex on components of* supp $B^b \setminus \{b\}$ *and* supp$\backslash C^c \backslash \{c\}$*, respectively and $f \circ A^a$ is either*

(i) uniformly convex on components of supp $A^a \setminus \{a\}$ *or*

(ii) strictly convex on components of supp $A^a \setminus \{a\}$

and if suppA^a *is unbounded then* $\lim\limits_{|x|\to\infty} \frac{f \circ A^a(x)}{|x|} = \infty$.

Then $A^a \oplus_T B^b = A^a \oplus_T C^c$ implies $B^b = C^c$.

Proof. Assume $f \circ B^b$, $f \circ C^c$ and $f \circ A^a$ verify conditions from theorem. Let $A^a \oplus_T B^b = A^a \oplus_T C^c$. This imply $f \circ A^a \,\square\, f \circ B^b = f \circ A^a \,\square\, f \circ C^c$ and by Zagrodny's results $f \circ B^b = f \circ C^c \Rightarrow B^b = C^c$, i.e. the cancellativity is valid. □

Remark 2. If A^a is LR fuzzy number we can require only strict convexity of $f \circ A^a$ on components of supp $A^a \setminus \{a\}$.

The strict convexity of $f \circ A$ is necessary condition for the cancellativity. See the following example.

Example 1. Consider the product t-norm T_P with additive generator $f(x) = -\ln x$ and fuzzy numbers $A_1(x) = e^{-x^2}$, $A_2(x) = e^{-|x|}$, $B_k(x) = e^{-k|x|}$, $k > 0$. Then T_P-sum of A_1 and B_k is given by

$$A_1 \oplus_{T_P} B_k(x) = \begin{cases} e^{-x^2}, & \text{if } |x| \leq \frac{k}{2} \\ e^{-k|x|+\frac{k^2}{4}}, & elsewhere\ . \end{cases}$$

and T_P-sum of A_2 and B_k for $k > 1$ is given by

$$A_2 \oplus_{T_P} B_k(x) = e^{-|x|}\ .$$

Evidently, $A_1 \oplus_{T_P} B_{k_1}(x) = A_1 \oplus_{T_P} B_{k_2}(x)$ if and only if $k_1 = k_2$ (observe that $f \circ A_1(x) = x^2$ is strictly convex). However, $A_2 \oplus_{T_P} B_{k_1}(x) = A_2 \oplus_{T_P} B_{k_2}(x)$ for all $k_1, k_2 \in]1, \infty[$ (observe that $f \circ A_2(x) = |x|$ is not strictly convex).

3.3 The cancellation law for sum based on a nilpotent t-norm

The case of nilpotent t-norm is more complicated. Conditions from Theorem 5 are deficient. See Example 2.

Example 2. Consider the Lukasiewicz t-norm T_L with additive generator $f(x) = 1 - x$ and LR-fuzzy numbers $A = (0, 1, 1)_{LL}$, $L = 1 - x^2$, $B = (0, 1, 1)$ and $C = (0, 0.9, 0.9)_{KK}$,

$$K = \begin{cases} 1 - 0.9x, & x \in [0, \frac{8}{9}] \\ 1.8 - 2x, & x \in (\frac{8}{9}, 1] \end{cases}\ .$$

Then components of $\operatorname{supp} A \backslash \{a\}$ are $(-1, 0)$ and $(0, 1)$, and $f \circ A$ is given on them by formula $f \circ A(x) = x^2$ (i.e., strictly convex function).
The components of $\operatorname{supp} B \backslash \{b\}$ are also $(-1, 0)$ and $(0, 1)$ and $f \circ B(x) = -x$ on $(-1, 0)$, $f \circ B(x) = x$ on $(0, 1)$ (i.e., convex functions).
Finally, the components of $\operatorname{supp} C \setminus \{c\}$ are $(-0.9, 0)$ and $(0, 0.9)$ and

$$f \circ C = \begin{cases} -2x - 0.8, & x \in (-0.9, -0.8] \\ -x, & x \in (-0.8, 0) \end{cases}\ .$$

respectively

$$f \circ C = \begin{cases} x, & x \in (0, 0.8] \\ 2x - 0.8, & x \in (0.8, 0.9) \end{cases}\ .$$

(i.e., convex functions).
However, the T_L-sum of A and B is the same as the T_L-sum of A and C.
$A \oplus_{T_L} B(x) = A \oplus_{T_L} C(x) = (0, 1.25, 1.25)_{MM}$

$$M = \begin{cases} 1 - \frac{25x^2}{16}, & x \in [0, 0.4] \\ 1.25(1 - x), & x \in (0.4, 1] \end{cases}\ .$$

Thus Theorem 5 is not valid in the case when T is a nilpotent t-norm, in general.
For nilpotent t-norms, we have only the following special cancellation theorems.

Theorem 6. *Consider a nilpotent t-norm T with normed additive generator f. Let A, B, C be fuzzy intervals, such that $f \circ A$, $f \circ B$ and $f \circ C$ are concave on the components of* $\operatorname{supp} A \setminus ker(A)$, $\operatorname{supp} B \setminus ker(B)$ *and* $\operatorname{supp} C \setminus ker(C)$, *respectively. Moreover, let $ker(A) = [a_1, a_2]$, $ker(B) = [b_1, b_2]$, $ker(C) = [c_1, c_2]$ and*

$$A(a_2 + x) \le \min\left(B(b_2 + x), C(c_2 + x)\right)$$
$$A(a_1 - x) \le \min\left(B(b_1 - x), C(c_1 - x)\right) \qquad \text{for all } x \in (0, \infty)\ .$$

Then $A \oplus_T B = A \oplus_T C \quad \Leftrightarrow B = C.$

The proof follows from the fact that under requirements of the theorem, $A \oplus_T B = A \oplus_{T_D} B$ and $A \oplus_T C = A \oplus_{T_D} C$, see [4, 7]. Note that the same claim is true also for strict t-norms. However then A, B, C have necessarily as support the whole real line $\mathbb{R}$.

Theorem 7. *Consider a nilpotent t-norm T with normed additive generator f. Let A, B, C be LR-fuzzy intervals, such that $f \circ L(x) = x^p$ and $f \circ R(x) = x^q$ for some $p, q \in (1, \infty)$. Then $A \oplus_T B = A \oplus_T C \quad \Rightarrow B = C.$*

Proof. Following [6], under requirements of the theorem, for $A = (a_1, a_2, \alpha_1, \alpha_2)_{LR}$, $B = (b_1, b_2, \beta_1, \beta_2)_{LR}$, $C = (c_1, c_2, \gamma_1, \gamma_2)_{LR}$, it holds

$$A \oplus_T B = (a_1 + b_1, a_2 + b_2, \delta_1, \delta_2)_{LR},$$

where $\delta_1^{\frac{1}{p}-1} = \alpha_1^{\frac{1}{p}-1} + \beta_1^{\frac{1}{p}-1}$ and $\delta_2^{\frac{1}{q}-1} = \alpha_2^{\frac{1}{q}-1} + \beta_2^{\frac{1}{q}-1}$.

Similarly,

$$A \oplus_T C = (a_1 + c_1, a_2 + c_2, \varepsilon_1, \varepsilon_2)_{LR}\,,$$

where $\varepsilon_1^{\frac{1}{p}-1} = \alpha_1^{\frac{1}{p}-1} + \gamma_1^{\frac{1}{p}-1}$ and $\varepsilon_2^{\frac{1}{q}-1} = \alpha_2^{\frac{1}{q}-1} + \gamma_2^{\frac{1}{q}-1}$.

Now, it is evident that $A \oplus_T B = A \oplus_T C$ if and only if $b_1 = c_1$, $b_2 = c_2$, $\beta_1 = \gamma_1$, $\beta_2 = \gamma_2$, i.e., $B = C$. □

Example 3. Consider the Yager's t-norm $T = T_Y^2$, with additive generator $f(x) = (1 - x)^2$ and linear fuzzy LR-numbers $A = (a, \alpha, \alpha)$, $B = (b, \beta, \beta)$, $C = (c, \gamma, \gamma)$. Then $f \circ L(x) = f \circ R(x) = x^2$, i.e., $p = q = 2$. Consequently

$$(a, \alpha, \alpha) \oplus_{T_Y^2} (b, \beta, \beta) = (a, \alpha, \alpha) \oplus_{T_Y^2} (c, \gamma, \gamma)$$
$$(a + b, \sqrt{\alpha^2 + \beta^2}, \sqrt{\alpha^2 + \beta^2}) = (a + c, \sqrt{\alpha^2 + \gamma^2}, \sqrt{\alpha^2 + \gamma^2})$$

$\Rightarrow\ b = c\ \&\ \beta = \gamma.$

4 Conclusion

We have discussed the cancellation law for T-based addition of fuzzy intervals. This genuine property of the classical addition of reals is preserved for T_M-based addition of fuzzy intervals. Consequently, every fuzzy linear equation (with real multiplicative constant) of one variable if solvable then it has unique solution.
In the case of a t-norm T different from T_M, the cancellation law for T-based addition of fuzzy intervals does not hold, in general. However, it is still valid under some specific requirements - for example in specific classes of fuzzy numbers determined by the choice of T.
Presented resuls can be generalized also for pseudo-convolutions introduced in [9], following the ideas presented in [3].

Acknowledgement. The work on this contribution was supported by Science and Technology Assistance Agency under the contract No. APVT-20-046402.

References

1. De Baets B, Marková - Stupňanová A, (1997) Analytical expressions for the addition of fuzzy intervals. Fuzzy Sets and Systems 91:203–213
2. Mareš M, (1994) Computation Over Fuzzy Quantities. CRC Press, Boca Raton
3. Marková - Stupňanová A, (1999) A note on the idempotent functions with respect to pseudo-convolution. Fuzzy Sets and Systems 102:417–421
4. Marková - Stupňanová A, (1997) A note to the addition of fuzzy numbers based on a continuous Archimedean t-norm. Fuzzy Sets and Systems 91: 253–258
5. Marková A, (1997) T-sum of L-R fuzzy numbers. Fuzzy Sets and Systems 85:379–384
6. Mesiar R, (1997) Shape preserving additions of fuzzy intervals. Fuzzy Sets and Systems 86:73–78
7. Mesiar R, (1997) Triangular-norm-based addition of fuzzy intervals. Fuzzy Sets and Systems 91:231–237
8. Moynihan R, (1975) On the Class of τ_T Semigroups of Probability Distribution Functions. Aequationes Mathematicae, Birk. Verlag Basel 12 (2/3):249–261
9. Pap E, (1995) Null-Aditive Preserved by Given Pseudo-Convolution * Set Functions. Ister Science & Kluver Academic Publishers, Dordrecht
10. Zagrodny D,(1994) The cancellation law for inf-convolution of convex functions. Studia Mathematika 110 (3):271–282

Generic View On Continuous T-Norms and T-Conorms*

Irina Perfilieva

University of Ostrava
Institute for Research and Applications of Fuzzy Modeling
30. dubna 22, 701 03 Ostrava 1, Czech Republic

Irina.Perfilieva@osu.cz

1 Introduction

Triangular norms were introduced in the framework of probabilistic metric spaces. However, they are widely applied in several other fields, e.g., in fuzzy set theory, fuzzy logic, and their applications. Therefore, the problem of compact representation of continuous t-norms using generating functions is of permanent interest.

2 Functional Representation of Continuous t-Norms and t-Conorms

Let I be a countable and linearly ordered family of indices and

$$\mathcal{F}_I = \{f_i, \quad i \in I\},$$

be a family of continuous and strictly monotonously increasing functions such that

$$f_i : [0,1] \longrightarrow [0, c_i], c_i \leq +\infty, \quad \text{and} \quad f_i(0) = 0$$

for all $i \in I$. Each $f_i, i \in I$, determines the function

$$g_i(x) = f_i(1-x)$$

which is an additive generator of some continuous Archimedean t-norm. Therefore, we will refer to functions from $\mathcal{F}_I$ as *generating functions*.

Definition 1. *For a countable and linearly ordered family of indices I, let $\mathcal{A}_I = \{a_i\}_{i\in I}$, $\mathcal{B}_I = \{b_i\}_{i\in I}$, be two families of nodes from the interval $[0,1]$ such that for all $i \in I$*

* This paper has been partially supported by grant IAA1187301 of the GA AV ČR.

I. Perfilieva: *Generic View On Continuous T-Norms and T-Conorms*, Advances in Soft Computing **2**, 377–382 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

$$a_i < b_i,$$

and for all $i, j \in I$

$$i < j \Rightarrow (a_i < a_j)\ \&\ (b_i < b_j)\ \&\ (b_i \leq a_j).$$

We say that the families $\mathcal{A}_I$ *and* $\mathcal{B}_I$ *determine the* partition $\mathcal{P}_{\mathcal{A},\mathcal{B},I}$ *of* $[0,1]$*:*

$$[0,1] = \bigcup_{i\in I}(a_i, b_i) \cup D$$

where

$$D = [0,1] \setminus \bigcup_{i\in I}(a_i, b_i).$$

On each subinterval $[a_i, b_i]$, $i \in I$, we define two linear transition functions $\varphi_i, \psi_i : [a_i, b_i] \longrightarrow [0,1]$ such that

$$\varphi_i(x) = \frac{b_i - x}{b_i - a_i}, \quad \psi_i(x) = \frac{x - a_i}{b_i - a_i}. \tag{1}$$

It is easy to see that the following equalities hold true for each $i \in I$ and for each $x \in [a_i, b_i]$:

$$\varphi_i(x) + \psi_i(x) = 1,$$
$$\varphi_i(x) = \psi_i(a_i + b_i - x).$$

The inverse functions $\varphi_i^{-1}, \psi_i^{-1} : [0,1] \longrightarrow [a_i, b_i]$ fulfil the following equality for arbitrary $x \in [0,1]$:

$$\varphi_i^{-1}(x) + \psi_i^{-1}(x) = a_i + b_i.$$

The following proposition immediately follows from the representation of an arbitrary continuous t-norm (t-conorm) as an ordinal sum of continuous Archimedean t-norms (t-conorms) (see [1]).

Proposition 1. *Let* I *be at most countable, linearly ordered family of indices,* $\mathcal{F}_I$ *a family of generating functions, and* $\mathcal{P}_{\mathcal{A},\mathcal{B},I}$ *be a partition of* $[0,1]$ *determined by families* $\mathcal{A}_I, \mathcal{B}_I$*. Let transition functions be defined by (1). Then*

$$T(x,y) = \begin{cases} \varphi_i^{-1}(f_i^{-1}(\min(f_i(\varphi_i(x)) + f_i(\varphi_i(y)), f_i(1)))), & \text{if } (x,y) \in (a_i, b_i)^2, \\ \min(x,y), & \text{otherwise} \end{cases} \tag{2}$$

is a continuous t-norm and

$$S(x,y) = \begin{cases} \psi_i^{-1}(f_i^{-1}(\min(f_i(\psi_i(x)) + f_i(\psi_i(y)), f_i(1)))), & \text{if } (x,y) \in (a_i, b_i)^2, \\ \max(x,y), & \text{otherwise} \end{cases} \tag{3}$$

is a continuous t-conorm. Moreover, each continuous t-norm (t-conorm) can be represented in the form (2) (resp. (3)).

From the properties of transition functions φ_i, ψ_i and their inverse, it is easy to prove that for the t-norm T and the t-conorm S given by (2) and (3), the following relation holds true for arbitrary $x, y \in (a_i, b_i)$:

$$S(x, y) = a_i + b_i - T(a_i + b_i - x, a_i + b_i - y).$$

3 Generalized Additive Generators of Continuous t-norms and their Residua

In this section, we will focus on the representation of continuous t-norms with help of generalized additive generators. By definition (see [1]), an additive generator $g : [0, 1] \longrightarrow [0, \infty]$ of a t-norm T is a strictly decreasing function which fulfils two conditions:

1. g is right-continuous at 0 and $g(1) = 0$,
2. $$T(x, y) = g^{-1}(\min(g(0), g(x) + g(y))). \tag{4}$$

It is known (cf. [1]) that it is possible to construct the representation (4) for continuous Archimedean t-norms. Moreover, if a t-norm has an additive generator then it is necessarily Archimedean. We will generalize representation (4) to the case of arbitrary continuous t-norm (see Theorem 1 below).

On the basis of Proposition 1, we may claim that a continuous t-norm T can be fully characterized by a pair $(\mathcal{F}_I, \mathcal{P}_{\mathcal{A},\mathcal{B},I})$ and moreover, each function f_i from $\mathcal{F}_I$ determines the additive generator

$$g_i(x) = f_i(\varphi_i(x)) \tag{5}$$

of the respective Archimedean part on interval $[a_i, b_i]$, $i \in I$. We will show that each continuous t-norm can be represented using truncated arithmetic sum of negative reals (see expression (7)). Furthermore, the operation of residuum of a continuous t-norm can be represented using truncated arithmetic subtraction of negative reals (see expression (8)).

We restrict ourselves to continuous non-Archimedean t-norms, because the representation (4) has been constructed in ([1]) for continuous Archimedean t-norms. Let a continuous non-Archimedean t-norm T be characterized by the pair $(\mathcal{F}_I, \mathcal{P}_{\mathcal{A},\mathcal{B},I})$ where $|I| \geq 2$, and additive generators of the respective Archimedean parts on intervals $[a_i, b_i]$ are given by (5). We consider the following set of couples of reals:

$$R_T^- = \bigcup_{a \in D} \{(a, 0)\} \cup \bigcup_{i \in I} \{b_i\} \times (0, -c_i)$$

and define the lexicographic order on R_T^-:

$$(x_1, y_1) \leq (x_2, y_2) \Leftrightarrow (x_1 < x_2) \vee (x_1 = x_2 \,\&\, y_1 \leq y_2)$$

as well as one-to-one mapping $g : [0,1] \longrightarrow R_T^-$:

$$g(x) = \begin{cases} (x,0), & \text{if } x \in D, \\ (b_i, -g_i(x)), & \text{if } x \in (a_i, b_i). \end{cases} \tag{6}$$

The construction of R_T^- is fully determined by the partition $\mathcal{P}_{\mathcal{A},\mathcal{B},I}$ of $[0,1]$ and therefore, by the choice of the t-norm T. This dependence is marked by the lower index in the denotation R_T^-.

Let us introduce the operations of truncated sum and truncated subtraction on R_T^- as follows:

$$(x_1,y_1)\dot{+}(x_2,y_2) = \begin{cases} \min((x_1,y_1),(x_2,y_2)), & \text{if } x_1 \neq x_2, \\ (x_1, \max(y_1+y_2, -c_i)), & \text{if } (x_1 = x_2)\ \&\ (y_1, y_2 \in (0,-c_i)), \end{cases}$$

and

$$(x_1,y_1) \dot{-} (x_2,y_2) = \begin{cases} (x_1, y_1 - y_2), & \text{if } x_1 = x_2\ \&\ y_1 < y_2, \\ (1,0), & \text{if } x_1 > x_2 \text{ or } x_1 = x_2\ \&\ y_1 \geq y_2, \\ (x_1,y_1), & \text{if } x_1 < x_2. \end{cases}$$

Then we can prove the following representation theorem.

Theorem 1. *Let a continuous t-norm T be characterized by the pair $(\mathcal{F}_I, \mathcal{P}_{\mathcal{A},\mathcal{B},I})$ where $|I| \geq 2$. Moreover, let I_T denote the respective residuum of the t-norm T. Then*

$$T(x,y) = g^{-1}(g(x) \dot{+} g(y)) \tag{7}$$

and

$$I_T(x,y) = g^{-1}(g(y) \dot{-} g(x)) \tag{8}$$

where g is the generalized additive generator (6) of the t-norm T.

We can prove even more if we consider a standard BL-algebra (see [2]) on $[0,1]$

$$\mathcal{L} = \langle [0,1], \vee, \wedge, *, \rightarrow, \mathbf{0}, \mathbf{1} \rangle$$

where $*$ is a continuous t-norm and $\rightarrow$ is its residuum.

Theorem 2. *Let $\mathcal{L}$ be a standard BL-algebra on $[0,1]$ and $*$ be a continuous t-norm characterized by the pair $(\mathcal{F}_I, \mathcal{P}_{\mathcal{A},\mathcal{B},I})$ where $|I| \geq 2$. Then additive generator g of $*$ defined by (6) establishes an isomorphic mapping between $\mathcal{L}$ and the following algebra on R_*^- with truncated operations:*

$$\mathcal{R}_*^- = \langle R_*^-, \vee, \wedge, \dot{+}, \dot{-}, (0,0), (1,0) \rangle.$$

4 Generalized Additive Generators of Continuous t-conorms

In this section, we will suggest the representation of continuous t-conorms with help of generalized additive generators. In Proposition 1, we have presented the representation of an arbitrary continuous t-conorm as an ordinal sum of continuous Archimedean t-conorms. Let us recall (see [1]) that the additive generator $g : [0,1] \longrightarrow [0,\infty]$ of a t-conorm S is a strictly increasing function $g : [0,1] \longrightarrow [0,\infty]$ which fulfils two conditions:

1. g is left-continuous at 1 and $g(0) = 0$,
2. $$S(x,y) = g^{-1}(\min(g(0), g(x) + g(y))). \tag{9}$$

Similar to t-norms, it is known [1] that it is possible to construct the representation (9) for continuous Archimedean t-conorms. We will generalize this kind of representation to the case of arbitrary continuous t-conorm (see Theorem 3 below).

Let a continuous t-conorm S be characterized by the pair $(\mathcal{F}_I, \mathcal{P}_{\mathcal{A},\mathcal{B},I})$ where $|I| \geq 2$, and moreover, each function f_i from $\mathcal{F}_I$ determines the additive generator

$$g_i(x) = f_i(\psi_i(x)) \tag{10}$$

of the respective Archimedean part on interval $[a_i, b_i]$, $i \in I$.

We consider the following set of couples of reals:

$$R_T^+ = \bigcup_{a \in D} \{(a,0)\} \cup \bigcup_{i \in I} \{b_i\} \times (0, c_i).$$

and define the lexicographic order on R_T^+:

$$(x_1, y_1) \leq (x_2, y_2) \Leftrightarrow (x_1 < x_2) \vee (x_1 = x_2 \,\&\, y_1 \leq y_2)$$

as well as the one-to-one mapping $g : [0,1] \longrightarrow R_T^+$:

$$g(x) = \begin{cases} (x, 0), & \text{if } x \in D, \\ (a_i, g_i(x)), & \text{if } x \in (a_i, b_i). \end{cases} \tag{11}$$

Let us introduce the operations of truncated sum on R_T^+ as follows:

$$(x_1, y_1) \dot{+} (x_2, y_2) = \begin{cases} \max((x_1, y_1), (x_2, y_2)), & \text{if } x_1 \neq x_2, \\ (x_1, \min(y_1 + y_2, c_i)), & \text{if } (x_1 = x_2) \,\&\, (y_1, y_2 \in (0, c_i)). \end{cases}$$

Theorem 3. *Let a continuous t-conorm S be characterized by the pair $(\mathcal{F}_I, \mathcal{P}_{\mathcal{A},\mathcal{B},I})$ where $|I| \geq 2$. Then*

$$S(x,y) = g^{-1}(g(x) \dot{+} g(y)) \tag{12}$$

where g is the generalized additive generator (11) of the t-conorm S.

5 Conclusion

We have suggested the representation of continuous t-norms and t-conorms with help of generalized additive generators. This representation shows similar origin of triangular operations (t-norms and t-conorms) as truncated sums of negative and positive reals, respectively. Moreover, we showed that a generalized additive generator establishes an isomorphic mapping between a standard BL-algebra $\mathcal{L}$ on $[0, 1]$ and the algebra on the set of couples of reals R_*^- with truncated operations.

References

1. Klement P, Mesiar R, Pap E (2001) Triangular norms. Kluwer, Dordrecht
2. Hájek P (1998) Metamathematics of fuzzy logic. Kluwer, Dordrecht

Invited Session

Intelligent Techniques for Knowledge Extraction and Management

Session Organiser:
Ernesto Damiani

Mining Class Hierarchies from XML Data: Representation Techniques

Paolo Ceravolo and Ernesto Damiani

Department of Information Technology
University of Milan
Via Bramante, 65 - 26013 Crema (Italy)
damiani, ceravolo@dti.unimi.it

1 Introduction

In this paper, we describe a technique for extracting patterns to a XML data flow; then, we show how such patterns can be developed into an ontology of classes. Also, we discuss the impact of different fuzzy representation techniques for XML data on the outcome of our procedure. One might wonder why all this is needed, since the semantics of XML data could in principle be satisfactorily represented via their associated XML schemata `ComplexTypes`. Unfortunately it turns out that standard XML schema definitions need to cover a wide repertoire of possible attributes. For this reason, optional elements are widely used, thus decreasing the expressiveness of XML schemata as descriptors of the content of single instances. Our approach relies on comparing fuzzy encodings of XML fragments. This comparison will allow us to define "typical" sets of attributes, that we shall consider hints to possible meaningful classes. Then, we shall evaluate fuzzy overlapping between candidate cluster heads in order to define a tentative class hierarchy. Our fuzzy modelling assumes that a domain expert has associated an importance degree in the $[0, 1]$ interval to vocabulary elements (i.e. tag names). As we shall see in the remainder of the paper, this burden is not excessive, since this importance assessment only needs to be carried out once, looking at the schema. At run time, each incoming XML fragment is mapped into a fuzzy set whose elements are the tag names [3]. Each element membership is computed by aggregating the vocabulary importance values of the tags lying on the path from it to the root. The topology of the individual XML tree is modelled by using an aggregation that takes into account nodes nesting level or nodes occurrence. Our procedure consists of the following steps:

- Identify and choose the "right" fuzzy comparison function for a given XML data flow;
- Classify the authentication data flow according to it;

P. Ceravolo and E. Damiani: *Mining Class Hierarchies from XML Data: Representation Techniques*, Advances in Soft Computing **2**, 385–396 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

- Build a tentative `is-a` hierarchy based on classes overlapping.

The comparison function M will be based on the degree of inclusion of the fuzzy sets modelling the XML fragments. These functions have been extensively studied in the literature [1], [4], [3]. The comparison function must to maintain a semantical coherence with assignment function adopted in evaluating the nodes membership values. For this reason in Section 3 we discuss on the right way of coupling weight assignments and comparison function. In most cases comparison function are not symmetrical and lack the mathematical properties of a distance [1] . As a consequence, no standard clustering technique can be used. Instead, our technique:

- Periodically scans portions of the incoming flow to select typical XML fragments as tentative "clusterheads" representing classes.
- Computes the comparison of each incoming fragment with the current clusterheads and add it to the class of the closest match.

After creating the classes, we evaluate the quality of the classification by computing each class *cohesion*, i.e. the comparison of the elements of a class with the elements in other classes whose distances from their own clusterhead is maximum. If the classification is satisfactory, i.e. the *cohesion* of all classes is high enough, we apply a suitable comparison measure M applied to clusterheads in order to estimate *class overlapping*. When the comparison between two classes A and B is higher than a pre-set threshold we assume a generalization relation exists, directed along increasing comparison score. Finally, the generalization hierarchy obtained in the previous steps is first converted into a metadata format suitable for ontologies like RDF [5] of OWL [6] and then presented to the user via a standard ontology editor. The paper is organized as follows: in Section 2, we introduce the basic notions of fuzzy representation of XML data and outline their application in the framework hierarchy mining. In Section 3, we discuss in further detail the role of weighting and comparison functions when XML data are represented as fuzzy sets. Section 4 discusses how fragments can be used to build a class hierarchy and, eventually, a complete ontology. Finally in Section 5 we draw the conclusion and give an outline of our future work.

2 A Fuzzy Pattern Extraction Technique

In this section we represent XML data items as flat fuzzy sets. For transforming an XML tree into a fuzzy set, we need to evaluate its structure for assigning relevance values to nodes in a way reminiscent of their original positions in the XML tree topology. In other words, we use the fuzzy membership

[1] This lack of symmetry is reasonable, since these functions model object comparison rather than a generic similarity.

degree for expressing the relevance of each XML node as it is suggested by its structural position in the original XML tree.
In a XML Schema a node can occur repetitively and in different positions of the tree. A collection of elements which may contain duplicates is called a *bag* (also called *multiset*).
We use the standard notation for crisp bags, as follows: $A =< a, b, b, c, d, d >=< 1 * a, 2 * b, 1 * c, 2 * d >$. Extending this notation to fuzzy bags, we obtain:

$$A = \{\{0.64/2\} * a, \{0.66/1, 0.82/1\} * b, \{0.49/1, 0.71/1\} * c\}$$

Where we used the standard notation putting each element's membership value before the slash character, followed by the number of occurrences associated to that value. We are now ready to describe the basic steps of our technique for comparing XML fragments; in the next sections, we shall propose a coupling of membership functions and matching measures suitable to take into account XML data items topology. Before computing matching between XML fragment representations, however, we have to:

- Represent XML trees in term of fuzzy sets. During this step we need to assign weight to nodes. Evaluating each node we assign two relevance values to it:
  - an explicit one, according to the relevance of its tag label in the current domain vocabulary.
  - an implicit one, taking into account its position in the tree structure.
- Aggregate these two values, obtaining a single membership degree.
- Match objects and get a measure of resemblance between them.

According to the application requirements, different techniques can be used for each step. In the following Table 1 we outline the main options:

Weighting	*Type*
`Nodes Depth`	Topology
`Fan-out`	Topology
`Domain Expert`	Tag
`Content Size/Type`	Tag

Table 1. *Weight Assignment Techniques*

Table 1 distinguishes among Topology- and Tag-based weight assignment techniques. Indeed, depth and fan-out are two simple techniques for evaluating the relevance of a node based on its topological properties. If relevance has to be estimated based on the node content two approaches are possible. First, one could ask a domain expert to assign the nodes weights; secondly, weights

can be computed based on the node content size and/or according to serialized content type [3].

Topology and Tag-based weights must be aggregated in a single value using a standard t-norm. An *aggregation function* allows us to tune the relevance we want to attribute to the tag position in the topology rather than to the node tag itself. In [3] a list of aggregation function are proposed. In this paper, we adopt a parametrical average-based norm, supporting fine tuning. Namely, we set:

$$F_k(x, y) = k(x + y - xy) + (1 - k)(xy) \quad (1)$$

Each XML data item can now be represented as a fuzzy set where elements membership is given by the combination of weights described above. This allows us to use a standard functions for objects comparison to compute clusters to be later refined into classes (Sect. 4). Fuzzy logic versions of standard comparison functions have been extensively studied in the literature. In [4], Damiani, Bosc, and Fugini describe comparison measures based on different type of implication; Bouchon-Meunier and Rifqi [1] propose a complete classification of comparison measures distinguish among measures of *satisfiability, resemblance and inclusion*. In the remainder of the paper, we will discuss coupling between some weighting and comparison measures depending on the semantics of XML file structure.

3 Combining Weighting and Comparison Functions

Let us ground our discussion in an example. An XML data item describing book references can have different structures. If you are describing a book for an e-business service you will use a closed set of information for each instance. If you are describing a book for a bibliography search engine you are interested in the number of citations per book, and this kind of information is not predetermined for all instances.
Obviously, different weighting functions must be used in these two cases. Also, comparing bibliography entries is an entirely different matter than comparing a publisher's book catalog entries. In this section, we will propose some associations between membership and comparison functions, depending on the purpose of their utilization.

3.1 Focusing on Nodes Depth

In the first case, we have to compare XML files where the structure is used for distinguishing general elements, i.e. internal nodes, from data level elements, i.e. tree leaves. From a schema point of view, high-level elements (i.e. the ones closer to the root) are more relevant than low-level ones. Fig. 1 shows an example of such an XML file [2].

[2] For the sake of clarity we omit the XML elements content, as it is not relevant to our discussion

```
<DOC>
<BOOK>
<TITLE></TITLE>
<AUTHOR></AUTHOR>
</BOOK>

<EDITION>
<PUBLISHER></PUBLISHER>
<EDITOR></EDITOR>
<ISBN></ISBN>
<PRICE></PRICE>
</EDITION>
</DOC>
```

Fig. 1. *The book references XML data item*

Such an XML tree can be represented assigning weights to elements according to a pre-classification of the domain vocabulary and to a weighting function W expressing their distance from the root. The domain vocabulary used in this example is shown in the second column of the Table 2. A simple weighting function is the following:

$$W = \frac{d_{max} - l}{d_{max}}$$

where d_{max} is the longest path in the tree from the root to a leaf node. The *level weighting* column in Table 2 shows weight assignments according to the *nodes depth function*. Now, values have to be aggregated by means of our *aggregation function* F_k, as explained in Section 2. The third column of the table shows the results of the function assuming to set parameter k to 0.4, i.e. giving more emphasis to the tag position in the tree than to the tag name pre-set importance stored in the vocabulary.

Element	*Term Weighting*	*Level Weighting*	*Aggregation Function*	*Label*
`Book`	1	0.67	0.8	B
`Title`	1	0.33	0.6	T
`Author`	0.9	0.33	0.55	A
`Edition`	0.7	0.33	0.46	En
`Publisher`	0.6	0	0.24	Pu
`Editor`	0.5	0	0.2	Er
`ISBN`	0.8	0	0.32	I
`Price`	0.6	0	0.24	Pr

Table 2. *Elements Weighting to the XML in Fig. 1*

We can now provide a representation of the XML file in Fig. 1 as a fuzzy set:

$F1 = \{0.8/B, 0.6/T, 0.55/A, 0.46/En, 0.24/Pu, 0.2/Er, 0.32/I, 0.24/Pr\}$

Following the same procedure, a second XML file (Fig. 2), with same tags but with the element `Price` in a different position, would be represented as follows:

$F2 = \{0.8/B, 0.6/T, 0.55/A, 0.46/En, 0.24/Pu, 0.2/Er, 0.32/I, 0.41/Pr\}$

```
<DOC>
<BOOK>

<TITLE></TITLE>
<AUTHOR></AUTHOR>

<EDITION>
<PUBLISHER></PUBLISHER>
<EDITOR></EDITOR>
<ISBN></ISBN>
</EDITION>

</BOOK>

<PRICE></PRICE>
</DOC>
```

Fig. 2. *Another XML data item*

3.2 Comparing XML Data Items Representations Based on Nodes Depth

Comparing fuzzy sets $F1$ and $F2$, we are interested in differences due to the different topological structures of the XML trees they encode. Disregarding pre-set importance values of tags stored in the vocabulary[3], our comparison function has to return a different score when we compare a set generated by a shallow XML tree to a set generated by a deeply nested one than viceversa. In other words, we need an asymmetrical comparison. Suppose that two fuzzy sets obtained encoding XML trees, A and B, have the same support, i.e. their difference is only due to the original topology of the trees they encode. Suppose also that the only difference between the two sets is due to an element x having a higher membership value in A, e.g. because node x was positioned

[3]In the remainder of the section, we assume all elements have the same pre-set importance and that all differences in membership are due to topology.

in $tree(A)$ closer to the root than it was in $tree(B)$[4]. Then, when x has a higher membership in A than in B, we need $M(A,B) < M(B,A)$. Note that $\Delta = |M(A,B) - M(B,A)|$ should depend on the types of elements involved, i.e. Δ should depend on the type of x[5].

Specifically, measure M allows us to cluster objects aiming to a hierarchy construction. In fact $M(A,B)$ returns us a different value then $M(B,A)$, and Δ can be used to estimate a direction in the hierarchy.
Following Bouchon-Meunier [1] we can define two basic measures satisfying our requirements.

Inclusion:

$$M(A,B) = \frac{A \cap B}{A} = \frac{A \cap B}{(A \cap B) + (A - B)} \tag{2}$$

That can be represented as follows

$$\frac{\sum min(\mu_a(x), \mu_b(x))}{\sum min(\mu_a(x), \mu_b(x)) + max(0, \mu_a(x) - \mu_b(x)} \tag{3}$$

Note that *Inclusion* is a typical monotonic measure. Using for instance the *Inclusion* measure for our example we get:

$$F1 \cup F2 = \{0.7/B, 0.4/T, 0.36/A, 0.46/En, 0.24/Pu, 0.2/Er, 0.32/I, 0.24/Pr\}$$

$$F1 - F2 = \{0.1/B, 0.2/T, 0.19/A, 0/En, 0/Pu, 0/Er, 0/I, 0/Pr\}$$

Hence

$$M(F1, F2) = 1.49$$

In order to normalize this result we can use a simple function such as $\frac{1}{x}$, getting a final value equal to 0.67. At contrary $M(F2, F1)$ produces a score of 0.92.

[4]We informally write $tree(X)$ to denote a (not unique) XML tree encoded by fuzzy set X, given a vocabulary and a value of k

[5]Of course, the opposite choice could also be made under different assumptions. For instance in a *Satisfiability* measure we have $M(A,B) > M(B,A)$: $M(A,B) = \frac{A \cap B}{B} = \frac{A \cap B}{(A \cap B) + (B - A)}$. To put it more clearly, a *Satisfiability* measure represents the fact that a phone is "more included" in a computer than a computer in a phone, because most phone features are also held by the computer, but the viceversa is not true. The strategy described in the paper is the one usually adopted for O-O schema design, i.e. the simpler structure is more general, and therefore more similar to the more specific ones than these specializations are similar to it.

3.3 Focusing on Nodes Fan-out

In some cases we want to compare XML files giving more emphasis to elements occurrence than to the original trees topology. With reference to our running example about books, we assume to work with a corpus of XML files describing bibliography information. In this application the number of citations per book, and the number of cited books are very important. For representing our objects we use a different domain vocabulary; also, we employ a different function for defining nodes relevance into the XML data item. We can think of all kind of functions taking into account the number of sub-nodes starting from a given node. In our example, we use a very simple function, dividing the number of children of a node on the basis of a given parameter.

$$F(x) = \frac{Fanout(x)}{Fanout(x) + \beta}$$

Taking β equal to 1 we represent the fuzzy set version of XML files in Fig. 3 and Fig. 4 according to the Table 3.

```
<DOC>
<BOOK>

<TITLE></TITLE>
<AUTHOR></AUTHOR>

<CITED>
<BOOK><TITLE></TITLE><AUTHOR></AUTHOR></BOOK>
<BOOK><TITLE></TITLE><AUTHOR></AUTHOR></BOOK>
</CITED>

<CITATION>
<BOOK><TITLE></TITLE><AUTHOR></AUTHOR></BOOK>
<BOOK><TITLE></TITLE><AUTHOR></AUTHOR></BOOK>
<BOOK><TITLE></TITLE><AUTHOR></AUTHOR></BOOK>
<BOOK><TITLE></TITLE><AUTHOR></AUTHOR></BOOK>
<CITATION>

</BOOK>
</DOC>
```

Fig. 3. *A Book Citation XML File*

It is also possible to use a *fuzzy bag* representation, taking into account multiple occurrences of the same element. In this case, we can express the XML file in Fig. 3 as follows:

```
<DOC>
<BOOK>

<TITLE></TITLE>
<AUTHOR></AUTHOR>

<CITED>
<BOOK><TITLE></TITLE><AUTHOR></AUTHOR></BOOK>
<BOOK><TITLE></TITLE><AUTHOR></AUTHOR></BOOK>
<BOOK><TITLE></TITLE><AUTHOR></AUTHOR></BOOK>
<BOOK><TITLE></TITLE><AUTHOR></AUTHOR></BOOK>
<BOOK><TITLE></TITLE><AUTHOR></AUTHOR></BOOK>
<BOOK><TITLE></TITLE><AUTHOR></AUTHOR></BOOK>
</CITED>

<CITATION>
<BOOK><TITLE></TITLE><AUTHOR></AUTHOR></BOOK>
<CITATION>

</BOOK>
</DOC>
```

Fig. 4. *A Different Version of the Book Citation XML File*

Element	*Term Weighting*	*Level Weighting*	*Aggregation Function*	*Label*
`Book`	1	0.96	0.97	B
`Title`	1	0	0.4	T
`Author`	0.9	0	0.36	A
`Cited`	0.8	0.86	0.8	Cd
`Citation`	0.7	0.92	0.78	Cn
`Book`	1	0.67	0.8	B
`Title`	1	0	0.4	T
`Author`	0.9	0	0.36	A

Table 3. *Elements Weighting to the XML in Fig. 3*

$F3 = \{\{0.97/1, 0.8/6\} * B, \{0.4/7\} * T, \{0.36/7\} * A, \{0.8/2\} * Cd, \{0.78/4\} * Cn\}$

Much in the same way, it is possible represents the XML in Fig. 4 as reported below.

$F3 = \{\{0.97/1, 0.8/7\} * B, \{0.4/8\} * T, \{0.36/8\} * A, \{0.8/6\} * Cd, \{0.78/1\} * Cn\}$

3.4 Comparing XML Files Representations Based on Nodes Fan-out

Comparing $F3$ and $F4$ we are interested extracting a difference on the basis of elements occurrences. An object A, represented by a bag where some elements have a high number of occurrences, must be similar to another object B, with the number of occurrences is roughly the same and the result of comparison have to be equal regardless if comparing A to B or B to A. In other words we need a symmetric measure of similarity.
Following again [1], the following measure can be used:

Similarity:

$$M(A,B) = \frac{A \cap B}{A \cup B} = \frac{A \cap B}{(A \cap B) + (A - B) + (B - A)}$$

that is, in a fuzzy representation:

$$\frac{\sum min(\mu_a(x), \mu_b(x))}{\sum min(\mu_a(x), \mu_b(x)) + max(0, \mu_a(x) - \mu_b(x) + max(0, \mu_b(x) - \mu_a(x)} \quad (4)$$

Using this function, the similarity degree between our sample sets $F3$ and $F4$ is equal to 0.57.

4 Building the Hierarchy

Once a set of patterns has been identified by computing the similarity values between all XML documents, building a hierarchy is not difficult. Similarity allows for defining clusters composed of documents whose representation is close to a pattern [6].

After creating the classes, we evaluate the quality of the classification we obtained by computing each class *cohesion*, i.e. the comparison score of the elements of a class with the elements in other classes whose distances from their own clusterhead is maximum. If the classification is satisfactory, i.e. the *cohesion* of all classes is high enough, we apply a suitable comparison function M applied to clusterheads in order to estimate *class overlapping*. When the

[6] Here we loosely use the term "cluster", even if this is not completely correct. This term is only appropriate for symmetrical comparison functions. When we use a *depth-based comparison function*, our blocks are not the result of clustering, since the chosen similarity lacks the mathematical property of a distance. Intuitively, however, patterns within a valid cluster are more similar to each other than they are to a pattern belonging to a different cluster

comparison score between two classes A and B is higher than the threshold α we assume a generalization relation exists between A and B, directed along increasing comparison score. The next step is to assign meaning to candidate classes; this task is manually done by domain experts who relate derived classes structure of data to domain theory, and, more specifically to the current problem solving situation. This process can also lead to discovery of new concepts that improve problem solving and extend domain theories.
We evaluate clustering validity in term of class cohesion or intra-class similarity. This is defined in terms of how individual documents match the prototypical description of the cluster they are assigned to by algorithm. Classes that exhibit high class cohesion lead to a better classification.
Once obtained classes representing our domain the last task of our work is discover relations between them. We represent classes as vertices of a direct graph. Every edge joins the class with lower cardinality with classes with higher cardinality. Labels associated to the edges express the average value between the similarity values resulting from the comparison between the interested classes (Fig. 5(a)).
The initial direct cyclic graph has to be converted in a direct acyclic graph that better express inheritance relations. If it is possible, in Fig. 5(b) and Fig. 5(c) two results of this process are shown. In Fig. 5(b) for example, we show the *Directed Acyclic Graph* (DAG) corresponding to threshold $\alpha = 0.5$. This situation shows how not always is possible to obtain a tree as we can instead see in Fig. 5(c), where choosing an higher threshold or based on a decision by a domain expert, the edge between CH3 and CH4 is removed.

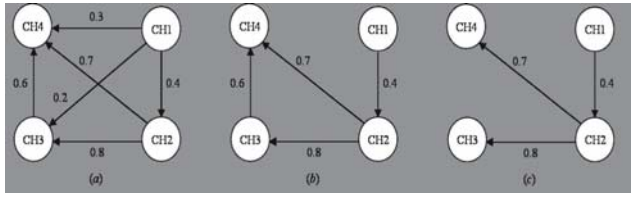


Fig. 5. (a): the DAG representing classes and relations between them. (b) and (c): the DAG and the tree resulting from the process of extraction of inheritance relations.

Finally, the generalization hierarchy obtained in the previous steps is (i) converted into a metadata format suitable for ontologies like RDF [5] of OWL [6] and (ii) presented to the user via a standard ontology editor. Clusters are expressed as constraints on the initial XML schema[7] and are provided to the ontology engineer for a final pruning. In other words such an ontology

[7]If no schema is available we assume to use the domain vocabulary as a first lossy approximation of a schema.

schema is builded mixing top-down and bottom-up approaches, and using semi-automated procedures, giving an important support for enabling knowledge representation in fine grained access control applications [2]. Periodically we scan portions of the incoming flow to reselect typical XML fragments as tentative clusterheads representing classes. Computes the comparison score of each incoming fragment with the current clusterheads and add it to the class of the closest match.

5 Conclusions

We have outlined a knowledge extraction technique aimed at extracting class hierarchies from flows of XML fragments. Preliminary results on sample data are encouraging; we are now in the process of implementing our approach and testing it on experimental data and refining the modelling of data items in order to fully preserve the topological and cardinality information associated with tags. This should allow us to produce hierarchies at a finer level of granularity. To this end we plan to model our data using *fuzzy bags* suitably extending the function definition.

6 Acknowledgments

The work reported in this paper has been partially supported by the Italian Ministry of Research FIRB program within the *KIWI* project. We acknowledge the project partners for useful discussion on this topic.

References

1. M. Rifqi B. Bouchon-Meunier and S. Bothorel. Towards general measures of comparison of objects. *ACM, Special issue dedicated to the memory of Professor Arnold Kaufmann*, 84(2), December 1996.
2. J. Davies, D. Fensel, and F. van Harmelen. *Towards the Semantic Web: Ontology-Driven Knowledge Management*. John Wiley & Sons, Ltd, 2002.
3. A. F. Fontana E. Damiani, L. Tanca. Fuzzy xml queries via context-based choice of aggregations. *KYBERNETIKA*, 36(6):635–655, 2000.
4. E. Damiani P. Bosc and M.G. Fugini. Fuzzy service selection in a distributed object-oriented environment. *IEEE Transactions on Fuzzy Systems*, 9(5), October 2001.
5. World Wide Web. *RDF Vocabulary Description Language 1.0: RDF Schema*, December 2003. http://www.w3.org/TR/rdf-schema/.
6. World Wide Web Consourtium. *OWL Web Ontology Language – Overview*, December 2003. http://www.w3.org/TR/owl-features/.

Generalizing Quantification in Fuzzy Description Logics

Daniel Sánchez[1] and Andrea G. B. Tettamanzi[2]

[1] University of Granada
Department of Computer Science and Artificial Intelligence
Periodista Daniel Saucedo Aranda s/n, 18071 Granada, Spain
daniel@decsai.ugr.es
[2] University of Milan
Department of Information Technologies
Via Bramante 65, I-26013 Crema (CR), Italy
andrea.tettamanzi@unimi.it

Abstract. In this paper we introduce $\mathcal{ALCQ}_F^+$, a fuzzy description logic with extended qualified quantification. The proposed language allows for the definition of fuzzy quantifiers of the absolute and relative kind by means of piecewise linear functions on $\mathbb{N}$ and $\mathbb{Q} \cap [0,1]$ respectively. These quantifiers extends the usual (qualified) $\exists$, $\forall$ and number restriction. The semantics of quantified expressions is defined by using method GD [4], that is based on recently developed measures of the cardinality of fuzzy sets.

1 Introduction

Description logics (DL) [1] are a family of logic-based knowledge-representation formalisms emerging from the classical AI tradition of semantic networks and frame-based systems. DLs are well-suited for the representation of and reasoning about terminological knowledge, configurations, ontologies, database schemata, etc.

The need of expressing and reasoning with imprecise knowledge and the difficulties arising in classifying individuals with respect to an existing terminology is motivating research on nonclassical DL semantics, suited to these purposes. To cope with this problem, fuzzy description logics have been proposed that allow for imprecise concept description by using fuzzy sets and fuzzy relations. However, these approaches have paid little attention to the quantification issue (only the semantics of $\exists$ and $\forall$ have been extended to the fuzzy case [8]).

This is an important lack by several reasons. On the one hand, number restriction is a kind of quantification that arises very frequently in concept description, so it is necessary to extend it to the fuzzy case. But another

D. Sánchez and A.G.B. Tettamanzi: *Generalizing Quantification in Fuzzy Description Logics*, Advances in Soft Computing **2**, 397–411 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

important reason is that not only concepts, but also quantifiers are imprecise in many cases (e.g. "around two", "most").

For example, suppose you are the marketing director of a supermarket chain. You are about to launch a new line of low-calorie products. In order to set up your budget, you need to project the sales of this new line of products. This can be done either by means of an expensive market research, or by means of some kind of inference based on your knowledge of customer habits. For instance, you could expect prospective buyers of this new line of products to be essentially faithful customers who mostly buy foods with low energy value. We have here all the ingredients of imprecise knowledge: a "faithful customer" is a fuzzy concept; "low" energy value is a linguistic value, which might be modeled as a fuzzy number; to "mostly" buy a given kind of product is equivalent to a quantified statement of the form "most of the bought products are of this kind", where "most" is an imprecise quantifier.

Zadeh [10] showed that imprecise quantifiers can be defined by using fuzzy sets, and by incorporating them into the language and providing the tools to define their semantics we can provide a very powerful knowledge representation tool, with greater expressive power, and closer to the humans' way of thinking. This is the objective of our work.

The paper is organized as follows: section 2 introduces briefly existing developments on fuzzy description logics. Section 3 is devoted to fuzzy quantifiers and the strongly linked issue of cardinality of fuzzy sets. Our proposal of fuzzy description logic with extended fuzzy quantification is described in section 4. Finally, section 5 contains our conclusions.

2 Fuzzy Description Logics

The idea of fuzzifying description logics to deal with imprecision is not new. Recently, a quite general fuzzy extension of description logics has been proposed, with complete algorithms for solving the entailment problem, the subsumption problem, as well as the best truth-value bound problem [8].

2.1 Fuzzy $\mathcal{ALC}$

The $\mathcal{ALC}$ description language is a basic yet significant representative of DLs. The syntax of the $\mathcal{ALC}$ language is very simple: a *concept* is built out of primitive (or *atomic*) concepts according to the grammar

$$
\begin{aligned}
\langle \text{concept\_description} \rangle ::= {} & \langle \text{atomic\_concept} \rangle \mid \\
& \top \mid \bot \mid \neg \langle \text{concept\_description} \rangle \mid \\
& \langle \text{concept\_description} \rangle \sqcap \langle \text{concept\_description} \rangle \mid \\
& \langle \text{concept\_description} \rangle \sqcup \langle \text{concept\_description} \rangle \mid \\
& \langle \text{quantification} \rangle \\
\langle \text{quantification} \rangle ::= {} & \langle \text{value\_restriction} \rangle \mid
\end{aligned}
$$

⟨existential_quantification⟩
⟨value_restriction⟩ ::= ∀ ⟨role⟩.⟨concept_description⟩
⟨existential_quantification⟩ ::= ∃ ⟨role⟩.⟨concept_description⟩

When necessary to avoid ambiguities, parentheses should be used. From a logical point of view, concepts can be seen as unary predicates, whereas roles can be interpreted as binary predicates linking individuals to their attributes.

A usual extension of the $\mathcal{ALC}$ language, called $\mathcal{ALCN}$, is obtained by allowing number restrictions of the form $\leq nR$ and $\geq nR$. It is highly relevant to this work since number restriction is a form of quantification. The syntax of $\mathcal{ALCN}$ can be described by the following additional production rules:

⟨quantification⟩ ::= ⟨value_restriction⟩ |
⟨existential_quantification⟩
⟨number_restriction⟩
⟨number_restriction⟩ ::= ⟨comparison_operator⟩⟨natural_number⟩⟨role⟩
⟨comparison_operator⟩ ::= ≤ | = | ≥

Fuzzy $\mathcal{ALC}$ [8] retains the same syntax as its crisp counterpart, only semantics changes. Semantics for fuzzy $\mathcal{ALCN}$ are part of the contribution of this paper and will be discussed in next sections.

2.2 Fuzzy Interpretation

A fuzzy interpretation $\mathcal{I}$ consists of a non-empty domain $U^{\mathcal{I}}$ (the universe of discourse), and an assignment $\cdot^{\mathcal{I}}$, which maps every atomic concept A onto a fuzzy subset $A^{\mathcal{I}}$ of $U^{\mathcal{I}}$, every atomic role R onto a fuzzy binary relation $R^{\mathcal{I}} \subseteq U^{\mathcal{I}} \times U^{\mathcal{I}}$, and every individual name a onto an element $a^{\mathcal{I}} \in U^{\mathcal{I}}$. The special atomic concepts $\top$ and $\bot$ are mapped respectively onto $U^{\mathcal{I}}$ (the function that maps every individual onto 1) and the empty set (the function that maps every individual onto 0).

The semantics of the intersection, disjunction, and negation of concepts is defined as follows: for all $a \in U^{\mathcal{I}}$,

$$(C \sqcap D)^{\mathcal{I}}(a) = \min\{C^{\mathcal{I}}(a), D^{\mathcal{I}}(a)\}; \tag{1}$$

$$(C \sqcup D)^{\mathcal{I}}(a) = \max\{C^{\mathcal{I}}(a), D^{\mathcal{I}}(a)\}; \tag{2}$$

$$(\neg C)^{\mathcal{I}}(a) = 1 - C^{\mathcal{I}}(a). \tag{3}$$

For the existential quantification, there is only one possible semantics that can be given in terms of fuzzy set theory, namely

$$(\exists R.C)^{\mathcal{I}}(a) = \sup_{b \in U^{\mathcal{I}}} \min\{R^{\mathcal{I}}(a,b), C^{\mathcal{I}}(b)\}. \tag{4}$$

The value restriction construct $\forall R.C$ of FDLs is interpreted in [9, 8] by translating the implication of the crisp interpretation into the *classical* fuzzy implication which directly maps the $P \supset Q \equiv \neg P \vee Q$ logical axiom:

$$(\forall R.C)^{\mathcal{I}}(a) = \inf_{b \in U^{\mathcal{I}}} \max\{1 - R^{\mathcal{I}}(a,b), C^{\mathcal{I}}(b)\}. \tag{5}$$

2.3 Fuzzy Assertions

An assertion can be either of the form $C(a) \leq \alpha$ (respectively $R(a,b) \leq \alpha$), or $C(a) \geq \alpha$ (resp. $R(a,b) \geq \alpha$), where C is a concept, R is a role, a and b are individual constants and $\alpha \in \mathbb{Q} \cap [0,1]$ is a truth degree. The two kinds of assertions are true in $\mathcal{I}$ if $C^{\mathcal{I}}(a^{\mathcal{I}}) \leq \alpha$ (resp. $C^{\mathcal{I}}(a^{\mathcal{I}}) \geq \alpha$), and false otherwise.

2.4 Fuzzy Terminological Axioms

Axioms and queries can be of two kinds: specializations and definitions.

A fuzzy concept specialization is a statement of the form $C \sqsubseteq D$, where C and D are concepts. A fuzzy interpretation $\mathcal{I}$ satisfies a fuzzy concept specialization $C \sqsubseteq D$ if, for all $a \in U^{\mathcal{I}}$, $C^{\mathcal{I}}(a) \leq D^{\mathcal{I}}(a)$.

A fuzzy concept definition is a statement of the form $C \equiv D$, which can be understood as an abbreviation of the pair of assertions $\{C \sqsubseteq D, D \sqsubseteq C\}$.

If $C \sqsubseteq D$ is valid (true in every interpretation), then we say that D *subsumes* C.

2.5 Fuzzy Knowledge Bases

A fuzzy knowledge base comprises two components, just like its crisp counterpart: a TBox and an ABox. While the TBox of a fuzzy knowledge base has formally nothing fuzzy with it, for the syntax of the terminological part of the fuzzy $\mathcal{ALC}$ language as defined in [8] is identical to the crisp $\mathcal{ALC}$ language, the ABox can contain fuzzy assertions.

For example, the knowledge base describing the business of running a supermarket chain could contain the following terminological axioms:

$$\begin{aligned} \mathsf{FaithfulCustomer} &\sqsubseteq \mathsf{Customer} \sqsubseteq \top \\ \mathsf{FoodProduct} &\sqsubseteq \mathsf{Product} \sqsubseteq \top \\ \mathsf{LowCalorie} &\sqsubseteq \mathsf{EnergyMeasure} \sqsubseteq \top \\ \mathsf{LowCalorieFood} &\equiv \mathsf{FoodProduct} \sqcap \forall \mathsf{energyValue.LowCalorie} \end{aligned}$$

The ABox describing facts about your supermarket chain might contain fuzzy assertions which we might summarize as follows:

- given an individual customer c and a product p, $\mathsf{buys}(c,p)$ might be defined as
$$\mathsf{buys}(c,p) = f(\text{weeklyrevenue}(c,p)),$$
where $f : \mathbb{R} \to [0,1]$ is nondecreasing, and $\text{weeklyrevenue}(c,p) : \mathsf{Customer}^{\mathcal{I}} \times \mathsf{Product}^{\mathcal{I}} \to \mathbb{R}$ returns the result of a database query which calculates the average revenue generated by product p on customer c in all the stores operated by the chain;

- given an individual customer c, $\mathsf{FaithfulCustomer}(c)$ might be defined as

$$\mathsf{FaithfulCustomer}(c) = g(\text{weeklyrevenue}(c)),$$

  where $g : \mathbb{R} \to [0,1]$ is nondecreasing, and $\text{weeklyrevenue}(c) : \mathsf{Customer}^{\mathcal{I}} \to \mathbb{R}$ returns the result of a database query which calculates the average revenue generated by customer c in all the stores operated by the chain;
- finally, $\mathsf{LowCalorie}(x)$, where x is an average energy value per 100 g of product measured in kJ, could be defined as

$$\mathsf{LowCalorie}(x) = \begin{cases} 1 & x < 1000, \\ \frac{2000-x}{1000} & 1000 \leq x \leq 2000, \\ 0 & x > 2000. \end{cases}$$

By using this knowledge base, you would be able, for example, to deduce the degree to which a given food product would be a low-calorie food, and other useful knowledge implied in the TBox and ABox.

However, it would be impossible to even express the notion of a "faithful customer who mostly buys low-calorie food", let alone using that concept in deductions!

3 Cardinality and Fuzzy Quantification

Crisp quantification is strongly linked to crisp cardinality since a crisp quantifier Q represents a crisp subset of absolute (values in $\mathbb{N}$) or relative (values in $\mathbb{Q} \cap [0,1]$) cardinalities, we call $S(Q)$. For example, $\exists$ represents the set of absolute cardinalities $\mathbb{N}\backslash\{0\}$ (equivalently the set of relative cardinalities $S(\exists) = \mathbb{Q} \cap (0,1]$)[3]. In the same way, $S(\forall) = \{1\} \subseteq (\mathbb{Q} \cap (0,1])$ and $S(\geq n) = \mathbb{N}\backslash\{0,\ldots,n-1\}$.

Hence, cardinality plays a crucial role in the assessment of crisp quantified statements. For example, let $D \equiv QR.C$ be a concept definition and let $\mathcal{I}$ be a crisp interpretation. Let $R_a^{\mathcal{I}}$ be the projection of relation $R^{\mathcal{I}}$ on individual a: for all $b \in U^{\mathcal{I}}$,

$$b \in R_a^{\mathcal{I}} \Leftrightarrow (a,b) \in R^{\mathcal{I}}.$$

Then $D(a)$ is true iff $\left|C^{\mathcal{I}} \cap R_a^{\mathcal{I}}\right| / \left|R_a^{\mathcal{I}}\right| \in S(Q)$ (when Q is relative), or $\left|C^{\mathcal{I}} \cap R_a^{\mathcal{I}}\right| \in S(Q)$ (when Q is absolute).

In summary, the evaluation of the truth degree of a quantified sentence consists in calculating the compatibility between cardinality and quantifier. Hence, in order to extend quantification to the fuzzy case, we must discuss first about cardinality of fuzzy sets.

[3]This is the only quantifier that can be represented both ways in the general case, i.e., when the cardinality of the referential is not known.

3.1 Fuzzy cardinality

We consider two different kinds of cardinalities: absolute and relative. "Absolute cardinality", or simply "cardinality", measures the amount of elements in a set, while "relative cardinality" measures the percentage of objects of one set that are in another set.

Absolute cardinality

The most widely used definition of fuzzy set cardinality introduced in [7] is the following: given a fuzzy set F,

$$|F| = \sum_{x \in U} F(x), \tag{6}$$

which is, in general, a real number, and not an integer as it is the case with classical set cardinality.

However, this definition lends itself to many objections and leads to paradoxes. On the one hand, it is well known that the aggregation of many small values can yield a value of cardinality that does not correspond to the amount of elements in the set. One usual solution is to add only values over a certain threshold, but this is not satisfactory.

Even with high membership degrees some unintuitive results can arise. Though this is true in general for fuzzy sets, let us illustrate it with an example in terms of FDLs: consider the fuzzy concept Blonde with the interpretation $1/\mathsf{MIKE} + 0.5/\mathsf{JOHN} + 0.5/\mathsf{TONY}$. According to Equation 6, there would be exactly two instances of Blonde in this set. But, who are they? Obviously, MIKE is Blonde (i.e., $\mathsf{Blonde}(\mathsf{MIKE}) = 1$), so the other one should be JOHN or TONY. But if we consider JOHN is Blonde, then we must accept TONY also is, because $\mathsf{Blonde}(\mathsf{JOHN}) = \mathsf{Blonde}(\mathsf{TONY})$. In other words, the cardinality of this set could never be two.

These problems have motivated research over the last twenty years, where several alternative definitions have been proposed. There is a wide agreement that the absolute cardinality of a fuzzy set should be a fuzzy subset of $\mathbb{N}$. In particular, this approach is employed in the definition of the fuzzy cardinality measure ED introduced in [3] as follows:

Definition 1 ([3]). *The fuzzy cardinality of a set G is the set $ED(G)$ defined for each $0 \leq k \leq |\mathrm{supp}(G)|$ as*

$$ED(G)(k) = \begin{cases} \alpha_i - \alpha_{i+1} & \alpha_i \in \Lambda(G) \text{ and } |G_{\alpha_i}| = k \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

with $\Lambda(G) = \{\alpha_1, \ldots, \alpha_p\} \cup \{1\}$ the level set of G, and $\alpha_i > \alpha_{i+1}$ for every $i \in \{1, \ldots, p\}$, and $\alpha_{p+1} = 0$.

It is easy to see that $ED(\mathsf{Blonde}^{\mathcal{I}})(2) = 0$ since no α-cut has cardinality 2.

Relative cardinality

In the crisp case it is easy to obtain relative cardinalities from absolute ones, i.e., the relative cardinality of set G with respect to set F (i.e. the percentage of elements of G that are in F) is

$$\text{RelCard}(G/F) = \frac{|G \cap F|}{|F|} \tag{8}$$

However, performing this quotient between fuzzy cardinalities is not easy and can even lead to misleading results, so it has been historically preferred to calculate it directly from the definitions of G and F.

The scalar approach based on Equation 6 poses the same problems commented before. To cope with this, a fuzzy cardinality measure called ER, that extends ED, was also introduced in [3] as follows:

Definition 2 ([3]). *The fuzzy relative cardinality of a set G with respect to a set F is the set $ER(G/F)$ defined for each $0 \leq q \leq 1$ as*

$$ER(G/F)(q) = \sum_{\alpha_i \mid C(G/F,\alpha_i)=q} (\alpha_i - \alpha_{i+1}) \tag{9}$$

with $\Lambda(F) \cup \Lambda(G \cap F) = \{\alpha_1, \ldots, \alpha_p\}$ and $\alpha_i > \alpha_{i+1}$ for every $i \in \{1, \ldots, p\}$, and $\alpha_0 = 1$, $\alpha_{p+1} = 0$.

ER is an extension of ED since, if F is crisp and $|F| = n$, then $ER(G/F)(k/n) = ED(G)(k)$ [3].

3.2 Fuzzy quantifiers

The concept of fuzzy linguistic quantifier is due to L. A. Zadeh [10]. Fuzzy quantifiers are linguistic labels representing imprecise quantities or percentages. It is usual to distinguish two basic types of fuzzy quantifiers:

- *Absolute quantifiers* express vague quantities (e.g., "Around 2") or quantity intervals (i.e., "Approximately between 1 and 3"). They are represented as fuzzy subsets of $\mathbb{N}$. For example, we could define

$$\text{"Around 2"} = 0.5/1 + 1/2 + 0.5/3$$
$$\text{"Approx. 1–3"} = 0.5/0 + 1/1 + 1/2 + 1/3 + 0.5/4$$

- *Relative quantifiers* express fuzzy percentages and they are represented by fuzzy subsets of the real unit interval, although in practice only rational values make sense. In this category belong the standard predicate-logic quantifiers $\exists$ and $\forall$, that can be defined as

$$\exists(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 1 \end{cases} \quad \forall(x) = \begin{cases} 0 & x < 1 \\ 1 & x = 1 \end{cases}$$

Some other examples are

$$(\text{Approx. Half})(x) = \begin{cases} 2x & x \leq 0.5 \\ 2(1-x) & x \geq 0.5 \end{cases}$$

$$(\text{Approx. Half or more})(x) = \begin{cases} 2x & x \leq 0.5 \\ 1 & x \geq 0.5 \end{cases}$$

3.3 Evaluation of quantified sentences

Quantified sentences are natural language sentences involving fuzzy linguistic quantifiers, and therefore they express claims about the (fuzzy) quantity or percentage of elements of a (possibly fuzzy) set that verify a certain imprecise property.

Following Zadeh [10] there are two main types of quantified sentences, whose general structure is the following:

Type I sentences: Q of X are G
Type II sentences: Q of F are G

where Q is a linguistic quantifier, X is a crisp finite set, and F and G are two fuzzy subsets of X that represent imprecise properties. Examples are

Type I: *Around 30 students are young*
Type II: *Most of the efficient students are young*

Type I sentences are suitable for both absolute and relative quantifiers, whilst type II sentences only make sense for relative quantifiers (i.e. a sentence like "Around 2 F are G" is in fact a type I sentence "Around 2 X are $F \cap G$").

The evaluation of a quantified sentence is the process of calculating its fuzzy accomplishment degree. There are several methods available in the literature (some methods are discussed in [4], recent developments are [6, 5]).

In order to extend fuzzy DLs with absolute and relative quantifiers, we shall employ method GD [4]. This method calculates the accomplishment degree of a quantified sentence "Q of F are G" as the compatibility degree between the fuzzy relative cardinality measure $ER(G/F)$ [3] and Q (if Q is relative) or between its absolute counterpart $ED(G \cap F)$ [3] and Q (if Q is absolute). A convenient formulation of GD is the following [4]:

Definition 3. *The method GD obtains the evaluation of a quantified sentence "Q of F are G" as*

$$GD_Q(G/F) = \sum_{\alpha_i \in \Lambda(G/F)} (\alpha_i - \alpha_{i+1})\, Q\left(\frac{|(G \cap F)_{\alpha_i}|}{|F_{\alpha_i}|}\right) \tag{10}$$

for relative quantifiers, and

$$GD_Q(G/F) = \sum_{\alpha_i \in \Lambda(G/F)} (\alpha_i - \alpha_{i+1})\, Q\left(|(F \cap G)_{\alpha_i}|\right) \tag{11}$$

for absolute ones, where $F \cap G$ is computed using the minimum, and $\Lambda(G/F) = \Lambda(G \cap F) \cup \Lambda(F)$. We label these values as $\Lambda(G/F) = \{\alpha_1, \ldots, \alpha_p\}$ with $\alpha_i > \alpha_{i+1}$ for every $i \in \{1, \ldots, p\}$ and $\alpha_{p+1} = 0$.

As discussed in [4], $GD_Q(G/F)$ is undefined for type II sentences when F is not normalized since $F_\alpha = \emptyset$ for at least one $\alpha \in [0, 1]$ (in particular for $\alpha = 1$). Possible solutions to this problem are to normalize F (applying the same factor to $F \cap G$ after) or to define the value of any relative quantifier when the relative cardinality is undefined (we note this value as $Q(u)$, by considering $|\emptyset|/|\emptyset| = u$). In this work we shall employ this last option. In particular, $\exists(u) = 0$ and $\forall(u) = 1$.

4 Fuzzy DLs with Fuzzy Quantification

Quantified expressions in description logics are expressions like $QR.C$, where Q is a quantifier, R is a role and C is a concept, called sometimes a *qualifier*. These expressions are useful for defining new concepts, like $D \equiv QR.C$, meaning that an individual a is an instance of concept D if Q of the individuals which fill its role R are instances of C.

In crisp description logics, quantification is limited to the classical (relative) quantifiers $\exists$ and $\forall$, as well as to the so-called number restriction, $\leq n$ and $\geq n$ (crisp absolute quantifiers). In this section we propose a fuzzy description logic able to represent and reason with general absolute and relative quantifiers. The goal is to be able to express such fuzzy definitions as, e.g., $D \equiv QR.C$, where Q is now a fuzzy quantifier, meaning that an individual a is an instance of concept D *to the extent that* Q of the individuals which fill its role R are instances of C.

4.1 The $\mathcal{ALCQ}_F^+$ Language

The extended language we introduce, for which we propose the name $\mathcal{ALCQ}_F^+$, in keeping with DL naming conventions[4], has the following syntax:

$$
\begin{aligned}
\langle \text{concept\_description} \rangle ::= {} & \langle \text{atomic\_concept} \rangle \mid \\
& \top \mid \bot \mid \neg \langle \text{concept\_description} \rangle \mid \\
& \langle \text{concept\_description} \rangle \sqcap \langle \text{concept\_description} \rangle \mid \\
& \langle \text{concept\_description} \rangle \sqcup \langle \text{concept\_description} \rangle \mid \\
& \langle \text{quantification} \rangle \\
\langle \text{quantification} \rangle ::= {} & \langle \text{quantifier} \rangle \langle \text{atomic\_role} \rangle . \langle \text{concept\_description} \rangle
\end{aligned}
$$

[4]The superscript plus is to suggest that, in addition to **q**ualified number restrictions available in the description logic $\mathcal{ALCQ}$ introduced by De Giacomo and Lenzerini [2], we provide also more general fuzzy linguistic **q**uantifiers. The subscript F means that the language deals with infinitely many truth-values, as in the language $\mathcal{ALC}_{F_M}$ of Tresp and Molitor [9].

⟨quantifier⟩ ::= "(" ⟨absolute_quantifier⟩ ")" | "(" ⟨relative_quantifier⟩ ")" | ∃ | ∀
⟨absolute_quantifier⟩ ::= ⟨abs_point⟩ | ⟨abs_point⟩ + ⟨absolute_quantifier⟩
⟨relative_quantifier⟩ ::= ⟨fuzzy_degree⟩/u | ⟨fuzzy_degree⟩/u + ⟨piecewise_fn⟩
⟨piecewise_fn⟩ ::= ⟨rel_point⟩ | ⟨rel_point⟩ + ⟨piecewise_fn⟩
⟨abs_point⟩ ::= ⟨val⟩/⟨natural_number⟩
⟨rel_point⟩ ::= ⟨val⟩/⟨[0,1]-value⟩
⟨val⟩ ::= [⟨fuzzy_degree⟩ ◁]⟨fuzzy_degree⟩[▷ ⟨fuzzy_degree⟩]

In this extension the semantics of quantifiers is defined by means of piecewise-linear membership functions. In the case of absolute quantifiers, the quantifier is obtained by restricting the membership function to the naturals.

The piecewise-linear functions are defined by means of a sequence of points. These points are expressed as $\alpha \triangleleft \beta \triangleright \gamma/x$, where x is the cardinality value, β is the membership degree of x, and α and γ are the limit when the membership function tends to x at the left and at the right, respectively. When the function is continuous, this can be summarized as β/x (since $\alpha = \beta = \gamma$), whereas discontinuities on the left ($\alpha \neq \beta = \gamma$) or right ($\alpha = \beta \neq \gamma$) can be summarized as $\alpha \triangleleft \beta/x$ and $\beta \triangleright \gamma/x$, respectively.

Obviously, for a membership function definition of the form $val_1/x_1 + val_2/x_2 + \cdots + val_p/x_p$ it is required $x_i \neq x_j \ \forall i < j$.

For relative quantifiers, as pointed out in the previous section, we should take into account the definition of a membership degree for the case "undefined" that arises when the referential set with respect to which the relative cardinality is calculated is empty[5]. This value could depend on the subjective view of the user or the application at hand, though it is well known and fixed for some quantifiers like ∃ and ∀, as we have seen.

Unless a different definition is provided explicitly, we shall assume that

- the points 0/0 and 1/1 are part of the definition of any relative quantifier, and
- the point 0/0 is part of the definition of any absolute quantifier. Also, let x_l be the greatest natural value in the definition of an absolute quantifier and let $\alpha_l \triangleleft \beta_l \triangleright \gamma_l$ be the values for x_l. Then, for any $x > x_l$ we assume γ_l/x.

The following is a set of absolute quantifiers and their corresponding (subjective) expressions using the proposed notation:

[5]Furthermore, the presence of the definition of a membership degree for the case "undefined" univocally identifies a quantifier as relative, even in those rare cases in which a doubt might arise.

(Around 2)	$(1/2 + 0/4)$
(Approx. between 1 and 3)	$(0.5/0 + 1/1 + 1/3 + 0/5)$
(Exactly 3)	$(0 \triangleleft 1 \triangleright 0/3)$
> 5	$(0 \triangleright 1/5)$
< 8	$(1/0 + 1 \triangleleft 0/8)$
(Around 7)	$(0/5 + 1/7 + 0/9)$

Some examples of relative quantifiers are the following, where $Q_M(x) = x$ is a quantifier sometimes called "Most"[6], and α_i represent user-defined values for the case "undefined":

$\forall$	$(1/u + 0 \triangleleft 1/1)$
$\exists$	$(0/u + 0 \triangleright 1/0)$
(Approx. half)	$(\alpha_1/u + 1/0.5 + 0/1)$
(Approx. half or more)	$(\alpha_2/u + 1/0.5)$
Q_M	(α_3/u) or $(\alpha_3/u + 0/0 + 1/1)$
(Around 75%)	$(\alpha_4/u + 0/0.25 + 1/0.75 + 0.5/1)$

The definition of quantifiers we have introduced generalizes the classical quantifiers $\exists$ and $\forall$ (particular cases of relative quantifiers), as we have just seen, so the symbols $\exists$ and $\forall$ are included in the language only by historical reasons and to preserve backward compatibility. In addition, the language employed to define quantifiers generalizes number restriction (particular cases of absolute quantifiers), since $\leq n$ translates to $(1/0 + 1 \triangleright 0/n)$ and $\geq n$ translates to $(0 \triangleleft 1/n)$.

Finally, in order to name quantifiers we shall employ the same notation used to name concepts, for example

$$(\text{Around 2}) \equiv (1/2 + 0/4).$$

As for the classical quantifiers $\exists$ and $\forall$, by including them in the $\mathcal{ALCQ}_F^+$ language we are assuming implicitly the definitions:

$$\forall \equiv (1/u + 0 \triangleleft 1/1);$$
$$\exists \equiv (0/u + 0 \triangleright 1/0).$$

When a quantifier is defined which is denoted by a single mathematical symbol (possibly followed by a single number), the parentheses around the quantifier name might be dropped without risk of confusion. For example,

$$\simeq 2 \equiv (1/2 + 0/4);$$
$$\widetilde{<}2 \equiv (1/0 + 1/2 + 0/4).$$

[6] Q_M could be called "Cardinal", because the membership degree is exactly the cardinality in the crisp case.

4.2 Semantics

Given a fuzzy interpretation $\mathcal{I}$, the semantics of the intersection, disjunction, and negation of concepts in our language keep being those introduced in section 2. For the general quantification, we can translate a general expression like $D \equiv (QR.C)$ into the usual notation of quantified sentences introduced in the previous chapter as follows: if Q is an absolute quantifier, the degree to which an individual a verifies concept D is the result of the evaluation of the quantified sentence

$$Q \text{ of } U^{\mathcal{I}} \text{ are } R_a^{\mathcal{I}} \cap C^{\mathcal{I}},$$

whilst for relative quantifiers, it is the result of the evaluation of the sentence

$$Q \text{ of } R_a^{\mathcal{I}} \text{ are } C^{\mathcal{I}},$$

where $R_a^{\mathcal{I}}$ is the projection of fuzzy relation $R^{\mathcal{I}}$ on individual a: for all $b \in U^{\mathcal{I}}$,

$$R_a^{\mathcal{I}}(b) = R^{\mathcal{I}}(a, b).$$

Hence, we consider two cases depending on whether the quantifier is absolute (12) or relative (13):

$$(Q_{\text{abs}}R.C)^{\mathcal{I}}(a) = GD_{Q_{\text{abs}}}((R_a^{\mathcal{I}} \cap C^{\mathcal{I}})/U^{\mathcal{I}}) \tag{12}$$

$$(Q_{\text{rel}}R.C)^{\mathcal{I}}(a) = GD_{Q_{\text{rel}}}(C^{\mathcal{I}}/R_a^{\mathcal{I}}) \tag{13}$$

where GD is the evaluation method introduced in definition 3. In particular the semantics for the quantifiers $\exists$ and $\forall$ is:

$$(\exists R.C)^{\mathcal{I}}(a) = \sum_{(R_a^{\mathcal{I}})_{\alpha_i} \cap (C^{\mathcal{I}})_{\alpha_i} \neq \emptyset} (\alpha_i - \alpha_{i+1}) \tag{14}$$

$$(\forall R.C)^{\mathcal{I}}(a) = \sum_{(R_a^{\mathcal{I}})_{\alpha_i} \subseteq (C^{\mathcal{I}})_{\alpha_i}} (\alpha_i - \alpha_{i+1}) \tag{15}$$

with $\alpha_i \in \Lambda(G/F)$. Following the properties of GD [4], the semantics of the existential quantification keep being as usual, i.e., Equation 14 is equivalent to

$$(\exists R.C)^{\mathcal{I}}(a) = \sup_{b \in U^{\mathcal{I}}} \min\{R_a^{\mathcal{I}}(b), C^{\mathcal{I}}(b)\}.$$

As a particular case, when the referential F is crisp, then GD verifies De Morgan's laws, i.e.,

$$1 - GD_{\forall}(G/F) = GD_{\exists}(\neg G/F)$$
$$1 - GD_{\exists}(G/F) = GD_{\forall}(\neg G/F)$$

Hence, if $R_a^{\mathcal{I}}$ is crisp then Equation 15 is equivalent to

$$(\forall R.C)^{\mathcal{I}}(a) = \inf_{b \in U^{\mathcal{I}}} \max\{1 - R_a^{\mathcal{I}}(b), C^{\mathcal{I}}(b)\}.$$

However, this is not true in general. In fact, in order to verify De Morgan's laws it is necessary that

$$R_a^{\mathcal{I}} \subseteq C^{\mathcal{I}} \Leftrightarrow R_a^{\mathcal{I}} \cap (\neg C)^{\mathcal{I}} = \emptyset$$

but this equivalence holds only when $R_a^{\mathcal{I}}$ (or $C^{\mathcal{I}}$) is crisp. Otherwise we could have a situation where both sets are fuzzy and $R_a^{\mathcal{I}} \subseteq C^{\mathcal{I}}$ (and hence the evaluation using $\forall$ is expected to yield 1), but $R_a^{\mathcal{I}} \cap (\neg C)^{\mathcal{I}} \neq \emptyset$ (and hence the evaluation using $\exists$ is expected to yield a value greater than 0). For example, suppose

$$R_a^{\mathcal{I}} = 1/b_1 + 0.6/b_2 + 0.4/b_3$$
$$C^{\mathcal{I}} = 1/b_1 + 0.9/b_2 + 0.5/b_3$$

then $R_a^{\mathcal{I}} \subseteq C^{\mathcal{I}}$ and $R_a^{\mathcal{I}} \cap (\neg C)^{\mathcal{I}} = 0.1/b_2 + 0.4/b_3 \neq \emptyset$, consequently $GD_{\forall}(C^{\mathcal{I}}/R_a^{\mathcal{I}}) = 1$ and $GD_{\exists}((\neg C)^{\mathcal{I}}/R_a^{\mathcal{I}}) = 0.4 \neq 0$.

4.3 An Example

Let us go back to our low-calorie product line example. By using the $\mathcal{ALCQ}_F^+$ language, it is now possible to express the notion of a faithful customer who mostly buys food with low energy value as

$$C \equiv \mathsf{FaithfulCustomer} \sqcap (\mathsf{Most})\mathsf{buys.LowCalorieFood},$$

where $(\mathsf{Most}) \equiv (0/u + 0/0.5 + 1/0.75)$.

A useful deduction this new axiom allows you to make is, for instance, calculating the extent to which a given individual customer or, more precisely, a fidelity card, say CARD0400009324198, is a C. For instance, you could know that

$$\mathsf{FaithfulCustomer}(\mathsf{CARD0400009324198}) = 0.8,$$

and, by querying the sales database, you might get all the degrees to which that customer buys each product. For sake of example, we give a small subset of those degrees of truth in Table 1, along with the energy values of the relevant products.

According to the semantics of $\mathcal{ALCQ}_F^+$,

$$C(\mathsf{CARD0400009324198}) \approx 0.742$$

i.e., the degree to which most of the items purchased by this customer are low-calorie is around 0.742. This seems to be in accordance with the data in table 1, where we can see that four products (those products p in rows 2, 4, 5, and 6) verify

$$\mathsf{buys}(\mathsf{CARD0400009324198}, p) \leq \mathsf{LowCalorieFood}(p)$$

Product	Energy [kJ/hg]	LowCalorieFood(·)	buys(CARD . . . , ·)
GTIN8001350010239	1680	0.320	0.510
GTIN8007290330987	1475	0.525	0.050
GTIN8076809518581	1975	0.025	0.572
GTIN8000113004003	1523	0.477	0.210
GTIN8002330006969	498	1.000	1.000
GTIN8005410002110	199	1.000	1.000
GTIN017600081636	1967	0.033	0.184

Table 1. The energy value, membership in the LowCalorieFood, and the degree to which customer CARD0400009324198 buys them for a small sample of products.

while for the products in rows 1 and 7 the difference between being purchased and being low-calorie food is not so high. Only the item in row 3 seems to be a clear case of item purchased but not low-calorie.

As another justification of why this result appears in agreement with the data, in Table 2 we show the percentage of purchased items that are low-calorie at α-cuts of the same level. At any other level, the percentage obtained is one of those shown in Table 2.

Level	Percentage
1.000	$1.000 = 2/2$
0.572	$0.667 = 2/3$
0.510	$0.500 = 2/4$
0.320	$0.750 = 3/4$
0.210	$0.800 = 4/5$
0.184	$0.667 = 4/6$
0.050	$0.714 = 5/7$
0.033	$0.857 = 6/7$

Table 2. Percentage of purchased items that are low-calorie at significant levels.

At many levels the percentage is above 0.75, therefore fitting the concept of Most as we have defined it. At level 0.050 the percentage is almost 0.75. The only level that clearly doesn't fit Most is 0.510, but at the next level (0.320) we have again 0.75 and $\mathsf{Most}(0.75) = 1$.

5 Conclusions

$\mathcal{ALCQ}_F^+$ allows for concept descriptions involving fuzzy linguistic quantifiers of the absolute and relative kind, and using qualifiers. It provides also semantics for crisp quantifiers like $\forall$, $\exists$, and number restriction in those cases where the roles and/or qualifiers employed are fuzzy.

Acknowledgments

We are much indebted to Piero Andrea Bonatti, of the Federico II University, Naples, for reading an early draft of this paper. His comments and observations helped us to make significant improvements.

References

1. Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider, editors. *The Description Logic Handbook: Theory, implementation and applications*. Cambridge, 2003.
2. Giuseppe De Giacomo and Maurizio Lenzerini. A uniform framework for concept definitions in description logics. *Journal of Artificial Intelligence Research*, 6:87–110, 1997.
3. M. Delgado, M.J. Martín-Bautista, D. Sánchez, and M.A. Vila. A probabilistic definition of a nonconvex fuzzy cardinality. *Fuzzy Sets and Systems*, 126(2):41–54, 2002.
4. M. Delgado, D. Sánchez, and M.A. Vila. Fuzzy cardinality based evaluation of quantified sentences. *International Journal of Approximate Reasoning*, 23:23–66, 2000.
5. F. Díaz-Hermida, A. Bugarín, P. Cariñena, and S. Barro. Voting model based evaluation of fuzzy quantified sentences: A general framework. *Fuzzy Sets and Systems*. To appear.
6. I. Glöckner. Fundamentals of fuzzy quantification: Plausible models, constructive principles, and efficient implementation. Technical Report TR2002-07, Technical Faculty, University Bielefeld, 33501 Bielefeld, Germany, 2002.
7. A. De Luca and S. Termini. A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory. *Information and Control*, 20:301–312, 1972.
8. U. Straccia. Reasoning within fuzzy description logics. *Journal of Artificial Intelligence Research*, 14:137–166, 2001.
9. Christopher B. Tresp and Ralf Molitor. A description logic for vague knowledge. In *Proceedings of the 13th biennial European Conference on Artificial Intelligence (ECAI'98)*, pages 361–365, Brighton, UK, 1998. J. Wiley and Sons.
10. L. A. Zadeh. A computational approach to fuzzy quantifiers in natural languages. *Computing and Mathematics with Applications*, 9(1):149–184, 1983.

Fuzzy Types: A First Step Towards Lazy Types in the .NET Framework

Fernando Berzal, Juan Carlos Cubero, Nicolás Marín, and M. Amparo Vila

IDBIS - Intelligent DataBases and Information Systems Research Group
Department of Computer Science and Artificial Intelligence
University of Granada
Granada - 18071 - Spain
{fberzal,JC.Cubero,nicm,vila}@decsai.ugr.es

Abstract. Many applications deal with data which are not easy to accommodate in conventional data models. This paper shows how the advanced features of modern programming platforms allow the transparent implementation of lazy typing capabilities which can then be easily used by programmers. As a first approximation to the problem, we show how this capabilities can be used to develop applications with fuzzy types.

Keywords: fuzzy types, lazy types, object-orientation.

1 Introduction

The arrival of the object-oriented data model broadened new horizons for data modelers. Object-oriented programming platforms make the development of complex applications easier, while object-relational and object-oriented database systems allow for the persistent representation of sophisticated data structures. This kind of systems can be used to represent data as complex as needed, but the structure of data must be unvarying and it must be built upon predefined basic data types.

Nevertheless, many applications in our current Information Society manage data which does not easily fit into current data models. Under some circumstances, data should be expressed with imprecision and uncertainty, due to their nature or just to their acquisition method. Moreover, data can also present structural irregularities, probably caused by differences in the representation schemata of the data sources.

Two trends have emerged to improve the modelling capability of the object-oriented data model with respect to aforementioned problems. On the one hand, *Fuzzy Object-Oriented Data Models* deal with the representation of data expressed with imprecision and uncertainty [5, 6]. On the other

F. Berzal et al.: *Fuzzy Types: A First Step Towards Lazy Types in the .NET Framework*, Advances in Soft Computing **2**, 413–418 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

hand, *Semistructured Data Management Systems* extend the capabilities of the object-oriented data model, allowing the representation of data whose structure is not completely regular [1, 4].

Recently, we proposed a Fuzzy Object-Oriented Data Model [3] rooted on the idea of using classical object-oriented capabilities in order to implement fuzzy object-oriented extensions. As part of this model, we considered the possibility of defining both the structure and the behavior of a class (i.e., its type) in a soft or fuzzy way. In our model, objects are created by using an appropriate $\alpha - cut$ of the fuzzy type definition that is associated to the class which the object belongs to.

Modern programming platforms, such as the .NET Framework or the Java programming platform, include advanced features that allow the implementation of proper solutions for the problems mentioned above. In this paper, we show how these features can be used to implement lazy typing capabilities. As a first step towards general lazy typing capabilities, we show how to develop applications with fuzzy types.

This paper is organized as follows. Section 2 describes the concept of lazy typing. Section 3 introduces the special case of fuzzy types. Section 4 illustrates the use of fuzzy types in C♯. Finally, some concluding remarks and guidelines for future work close our paper.

2 Lazy typing for data representation

Object-orientation modelling capabilities enable more flexible solutions for data representation than previous data models. However, the handling of irregular data is still difficult in the context of this (now conventional) typing environment. Problems where entities need to be managed with different levels of precision, or where entities can present diversity in their structures, require the use of more expressive and powerful techniques to define the type of a certain class of objects.

Figure 1(i) graphically explains the situation. The figure represents three objects belonging to a given class. The data which describes each object is conformed by the values of the attributes which are within the line that represents the object. As it can be seen, though the objects belong to the same class and represent occurrences of the same semantic entity, their representation is irregular, and the use of conventional data types will cause the appearance of plenty of null values.

Using current methods, we should need to create different types in order to solve this problem, one type for each different structure observed in data. The size of such perfect type mapping might be quite large, which would make its use difficult, and the resulting model would be distorted from the semantic point of view (many artificial classes would appear in the schema). Lazy or approximate typing techniques have begun to appear in order to improve the representation of this kind of classes. Using these techniques, an object does

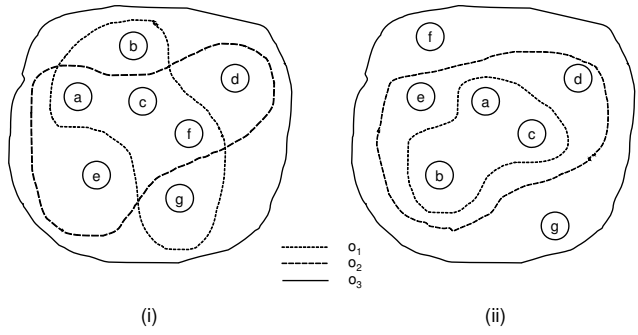


Fig. 1. The motivation behind lazy typing.

not have to fit its type definition perfectly. The object will only incorporate the properties it really needs from the type of its class.

3 Fuzzy Types: A particular example of lazy types

The three objects depicted in Figure 1(ii) would also benefit from lazy typing. However, in this case, the irregularity of the structure is lower, because the different sets of attributes fulfill an inclusion relation among them. In other words, the class the objects belong to represents a semantic entity whose occurrences can be expressed with different precision levels. Object o_1 is described with the lowest precision level, object o_3 employs the highest precision level, and object o_2 lays in an intermediate position.

Fuzzy types [7, 8] have been introduced to handle concepts with different levels of precision. In a fuzzy type, properties belong to the type with a certain membership degree. This membership degree allow us to organize the properties of the type in suitable α-cuts, each of one describing the type with a different level of precision. A fuzzy constructor method allows the creation of objects incorporating the desired α-cut of properties.

The use of a fuzzy type to represent the structure of classes like the one depicted in Figure 1(ii) allows us to keep a unique type in the schema, avoiding the multiplication of types and the appearance of null values, reasons which make our model desirable from the semantic and the implementation points of view, respectively.

Advanced features included in modern programming platforms make possible the implementation of frameworks which can be developed so that the programmer of soft computing applications can use fuzzy types in a transparent way. In particular, our framework makes use of reflection -the ability of an executing program to examine or "introspect" itself-, the dynamic creation of types, and metadata describing user-defined classes.

Our framework currently not only enables the use of fuzzy types in user-defined classes, but also can include imprecise domains for the values of the attributes which conform these types. As an example, the following section shows how fuzzy types can be used in the C♯ programming language.

4 A case study in C♯

Olive oil is one of the main products obtained from the Andalusian agrarian industry. In fact, large stretches of land are devoted to the cultivation of olive trees. The area and the number of trees in each plot of land are of remarkable interest for governmental agencies, since they can use that information to forecast the annual production and estimate the necessary funds (usually destined to the subvention of this farming activity).

Let us consider, for instance, the usual process of a survey aimed at obtaining information about each plot of land. First, the surveyor could obtain the approximate area of each farm directly from the farmers. Afterwards, the surveyor could consult the Property Registry for information about the polygon describing the exact limits of the plot of land. Finally, the surveyor could obtain an aerial image of the farms to check their current exploitation.

At a given moment during this survey process, each plot of land can be at a different stage, so the database will contain farms described with three different precision levels. Obviously, the way the actual area and the number of trees in a farm can be obtained depends on the precision level it is described with. While there is no other alternative than using the declared area of the farm when no other information is available, the availability of its precise perimeter allows a more accurate computation of the area. Also, an aerial photograph can be used to perform a "visual count" of the actual number of trees.

Our framework allows the representation of this problem by means of a fuzzy type in C♯ as figure 2 shows.

The fuzzy type `Lot`, representing a plot of land, is organized into three levels of precision (with degrees 1.0, 0.9, and 0.8[1]). C♯ metadata attributes are used to express the membership degree of each attribute and to create alternatives for the class methods `GetArea` and `GetTree`. Our framework will automatically determine the proper alternative to use over a given instance analyzing the use of attributes at different precision levels.

The creation of fuzzy objects can be done just by specifying an α-cut for the type when invoking a generic fuzzy object factory:

```
Lot lotObject = (Lot) FuzzyFactory.Create(typeof(Lot), degree);
```

[1]The concrete values used for α may not be very important since their only purpose is to organize a structure in a certain number of precision levels. In some situations, however, giving some semantics to those values might be necessary. For example, if the structure is being inferred from a set of instances.

```
[Fuzzy]
public class LotImplementation: Lot {

  // Fuzzy structure

  [MembershipDegree(1.0f)]
  private Area observedArea;
  [MembershipDegree(0.9f)]
  private Polygon perimeter;
  [MembershipDegree(0.8f)]
  private Image photo;

  // Fuzzy behavior

  public Area GetArea () {
    return observedArea;
  }

  [AlternativeImplementation("GetArea")]
  public Area GetAreaFromPerimeter () {
     return perimeter.GetArea();
  }

  public FNumber GetTrees () {
    return (FNumber) ( GetArea()* AverageTreeDensity );
  }

  [AlternativeImplementation("GetTrees")]
  public FNumber GetTreesFromPhoto () {
    return ImageMorphologyAnalyzer.GetObjectCount
               (photo, perimeter, AverageTreeSize);
  }
  ...  // Other standard class fields and methods
}
```

Fig. 2. Fuzzy type Lot.

Once the object attributes have been initialized, the lot area and its number of trees can be obtained just by invoking the corresponding methods (the proper implementation of each method will be automatically used depending on the object precision level):

```
lotObject.getArea();
lotObject.getTrees();
```

5 Conclusions and further work

In this paper we have presented a framework which allows the development of applications with fuzzy types, as a first approximation to the world of lazy typing. As a first approximation to the problem, we show how this capabilities can be used to develop applications with fuzzy types.

Advanced features of modern programming platforms, like reflection and metadata attributes, allow the transparent implementation of these typing capabilities. Programmers can easily use these capabilities without having to change their development environment nor programming language.

This framework is currently being generalized to allow for lazy typing in a more general range of semistructured data problems, and it can be integrated with a previous framework which allows the management of fuzzily described objects[2].

This paper has been supported in part by the Spanish "Comisión Interministerial de Ciencia y Tecnología" under grants TIC2003-08687-C02-02, TIC2002-00480, and TIC2002-04021-C02-02.

References

1. Serge Abiteboul. Querying semi-structured data. In *ICDT*, volume 6, pages 1–18, 1997.
2. F. Berzal, J.C. Cubero, N. Marín, and O. Pons. Enabling fuzzy object comparison in modern programming platforms through reflection. In *Lecture Notes in Artificial Intelligence*, pages 660–667, 2003.
3. F. Berzal, N. Marín, O. Pons, and M. A. Vila. Using classical object-oriented features to build a foodbs. In Jonathan Lee, editor, *Software Engineering with Computational Intelligence*. Springer Verlag, 2003.
4. Peter Buneman. Semistructured data. In *PODS'97*, 1997. Invited Tutorial.
5. Rita De Caluwe. *Fuzzy and Uncertain Object-Oriented Databases: Concepts and Models*, volume 13 of *Advances in Fuzzy Systems-Applications and Theory*. World Scientific, 1997.
6. J. Lee, J-Y. Kuo, and N-L. Xue. A note on current approaches to extent fuzzy logic to object oriented modeling. *International Journal of Intelligent Systems*, 16:807–820, 2001.
7. N. Marín, M. A. Vila, and O. Pons. Fuzzy types: A new concept of type for managing vague structures. *International Journal of Intelligent Systems*, 15:1061–1085, 2000.
8. N. Marín, M. A. Vila, and O. Pons. A strategy for adding fuzzy types to an object-oriented database system. *International Journal of Intelligent Systems*, 16:863–880, 2001.

Fuzzy Induction via Generalized Annotated Programs

Tomáš Horváth and Peter Vojtáš

Faculty of Science, University of Pavol Jozef Šafárik Institute of Computer Science
Jesenná 5, 040 01 Košice, Slovakia
thorvath@cs.science.upjs.sk
vojtas@kosice.upjs.sk

Abstract. The aim of this paper is to describe the method of induction of generalized annotated programs called IGAP what is a special case of inductive fuzzy logic programming for monotonely classified data. This method is based on the multiple use of two valued ILP and the syntactical equivalence of fuzzy logic programs and a restricted class of generalized annotated programs. Finally we compare our method with several fuzzy ILP methods.

Introduction and motivation

In a standard logical framework, we are restricted to representing only facts that are true or false absolutely. Thus, this framework is unable to represent and reason with uncertain and noisy information. Uncertainty is unavoidable in the real world: our information is often inaccurate and always incomplete, and only a few of the "rules" that we use for reasoning are true in all (or even most) of the possible cases.

R. Fagin in [4] mentions a nice example: the user wants to get information about restaurants. The user has an aggregation function that gives an overall score to each restaurant based on how good it is, how inexpensive it is, and how close it is – these are inherently fuzzy attributes.

This is a typical example of multifeature querying (featuring different properties of restaurants). What is moreover specific, there are no clearly good and bad restaurants, rather, there is a monotone hierarchy of better, cheaper,…restaurants. The users overall preference is monotonically dependent on the grade of fulfillment of single features.

Main point of interest of this paper is to learn the (user dependent) aggregation function, which gives an overall score to a graded fulfillment of

T. Horváth and P. Vojtáš: *Fuzzy Induction via Generalized Annotated Programs*, Advances in Soft Computing **2**, 419–433 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

all user's requirements. This appears also in multicriterial and/or multi-user decision making and also in graded classification (where the monotonicity of dependence can be more problematic).

Inductive logic programming ILP has been successfully applied to classification problems for domains with non-numerical data. We present a method for induction of generalized annotated programs in the case of monotonely classified data what is a special case of fuzzy ILP task. We have chosen multivalued logic, especially GAP [5] because it provides us with a comparative notion of truth which models user preferences. Our construction is based on a syntactical equivalence of fuzzy logic programs FLP [16] and a restricted class of generalized annotated programs GAP.

We formulate our induction problem as preserving the order of overall scoring (or more relaxed, we require the learning does not invert classification).

We show learned rules from a synthetic dataset with distribution typical for web querying and Information Retrieval. Finally we compare our method with other fuzzy ILP systems (methods) [8,10].

Fuzzy Logic Programming

We formulate properties of residuation.

Definition. Let C a conjunctor and I an implicator. In what follows, b, h, r are universally quantified and range through [0,1]. We define following properties of C and I:

1. *(a)(C,I)* $r \le I(b,h)$ iff $C(b,r) \le h$
2. *ϕ 2(C,I)* $C(b,I(b,h)) \le h$
3. *ϕ 3(C,I)* $r \le I(b, C(b,r))$

Observations.

- (a)(C,I) iff (ϕ2(C,I) and ϕ3(C,I))
- Assume (a)(C,I) then
  $I(b,h) = \sup\{r: C(b,r) \le h\}$ and $C(b,r) = \inf\{h: I(b,h) \ge r\}$.
- Given C, then there is an I such that (a)(C,I) iff C is left continuous in r.
- Given I, then there is a C such that (a)(C,I) iff I is right continuous in h.

In our computational model, we have conjunctors $C_1,...,C_n$ which are residual to above implications. These need not be truth functions of any conjunctions in our language. We assume conjunctors are left continuous.

Any formula built from atoms using conjunctions, disjunctions and aggregations is called a *body*. Every composition of conjunctors, disjunctors and aggregation operators is again an aggregation operator. Hence, without a loss of generality, we can assume that each body is of the form B =

$@(B_1,\ldots,B_n)$. A *rule* of FLP is a graded implication $(A \leftarrow (B_1,\ldots,B_n).r)$, where A is an atom, $@(B_1,\ldots,B_n)$ is a body and $r \in Q \cap [0,1]$ is a rational number. $(A \leftarrow @((B_1,\ldots,B_n))$ is the logical part of the rule and r is the quantitative part of the rule). A *fact* is a graded atom (B.b).

A finite set P of positively graded FLP rules and facts is said to be a *fuzzy logic program* if there are no two rules (facts) with the same logical parts and different quantitative parts. It can be represented as a partial mapping P: Formulas $\rightarrow$ (0,1] with the domain of P, dom(P) consisting only of atoms and logical parts of FLP rules of the form $A \leftarrow @(B_1,\ldots,B_n)$. The quantitative part of the rule is $r = P(A \leftarrow @((B_1,\ldots,B_n))$.

Let B_L be the Herbrand base. A mapping $f:B_L \rightarrow [0,1]$ is said to be a fuzzy *Herbrand interpretation*. Our fuzzy logic is truth functional i.e. f can be extended to all formulas by $\underline{f}$ along the complexity of formula using the truth function of connectives. A graded formula $(\varphi.x)$ is true in an interpretation f if $\underline{f}(\varphi) \geq x$. For a rule this means that

$$\underline{f}(A \leftarrow @((B_1,\ldots,B_n)) = \leftarrow^{\bullet}(\underline{f}(A), @^{\bullet}(\underline{f}(B_1),\ldots,\underline{f}(B_n))) \geq r \qquad (1)$$

Recall the many-valued modus ponens

$$\frac{(B.b), (A \leftarrow_i B.r)}{(A.C_i(b,r))} \qquad (2)$$

We base our procedural semantics on the "backward usage of modus ponens" (no refutation nor resolution is applied here). We know by the residuality of C_i that this is a sound rule[15].

Definition. Assume P is a fuzzy logic program. Then

$$T_P(f)(A) = \max\{\sup\{C_i(\underline{f}(B),r): (A \leftarrow_i B.\ r) \text{ is a ground instance of a rule in the program P}\}, \sup\{b: (A.\ b) \text{ is a ground instance of a fact in the program P}\}\}. \qquad (3)$$

We know that this operator is continuous [14] and it's fixpoint is the minimal model of the definite program P.

Generalized Annotated Programs

M. Kifer and V. S. Subrahmanian ([5]) introduced generalized annotated logic programs (GAP) that unify and generalize various results and treatments of multivalued logic programming (like fuzzy, possibilistic, signed, …).

In multivalued logic the set of truth values is modeling our set of preferences. The whole theory of GAP is developed in a general setting for truth value set being lattices. We restrict here ourselves to finite subsets of the unit interval of real numbers [0,1]. Namely, it can be a result of discretization of data but also of granulation of user preferences (psychology is learning us that user can distinguish 7 +-2 classes). Indeed for different attributes the users granulation can be different (e.g. one can distinguish 5 degrees of quality of restaurants and maybe up to 10 classifications of distance). In this paper we do not concentrate on this, our training data say what is the range of classification (which can be always embedded into [0, 1]).

Most of many valued logical systems are truth functional – i.e. assigning the truth value to formula using truth functions of connectives along the complexity of formula. This can cause serious problems when trying to learn the semantics from data. There is an extensive research done in fuzzy ILP systems, nevertheless we do not go this direction (FILP in [11]).

Annotated logic, on the other hand, appeared to associate truth values with each component of a formula (typically implication in IF-THEN rules) rather than the implication as a whole. This implication is interpreted in a "classical logic" fashion. This is a substantial improvement in learning. We do not have to learn (an unknown) set of truth functions of connectives, instead we learn the aggregation function.

Definition. A function a: $[0,1]^i \to [0,1]$ is an *annotation function* if it is left continuous and order preserving in all variables.

The language of annotated programs consists of a qualitative and a quantitative part. The qualitative part of GAP consists of the usual language of predicate logic (with object variables, constants, predicates and function symbols). The quantitative part of the language has annotation variables and annotation constants and a set of basic *annotation terms* of different arity. Every basic annotation term ρ (considered as a symbol of our alphabet) is assigned an annotation function $\rho^\bullet$. In [5] it is assumed that all $\rho^\bullet$'s are "total continuous (hence monotonic) functions" - in the sense of lattice theory. This lattice continuity means that all annotation functions are nondecreasing and left continuous in the topology of real line. More complex annotation terms (and functions) are built from these building blocks preserving arity. Notice, that $\rho^\bullet$ can be considered as the truth function of an aggregation operator.

If A is an atomic formula and α is an annotation term, then $A : \alpha$ is an *annotated atom*. When α is an annotation variable, then $A : \alpha$ is said to be variable-annotated (v-annotated).

Definition (FLP and GAP transformations [16]). Assume $C = A{:}\rho \leftarrow B_1 {:}\mu_1 \,\&\, ... \,\&\, B_k {:}\mu_k$ is an annotated clause. Then flp(C) is the fuzzy rule $A \leftarrow \rho(B_1,...,B_k).1$, here ρ is understood as an n-ary aggregator operator.

Assume $D = A \leftarrow_i @(B_1,...,B_n).r$ is a fuzzy logic program rule. Then gap(D) is the annotated clause $A{:}C_i(@(x_1,...,x_n),r) \leftarrow B_1{:}x_1,...,B_n{:}x_n$

Theorem. Assume C is an annotated clause, D is a fuzzy logic program rule and F is a fuzzy Herbrand interpretation. Then
f is a model of C iff f is a model of flp(C)
f is a model of D iff f is a model of gap(C)

In Kifer-Subrahmanian there is developed the deductive (procedural and declarative part of GAP system). So far we do not know about any inductive GAP system (also personally confirmed by V.S. Subrahmanian).

Induction of generalized annotated programs (IGAP)

In ILP, given is a set of examples $E = E^+ \cup E^-$, where E^+ contains positive and E^- negative examples, and background knowledge B. The task is to find a hypothesis H such that $\forall e \in E^+$: $B \wedge H \models e$ (H is complete) and $\forall e \in E^-$: $B \wedge H \not\models e$ (H is consistent). This setting, introduced by Muggleton ([3, 6]), is also called learning from entailment.

In order to search the space of relational rules (program clauses) systematically, it is useful to impose some structure upon it, e.g. an ordering. One such ordering is based on subsumption (clause C subsumes C' if there exist a substitution θ, such that $C\theta \subseteq C'$). Notice, that if C subsumes D then $C \models D$. The converse does not hold always.

Our system is based on a translation of our IGAP problem to multiple use of classical ILP system with additional monotonicity axiom in the background knowledge. We provide experiments on synthetic benchmark data with distribution typical for web querying and Information Retrieval.

Although our approach is multirelational, in experiments (for better visualization) we skip this and use simplified synthetic data with two features (although features of objects have to be joined from different relations). Aggregation functions play important role in our models. Here we mention only some of them, the arithmetic mean, weighted average, ordered weighted averaging, an unordered variant of weighted mean, ad hoc aggregation functions. For more see [12]. These aggregation operators can be helpful in comparing composite conditions. Training data (with different distribution as above) depicting the influence of aggregation function $@_{ws}(x_1,x_2)=(x_1+2x_2)/3$ are shown bellow (on the Fig. 1 @ is marked by y).

Since ILP system understands numbers syntactically, we discretize the values to several classes. Notice, that the classification (learning) is better if we use finest discretization (we mention it later).

Because of monotonicity axioms we expect that the used ILP system can work with rules in the background knowledge. For this purposes we use ALEPH [1, 9], which is based on the inverse entailment [7].

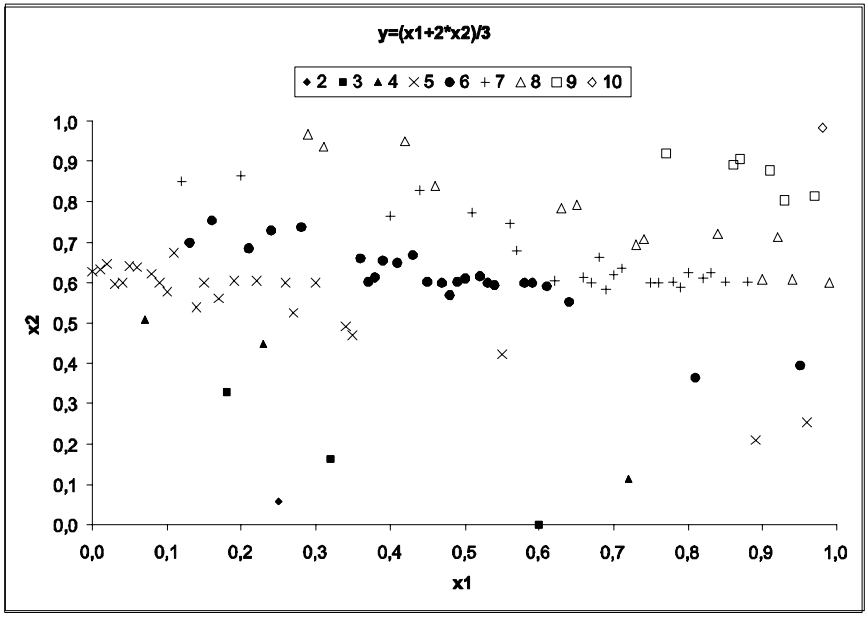


Fig. 1. Training data – classes of the aggregation function after discretization: 1 means values from the interval <0;0.1), 2 means <0.1;0.2), 3 is <0.2;0,3), …, and 10 means numbers from <0.9;1>. The class 1=<0;0.1) contains no members, so it don't figure in the legend.

The two valued ALEPH works as follows: Given background knowledge B and examples E and the hypothesis H. It must hold, that $(B \wedge H) \models E$. If we rearrange the above using the law of contraposition we get the more suitable form $(B \wedge \neg E) \models \neg H$. In general B, H and E can be arbitrary logic programs but if we restrict H and E to being single Horn clauses, $\neg H$ and $\neg E$ above will be ground skolemized unit clauses. If $\neg\bot$ is the conjunction of ground literals which are true in all models of $B \wedge \neg E$ we have $(B \wedge \neg E) \models \neg\bot$. Since $\neg H$ must be true in every model of $B \wedge \neg E$ it must contain a subset of the ground literals in $\neg\bot$. Hence $(B \wedge \neg E) \models \neg\bot \models \neg H$ and so $H \models \bot$. The complete set of candidates for H could in theory be

found from those clauses which imply $\perp$. A subset of the solutions for H can then be found by considering those clauses which subsume $\perp$. ALEPH searches the latter subset of solutions for H that subsume $\perp$. $\perp$ is called saturation of example.

The multiple use of classical ILP system

We transfer the problem of annotated ILP with graded background knowledge and a graded set of examples to several crisp (crisp = two valued) ILP problems. First, assume B is a annotated background knowledge consisting of annotated facts (so far we do not discuss the case of annotated rules in B). Then c(B) is the crisp knowledge acquired from B by adding an additional attribute for the truth value to each predicate (e.g. we transform A(c).x to A(c,x)). Second, for every $\alpha \in [0, 1]$ in the range of our annotated set of examples, we put E_α^+ to be the upper and E_α^- to be the lower α cut of the annotated set E. Assume the classical ILP system for each α and (E_α^+, E_α^-) and c(B) returns a crisp set of hypothesis H_α . Our aim is to have a formal model allowing us correctly to answer the annotated ILP problem (E,B) with the annotated set of hypothesis H, such that for all α, the α cut of H is exactly H_α in other words, the annotated hypothesis fulfils $H|H_\alpha \equiv \alpha$.

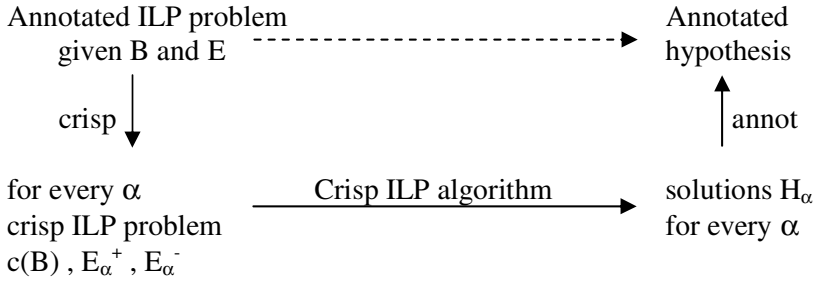


Fig. 2. The transformation of annotated ILP task to multiple use of crisp ILP task

The ILP graded classification task

Given the annotated set of examples E and a background knowledge B containing annotated atoms. Our task is to find a Generalized Annotated Program H such that the following holds:

If f_m is a Herbrand interpretation which is a minimal model of both B and the GAP program H then we have for all e_1 and e_2 in the domain of E:

$$E(e_1) < E(e_2) \quad \text{implies } f_m(e_1) \leq f_m(e_2) \tag{4}$$

and

$$E(e_1) > E(e_2) \quad \text{implies } f_m(e_1) \geq f_m(e_2) \tag{5}$$

That is, our classification learned by minimal model of B and H does not contradict the original classification E, or in other words, H does not declare an object e_1 strictly better than e_2 if in E is just the opposite, i.e. e_2 strictly better than e_1.

We can base the cover relation of our ILP system on the fixpoint semantics of FLP Cover(P) = $T_P^{\omega}(0)$ – from Eq. (3).

The monotonicity axiom in the background knowledge

We transfer the problem of graded ILP with annotated (graded) background knowledge and annotated set of examples (graded examples) to several classical ILP problems, so that c(B) is the knowledge acquired from B by adding an additional attribute for the truth value. The target predicate is translated to classical ILP differently. For every $\alpha \in T$ let E_α^+, E_α^- are the upper and lower cuts of the graded (annotated) set E.

For each α consider the classical ILP task with E_α^+ and E_α^- as positive and negative examples and c(B). System returns a set of hypothesis H_α. Our aim is to have a formal model allowing us correctly to answer the GAP ILP problem (E,B) with the GAP set of hypothesis H, such that for all α, the α cut of H is exactly H_α.

Qualitative constraints

There is a small problem, namely, we are learning under a qualitative constraint about our orderings of domains of attributes. Namely, when on a certain level α, the classical ILP system provides us with a logic programming rule

$$H(x) \leftarrow B_1(x, \alpha_1),...,B_n(x, \alpha_n)$$

we will transform it to a GAP rule

$$H(x): \alpha \leftarrow B_1(x):\alpha_1,...,B_n(x):\alpha_n$$

Now it can happen that in our data there is an object constant d with body predicates annotated by constants $\beta_1, \ldots, \beta_n$ such that for all $i \leq n$ we have $\alpha_i \leq \beta_i$. Then the body of the above rule is fired for object d, and we have to guarantee that H(d): α is also true, that is, in our training data there is constant annotated atom H(d):β with $\beta \geq \alpha$.

We offer one solution, namely it is enough to add to the classical background knowledge c(B) the monotonicity axiom. i.e. for all predicates p

$$p(d,y) \leftarrow y \leq x \, , \, p(d,x). \tag{6}$$

Assuming the $\leq$ is definable. Then the classical ILP is learning with H(d) in E_α^+ and with both $B_1(d, \alpha_1),\ldots,B_n(d, \alpha_n)$ and $B_1(d, \beta_1),\ldots,B_n(d, \beta_n)$ in the background knowledge.

Implementation Results

In this section we introduce an example of rules induced by repetitive usage of Aleph under the monotonicity axiom in the background knowledge. The found rules (depicted on the Fig. 3) have a form (we mentioned that we discretized the values of x1,x2 and y to ten classes):

For α=9
y(A) :- x1(A,8), x2(A,9).
For α=8
y(A) :- x1(A,9), x2(A,7). y(A) :- x1(A,7), x2(A,8).
y(A) :- x2(A,9).
For α=7
y(A) :- x2(A,9). y(A) :- x1(A,7), x2(A,7).
y(A) :- x1(A,4), x2(A,8). y(A) :- x1(A,8), x2(A,6).
For α=6
y(A) :- x1(A,7), x2(A,4). y(A) :- x1(A,4), x2(A,7).
y(A) :- x1(A,5), x2(A,6). y(A) :- x2(A,8).
For α=5
y(A) :- x1(A,1), x2(A,6). y(A) :- x2(A,7).
y(A) :- x1(A,4), x2(A,4). y(A) :- x1(A,5), x2(A,3).
For α=4
y(A) :- x1(A,3), x2(A,3). y(A) :- x1(A,5), x2(A,1).
y(A) :- x2(A,5).
For α=3
y(A) :- x1(A,4). y(A) :- x2(A,1).

The class 10 contains only one member, so we do not learn it and the class 1 contains no member, so the lowest grade (class) we can learn is 3. Regarding to discretization these rules have a form:

For α=0.8

```
y(A) :-  x1(A,0.7), x2(A,0.8).
For α=0.7
y(A) :-  x1(A,0.9), x2(A,0.6).    y(A) :- x1(A,0.6), x2(A,0.7).
y(A) :-  x2(A,0.9).
For α=0.6
y(A) :-  x2(A,0.8).               y(A) :- x1(A,0.6), x2(A,0.6).
y(A) :-  x1(A,0.3), x2(A,0.7).    y(A) :- x1(A,0.7), x2(A,0.5).
For α=0.5
y(A) :-  x1(A,0.6), x2(A,0.3).    y(A) :- x1(A,0.3), x2(A,0.6).
y(A) :-  x1(A,0.4), x2(A,0.5).    y(A) :- x2(A,0.7).
For α=0.4
y(A) :-  x1(A,0.1), x2(A,0.5).    y(A) :- x2(A,0.6).
y(A) :-  x1(A,0.3), x2(A,0.3).    y(A) :- x1(A,0.4), x2(A,0.2).
For α=0.3
y(A) :-  x1(A,0.2), x2(A,0.2).    y(A) :- x1(A,0.4), x2(A,0.1).
y(A) :-  x2(A,0.4).
For α=0.2
y(A) :-  x1(A,0.3).               y(A) :- x2(A,0.1).
```

The GAP forms of these rules are:

```
y(A):0.8 :-  x1(A):0.7, x2(A):0.8.      y(A):0.7 :-  x1(A):0.9, x2(A):0.6.
y(A):0.7 :-  x1(A):0.6, x2(A):0.7.      y(A):0.7 :-  x1(A):0,   x2(A):0.9.
y(A):0.6 :-  x1(A):0,   x2(A):0.8.      y(A):0.6 :-  x1(A):0.6, x2(A):0.6.
y(A):0.6 :-  x1(A):0.3, x2(A):0.7.      y(A): 0.6 :- x1(A):0.7, x2(A):0.5.
y(A): 0.5 :- x1(A):0.6, x2(A):0.3.      y(A): 0.5 :- x1(A):0.3, x2(A):0.6.
y(A): 0.5 :- x1(A):0.4, x2(A):0.5.      y(A): 0.5 :- x1(A):0,   x2(A):0.7.
y(A): 0.4 :- x1(A):0.1, x2(A):0.5.      y(A): 0.4 :- x1(A):0,   x2(A):0.6.
y(A): 0.4 :- x1(A):0.3, x2(A):0.3.      y(A): 0.4 :- x1(A):0.4, x2(A):0.2.
y(A): 0.3 :- x1(A):0.2, x2(A):0.2.      y(A): 0.3 :- x1(A):0.4, x2(A):0.1.
y(A): 0.3 :- x1(A):0,   x2(A):0.4.      y(A): 0.2 :- x1(A):0.3, x2(A):0.
y(A): 0.2 :- x1(A):0,   x2(A):0.1.
```

and they mean (for example the first rule)
IF x1 is at least 0.7 AND x2 is at least 0.8 THEN y is at least 0.8
In FLP this rule is
y(A) :- ρ(x1, x2).1, where the value of ρ(0.7,0.8) is 0.8

Preprocessing

In this section we mention two preprocessing methods what have impact on the quality of learning – discretization and monotonicity checking.

Discretization

We can understand the grades – classes - like truth values or annotated constants (these values can be e.g. letters A,B,C,D,E like grades of stu-

dents in the school or words LOW, MEDIUM, HIGH, our method can work with it), important is the monotonicity (e.g. the student with grade B fulfills teachers requirements for the grades C, D, E (the worse) and does not fulfill requirements for the grade A (better)). The discretization of data set specifies the number of grades, so it is obvious that the completeness of the hypothesis grows with the finest discretization. On the other side the complexity of the computing grows, too.

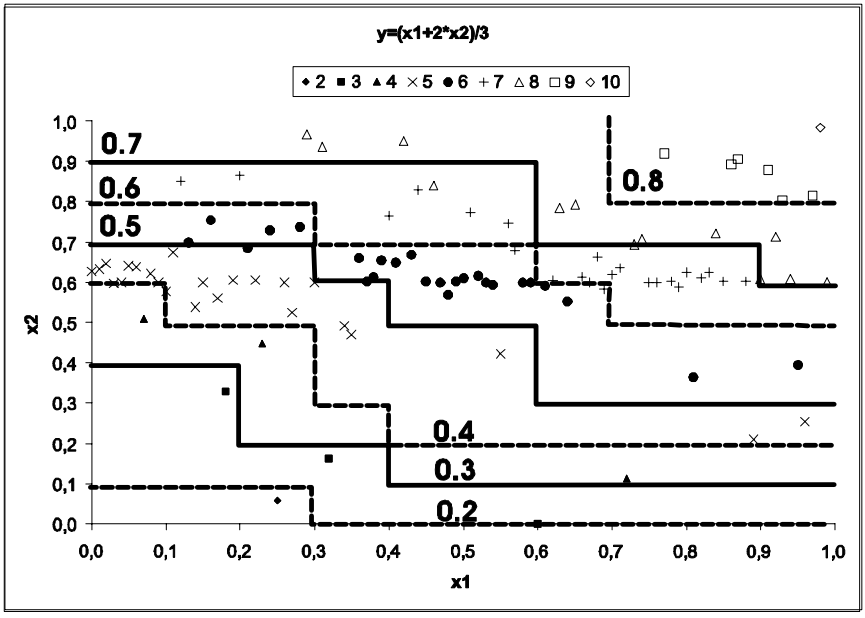


Fig. 3. The results for the discretization of x_1,x_2 and y to 10 classes for the grades – classes (9, 8, ..., 4, 3) of aggregation function. The squares means the rules from the hypothesis.

Checking the monotonicity

In our case (dataset) – from the selected aggregation function - it is clear that the attribute values of x1 and x2 contribute positively to higher classification in the natural ordering of the domain. In many cases these dependencies are not obvious or there can be the dependence preferring medium values and/or there is no monotone dependence at all. For checking these dependencies we can use several techniques – statistical methods (regression, …), methods for qualitative learning [2], etc.

We can use for this checking our method, too, only just with checked attribute (and the monotonicity rules) in the background knowledge. Our IGAP gave quite satisfactory results, violation of classification ordering was bellow 2% and comparison to other qualitative learning (e.g. such as [2]) showed comparable results.

Factors of complexity

The size of examples and the number of predicates in background knowledge contribute to the size of the search space. Moreover the size increases as our set of truth values (the number of classification classes) increases.

There are basically two optimization heuristics in our approach – to reduce the size of example (from above) set and to reduce the size of predicates in the background knowledge (from bellow).

Reducing the background knowledge

In the example above, the system has found rules for class 0.4 (hence all examples classified with 0.4 and higher)

y(A) :- x1(A,0.1), x2(A,0.5).
y(A) :- x2(A,0.6).
y(A) :- x1(A,0.3), x2(A,0.3).
y(A) :- x1(A,0.4), x2(A,0.2).

Here we see, that all rules have the condition on x2 at least 0.2. Hence we can cut the background knowledge and delete all predicates with x2 < 0.2 in all ILP tasks for classes higher than 0.4. This is possible because we are learning an aggregation function which is monotone in all coordinates and all rules characterizing class $E^+_{0.4}$ require x2 to be at least 0.2 (in graphical visualization we see all data points belonging to higher classes have x2 bigger than 0.2). Notice that on x1 there is no reduction, this is a property of the learned function and the data set.

Moreover, notice, this optimization can be used from bellow, in general first we learn the lowest class $E^+_{0.1}$, finding the hypothesis $H_{0.1}$ we can form for every feature attribute a threshold

$t^i_{0.1} = \min\{a$: there is a rule y(A, 0.1) :- $x_i(A,a)$, rest_of_body in $H_{0.1}\}$

Then the reduced background knowledge for further induction will be

$B_{0.1} = \{x_i(A,a)$: $x_i(A,a) \geq t^i_{0.1}\}$

and similarly for further steps. Experimental results, showing interdependence between aggregation function to be learned and optimization techniques follow.

Reducing the example set

The idea of reducing the example set resembles the ALEPH optimization "induce", nevertheless it differs.

For E^+_a and E^+_b with a < b we have inclusion $E^+_a \subseteq E^+_b$. Hence rules in H_b cover also examples in E^+_a and hence instead of learning with it we learn only with E^+_a- H_b . This is a significant help.

Reducing B and E at the same time

Combination of both reduction is not straight forward. Reduction of B-reduction has to be applied from bellow (learning lower classifications) whereas E-reduction has to be applied from above.

We have tried several heuristics. E.g. zigzag combination of E and B reduction for classes: E-reduce after learning for 1,0; then B reduce for 0,1 (learning with E+0.1- H1.0); then E-reduce (learning E+0.9- H1.0 over B0.1), and so on. Another heuristics has shown to be useful, start from the middle, first learn e.g. H0.6 for E+0.6 and then proceed with up and down appropriate reduction (we have experimented with several variants of this method, but it is out of the scope of this paper).

This methods have shown to be dependent on data and the learned aggregation function (E-reduction is better for or-like aggregations, B-reduction for and-like) and shows that some prior information on data distribution (e.g. histograms) can be useful for better learning.

All this together show substantial improvement of learning complexity, especially when summing up the whole learning.

Conclusions and comparison

A good overview on fuzzy ILP is in [8] where a system enriching relational learning with several types of fuzzy rules was introduced based on fuzzy FOIL. Our aim in this paper is two fold, to avoid arbitrary discretization and it enlarges the expressive power of what is learned by considering different types of fuzzy rules, which may describe gradual behaviors of related attributes or uncertainty pervading conclusions – this is achieved mainly by using fuzzy linguistic variables. Second difference is that authors of [8] cut the whole example set and the background knowledge by same α (in one case examples by α and background knowledge by 1-α) and then calculate the confidence factor of the whole rule by aggregation weights of association rules. The main difference with our approach is that our background knowledge for learning on level α contains information for

all possible $\alpha_1, \ldots, \alpha_n$ in the range of predicates and our system induces rules with different weights in body and head. Moreover, as mentioned above, we do not have to consider many valued connectives, GAP uses for satisfaction of complex formulas two valued logic.

To create a more suitable set of rules using ILP in [11] developed an algorithm called FS-FOIL, that extends the original FOIL algorithm. While FOIL was developed to find Horn clauses, they modified it to be able to handle first order fuzzy predicates where cover compares confidence and support of fuzzy predicates.

In [10] another fuzzy variant of the ILP method FOIL is used for a crisp classification of „good arch“ in civil engineering using vague linguistic hedges. Their system FCI uses min-max logic with Lukasiewicz implication and creates only crisp hypothesis. Our system enables to describe more general dependencies (our learned function - from the annotation - is not expressible using min, max). FCI search of hypothesis tries to cover positive examples with degree at least μ^+ and avoid covering of negative examples with degree bellow μ^-. We can model this approach in our method if we define the example sets as $E^+ = \{e: E(e) \geq \mu^+\}$ and $E^- = \{e: E(e) < \mu^-\}$ and leave out monotonicity axioms from the background knowledge. Notice, that we have learning „with holes“ (between positive and negative example sets) in case of $\mu^- < \mu^+$.

We have good results on several experiments with this our method (student’s valuation in school, auto-mpg data from UCI repository, business competitiveness).

We have studied aggregating multifeature querying which appears in user preferences, rating in information retrieval, semantic web and multimedia databases. We introduced IGAP - Inductive Generalized Annotated Programming method based on multiple use of classical ILP method. Further work requires full implementation of these techniques.

Acknowledgements

The paper was partially supported by VEGA 1/0385/03.

References

1. ILP system ALEPH www.comlab.ox.ac.uk/oucl/research/\newlineaeas/machlearn/Aleph/aleph.pl.
2. Bratko I, Šuc D (2003) Learning qualitative models. AI Magazine 24:107-119

3. Džeroski S, Lavrač N (2001) An introduction to inductive logic programming. In: Džeroski S, Lavrač N (eds) Relational data mining. Springer, Berlin.
4. Fagin R (2002) Combining Fuzzy Information: an Overview. In: ACM SIGMOD Record 31: Database principles column, pp 109-118.
5. Kifer M, Subrahmanian VS (1992) Theory of generalized annotated logic programming and its applications. J. Logic Programming 12: 335-67
6. Muggleton S (1991) Inductive logic programming. New Gen Comp 8: 295-318
7. Muggleton S (1995) Inverse entailment and Progol. New Gen Comp 13: 245-86
8. Prade H, Richard G, Serrurier M (2003) Enriching relational learning with fuzzy predicates. In : Proc. PKDD'03, LNCS 2838, Springer, pp 399-410
9. Srinavasan A (2000). The Aleph Manual. Tech Rep Comp Lab Oxford Univ. at http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/aleph.html
10. Shibata D et al (1999) An induction algorithm based on fuzzy logic programming. In: Zhong N, Zhou L (eds) Proc. PAKDD'99, LNCS 1574, Springer, pp 268-273
11. Winiwarter W, Bodenhofer U and Drobics M (2000) Interpretation of self-organizing maps with fuzzy rules. In: ICTAI, IEEE.
12. Zimmermann H. J. (1996) *Fuzzy Set Theory*. Kluwer Academic Publishers, Boston, 1996
13. Vojtáš P, Horváth T, Krajči S, Lencses R (2004) An ILP model for a monotone graded classification problem. Kybernetika 40,3
14. Pokorný J, Vojtáš P (2001) A data model for flexible querying. In: Caplinskas A and Eder J (eds) Proc. ADBIS'01, Lecture Notes in Computer Science 2151, Springer Verlag, Berlin, pp.280-293
15. Vojtáš P (2001) Fuzzy logic programming. Fuzzy Sets and Systems. 124,3: 361-370
16. Krajči S, Lencses R, Vojtáš P (2004) A comparison of fuzzy and annotated logic programming. Fuzzy Sets and Systems, 144 : 173–192

Evaluating Fuzzy Association Rules on XML Documents

Carlo Combi, Barbara Oliboni, and Rosalba Rossato

Dipartimento di Informatica
Università degli Studi di Verona
Ca' Vignal 2 - Strada Le Grazie 15
37134 Verona - Italy
{combi,oliboni,rossato}@sci.univr.it

Abstract. In this work we propose a flexible approach to evaluate association rules on XML documents. In particular we describe evaluation techniques in order to assign a satisfiability degree to *structural association rules*, which allow one to evaluate the similarity of the XML document with respect to a given structure, and to *value association rules* which allow one to capture the similarity between the information contained in the XML document and the required information.

1 Introduction

During the last years XML has became very popular as a standard for representing, exchanging, and publishing information [1, 15]. XML is a markup language which is suitable for representing semistructured data; usually in this kind of information no pre-imposed schema or type is needed for data interpretation itself. The need of describing association rules over XML documents has arisen in some work [6, 14]. Indeed, even without a document with a fixed structure, it could be interesting to be able to identify in an XML document some recurrent situations: for example, in a document which describes information about employees, it could be useful to discover that an "employee" element usually contains a "salary" element. Moreover, it could be useful to discover that most "salary" subelements (of the elements "employee") have the content "900".

In this work we propose a flexible approach to evaluate association rules on XML documents. In [9] we proposed two kinds of association rules: *structural* and *value* associations, which allow one to consider the structure and the content of an XML document, respectively. The evaluation of structural association rules returns a similarity degree taking into account the structure of the document. The evaluation of value association rules returns a similarity degree taking into account the document content. In this work we will

C. Combi et al.: *Evaluating Fuzzy Association Rules on XML Documents*, Advances in Soft Computing **2**, 435–448 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

propose different kinds of evaluation techniques both for structural and value association rules.

The structure of the paper is as follows. In Section 2 we discuss related work for the description of association rules, integrity constraints and functional dependencies for XML documents. In Section 3 we first describe two kinds of *fuzzy association rules* called *structural* and *value* associations rules and then we propose several evaluation techniques in order to establish their degree satisfaction. In Section 4 we report the conclusion and draft some future work.

2 Related Work

In the context of semistructured data a main issue is related to the description of functional dependencies over XML documents. The information contained in an XML document could be partial and incomplete and moreover the document could be without a DTD (Document Type Definition): these specific features of XML documents make the description and the evaluation of functional dependencies more complex than in the context of relational databases. The possibility of missing information in the XML document can involve the violation of the required dependencies. The problem of describing functional dependencies for XML is still an open problem but in the literature there are some proposals which deal with this topic [5, 11, 12].

In [5] the authors have proposed the first definition of functional dependencies for XML documents, defining also a normal form, based on the proposed dependencies. In [12] the authors try to overcome the problems due to the nature of XML data and give a precise definition of functional dependencies without assuming the existence of a DTD. In [11] an XML-based language to define functional dependencies for XML documents have been proposed. In these approaches functional dependencies for XML are described in term of implication between paths (starting from the root) and their satisfaction is evaluated taking into account the reachable values (w.r.t. the considered paths).

Another research topic is related to the description of integrity constraints for semistructured data and XML documents. In [8] the need of a formal definition of integrity constraints have been highlighted and the most important categories of constraints for XML have been defined. In [7] the authors study absolute and relative keys for XML, and investigate their associated decision problems. They also propose a new key constraint language for XML which can handle keys with a complex structure. In general, a key is described by means of a path on a (sub)tree with a specific root. In [10] the authors adopt the formal definition of keys described in [7] and propose a technique to obtain a compact set of keys from an XML document.

A recent research direction in the context of XML data is related to the extraction of association rules from XML documents [6, 13, 14]. An association

rule describes the co-occurrence of data items in a large amount of collected data [4]. Rules are usually described by means of implication in the form $X \Rightarrow Y$, where X and Y are two arbitrary sets of data items such that $X \cap Y = \emptyset$. The quality of an association rule is described by means of two parameters: *support* and *confidence*. Support corresponds to the frequency of the set $X \cup Y$ in the dataset, while confidence corresponds to the conditional probability $p(Y|X)$, i.e. the probability of finding Y having found X. Several works deal with the problem of mining association rules in large databases [2, 3, 4].

With respect to the XML context, in [14] the authors show a technique which allows one to extract association rules by using XQuery. In [6] association rules for XML documents are described by introducing an XML-specific operator, called XMINE RULE, which is based on the use of XML query languages.

Our approach is included in the context related to the extraction of association rules but, unlike the previous mentioned proposals, it concerns the description of flexible association rules for XML. In [9] we have proposed a first approach for the description of association rules on XML documents; in this work we will introduce a set of different evaluating techniques for the evaluation of the proposed association rules by associating to each rule a similarity degree. The similarity degree we propose is quite different from the standard parameters used for association rules, such as support and confidence, but we think it is important to evaluate, in a flexible way, the satisfiability of an association rule on an XML document. The introduced flexibility is related to the usage of different evaluation techniques which will assign a satisfiability degree by focusing on different aspects related both to the structure and to the content of an XML document.

3 Fuzzy association rules

In this work we will consider association rules on XML documents; their evaluation will be realized by taking into account their graphical forms.

As proposed in [9], we choose to represent XML documents by means of *XML graphs*; in particular, XML elements are represented by means of nodes, and their containment relationships by means of non-labeled edges. We assume to have general XML documents, thus we allow the presence of mixed XML elements. The *XML graphs* are composed by three kinds of nodes: *simple*, *complex*, and *mixed* nodes.

- A *simple* node is a leaf and has a specific value.
- A *complex* node has at least one outgoing edge.
- A *mixed* node is a complex node with a specific value.

In Figure 1 is reported a well-formed XML document containing information about employees. Figure 2 shows the graphical form of the XML docu-

ment of Figure 1. In particular, it is possible to observe that the XML graph has a root, described by means of a complex node with name Person. The nodes Employee, Department, and Project are mixed nodes while the nodes Salary, Bonus, and Office are simple nodes. The Information nodes are complex nodes because they have no specific values.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Person>
  <Employee>  Annie Bown
    <Information>
      <Salary> 800 </Salary>
      <Bonus> 100 </Bonus>
    </Information>
  </Employee>
  <Employee>  Franck Copperfield
    <Salary> 750 </Salary>
  </Employee>
  <Department> Computer Science
    <Employee> Roger Moore
      <Information>
        <Salary> 800 </Salary>
        <Office> Room 72 </Office>
      </Information>
      <Project> Pr.204
        <Bonus> 200 </Bonus>
      </Project>
    </Employee>
  </Department>
  <Employee>  Marc Valjavec
    <Information>
      <Salary> 900 </Salary>
    </Information>
  </Employee>
  <Department> Scientific and Technologic
    <Employee> Ambeta McGee
      <Information>
        <Salary> 900 </Salary>
      </Information>
      <Project> Pr.405
        <Bonus> 200 </Bonus>
      </Project>
    </Employee>
    <Employee>  Jack Tonkov
      <Information>
        <Salary> 900 </Salary>
        <Office> Room 74 </Office>
      </Information>
      <Project> Pr.405
        <Bonus> 200 </Bonus>
      </Project>
    </Employee>
  </Department>
  <Employee> Susan Sarandom
    <GeneralInfo>
      <Phone> 033 58 56 942 </Phone>
      <WorkInfo>
        <Salary> 750 </Salary>
      </WorkInfo>
    </GeneralInfo>
  </Employee>
</Person>
```

Fig. 1. A well-formed XML document.

In our proposal, it is possible to describe two kinds of association rules on XML documents:

- *structural* associations, which allow us to evaluate the similarity of an XML document with respect to a proposed structure;
- *value* associations, which allow us to establish the similarity between the information contained in an XML document and a given request.

These kinds of associations over XML documents can be described by means of a logical notation:

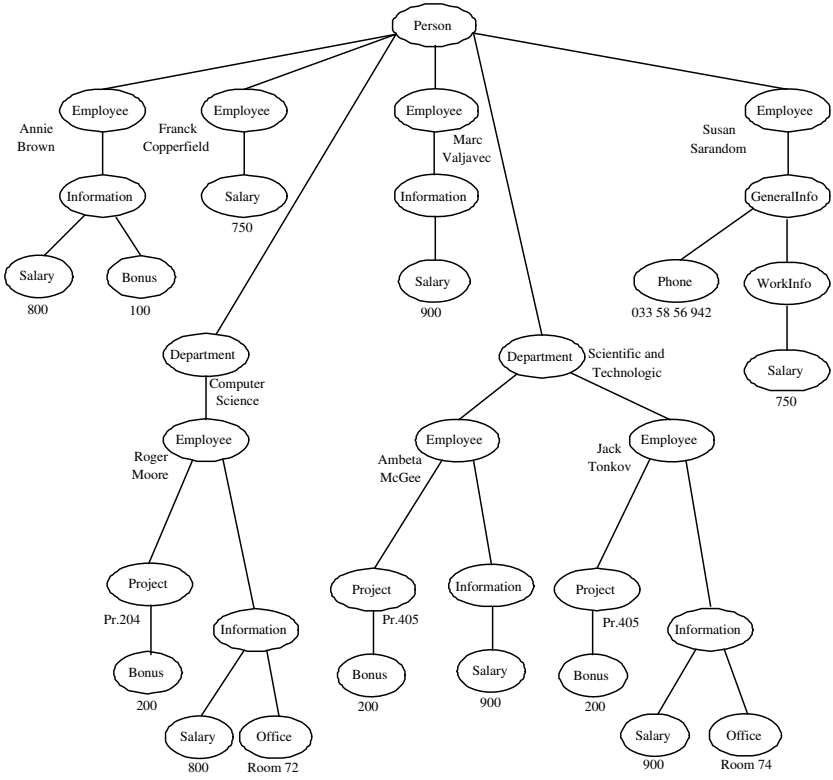


Fig. 2. The XML graph representing the XML document of Figure 1.

$$\mathsf{Elements_S} \rightarrow \mathsf{Element_D}$$

where $\mathsf{Elements_S}$ is called *starting element* and $\mathsf{Element_D}$ is called *destination element*. The logical notation describes the fact that in the XML document $\mathsf{Elements_S}$ and $\mathsf{Element_D}$ are related by means of a "relationships". In the case of *structural association* the required relationship takes into account the structure of the XML-subdocument (having as root the XML element $\mathsf{Elements_S}$), while in the case of *value association* the relationship takes into account the information contained in the XML-subdocument itself. In particular, it is possible to require that the destination element $\mathsf{Element_D}$ must have a specific value $\mathsf{Value_D}$ by using the logical notation:

$$\mathsf{Elements_S} \rightarrow \mathsf{Element_D}(\mathsf{Value_D})$$

We choose to evaluate association rules on the graphical form of an XML document, thus it is reasonable to consider also the graphical representation

of the association rules themself. The graphical representation of an association rule is a direct graph which describes the fact that the destination element, represented by means of a node called *destination node* (having as name $\mathsf{Element_D}$) can be reached by the starting element, represented by means of a node called *starting node* (having as name $\mathsf{Element_S}$) (Figure 3). In the following, the starting and destination node of an association rule will be denoted as $Node_S$ and $Node_D$ respectively. In the graphical form of value associations, the value related to the destination node is reported under the destination node (Figure 7).

3.1 Structural association rules

A *structural association* allows one to evaluate the similarity of an XML document with respect to a proposed structure. With respect to the XML document reported in Figure 1, a structural association rule could be:

$$\mathsf{Employee} \rightarrow \mathsf{Project}$$

The proposed association rule describes the fact that an employee is associated to a project. The graphical representation of this association is reported in Figure 3, and describes the fact that the node Project can be reached starting from the node Employee.

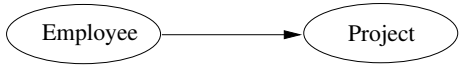


Fig. 3. The graphical representation of a structural association.

In order to establish the similarity degree of the XML graph with respect to the rule, it is possible to consider several kinds of *evaluation techniques*.

In the case of structural associations, the suitable techniques have to consider the structure of the XML graph; in particular, given a structural association rule, it is needed to check whether the XML graph satisfies the proposed structure in a more or less precise way. In this work we propose two kinds of *structural evaluation techniques*:

- *direct reachability*: it allows one to evaluate whether the starting node is directly connected to the destination node;
- *undirect reachability*: it allows one to evaluate whether the destination node can be reached from the starting node with a path composed by more than one edge.

The evaluation of a structural association rule over an XML graph can be realized with the following steps:

1. **subgraph extraction**: during this step, by means of suitable graph-search algorithms, the subgraphs having the starting node as root and containing the destination node are extracted.
2. **subgraph weight**: in the case of evaluation of a structural association rule it is possible to assign a weight to each extracted subgraph. The weight associated to the i-th extracted subgraph is denoted as ssd_i and it can be assigned by considering one of the different *evaluation techniques* proposed above.
   - Using the *direct reachability*, the weight associated to each subgraph will be 1 if the destination node is directly connected to the starting node (i.e. $Node_D$ is a child of $Node_S$) and 0 otherwise. In this case, the ssd_i can be calculated by means of the function *Direct-Reach*:

$$ssd_i = \textit{Direct-Reach}(Node_{S_i}, Node_{D_i}) = \begin{cases} 1 & \text{if } Node_{D_i} \text{ is a child of } Node_{S_i} \\ 0 & \textit{otherwise} \end{cases}$$

     where $Node_{S_i}$ and $Node_{D_i}$ are respectively the starting node and the destination node of the i-th retrieved subgraph.
   - Using the *undirect reachability*, the weight associated to each subgraph will take into account the number of edges needed to reach the destination node from the starting node. In this case, the ssd_i can be calculated by means of the function *Undirect-Reach*, which returns the number of edges between the starting node ($Node_{S_i}$) and the destination node ($Node_{D_i}$) of the i-th subgraph. The weight associated to a subgraph is in inverse proportion to the number of edges in the path between the starting node and the destination node and its precise value is:

$$ssd_i = \frac{1}{\textit{Undirect-Reach}(Node_{S_i}, Node_{D_i})}$$

3. **structural satisfiability degree**: in this step it is possible to calculate the *structural satisfiability degree* (*ssd*) of the XML graph with respect to the structural association. The value of *ssd* is in $[0, 1]$; the value 0 represents the fact that the XML graph does not satisfy the proposed structural association, while the value 1 describes the fact that the structural association is always satisfied in the XML graph, i.e. each extracted subgraph satisfies the proposed structure. Both in the case of *direct reachability* and in the case of *undirect reachability*, the *ssd* can be calculated with the formula:

$$ssd = \sum_{i=1}^{n} ssd_i \cdot \frac{1}{n}$$

   where n is the number of subgraphs extracted at step 1.

The evaluation of the structural satisfiability degree is determinated in the subgraphs which satisfy the required structure, i.e. in the subgraphs connecting in some way the destination node to the starting node. Thus, we do not compute the standard parameters such as *support* and *confidence* but it is possible to obtain them by counting the number of subgraphs having as root the required starting node and dividing this value for the number of subgraphs which satisfy, both in direct and undirect way, the required structure.

As an example, we consider the structural association Employee → Project of Figure 3 on the XML graph of Figure 2. At step 1, the graph-search returns the subgraphs which connect the node Employee with the node Project. In Figure 4 are shown the extracted subgraphs (included in dashed regions). In this case, both with *direct* and *undirect reachability* techniques, the value of *ssd* is 1. Indeed, in each extracted subgraph the node Project is directly connected to the node Employee; for these reasons, both the function *Direct-Reach* and *Undirect-Reach* assign value 1 to each extracted subgraph.

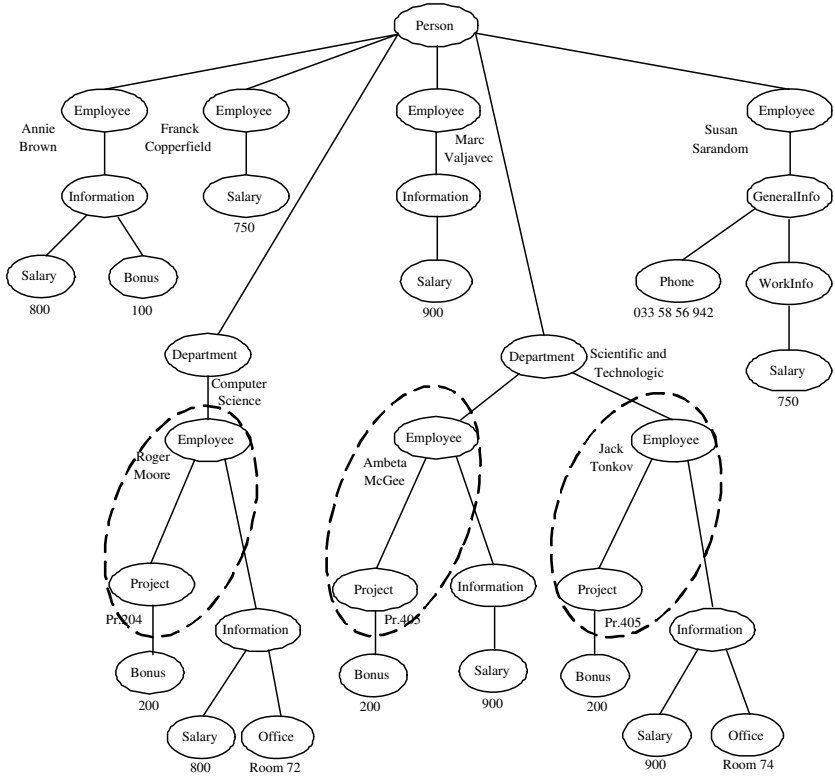


Fig. 4. The subgraphs which satisfy the structural association rule Employee → Project.

As another example, let us now consider the structural association rule Employee→ Salary of Figure 5, which describes the fact that an employee has a salary. In Figure 6 are shown, in dashed regions, the extracted subgraphs.

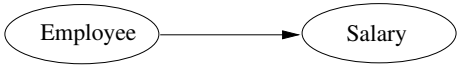


Fig. 5. The graphical representation of a structural association.

In order to evaluate the *ssd* of the XML graph reported in Figure 2, with respect to the proposed structural association (Figure 5), we suppose to apply the *direct reachability* technique. With this technique only the second retrieved subgraph (associated to the employee *Frank Copperfield*) has as weight 1. In all the other subgraphs the node Salary is not directly connected to the node Employee and thus the related weights are 0 according to the definition of the function *Direct-Reach*. In this example, the final value of *ssd* by using the *direct reachability* technique, is:

$$ssd = \sum_{i=1}^{n} ssd_i \cdot \frac{1}{n} = 1 \cdot \frac{1}{7} + 0 \cdot \frac{6}{7} = \frac{1}{7}$$

The value $\frac{1}{7}$ describes the fact that one subgraph on seven satisfies the proposed structural association, i.e. in one retrieved subgraph the node Salary is directly connected to the node Employee.

We now evaluate the structural association rule Employee → Salary (Figure 5) by using the *undirect reachability* technique. In this case, the weight associated to each extracted subgraph takes into account the number of edges between the nodes Employee and Salary. According to the function *Undirect-Reach*, the first subgraph (having value *Annie Brown* for the node Employee) has weight $\frac{1}{2}$ because the path between the considered nodes (Employee and Salary) has length 2. For the same reason also the third, fourth, fifth, and sixth subgraphs (associated to the employee *Roger Moore*, *Marc Valjavec*, *Ambeta McGee*, and *Jack Tonkov* respectively) have weight $\frac{1}{2}$. The second subgraph (having value *Franck Copperfield* for the node Employee) has weight 1, as just described in the application of the previous technique. Finally, the seventh subgraph (associated to the employee *Susan Sarandom*) has weight $\frac{1}{3}$ because the path between the considered nodes is composed by 3 edges. In this example, the final value of *ssd* by using the *undirect reachability* technique, is:

$$ssd = \sum_{i=1}^{n} ssd_i \cdot \frac{1}{n} = \frac{1}{2} \cdot \frac{1}{7} + 1 \cdot \frac{1}{7} + \frac{1}{2} \cdot \frac{4}{7} + \frac{1}{3} \cdot \frac{1}{7} = \frac{23}{42} = 0.548$$

The value 0.548 represents the structural satisfiability degree of the XML graph (Figure 2) with respect to the structural association Employee → Salary

(Figure 5). This value describes the fact that the extracted subgraphs do not respect in exact way the required structure, i.e. there are subgraphs where the paths between the considered nodes have length greater than 1.

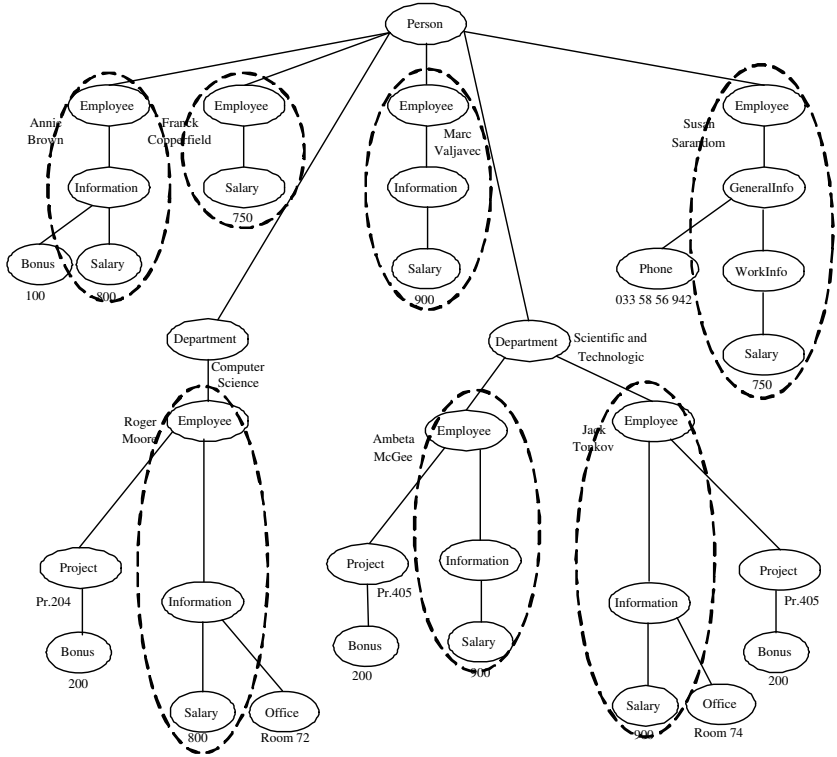


Fig. 6. The subgraphs which satisfy the structural association rule Employee → Salary.

3.2 Value associations

A *value association* allows one to check the similarity of the information contained in an XML document with respect to a proposed scenario. For example, a value association on the XML graph reported in Figure 2 could be Employee → Salary(900). The proposed association describes the fact that the employees have a salary with value 900; the graphical representation of the proposed value association is shown in Figure 7.

When we evaluate the satisfiability of a value association rule with respect to an XML graph, we return the *value satisfiability degree* (*vsd*) which can

Fig. 7. The graphical representation of a value association.

be calculated by using several evaluation techniques. In this work we propose the following *value evaluation techniques*:

- *exact value technique*: it allows one to check whether the values of the destination nodes, in the extracted subgraphs, are equal to the value required in the value association rule;
- *shift technique*: it allows one to associate a different weight to each extracted subgraph. The weight is calculated by taking into account the distance between the value of the destination node of each retrieved subgraph and the value required by the rule;
- *trend technique*: it allows one to check whether the value of the destination node of each retrieved subgraph and the value proposed by the value association rule are in a given relationship. The considered relationships are described by means of comparison operators in the set $\Theta = \{<, >, \leq, \geq\}$. Differently from the shift technique, the trend technique can be adopted also for checking textual data. In this case, we suppose to have a suitable set of comparison operators for the description of the lexicographical arrangement: in the following, with abuse of notation, we will use an operator $\theta \in \Theta$ both for textual and numerical comparisons.

The evaluation of a value association rule, over an XML graph, is realized with the same steps proposed for the evaluation of structural associations, with a difference in the step 2: in this case the subgraph weight is determinated by using one of the proposed *value evaluation techniques*.

- In the case of *exact value* technique, the weight associated to each subgraph will be calculated by using the function *Value*:

$$Value(Node_{S_i}, Node_{D_i}, val) = \begin{cases} 1 & \text{if } Node_{D_i} \text{ has value } val \\ 0 & \text{otherwise} \end{cases}$$

  If the destination node of the i-th subgraph is a complex node (without a specific value), the function *Value* returns 0.
- In the case *shift technique*, the function *Shift* allows one to evaluate the weight to associate to the i-th subgraph:

$$Shift(Node_{S_i}, Node_{D_i}, val,) = \begin{cases} \frac{val_{D_i}}{val} & \text{if } val_{D_i} \leq val \\ \frac{\lceil \frac{val_{D_i}}{val} \rceil \cdot val - val_{D_i}}{val} & \text{otherwise} \end{cases}$$

The function *Shift* returns value 1 if and only if the value of the destination node (val_{D_i}) is equal to *val*. In the other cases, it returns a value, in the interval [0,1], which is in direct proportion with val_{D_i} and in inverse proportion with *val*. For example, if the value required in the rule is 500 and the value retrieved in a subgraph is 250, then the value returned by the function *Shift* is 0.5; the same weight is assigned to a subgraph having value 750 (by applying the second equation of the definition).

- Finally, in the case of the *trend technique* it is possible to assign a weight 1 to the i-th subgraph if the value retrieved in its destination node (Val_{D_i}) satisfies a specific relationship with respect to the value required in the rule. In particular, if θ is a relationship ($\theta \in \Theta$), then the weight associated to the i-th subgraph can be calculated by means of the function *Trend*:

$$Trend(Node_{S_i}, Node_{D_i}, val, \theta) = \begin{cases} 1 & \text{if } Val_{D_i} \text{ is in } \theta\text{-relation with } val \\ 0 & \text{otherwise} \end{cases}$$

In the case of evaluation of a value association rule the structure of the extracted subgraphs is not taken into account.

We now consider some examples related to the evaluation of value association rules on the XML graph of Figure 2.

Given the value association rule Employee → Salary(900), shown in Figure 7, the related extracted subgraphs are shown, included in dashed regions, in Figure 6. We suppose to apply the *exact value technique* in order to evaluate the *vsd* of the XML graph with respect to the scenario given by the rule in Figure 7. The function *Value* will assign weight 1 to each subgraph having value 900 for the node Salary and 0 otherwise. Thus, the value satisfiability degree is:

$$vsd = \sum_{i=1}^{n} vsd_i \cdot \frac{1}{n} = 0 \cdot \frac{4}{7} + 1 \cdot \frac{3}{7} = \frac{3}{7} = 0.428$$

where n is the number of the retrieved subgraphs. Note that the first, second, third and seventh subgraphs have weight 0 because the values assumed by the nodes Salary are not 900, while the fourth, fifth and sixth subgraphs have weight 1. The final result describes the fact that three subgraphs (on seven) satisfy the proposed value association in an exact way.

We now apply the *shift technique* in order to evaluate the value association Employee → Salary(900) (Figure 7) on the XML graph shown in Figure 2. In this case, the function *Shift* will assign a weight in the interval [0,1] to each extracted subgraph (Figure 6) by taken into account the proportion between the required value and the retrieved one.

For example, as shown in Figure 6, in the first and third subgraph the value of the node Salary is 800 while the required value is 900. The weight assigned to these subgraphs, by applying the first equation of the function *Shift*, is:

$$\frac{800}{900} = 0.\overline{8}$$

The weight associated to the second and seventh subgraph is $0.8\overline{3}$, while the weight associate to the fourth, fifth, and sixth subgraph is equal to 1. Thus, the final value of *vsd* is:

$$vsd = \sum_{i=1}^{n} vsd_i \cdot \frac{1}{n} = 0.\overline{8} \cdot \frac{2}{7} + 0.8\overline{3} \cdot \frac{2}{7} + 1 \cdot \frac{3}{7} = \frac{58}{63} = 0.92$$

where n is the number of the retrieved subgraphs. Note that the value 1 for the *vsd* can be reached only when all the extracted subgraphs have the requested value (900) for the node Salary.

We now consider some examples of value association rules related to the usage of the *trend technique*. When we consider this kind of evaluation technique, we have to specify the relationship we want to check between the value of the destination node and the value required in the rule.

For example, if we evaluate the value association Employee $\rightarrow$ Salary(700) by using the relationship $\geq$ on the XML graph shown in Figure 2, then the final value of *vsd* is 1 because in each extracted subgraph (Figure 6) the value of the node Salary is greater than 700. In the case of the evaluation of the value association Employee $\rightarrow$ Salary(800) by using the relationship $\leq$, the final value of *vsd* is $\frac{4}{7}$; indeed the first, second, third and seventh subgraph have values no greater than 800.

4 Conclusions and Future Work

In this work we have propose a flexible approach to describe association rules on XML documents. The proposed association rules can assume two forms: *structural* and *value* associations. We have described a set of different evaluation techniques in order to establish the similarity degree of an XML document with respect to the proposed rules; the similarity degree of a rule has a meaning quite different from the traditional parameters used for classical association rule (such as support and confidence).

As future work we aim to study possible complex rules obtained by the composition of structural and value association rules. For example, it could be useful to describe a complex rule composed by value associations: in this case the complex rule could represent a *flexible functional dependency*, i.e. the satisfaction of the dependency is described by means of a parameter which reveals its satisfiability instead of a boolean answer.

References

1. S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web: from relations to semistructured data and XML.* Morgan Kaufman, 2000.

2. R. Agrawal, T. Imielinski, and A. Swami. Database mining: A performance perspective. *IEEE Transactions on Knowledge and Data Engineering*, pages 914–925, 1993.
3. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In Peter Buneman and Sushil Jajodia, editors, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, 1993.
4. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *Proceedings of the 20th International Confernece on Very Large Data Bases*, pages 478–499. Morgan Kaufmann, 1994.
5. M. Arenas and L. Libkin. A normal form for XML documents. In *Proceedings of the 21th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 85–96. ACM Press, 2002.
6. D. Braga, A. Campi, S. Ceri, M. Klemettinen, and P. Lanzi. Discovering Interesting Information in XML Data with Association Rules. In *SAC 2003*, pages 450–454. ACM, 2003.
7. P. Buneman, S.B. Davidson, W. Fan, C.S. Hara, and W.C. Tan. Reasoning about keys for XML. *Information Systems*, 28(8):1037–1063, 2003.
8. P. Buneman, W. Fan, J. Simn, and S. Weinstein. Constraints for Semi-structured Data and XML. *SIGMOD Record*, 30(1):47–55, 2001.
9. C. Combi, B. Oliboni, and R. Rossato. Towards association rules for XML documents. Technical Report RR 21/2004, Department of Computer Science, University of Verona, 2004.
10. G. Grahne and J. Zhu. Discovering approximate keys in XML data. In *Proceedings of the 11th International Conference on Information and Knowledge Management*, pages 453–460. ACM Press, 2002.
11. M. Li Lee, T.W. Ling, and W.L. Low. Designing Functional Dependencies for XML. In *Extending Database Technology*, pages 124–141, 2002.
12. M.W. Vincent and J. Liu. Functional dependencies for XML. In Y.Zhang X. Zhou and M.E. Orlowska, editors, *APWeb 2003*, volume 2642 of *LNCS*, pages 22–34, 2003.
13. J. W.W. Wan and G. Dobbie. Extracting association rules from XML documents using XQuery. In *DASFAA 2004*, pages 110–112, 2003.
14. J. W.W. Wan and G. Dobbie. Mining association rules from XML data using XQuery. In *DMWI 2004. To appear*, 2004.
15. J. Widom. Data management for XML-research directions. *IEEE Data Engineering Bulletin*, 22(3):44–52, 1999.

Plenary Talk

Ubiquitous Robot

Jong-Hwan Kim

Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology
johkim@rit.kaist.ac.kr.
http://rit.kaist.ac.kr

Abstract. In the near future human beings will be living in a ubiquitous world where all objects such as electronic appliances are networked to each other and robots will provide us with services of every variety by any device through any network, at any place anytime. However, current robots have a number of constraints to become wholly ubiquitous. This research attempts to eliminate the spatial limitations by introducing virtual robots into the physical world. Ubiquitous robot, Ubibot, is introduced to integrate three forms of robots: software robot (Sobot), embeded robot (Embot) and mobile robot (Mobot). Sobot is a virtual robot, which has the ability to move to any place or connect to any device through a network, in order to overcome the spatial limitation. It has the capability to interpret the context and thus interact with the user. Embot is embedded within the environment or in the Mobot. An Embot can recognize the locations of and authenticate the user or robot, and synthesise sensing information. Mobot provides integrated mobile services. The services that will be provided by Ubibot will be seamless, calm and context-aware.

This talk will address the basic concepts of Ubibot. A Sobot, called Rity, will be introduced in order to investigate the usability of the proposed concepts. Rity is a 3D synthetic character which exists in the virtual world, has a unique IP address and interacts with human beings through an Embot implemented by a face recognition system using a USB camera.

1 Introduction

The term Ubiquitous Computing, UC, was first coined by Mark Weiser [1]. The basic concepts include the characteristics, such as every device should be networked; user interfaces should operate calmly and seamlessly; computers should be accessible at anytime and at any place; and ubiquitous devices should provide services suitable to the specific situation such as location, equipment, ID, time, temperature and weather. He also addressed the evolution of computer technology in terms of the relationship between the technology and humans [2, 3]:

J.-H. Kim: *Ubiquitous Robot*, Advances in Soft Computing **2**, 451–459 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

- The first generation - the Mainframe Era, where a large elaborate computer system was shared by many terminals;
- The second generation (still current) - the Personal Computer era where a human uses a computer as a standalone or networked system, in a work or home environment; and
- The third generation - the ubiquitous computing era, where humans use various networked devices which pervade their environment unobtrusively.

Along with the ubiquitous revolution, robotics is also undergoing a paradigm shift. The first generation of robotics was dominated by industrial robots followed by the second generation in which personal robots became widespread. As a third generation, ubiquitous robot can be thought of. Figure 1 shows the comparison of the paradigm changes between the personal robot and ubiquitous robot eras. The personal robot era is based on individual robot systems. However, in the future multiple robot system will prevail.

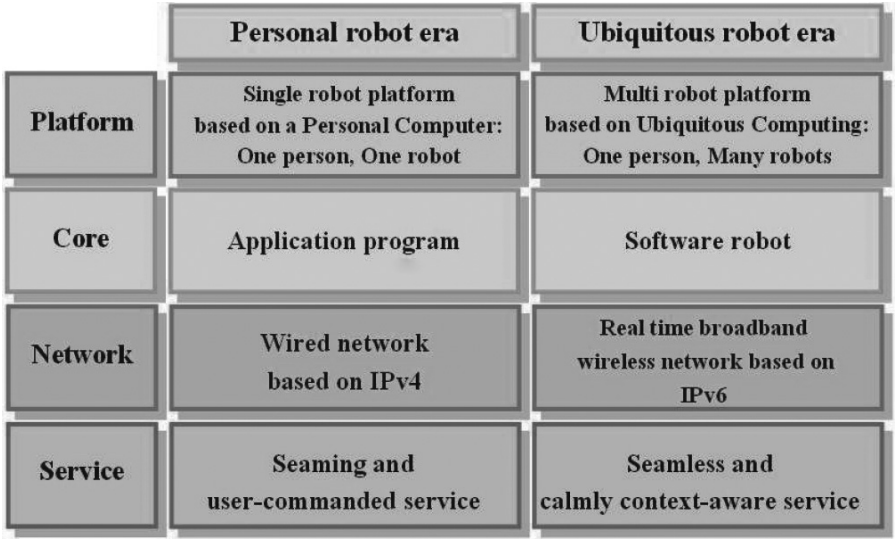


Fig. 1. Comparison between the personal robot and ubiquitous robot eras

Within the ubiquitous environment, a number of robots, such as Embots, Sobots, and Mobots, will provide a human with various services. In the current level of second generation robotics, the application software controlling the robots occupies the core. However, in the future, software robots will form the core and control the hardware robots which will then take on the more practical roles.

The intelligence of personal robots still mainly relies on a user-directed service system, which means it functions at a very low level. Therefore, when a user gives a command, s/he must wait until the robot understands and

interprets the command, then acts on it. Third generation ubiquitous robots will be able to understand what the user needs, wants or prefers and supply continuous and seamless service. This technology will be made possible by the use of IPv6 format and broadband wireless network technology.

The Ubibot (Ubiquitous Robot) has been developed based on UC and robot technology [4]. For humans, the future will present a ubiquitous world where all objects and devices are networked. In ubiquitous space, u-space, a Ubibot will provide the user with various services anytime, at any place, by any device, through any network. As demonstrated in the general concept of UC, Ubibot will be seamless, calm, context-aware and networked. Ubibot could be classified as three integrated robot systems: Sobot, Embot and Mobot. Sobot, the Software Robot, can be transmitted and connect to any device, at any time and any place. It is context-aware, and will automatically and calmly provide continuous cooperation with the user. Embot, the Embedded Robot, is embedded within the environment of a Mobot. It detects the location of the robot or the user, recognizes and authenticates them, and collects and synthesises the various sensing information. Mobot, the Mobile robot, provides general users with integrated services.

In this talk, Rity, developed at KAIST's RIT laboratory will be introduced as an example of a Sobot. Rity is a 3D virtual pet [5]. It has its own unique IP address and communicates and interacts with the user in the real world through an Embot implemented by a physical device such as a USBconnected camera.

2 Ubiquitous Robot: Ubibot

Ubibot is created and exists within ubiquitous space (uspace) which will be developed as an essential component of Ubibot.

2.1 U-space and UbiBot

It is anticipated that, in the future, the world will consist of numerous u-spaces, where each u-space will be based on the IPv6 or similar system and be connected to each other through broadband, wired or wireless, networks in real-time, Figure 2 [6]. A robot working within the u-space is defined as a Ubibot, which, in other words, can be used for any service through any terminal and any network by anyone at anytime and anywhere in a u-space.

Ubibot exists within u-space, Fig.3, and consists of both software and hardware robots. Sobot is a type of a software system whereas Embot and Mobot belong to a hardware system. Embots are located within the environment, human or otherwise, and are embedded in many devices. Their role is to sense and communicate with other Ubibots. Mobots are mobile. They can move both independently and cooperatively, and provide practical services. Each ubibot has specific individual intelligence and roles, and communicates

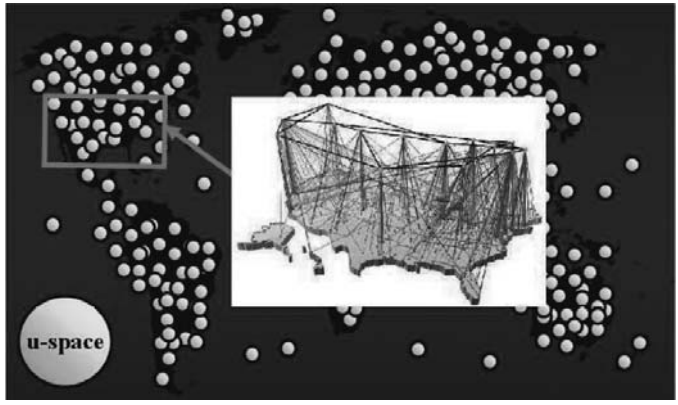


Fig. 2. Real world composed of billions of u-spaces

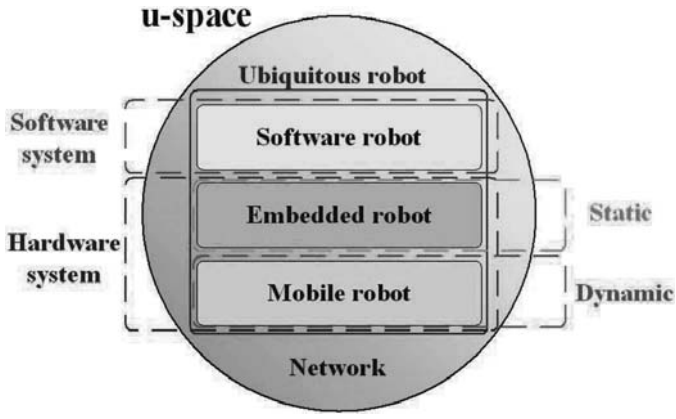


Fig. 3. Ubiquitous robot in u-space

information through networks. Sobot is capable of operating as an independent robot. However, it can also become the master system, which controls other Sobots, Embots and Mobots residing in other platforms as slave units.

2.2 Software Robot: SoBot

Because Sobot is software-based, it can move within the network and connect to other systems anytime and at any place. It can assess situations and interact with the user seamlessly. Sobot can be implanted into any environment and also other robots as a core system. It can control or, at an equal level, cooperate with Mobots. It can operate as an individual entity, without any help from other Ubibots. Figure 4 shows three main characteristics of Sobot: context-aware self-learning, context-aware intelligence and calm and seamless interaction [7, 8].

Fig. 4. Essential characteristics of SoBot

Context-aware self-learning

Sobot is autonomous and can determine and control its behaviour without external commands. Within the u-space, it can represent the user in behaviour and communication with others. Sobot also has the ability to "learn." It can assimilate objects, motions and situations. This "learning" process can be continuously developed. When a new user is introduced, Sobot may need to adjust behaviours within the u-space.

Context-aware intelligence

Sobots are proactive software robots. They operate with clear goals and well-constructed plans. The plans are constantly reviewed. When new situations arise, Sobots can adjust and react to the situation rather than use a predetermined reaction to a possible, or expected, situation. Sobots are able to demonstrate rational behaviour in the fact that they do not just "repeat" the same tasks, but can adjust to each task as necessary and produce a suitable outcome. Sobots remain context-aware at all times. They can modify and adapt themselves to the context, whenever needed. When they perceive a problem, they can locate and recognize the user automatically. By observing the user and user behaviour, they "learn" about the user and can easily adapt themselves to the user's preferences and interests.

Calm and seamless interaction

All activities are implemented calmly and seamlessly. Each Sobot is distributed and independent. A Sobot can be embedded and work inside an Embot or a Mobot. However, its behaviour is often limited by the resources of the software or hardware in which it is embedded.

Sobots can communicate with the environment and other Ubibots. There will be a higher level communication language, which need not have a predefined message or set of rules for the communication. Ubibot is omni-present. It exists everywhere in the u-space and will provide us with seamless services, at any time and at any place. These continuous and seamless services are performed through the network connected to many other devices. Sobot has continuous interface between the physical world and the virtual world.

The interaction mode used for Sobots is multi-modal. This will allow for greater convenience and flexibility in user interactions and communications. Because of the complexity of such a system, it is demanded to have compatible application environment for Sobot. This will allow many different types of robots to communicate, to continuously develop new functions and to control the robots remotely and cooperatively.

2.3 Embedded Robot: EmBot

EmBot is implanted in the environment or Mobots. In cooperation with various sensors, Embot can detect the location of the user or a Mobot, authenticate them, integrate assorted sensor information and understand the environmental situation. An Embot may include all the objects which have network and sensing functions, and be equipped with microprocessors to control Sobots. Embots generally have three major characteristics: calm sensing, ucommunication and information processing, Figure 5 [9].

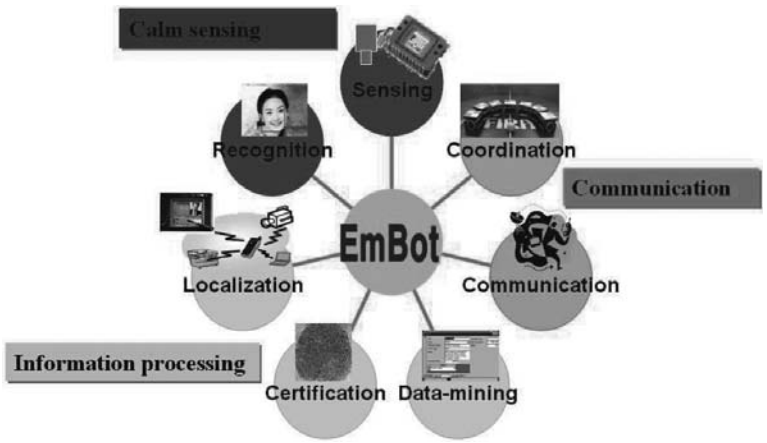


Fig. 5. Essential characteristics of EmBot

Calm sensing

Using various sensors, an Embot can detect and recognize objects even in crowded areas. This function works in cooperation with many other sensors embedded in humans, Mobots and other devices. Embots can calmly and unobtrusively sense human behaviour, status, preferences, relationships and neighbouring environments.

Embots recognize the patterns of human behaviour and status. Embots can also recognize environmental elements such as weather, time and climate which can impact on the daily behaviour and status of a human. They can also identify the different distances that exist between humans and may affect their behaviours. The distances are classified as contact distance, individual distance, social distance and public distance. Embots can identify human relationships, such as the social relationships between family members and visitors; friendship or personality. Embots can perceive the interactions between other humans and the main user. For example an Embot will know who is pointing, facing or gazing at the user.

Information processing

Embots can identify the location of Mobots, humans, objects and other environments. Embots can also assess and authorize a human to use a robot, authenticate various types of robots and robot users. Embots also possess data-mining abilities. They collect information about human behaviours, status and environment. Data created through the datamining process can be used to enhance the information search process.

Communication

Embots provide the user with assistance in various forms including: voice communication, behaviour, information transfer and motion guidance. This is done in accordance with the situation and through a network, in order to augment the communication with humans. Embots also facilitate and help multiple robots to cooperate, by using calm sensing and information processing. They can also provide services to hundreds of people through the sensor network.

Embots can recognize the location of humans and robots, behaviour of humans and the adjacent environment automatically. This is done using sensors embedded in the environment together with Embots' ability to certify, store and use previously gathered information. These features will allow easy communication between robot and humans and will be applied widely in homes, offices, and government buildings. This will also improve security and information searching systems, human research management and public resource management.

2.4 Mobile robot: Mobot

Mobot offers both essential services for general users and specific functions which are dedicated to the u-space. On the other hand, Embot is generally considered to have limited functions and resources. However, a Mobot will provide mobility and service; operates in the u-space; has Embot functions, Fig 6; and works together with Sobots neighbouring Embots.

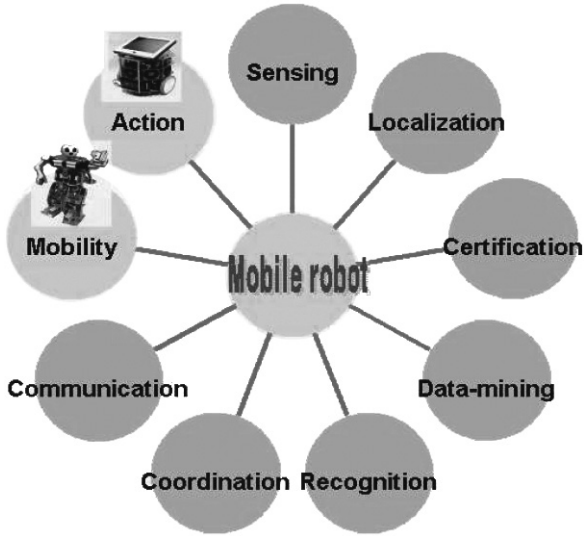


Fig. 6. Essential characteristics of mobile robot

Figure 6 shows the characteristics of Mobots, where mobility can be implemented in various types such as wheel and biped types, and action provides personal, public, and field services using manipulator and various devices.

Mobot communicates with Sobot in order to provide practical services based on information given by Embot. Mobot will be implemented as a multipurpose service robot in the u-space such as home, work, public organization, amusement, traffic and public facilities.

3 Conclusion

In considering the Ubibots of the future, it is anticipated that, Sobots, Embots and Mobots will co-exist with humans and will provide us with seamless, calm and context-aware services anytime and at any place through the ubiquitous network. Based on their own functions and intelligences, each Sobot, Embot and Mobot will have an individual role, as well as working cooperatively with other Ubibots. In the new ubiquitous era, our future world will

be composed of millions of u-spaces, each of which will be closely connected through ubiquitous networks. The future of ubiquitous robotics has a lot to offer. However, in order to ensure efficient and effective communication and smooth integration of robot systems it is essential to provide a standardised language and a uniform set of protocols.

Acknowledgement

This work was supported by the ITRC-IRRC (Intelligent Robot Research Center) of the Korea Ministry of Information and Communication in 2003.

References

1. Weiser, M., "Some computer science Problems in ubiquitous computing," Communications of ACM, Vol. 36, No.7, pp. 75-84, Jul, 1993
2. Weiser, M., http://www.ubiq.com/hypertext/weiser/ UbiHome.html
3. Weiser, M., "The computer for the 21st century," Scientific American, Vol. 265, No. 3, pp. 94-104, Sep., 1991.
4. Kim, J.-H, "IT-based UbiBot," (theme special lecture), The Korea Electronic Times, May 13, 2003.
5. Kim, Y.-D. and Kim, J.-H., "Implementation of Artificial Creature based on Interactive Learning," FIRA Robot World Congress, pp. 369-374, May 2002.
6. Ha, W.K., Kim, D.H., and Choi, N.H., "The Ubiquitous IT Revolution and the Third Space," The Korea Electronic Times, 2002.
7. Jennings, N. and Wooldridge, M., IEE Review, Volume: 42, Issue: 1, pp. 17-20, Jan. 1996.
8. Wooldridge, M., Software Engineering. IEE Proceedings, Volume: 144, Issue: 1, pp. 26- 37, Feb. 1997
9. ATR, http://www.ktab.go.jp/new/16/0413.htm

Fuzzy Image Processing

Combining Fuzzy Logic and Kriging for Image Enhancement

Anwar Majeed Mirza and Badre Munir

Faculty of Computer Science and Engineering, GIK Institute, Topi-23460, Swabi, Pakistan.
{mirza, badr}@giki.edu.pk

Abstract. We propose a fuzzy logic based punctual kriging technique for enhancing images corrupted by Gaussian noise. Punctual kriging is used to generate kernel weights employing the semivariances in the neighborhood of a pixel and empirically determined global semi-variogram. Semivariance is a measure of the degree of spatial differences between samples (pixel values). Superiority of kriging over other methods for noise cancellation in 1-D signals has been established. A quantitative analysis of the kriging technique, for image enhancement as compared to the Wiener filter shows that kriging performs inferior to Wiener filtering for image enhancement. We have proposed a new fuzzy logic based method which substantially improves the performance of the kriging for image enhancement. Experimental results are presented to illustrate the improvement in the results and the effectiveness of the new technique.

1 Introduction

Kriging is an interpolation technique, named after its developer, D. G. Krige [4]. The method has its origin in Geostatistics. Kriging has proven to be a nonlinear predictor in signal processing.

Costa et al [1] suggest kriging as an efficient tool for nonlinear filtering. They have shown through various simulated examples, that kriging predictor gives better results than the Volterra [8] and NARMAX [5] predictors on a nonlinear noise-cancellation problem in underwater acoustics.

Pham and Wagner [7] have used kriging along with fuzzy sets for Image Enhancement of images corrupted by Gaussian noise. They have modeled *soft-thresholding* by fuzzy sets. In their method, the pixel value in the processed image is a weighted sum of two values: the original (noisy) and the estimated (by kriging). The weighting is done using two fuzzy sets having S-shaped membership functions. The overlap between the membership functions of the two sets is tuned by a parameter β which can be var-

A.M. Mirza and B. Munir: *Combining Fuzzy Logic and Kriging for Image Enhancement*, Advances in Soft Computing **2**, 463–473 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

ied between 0 and 1. The authors [7] do not present any quantitative analysis of their scheme.

A number of image processing problems including image enhancement have been solved by employing fuzzy technique. Farbiz and Menhaj [3] have introduced an approach of image filtering based on fuzzy logic control. They have shown it to remove impulsive noise, smooth out Gaussian noise while, simultaneously, preserving image details and edges efficiently.

The remaining of this paper is organized as follows. Section 2 deals with kriging along with its mathematical formulation. Semi-variograms which are at the heart of kriging are explained, followed by a brief derivation of kriging equations. A discussion on the quantitative results of implementing the suggested method by Pham and Wagner is presented in section 3, along with observed problematic issues. Section 4 discusses the problems of Pham and Wagner's method and explains the proposed method of combining fuzzy logic with kriging. The results of the proposed method are presented in section 4.1.

2 Kriging

Kriging gives the *best linear unbiased estimate* of an unknown point on a surface [11, 6]. This estimate is the weighted sum of the known neighboring values around the unknown point. The weights are calculated as to minimize the variance of the estimation-error. To achieve this, kriging uses the semi-variogram, a related concept from Geostatistics. Based on the semi-variogram model chosen, known values are assigned optimal weights to calculate the unknown value.

2.1 Semi-variograms

Given a set of samples, the semivariance can be simply defined as half the variance, of the differences between all possible samples, spaced a constant distance 'd' apart. Semivariance gives a measure of spatial dependence between samples. The magnitude of the semivariance between two points depends on the distance between them. The plot of the semivariance as a function of distance is referred to as a semivariogram or a variogram. The semivariance at d=0, is zero [1]; as there are no difference when points are compared to themselves.

The experimental variogram [1, 6] for distance 'd' can be estimated by Eq. 1

$$\gamma(d) = \frac{1}{2N(d)} \sum_{i=1}^{N(d)} (z_i - z_{i+h})^2 \tag{1}$$

In Eq. 1, z_i is the sample value at location *i*, z_{i+h} is the sample value at a distance 'd' from *i*. *N(d)* is the total number of sample-pairs in the available data that are separated by distance 'd'

2.2 Kriging Process

Kriging uses the semi-variogram to calculate estimates. The first step in kriging involves finding a semi-variogram for the data at hand. Based on this semi-variogram, the set of optimal weights for estimating the unknown values are found. These weights give the best linear unbiased estimate of the unknown values [1].

Let z be the actual value at a point and $\hat{z}$ be the estimate of this value. The error e is then:

$$e = z - \hat{z} \tag{2}$$

And $\hat{z}$,the *kriging estimate*, is given by:

$$\hat{z} = \sum_{i=1}^{n} w_i z_i \tag{3}$$

In Eq. 3 w_i are the *kriging weights* and the z_i are the neighboring known (sampled) values around z.

The variance of the error can be worked out to be:

$$\begin{aligned} var(e) &= \mathrm{var}(z - \hat{z}) \\ &= var(z) - 2 \cdot \sum_j w_j \cdot \gamma(d_j) + \sum_j \sum_k w_j w_k \gamma(d_{jk}) \end{aligned} \tag{4}$$

In Eq. 4, d_j is the distance between current point and its neighbor 'j'. And d_{jk} is the distance between neighbors 'j' and 'k' of the current point.

To make the estimate unbiased, method of Lagrange multipliers is used to minimize *variance of error* by the unbiased-ness condition specified by Eq. 5.

$$\sum_{i=1}^{n} w_i = 1 \tag{5}$$

Differentiation of the Lagrange function obtained by combining Eqs. 4 and 5 gives a system of equations that can be written in matrix form [A]{w}={b} as:

$$\begin{pmatrix} \gamma(d_{11}) & \gamma(d_{12}) & \cdots & \gamma(d_{1n}) & 1 \\ \gamma(d_{21}) & \gamma(d_{22}) & \cdots & \gamma(d_{2n}) & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma(d_{n1}) & \gamma(d_{n2}) & \cdots & \gamma(d_{nn}) & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \\ \lambda \end{pmatrix} = \begin{pmatrix} \gamma(d_1) \\ \gamma(d_2) \\ \vdots \\ \gamma(d_n) \\ 1 \end{pmatrix} \tag{6}$$

In Eq. 6 'n' is the number of neighboring samples of the current unknown point used in the estimation process. The above matrix '**A'** is symmetric with a zero diagonal. The elements of this matrix are taken from the semi-variogram.

Solving Eq. 6, yields $\{w_1, w_2 \ldots w_n\}$, which are the optimal *kriging weights* to calculate the *kriging estimate* $\hat{z}$ by Eq. 3.

3 Issues in Pham and Wagner's Method

3.1 Quantitative Analysis of Pham and Wagner's method

This section presents the quantitative results of Pham and Wagner's method [7] and discusses the observed problematic issues in this method.

Instead of the peak signal-to-noise ratio (PSNR), a recent metric by Wang et al. [10] has been employed for measuring quality of the processed images. This metric is called the Structural Similarity (SSIM) Index. The authors have experimentally shown that, unlike PSNR, the SSIM Index is highly consistent with the Human Visual System. Given the original image and degraded test image, the SSIM Index returns a value between 0 and 1. Values of SSIM Index close to 1 indicate that the test image is close to the

original image and vice verse. Wang et al. also demonstrate that PSNR is a bad measure of image quality.

The kriging based technique of Pham and Wagner [7] was implemented in Matlab. Degraded versions of the 'Blood-cells' image (shown in Fig. 1), were used to compare the performance of Wiener filter and Pham and Wagner's method. The degraded versions were obtained by corrupting the 'Blood-cells' image with zero-mean Gaussian noise. For the degraded versions, the variance of the Gaussian noise was varied between 0.01 and 0.1. The degraded versions were enhanced by both the Wiener filter and Pham and Wagner's method. The SSIM index values of the enhanced images are shown in Table 1. Indeed, Wiener filtering is much ahead of Pham and Wagner's method for all cases. The parameter β in Pham and Wagner's method was set to 0.5.

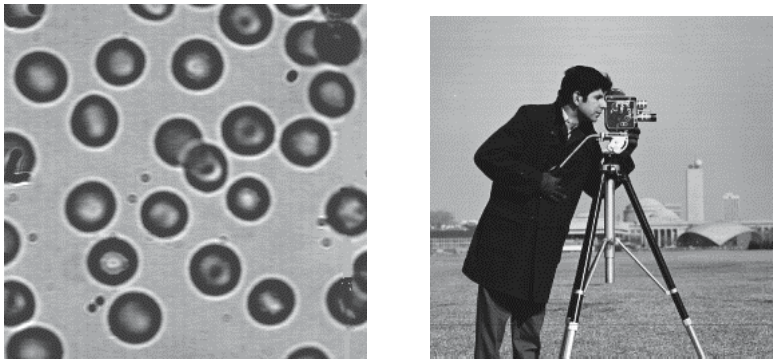


Fig. 1. Images used in the current work: Blood-cells and Cameraman

3.2 Discussion

This section details the observations made about the implementation and behavior of Pham and Wagner's method.

First, Pham and Wagner have chosen to use the experimental semi-variogram (obtained by Eq. 1) instead of one of the model semi-variograms: Linear, Spherical, Exponential and Gaussian. They claim to have found that using these models over-smoothes the data and thus advocate using the experimental semi-variogram.

Second, the set of equations for determining the *kriging weights* cannot always be solved. For a pixel, its 3x3 neighborhood is used to find its *kriging estimate*. For this case, matrix 'A' in Eq. 6 is a 9x9 matrix. Indeed,

it has been found that, at times, this matrix comes up as *singular* and thus kriging cannot be used for such pixels.

Table 1. SSIM Index values: Enhancement of the Blood-cells image by Wiener Filter and Pham & Wagner's Method.

Noise Variance	Noisy Image	Wiener Filter	P&W 's Method
0.010	0.4091	0.7236	0.4548
0.025	0.2790	0.5703	0.3201
0.050	0.1981	0.4584	0.2314
0.075	0.1558	0.3917	0.1829
0.100	0.1352	0.3633	0.1587

P&W Pham and Wagner's Method

Third, kriging is a computationally expensive procedure [6]. Because the optimal *kriging weights* are determined for each pixel, the system of equations in Eq. 6 must be solved for every pixel. However, it was observed that there are pixels for which the *kriging estimate* is not very different from the *mean* of the neighboring values. Indeed, blindly kriging all pixels is quite inefficient. A better approach would be to selectively perform kriging only on pixels where it may give better estimates than the mean.

4 The New Approach

In this section, a brief explanation of the observed issues in section 3.2 is followed by a description of the motivation behind combining kriging with fuzzy logic.

First, it must be noted that the experimental semi-variogram is just an *estimate* of, the true semi-variogram of the given data [6]. One reason may be that the semivariance is *estimated* by taking it simply a function of the distance between sample points. The *estimation* of the semi-variogram itself, inherently introduces an element of uncertainty in the overall processing. Fuzzy logic is well-suited for dealing with such problems. Several traditional image processing techniques have been extended with fuzzy logic giving much improved results, without adding to the complexity of implementing the overall process. In particular, Farbiz and Menhaj [3] have demonstrated the use of fuzzy logic to *simultaneously* take care of the conflicting tasks of noise-smoothing and edge preservation.

Second, the situation of singular kriging matrix is inherently *unpredictable* as it depends on the semi-variogram. But the semi-variogram itself depends on neighboring values of a pixel. Such scenarios could be taken care of by processing such pixels with by 'averaging' or 'median' –filter, which ever makes the error variance *small*.

Third, the high computational cost of kriging. It should be noted that the primary use of kriging was for estimating values of an un-sampled points on a geological terrain [6]. The reason for not sampling some points could be inherent practical limitations in the sampling process. In digital images, however, this is not the case. However, for Image enhancement blindly kriging all pixels may be not needed at all. It was observed that highly homogenous (smoother) regions of the image are enhanced more or less the same by kriging or simple averaging (mean).

It also follows that the decision of when to use kriging or not could be related to the smoothness or homogeneity of a region in the image. If a region in the image is *highly* homogeneous, then averaging would give almost the same results as kriging but with much less computations. The qualitative concept *highly-homogeneous* is best described using fuzzy logic. Thus, it seems sensible to have a fuzzy logic rule-based system to decide when to perform kriging.

The above suggested solutions are best described qualitatively. This human linguistic reasoning is efficiently and elegantly captured by a fuzzy logic rule-based system. The next section proposes a method that combines kriging with fuzzy logic, based on the above discussion.

4.1 The Proposed Method

In the proposed method, all pixels are not blindly kriged. Rather, based on the homogeneity of its local neighborhood, a pixel is selected for kriging by a fuzzy logic rule-based system. This fuzzy system is called the *Fuzzy Decider* in our work. The inputs to the *Fuzzy Decider* are: a measure of homogeneity, mean and variance, of the 3x3 window of the current pixel. The *degree of homogeneity* is estimated by Eq. 7, discussed by Tizhoosh in [9]. The numerator in Eq. 7 is the difference of the maximum and minimum gray values in the region comprised of the 3x3 local window around a pixel where as the denominator is the difference of the maximum and minimum gray values in the whole image.

$$\mu_H = \frac{g_{\max}^{local} - g_{\min}^{local}}{g_{\max}^{global} - g_{\min}^{global}} \tag{7}$$

The *Fuzzy Decider* is a basic *Mamdani-type* fuzzy logic system comprised of the rules given in Fig. 2 The *DAMdistance* in the rules is simply the difference between the gray value of current pixel and the mean gray value of its neighbors. *Homogeneity* in the rules is the value calculated by Eq. 7

If *Homogeneity* is *High* or *DAMdistance* is *Acceptable*
then Do Not Perform Kriging

If *Homogeneity* is *Low* or *DAMdistance* is *Very-High*
then Perform Kriging

Fig. 2. Rule base of the Fuzzy Decider. The terms in italics are explained in section 4.1

The proposed method has three stages. In the first stage, the noisy image is presented to the *Fuzzy Decider* which generates a binary image called the *decision map*. For a given pixel position in the decision map, a value of 1 indicates 'Perform Kriging' and 0 indicates that the *Fuzzy Decider*'s verdict is 'Do Not Perform Kriging'. The decision maps for the noisy versions of the Cameraman image are shown in Figs. 3 and 4, with noise variance set to 0.02 and 0.06 respectively. White dots in Figs. 3 and 4 correspond to pixels that have been selected by the *Fuzzy Decider* for kriging. As shown in Fig. 3, few pixels are selected for kriging in the highly homogenous region of the *Black Coat*. As increasing the Gaussian noise variance causes increased degradation. For such an image, much more pixels are selected for kriging by the *Fuzzy Decider* in the originally homogeneous region of the *Black Coat* of the Cameraman. This is evident in Fig. 4.

In the second stage, an attempt is made to find a *kriging estimate* for pixels selected by the *Fuzzy Decider*. If the attempt fails, the original pixel value is taken as the processed final value. The attempt to find a *kriging estimate* was found to fail due to two broad reasons: *singular kriging matrix* and *negative-weights*. For pixels that were rejected for kriging by the *Fuzzy Decider*, the *mean* of 3x3 neighbors is taken as the processed value. After this stage the processed image contains two types of value based on the decision map: *kriging estimate* and *mean value*. In the third and final stage 3x3 averaging is applied, on the image produced by the second stage, to yield the final processed image

The results of the above proposed scheme are shown in Table 2 and Table 3 for the images Blood-cells and Cameraman, respectively. The noise variance was varied between 0.01 and 0.1. Clearly, the proposed scheme performs better than the Wiener Filter for both images.

Fig. 3. Decision map for Cameraman image degraded with Gaussian noise having variance = 0.02.

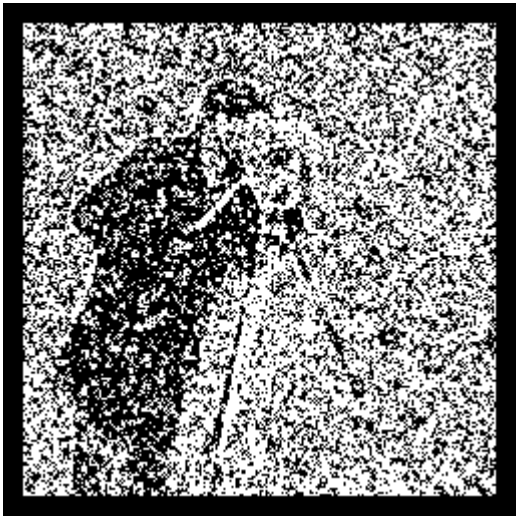


Fig. 4. Decision map for Cameraman image degraded with Gaussian noise having variance = 0.06.

Table 2. SSIM Index values: Enhancement of the Blood-cells image by Wiener Filter and the Proposed Method.

Noise Variance	Noisy Image	Wiener Filter	Proposed Method
0.010	0.4103	0.7240	0.8006
0.025	0.2783	0.5688	0.6736
0.050	0.1983	0.4549	0.5652
0.075	0.1597	0.4023	0.5053
0.100	0.1343	0.3581	0.4575

Table 3. SSIM Index values: Enhancement of the Cameraman image by Wiener Filter and the Proposed Method.

Noise Variance	Noisy Image	Wiener Filter	Proposed Method
0.010	0.3422	0.6325	0.6797
0.025	0.2343	0.4619	0.5273
0.050	0.1698	0.3425	0.4026
0.075	0.1386	0.2871	0.3372
0.100	0.1190	0.2549	0.3072

5 Conclusions

A scheme has been presented for enhancing images corrupted by zero-mean Gaussian noise. Punctual or Ordinary Kriging has been used to enhance Gaussian noise corrupted Images. The proposed method suggests a scheme for combining Fuzzy Logic with Kriging. For quantitative analysis a recent metric for measuring image qualtiy, called the SSIM Index has been employed. The SSIM image quality index is highly consistent with the Human Visual System. The experimental results show that the new proposed method performs better than Wiener filter for enhancing images corrupted by zero-mean Gaussian noise.

References

1. Clark I (1979) Practical Geostatistics. Applied Science, London
2. Costa JP, Pronzato L, Thierry E (2000) Nonlinear prediction by kriging, with application to noise cancellation. Signal Processing 80:553–566
3. Farbiz F, Menhaj MB (2000) A fuzzy logic control based approach for image filtering. In: Fuzzy Techniques in Image Processing Vol 52, Studies in Fuzziness and Soft Computing. Springer-Verlag, New York, pp. 194–221.
4. Krige D (1951) A Statistical approach to some mine valuation and allied problems on the Witwatersrand. Master Thesis, University of Witwatersrand
5. Leontaritis I, Billings S (1985) Input-output parametric models for nonlinear systems part 2: stochastic nonlinear systems. Int J Control 41(2):329-344
6. Naser El-Sheimy (1999) Digital Terrain Modeling, ENGO 573. The University of Calgary, Canada.
7. Pham TD and Wagner M (2000) Image Enhancement by Kriging and Fuzzy Sets. Int J Pattern Rec and Artificial Intelligence 14(8):1025-1038
8. Rugh WJ (1981) Nonlinear System Theory: The Volterra/Wiener Approach. The Johns Hopkins University Press, Baltimore
9. Tizhoosh H (2000) Fuzzy Image Enhancement: An Overview. In: Fuzzy Techniques in Image Processing, Studies in Fuzziness and Soft Computing, vol 52. Springer-Verlag, New York, pp. 137–171.
10. Wang Z, Bovik AC, Shiekh HR, Simoncelli EP (2004) Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Trans on Image Processing 13(4):600-612

Optical Quality Control of Coated Steel Sheets Using Fuzzy Grey Scale Correlograms

Dirk Lühning

University of Dortmund, Department Computer Science I, D-44221 Dortmund
Dirk.Luehning@udo.edu

Abstract. In this paper a method for optical quality control of coated steel sheets is proposed. A frequently used test for examining coating quality is the impact resistance test which results in a rapid deformation of a steel sheet sample. The quality of the coating with respect to adhesion can then be rated from an image of the deformed region by examination of the texture on the surface of the steel sheet. In the approach described here the texture analysis is carried out using the fuzzy grey scale correlogram which is a modified version of the colour correlogram developed in the context of content-based image retrieval. The correlogram data provides the basis for a classification using a fuzzy rule base. Experimental results indicate that the fuzzy grey scale correlogram is applicable to the problem discussed in this paper.

Key words: Colour correlogram, fuzzy grey scale correlogram, galvanised steel, impact resistance test, optical quality control, texture analysis.

1 Introduction

Steel sheet coatings are to protect the coated material against environmental stress. As the steel sheets are exposed to deformation by bending or impact in production and duty the adhesion of the coating is important. To determine the coating quality in terms of adhesion the impact resistance test has been developed among other test procedures [1, 4]. It comprises a steel ball with a weight applied to it being dropped from a defined height onto the coated steel sheet. This results in a rapid deformation of the sheet (see Fig. 1). The extent to which the coating flakes off can be interpreted as a measure for adhesion of the coating.

In this paper hot-dip galvanised steel sheets are considered. The coating quality of such steel sheets is analysed based on the results of an impact resistance test. An image (1024×1024 pixels, 8 bit grey scale) is taken of the deformed sheet. Each tested steel sheet is then assigned to one of three quality

D. Lühning: *Optical Quality Control of Coated Steel Sheets Using Fuzzy Grey Scale Correlograms*, Advances in Soft Computing **2**, 475–480 (2005)
www.springerlink.com © Springer-Verlag Berlin Heidelberg 2005

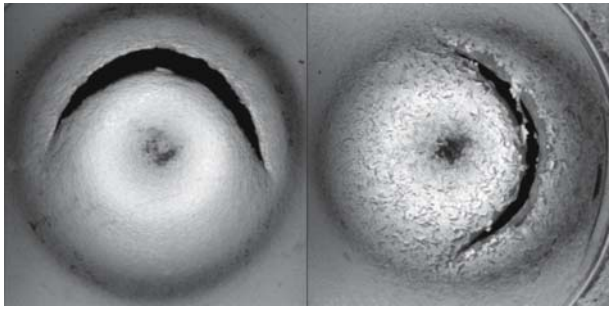


Fig. 1. Steel sheets with good (left) and bad (right) adhesion of coating

classes with class I describing the best quality. Sheets from class II can still be used for certain products while quality class III is rejected.

The steel sheets are rated with respect to the amount of flaking due to the impact resistance test. Spots where the coating flaked off can be identified in the image on the basis of their darker colour compared with the coating's lighter grey. Thus the classification has to take into account the texture of the image, i.e. the spatial correlation of grey values in the image.

2 Texture Analysis

An image feature that describes the spatial colour correlation in an image I is the colour correlogram proposed in [5]. It can be seen as a table indexed by pairs of colours. The k-th entry for the colour pair (c_i, c_j) specifies the probability of a pixel of colour c_j within a distance of k from a pixel of colour c_i:

$$\gamma^{(k)}_{c_i,c_j}(I) = \mathrm{P}(p_2 \in I_{c_j} | p_1 \in I_{c_i}, |p_1 - p_2| = k) \tag{1}$$

Here I_c denotes the set of all pixels with colour c. The distance $|p_1 - p_2|$ between two pixels is calculated according to the L_∞-norm.

2.1 Computation of the Colour Correlogram

The calculation of the colour correlogram described in [5] comprises three steps. In the first step the number of pixels of a given colour c within a given distance k from a fixed pixel $p = (x, y)$ is determined for positive horizontal and vertical direction. For horizontal direction the following equation holds:

$$\lambda^{c,h}_{(x,y)}(k) = \lambda^{c,h}_{(x,y)}(k-1) + \lambda^{c,h}_{(x+k,y)}(0) \tag{2}$$

with the initial condition

$$\lambda^{c,h}_{(x,y)}(0) = \begin{cases} 1 & \text{if}(x,y) \in I_c \\ 0 & \text{else} \end{cases} \tag{3}$$

Again I_c denotes the set of all pixels of colour c in the image I. For the vertical direction the calculation is similar. In the second step the number of pixels of colour c_j within distance k from some pixel of colour c_i is determined:

$$\Gamma^{(k)}_{c_i,c_j}(I) = \sum_{(x,y)\in I_c} (\lambda^{c_j,h}_{(x-k,y+k)}(2k) + \lambda^{c_j,h}_{(x-k,y-k)}(2k) + \lambda^{c_j,v}_{(x-k,y-k+1)}(2k-2) + \lambda^{c_j,v}_{(x+k,y-k+1)}(2k-2)). \quad (4)$$

In the last step the correlogram is calculated using the following equation:

$$\gamma^{(k)}_{c_i,c_j}(I) = \frac{\Gamma^{(k)}_{c_i,c_j}(I)}{8k h_{c_i}(I)}. \quad (5)$$

The factor $8k$ is due to the $8k$ pixels at distance k around any pixel while $h_c(I)$ denotes the colour histogram of the image I which is defined as

$$h_c(I) = |I_c|. \quad (6)$$

It has to be kept in mind that the border of the image is not considered by the algorithm in the described form.

2.2 Fuzzy Grey Scale Quantisation

Considering single grey values in the computation of the correlogram leads to extensive calculations and high memory consumption ($256^2 d$ correlogram entries where d is the maximum distance considered). Therefore it is reasonable to define classes of grey values instead. Such a definition also reflects human perception: A member of the quality control staff will not look at single grey values in the image. Instead he will describe the spatial correlation of grey values using terms like *light grey* or *very dark grey*.

These vague terms lead to some kind of quantisation of the grey scale using fuzzy sets [2, 6]. Chung and Fung proposed an algorithm for fuzzy colour quantisation which consists basically of a uniform quantisation of each of the RGB channels using fuzzy sets [3].

A similar approach was chosen for quantisation of the grey scale in this paper. The grey scale $[0, 255]$ was quantised into seven grey classes which were defined using fuzzy sets. In contrast to [3] a non-uniform quantisation has been chosen to match with human perception. The fuzzy sets can be considered as representing terms of a linguistic variable [7] *Grey Value*. A graphical representation of the defined fuzzy sets is shown in Fig. 2.

2.3 Fuzzy Grey Scale Correlogram

The original algorithm for the computation of the colour correlogram as described in [5] works only with single colour values. Thus the utilisation of

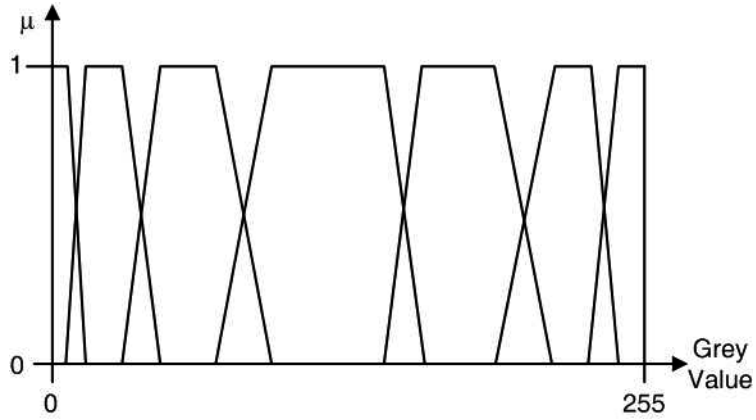


Fig. 2. Definition of classes of grey values as fuzzy sets: black, almost black, dark, middle, light, almost white, white (from left to right)

the fuzzy grey values proposed in Sect. 2.2 requires some modifications of the original algorithm.

Equation (3) from the first step of the calculation can also be written as

$$\lambda^{c,h}_{(x,y)}(0) = \chi_{I_c}(x,y) \tag{7}$$

with the characteristic function χ_{I_c} of the set I_c. In the calculation of the fuzzy grey scale correlogram the characteristic function is replaced by the membership function of the fuzzy set C of grey values:

$$\lambda^{c,h}_{(x,y)}(0) = \mu_C(I(x,y)) \tag{8}$$

Here $I(x,y)$ denotes the grey value of a pixel $p = (x,y)$. The second step of the algorithm conforms to the one given in [5]. In the last step of the calculation the color histogram is used. The definition of the histogram (6) can also be written as

$$h_c(I) = \sum_{p=(x,y)\in I} \chi_{I_c}(x,y) \tag{9}$$

As in the first step the characteristic function of the set I_c has to be replaced by the membership function of the fuzzy set C:

$$h_C(I) = \sum_{p=(x,y)\in I} \mu_C(I(x,y)) \tag{10}$$

3 Automatic Image Analysis

The proposed method for automatic analysis of images of steel sheets deformed by an impact resistance test includes several processing steps:

- contrast enhancement,
- calculation of the fuzzy grey scale correlogram,
- classification using a fuzzy-if-then-rule base.

In the first step a contrast stretching operator is applied to the images to compensate for different lighting conditions. Naturally, this step works on the crisp image data as delivered by the camera system. After this step the full range of grey values is utilised.

The second step calculates the fuzzy grey scale correlogram which was described in Sect. 2.3. To speed up calculations the membership degrees of all 256 possible grey values to the seven fuzzy grey values are precomputed and stored in an array for look-up.

In the last step the classification is carried out using a fuzzy-if-then-rule base. It has turned out that it is sufficient to consider only a few significant entries of the fuzzy grey scale correlogram. These are used as linguistic variables for the formulation of fuzzy-if-then-rules of the form

$$\text{IF } (\gamma^{(k)}_{c_i,c_j}(I) = T_1) \text{ AND } (\gamma^{(l)}_{c_m,c_n}(I) = T_2) \text{ AND } \ldots \text{ THEN Quality} = Q.$$

Here T_1 and T_2 denote terms of the linguistic variables used in the rule. The term sets include the terms VERY_SMALL, SMALL, MEDIUM, BIG and VERY_BIG though the corresponding fuzzy sets differ between the linguistic variables. Q is one of the three quality classes.

The method proposed here has been implemented and used as an experimental system for classification of steel sheets with respect to coating adhesion. A set of pre-classified images was used for the experiments. Half of the images of each class were used in defining the fuzzy-if-then-rules. The other images were then classified using the described method. At the end of the experimental phase the system classified all evaluation images correctly.

4 Conclusions and Further Work

The colour correlogram was designed to overcome the problems with color histograms in the field of image indexing and content-based image retrieval. Nevertheless the experiments indicate that the fuzzy version of the correlogram is also applicable to the problem discussed in this paper.

The results of the experimental phase are very promising but so far there were only very few examples of medium or bad quality sheets. Therefore further evaluation is necessary especially regarding the lower qualities.

Work will also be done with respect to the number of quality classes. As mentioned in the introduction sheets of class II can still be used for certain products. Therefore it is conceivable to define subclasses corresponding to the product requirements regarding the coating quality. Possibly it is helpful to include an analysis of the fissure that results from the deformation of the steel sheet because the regularity of the fissure seems to correspond to the quality of the coating. Also results of other test procedures can be included.

References

1. ASTM D2794-93 (1993) Standard Test Method for Resistance of Organic Coatings to the Effects of Rapid Deformation (Impact).
2. Buckley, J.; Eslami, E. (2002) An Introduction to Fuzzy Logic and Fuzzy Sets. Physica; Heidelberg, New York.
3. Chung, F.; Fung, B.Y.M. (2003) Fuzzy Color Quantization and its Application to Scene Change Detection. In: Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval (Berkeley, California, November 7, 2003):157–162.
4. DIN EN ISO 6272 (1993) Lacke und Anstrichstoffe - Prüfung durch ein fallendes Gewichtsstück.
5. Huang, J.; Kumar, S.R.; Mitra, M.; Zhu, W.-J.; Zabih, R.: Image indexing using color correlograms (1997) In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Puerto Rico, June 17-19, 1997):762–768.
6. Zadeh, L.A. (1965) Fuzzy Sets. In: Information and Control, vol. 8, no. 3:338–353.
7. Zadeh, L.A. (1975) The Concept of a Linguistic Variable and its Application to Approximate Reasoning I. In: Information Sciences, vol. 8:199–249.

Plenary Talk

Fuzzy Methods in Knowledge Discovery

Eyke Hüllermeier

Department of Mathematics and Computer Science
Marburg University
eyke@informatik.uni-marburg.de

As a response to the progress in digital data acquisition and storage technology, along with the limited human capabilities in analyzing and exploiting large amounts of data, the field of *knowledge discovery in databases* (KDD) has recently emerged as a new research discipline, lying at the intersection of statistics, machine learning, data management, and other areas. According to a widely accepted definition, KDD refers to the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable structure in data. The central step within the overall KDD process is *data mining*, the application of computational techniques to the task of finding patterns and models in data. Still, KDD also involves further important steps, notably data preparation, data cleaning, incorporation of prior knowledge, and interpretation of data mining results.

Fuzzy sets have the potential to support all of the steps comprising the KDD process. Their capability to interface quantitative patterns with qualitative knowledge structures expressible in terms of natural language can considerably improve the comprehensibility of extracted patterns, which is a point of major importance in data mining. Fuzzy information granulation allows for trading off accuracy against efficiency and understandability of models. Amongst other things, fuzzy sets can also be useful in data reduction, in dealing with incomplete and heterogeneous data, in modeling prior knowledge, or in interactive data mining, where the mining process is under partial control of the analyst.

The talk will focus on fuzzy data mining in the sense of searching for vague (fuzzy) patterns in data. This problem will be illustrated and motivated by several examples justifying the use of fuzzy techniques for representing and handling patterns in a formal way. Moreover, the problem of evaluating vague patterns will be highlighted, and a systematic approach for constructing corresponding scoring functions will be outlined. Finally, some pitfalls and difficulties that come along with fuzzy extensions of data mining methods will be pointed out.

E. Hüllermeier: *Fuzzy-Methods in Knowledge Discovery*, Advances in Soft Computing **2**, 483–483 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

Invited Session

Evolutionary Algorithms

Session Organiser:
Carlos Cotta

Action Games: Evolutive Experiences

Antonio J. Fernández and Javier Jiménez González

Dept. de Lenguajes y Ciencias de la Computación, Univ. de Málaga, Spain*
afdez@lcc.uma.es

Abstract. This paper defends the employment of Evolutive Algorithms (EAs) in action games by showing their virtues for both offline and online opponent controlling. The paper proposes (and also compares) several EAs applied offline in the solving of a classical path finding problem and used to provide intelligence to autonomous agents (e.g., the opponents) in an action computer game. The paper also presents an EA that has been successfully employed in real time (i.e., online) in an action game in which a player controls a military vehicle in a hostile enemy region.
Keywords: Evolutive algorithms, game programming, real time.

1 Introduction and Related Work

Game programming [1] is an exciting research field because of two main reasons: (1) computer games (CGs) are a business that generates millions of dollars each year, and (2) game programming has a heterogeneous nature that involves very different computer science techniques such as graphical techniques, artificial intelligence algorithms, interactive music, etc.

CGs are becoming one area of increasing interest, and proof of it is the fact that the traditional relation between cinema industry and computer game world is now being reversed, that is to say, in the past, many CGs were directly based on film scripts whereas nowadays, many films are directly based on CGs stories (e.g., Resident Evil, Mortal Kombat, Final Fantasy series, Street Fighter, Tomb Raider, Super Mario, ..., just to name a few).

The aim of CGs is to provide entertainment to the player(s), and in the past, research on commercial CGs was mainly focused on having more realistic games by improving graphics and sound (i.e., having higher resolution textures, more frames-per-second, ...etc). However, in recent years, hardware components have experienced exponential growth and players, with higher

* This work has been partially supported by projects TIC2001-2705-C03-02, and TIC2002-04498-C05-02 funded by both the Spanish Ministry of Science and Technology and FEDER.

A.J. Fernández and J.J. González: *Action Games: Evolutive Experiences*, Advances in Soft Computing **2**, 487–501 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

processing power computers, demand high quality opponents exhibiting intelligent behavior.

In many action commercial CGs, the opponent (i.e., the enemy) attitude is basically controlled by a fixed script previously programmed that often comprises hundreds of rules, in the form `if condition C if true then execute action A`. This is really a problem as these scripts are often complex programs that contain "holes" easily detected by an experienced player. As consequence, the reality simulation is drastically reduced and thus the interest of the player. This problem relies in the category of "artificial stupidity" [2]. To solve the problem, existing games employ some kind of artificial intelligence (AI) technique with the aim of giving intelligence to opponents and making the game more attractive to increase the satisfaction of the player.

In this context, AI plays an important role in the success or failure of a game and some major AI techniques have already been used in existing CGs [3]. Evolutionary algorithms[2] (EAs) offer interesting opportunities for creating intelligence in strategy or in role-playing games [4] and, in Internet, it is possible to find a number of articles related with the use of evolutive techniques in CGs [4], although, in general, most of them are dedicated to show the excellence of the use of EAs in Game Theory and, particularly, in the solving of multi-person decision problems [5, 6]. E.g., [6] shows how to use GAs for evolving control sequences for game agents although the focus is on bot navigation (i.e., exploration and obstacle avoidance). EAs have also been used in games for pathfinding [7].

Surely, EAs are one of the least-understood technologies in the game AI field, and scientific literature lacks proposals of using EAs in realistic action CGs (those in which we are interested on). For us, there are two main reasons for it: (1) game AI and academic AI often evolve separately in spite of the fact that game and academic communities have much to learn one from each other, and (2) EAs are resource intensive and require much time for development and tuning which does not make them adequate for in-game learning [7]. In fact, in game development, each resource is very important to allow the simulation in real time, and EAs are computationally expensive, so that traditionally, game developers have preferred another AI techniques such as Artificial Life, Neural Networks, Finite State, Fuzzy Logic, Learning and Expert Systems, among others [3, 7, 8].

In spite of their cost, EAs are considered as a promising technique that is on the forefront of game AI [9, 10, 11]. Moreover, game programming has already made successful use of EAs offline, that is to say, the EA works on the user computer (e.g., to improve the operational rules that guide the opponent actions) whereas the game is not being played and the results (e.g., improvements) can be used further online (i.e.. during the playing of the game). Through offline evolutionary learning, the quality of opponent intelligence in

[2] We use this term in a broad sense making reference to any kind of evolutive procedure, including genetic algorithms and genetic programming.

commercial games can be improved, and it has been proved that it is more effective than opponent-based scripts [12]. Also, genetic algorithms have been used to evolve combat strategies for agents or opponents between games (i.e., an offline learning) as it was done in the games *Unreal Tournament* [13] and *bSerene* [14]. In this sense, some realistic CGs that have used genetic algorithms are *return Fire II*, *The Creatures Series*, *Sigma* and *Cloak, Dagger and DNA* and *Quake III* [15, 16].

With respect to the online use of EAs in action computer game programming (i.e., EAs are executed in real time at the same time that the game is being played and simulated graphically in a computer), literature lacks studies about it. Perhaps the reason is that realistic action computer games require a real time graphical simulation that consumes a lot of computational resources (e.g., sound, music, and graphics) and EAs are cost; the direct consequence is that their combination seems to be too expensive computationally. Thus, to make a correct use of EAs in game programming one has to consider all these aspects and specifically has to know that the game AI and the game graphical simulation have to be executed together in real time. Speaking about the online evolutive learning, we have only found one paper in this sense, [17], that proposes methods and strategies for the online coevolution of agents in an action game.

In this paper, we encourage the use of EAs to provide, in action games, AI to autonomous agents controlled offline as well as online. We describe our experience on the employment of EAs on a specific military action game where the scenario conditions change dynamically. We describe (and compare) some EAs used for offline control in this game and, also implement an EA that can be used online: the secret consists of simplifying the set of actions executed by the opponents in order to reduce drastically the search space and accelerate the computation process.

2 The Game

We have implemented a game [18] that recreates an armed forces plan to transport a military vehicle (i.e., the user vehicle), placed in a hostile enemy region (i.e., the scenario), from an origin position to a destination one. Each *scenario* (also called indistinctly *region* or *world*) consists of a two-dimensional non-toroidal heterogeneous hostile dynamic grid-world. The world is *heterogeneous* because the terrain is not uniform, *hostile* because there exist enemy agents whose mission is to destroy the user vehicle, and *dynamic* because the solution search tree continuously change depending on the actions executed by the vehicle and the enemy agents. The purpose of the game is to move the military vehicle from the origin location to the destination position avoiding the natural obstacles of the world and the direct confrontation with the enemy agents in order to prevent the vehicle destruction. To do so, the vehicle is capable to execute some elementary actions such as go straight ahead one single grid square and turning 90° to its left or to its right.

Initially, the vehicle has some resources that must be kept in a controlled way until reaching the target. Another objective of the game is to optimize the expense of these resources: fuel, resistance and time. (The multi-objective consists of minimizing the time and maximizing both fuel and resistance at the end of the game.) The resources decrease during the game: time continuously decreases, fuel goes down according to the action executed by the vehicle (e.g., 4 units to cross a flat ground, 1 unit to turn 90°, 8 units to cross a terrain with gravel,...,etc) and resistance decreases with the attacks from the enemies. There exists also a *timeout* to limit the time to reach the target.

The game offers three kind of worlds with different sizes: 15×15, 30×30 and 50×50, and 4 scenarios for each of them. For each scenario, we have also generated 5 test cases where the origin location and the target position differ from one case to each other. Thus the game manages 60 different worlds; the timeout is constant (300 units) and the fuel is variable according to the world size; 250, 500 and 800 respectively for the worlds with size 15, 30 and 50.

The mission of the enemies is to destroy the vehicle or, alternatively, avoid the vehicle reach the destination grid before timeout. If an enemy crashes into the vehicle, then the vehicle is immediately destroyed. Each enemy has also associated an *area of shooting*, that is to say a rectangular region around the enemy, in which the vehicle is visible and the enemy can shoot the vehicle to decrease its resistance.

3 Off-line Evolutive Experiences

Our first trial consisted of transforming the game in a simpler (non-interactive) simulation in which the vehicle is an autonomous intelligent agent guided by some EA. The simulation emulates the game on different scenarios to evaluate the quality of our EAs to be used on action games. This means an *offline* intelligence as there is no human player to control the vehicle in real time.

During the game simulation, the vehicle does not know exactly the target position and is equipped with a *position sensor* that divides any world in four zones by the current position of the vehicle and indicates the zone where the target is (see Figure 1(left); the star points out the destination position). The vehicle also owns a *proximate sensor* that provides extra information (e.g., enemy presence, obstacle absence and terrain nature) about the adjacent grids located in the front of the vehicle as it is shown in Figure 1(medium).

The enemies are *dynamic* in the sense that they execute patrols following the cycle North-East-South-West as it is indicated in Figure 1(right).

In the following, by simplicity, if a is a variable belonging to a data type T, we write $a : T$, and if a is a value belonging to some data type T we write $a \in T$. Also, we write $T = (\text{field}_1 : T_1, \ldots, \text{field}_n : T_n)$ to indicate that T is the type composed by the set of all the tuples $(a_1, \ldots, a_n)$ where $a_1 \in T_1, \ldots, a_n \in T_n$. Moreover, if $a : T$ and $a = (a_1, \ldots, a_n)$, then $a.\text{field}_i$ denotes a_i $(1 \leq i \leq n)$. Below we show some of the data types used in the code of our offline EAs:

position = $(val\_x : \mathbb{N}, val\_y : \mathbb{N})$
resources = $(t : \mathbb{N}, f : \mathbb{N}, r : \mathbb{N},)$;
state = {success, timeout, death, withoutfuel, abort, outofrange, impassable};
terrain = {flat, target, river, fine gravel, puddle, mountain, bush, tree, hill};
grid = $(nature : terrain, time\_cost : \mathbb{N}, fuel\_cost : \mathbb{N})$
enemy = $(pos : position, \ldots)$
vehicle = $(pos : position, res : resources, \ldots)$
W_n = ARRAY $[1..n, 1..n]$ OF grid;

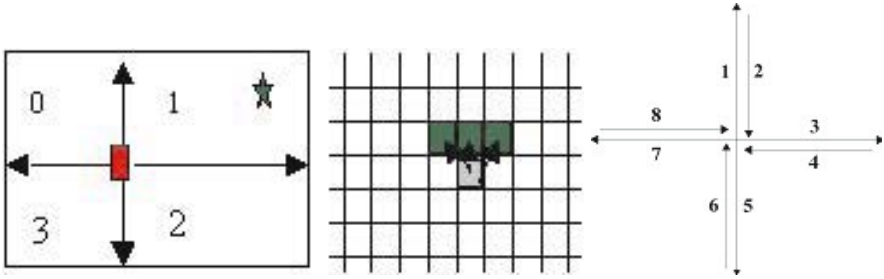


Fig. 1. (Left) Position sensor (Medium) Proximate sensor (Right) Patrol cycle

Basically, each world W_n is a grid bi-dimensional array of size $n \times n$ and, abusing the notation, we write $\alpha_{i,j} \in W_n$ to denote that $\alpha_{i,j}$ is the grid positioned in row i and column j in the world W_n. Grids can be represented as tuples or structures that contain the information about their nature (i.e., type of terrain) and some cost values (i.e., on time and fuel) that indicate the amount of resources spent by the vehicle in case that this crosses a specific grid. Each enemy has associated a position in the world that varies dynamically during the game, and the vehicle is identified, in each instant of the game, by a position inside the world and an amount of resources, i.e. time (t), fuel (f) and resistance (r). Both the vehicle and each enemy maintain also another information, not shown above, to register their actions for a further visualization of the game simulation.

The game objective consists of finding one (optimal) sequence of movements (i.e., a path, represented by a grid sequence associated to the vehicle) that enables the vehicle to reach the destination position safely.

All our offline EAs were implemented without using crossover operators. The reason is in the nature of the problem itself: observe that, in each stage of the game, resource consumption has to be taken into account as this influences the rest of the play. Thus, it is not easy to cross different paths since we have to consider not only segments of the paths (e.g., as in a classical one point or two point crossover) but also the correspondence between the resource consumption in the crossover points. A discussion about the usefulness of the crossover operator is beyond this paper and the reader is referred to [19].

Our offline EAs used these parameters: $pop\_size = 80$, $max\_generations \in \{300, 400, 500\}$ (depending on the world size), $mutation\_rate = 0.01$ and maximum length of chromosome = 250 (i.e., this parameter restricts the maximum number of movements as it is indicated below).

3.1 A Rule-Based Prototype

Fisrtly, we implemented a rule-based prototype (RBP) that defines a fixed script to control the vehicle. This script is composed by a set of behavior rules obtained by following the approach[3] given in [20]. The rules have the form *if direction sensor detects the target in zone i and proximate sensor informs about the nature of adjacent grid j then do action k*. Then information about the terrain nature and the resource cost associated to a grid in the world was used to optimize our set of behavior rules according to [20].

$k \leftarrow 0$
do{ $v[k+1].pos \leftarrow$ move_vehicle_according_rules($v[k].pos$)
$en[k+1] \leftarrow$ move_enemies_according_cycles($en[k]$)
$v[k+1].res \leftarrow$ update_resources($v[k].res, st$)
$i \leftarrow i + 1$
}**while** $(v[k-1].pos \neq (i_d, j_d) \wedge$ % Success
$v[k-1].pos \neq en[k-1, j].pos\ (\forall j \in \{1, \ldots, e\}) \wedge$ % Death
$res[k-1].r > 0 \wedge res[k-1].t <$ timeout $\wedge\ res[k-1].f > 0)$
% Termination: some resource is exhausted

Fig. 2. Basic schema of the RBP algorithm

The basic RBP schema is shown in Figure 2. Variable $k : \mathbb{N}$ denotes the iteration number and an expression $a \leftarrow b$ indicates the assignment from b to a; $\alpha_{i_d,j_d} \in W_n$ identifies the destination grid; *st:state* will contain the reason of the termination of the algorithm that is identified by one of the termination conditions of the loop *do-while*; v, and en are arrays of variable length where $v[k]$ and en[k,j] contain in the k'th-iteration, the information associated, respectively, with the vehicle (i.e., position and resources amount) and with the j'th-enemy, assuming that there are e enemies (for $1 \leq j \leq e$). Note that $v[0].pos$ corresponds with the starting position. Functions *move_vehicle_according_rules/1, move_enemies_according_cycles/1* and *update_resources/2* should be clear from their identifiers and basically update the state of the game in each iteration.

[3] It is a modification since the EA proposed in [20] is applied on static worlds without enemy agents i.e., basically it derives behavior rules to optimize a classical path finding problem.

As in the rest of all our offline EAs, we use this representation based on arrays not only to solve the posed problem but also to keep a trace of the movements executed by the vehicle and all the enemies in each instant of the game with the further aim of visualizing the simulation of the game[4]

Results. This approach solved 29 of the 60 worlds (i.e., 48.3%), a poor result although we also noted that this approach behaved much better on simple worlds (i.e., those not too heterogeneous) than on complex ones.

3.2 An Evolutive Hybrid Solution

We also developed an Evolutive Program (EP) based on the RBP procedure. Below we partially specify some new data types defined for the EP.

gen =(v : vehicle, en : ARRAY OF enemy);

chromosome = (path : ARRAY OF gen, length : $\mathbb{N}$, reason : state, fitness : $\Re$);

A *gen* is a tuple that provides information about the vehicle and the enemies in certain instant of the game and a *chromosome* is a 4-tuple that contains (1) a variable length sequence of genes that represents one of the possible path associated to one of the (possibly intermediate) states of the game[5], (2) the length of this path (i.e., number of stages or instants of the game played so far), (3) a value of termination (in case that the game is over), and (4) a fitness value that measures the path quality with respect to both target proximity and remaining resources amount. Note that, the last gen stored in the path (i.e., *ch.path*[*ch.length*], for some *ch* : *chromosome*) contains information about the last instant of the game played so far. The EP schematic code is shown in Figure 3.

```
For currentGeneration ← 1 to max_generations
    For i ← 1 to pop_size
        Evaluate(pop[i],, α_{i_d,j_d});
        if better_chromo(better,pop[i]) then better ← pop[i];
    Mutate(pop);
```

Fig. 3. Simplified schema of the EP algorithm

The variable pop contains the chromosome population where `pop[i].path[1].v.pos` identifies the origin grid ($1 \leq i \leq$ pop_size), $\alpha_{i_d,j_d} \in W_n$ is as in the RBP, `better:chromosome` initially contains the best

[4] E.g., the vehicle information is stored in $v[0], v[1], v[2] \ldots$ and so on.

[5] In fact a chromosome represents the "story happened so far" i.e., if *ch* : *chromosome*, then *ch.path*[*i*] and *ch.path*[$i-1$] keep information about consecutive instants of the game, where an *instant of the game* is each of the discrete stages that produce some change in the game state. Thus i numbers each of the different game stages.

path calculated by the RBP, and the procedure `Evaluate/2` applies the RBP algorithm taking as origin the last gen kept in chromosome $pop[i]$, that is to say, it applies the RBP procedure taking $v[0] = pop[i].path[pop[i].length].v$ and $en[0] = pop[i].path[pop[i].length].en$ (see Figure 2). Observe that this means a trial to complete the path stored so far in pop[i] in order to reach the target. Of course, this EP is very dependant on the RBP algorithm and thus on the quality of behavior rules used.

The fitness is calculated as the inverse of the Euclidean distance between the position stored in the last gen of the chromosome and the destination position[6]. A function call `better_chromo(c1,c2)` defines a stratified fitness function that returns true if $c_1.fitness > c_2.fitness$, or $c_1.fitness = c_2.fitness$ and $c_2.path[c_2.length].v.res \leq_r c_1.path[c_1.length].v.res$ (i.e., c_1 keeps more resource values than c_2), and false otherwise, and where $(t_2, f_2, r_2) \leq_r (t_1, f_1, r_1)$ iff $t_2 > t_1$, or $t_2 = t_1$ and $f_2 < f_1$, or $t_2 = t_1$ and $f_2 = f_1$ and $r_2 \leq r_1$ (i.e., a lexicographic ordering). At the end of the execution, *better.path* contains the sequence of genes that compose the better path obtained, and *better.reason* indicates if the simulation was successful (i.e., destination reached) or not.

The mutation is a grid offset: for some chromosome $pop[i]$, one position m (with $2 \leq m \leq pop[i].length$), where $pop[i].path[m].v.pos$ differs from the origin location, is chosen randomly and transformed to an adjacent position (if necessary, this is linked with the gen $pop[i].path[m-1]$ via the addition of gens to the path to assure its continuity). Of course, the enemies have to be taken into account in this mutation process. Basically, the mutation process consists of replacing the gen $pop[i].path[m]$ by a new gen $g : gen$ where

$$
\begin{aligned}
&g.v.pos.val\_x \leftarrow pop[i].path[m].v.pos.val\_x \pm \{0, 1\}\\
&g.v.pos.val\_y \leftarrow pop[i].path[m].v.pos.val\_y \pm \{0, 1\}\\
&g.en \leftarrow \text{move\_enemies\_according\_cycles}(pop[i].path[m-1].en)\\
&\qquad \text{and also}\\
&pop[i].length \leftarrow m
\end{aligned}
$$

The last sentence shown above truncates the path stored in $pop[i]$ from the mutated position so that, in next iterations, this can be completed by procedure `Evaluate/2` as explained above. Note that thsi is basically a path reconstruction process.

Results. EP solved 45 worlds from 60 (i.e., 75%). Figure 3.2 shows the time difference between the executions of the RBP (dark color line) and the EP (light color line). The x-axis identifies the name of the distinct worlds whereas the y-axis indicates the time (in seconds) spent in the simulation execution. The gaps in the graphic indicate that the simulation finished unsuccessfully; note that EP solves more cases and takes less time than RBP.

[6] i.e., for some $ch : chromosome$, $ch.fitness$ contains the inverse of $\sqrt{(ch.path[ch.length].v.pos.val\_x - i_d)^2 + (ch.path[ch.length].v.pos.val\_y - j_d)^2}$.

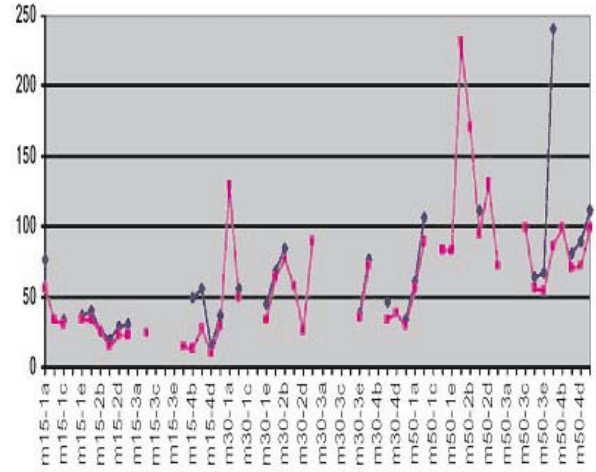


Fig. 4. Comparison of time (measured in seconds) (RBP-EP)

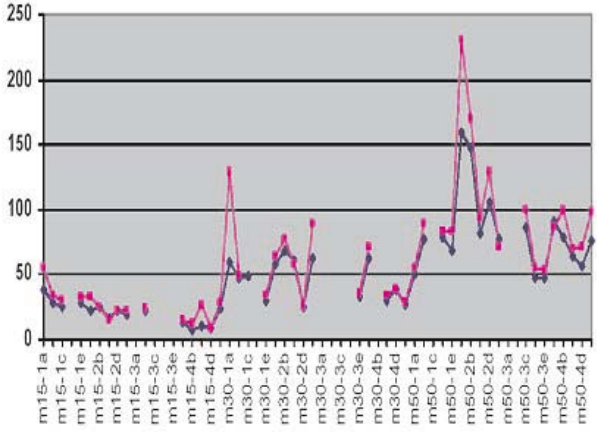


Fig. 5. Comparison of time (measured in seconds): Left (RBP-EP); Right (EP-EP+)

3.3 An Improved Evolutive Solution

We modified EP by defining a new mutation operator implemented as a roulette wheel selection based on the number of remaining generations (i.e., $max\_generations - current\_generation$). This means that the gen to mutate is not more chosen randomly with the same probability for all the candidate genes (as done in the EP algorithm), and its probability to be chosen increases according to the relative position of the gene in the path: higher position, higher probability. As consequence, it is likely that the first part of the path will remain intact and the path will be reconstructed, via function `Evaluate/2`, from a nearer-to-the-target gen. This technique often gives good

results as the different paths stored in the population "differ less" in their initial segments than in their final ones. We call this algorithm EP+.

Results. EP+ improved EP notably: from 60 worlds, 40 (i.e., 66'6%) were improved, 15 (i.e., 25%) largely improved (we speak of an improvement about 85% or of reaching a solution in a world that previously was not solvable with EP), 4 cases (6'6%) made worse (i.e., they needed more time although it was only 1.09% more in the worst case) and 1 case (1'6%) exactly equal. Figure 5 shows a comparative graphic wrt. time between the EP (light color line) and EP+ (dark color line).

3.4 Hybrid (EP+)-A*

A classical approach for solving path finding problems to optimality is the well known A* algorithm. We applied the A* algorithm to a modified version of the game (that we call *static*) where the enemies do not execute patrols but they maintain a fix position during the game simulation. The heuristic function used was the Euclidean distance from a grid to the target one. To the real cost, we added the time cost to cross a grid. As it was expected, we obtained an optimal solution to the static version of the game[7].

We then proposed the (EP+)-A* algorithm that hybridizes the EP+ and A* algorithms to combine the excellent behavior of A* in static worlds with the flexibility of the EP+ in dynamic worlds. Initially, (EP+)-A* considers, as starting point, the optimal static solution calculated by using A* in the static version, and then applies an EP+ modification consisting of replacing, in the *Evaluate* call in Figure 3, the application of the RBP algorithm by the A* algorithm to re-construct the path as explained in Section 3.2. The path variations obtained by this modification generate better paths.

Results. This proposal gave excellent results as we obtained a successful simulation in 60 worlds (i.e., 100%). Figure 6 shows the time comparison between the EP+ algorithm (dark color line) and this new hybrid algorithm (EP+)-A* (light color line). Note that, in most of the cases, the time is also optimized. There is just one draw case (identified as m30-4e) wrt. time; however the fuel resource was optimized using the hybrid algorithm (EP+)-A*.

We also created 10 new complex worlds with very extremely difficult cases on which the RBP, EP and EP+ algorithms did not find a solution. However, the (EP+)-A* algorithm returned a solution in all these cases.

Although successful our offline proposals were computationally cost (wrt. time) to be employed online in a real game.

4 Online Evolution for the Game

The experience accumulated so far with our offline EA versions highlighted that we had to follow an alternative strategy for using EAs online in a real

[7] In fact, the static version of the game consisted in a classical path finding problem.

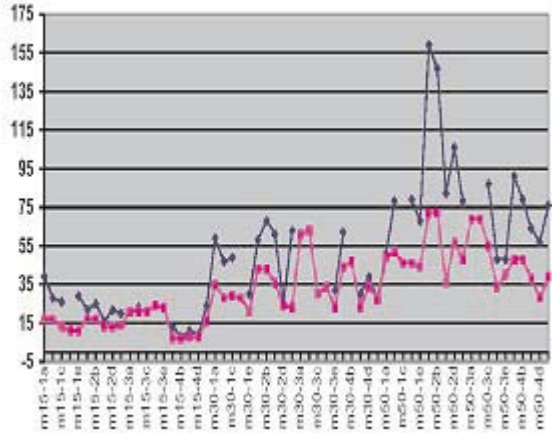


Fig. 6. Comparison of time (measured in seconds): EP+ vs.(EP+)-A*

game. Firstly, we replaced the offline game simulation by an online version where we let a human player control the vehicle and decide its actions in real time at the same time that each of its enemies is driven by a very simple evolutive algorithm whose purpose is to avoid, as soon as possible (i.e., by minimizing the game time), that the (player-controlled) vehicle reaches the target. In each stage of the game, each enemy has information about its (Euclidean) distance to both the human player (i.e., the vehicle) and the target position. This information is then translated into three degrees that depend on the world size. For a given world W_n and distance d, the degrees are: `near` ($d \in [0.0, n/6.0]$), `medium` ($d \in (n/6.0, n/3.0]$) and `far` otherwise. As consequence, each enemy receives information only in three discrete degrees. For each enemy, there are two targets (i.e., the player position and the destination location) so that there exist only 8 combinations of rules in the form of: *if human is at degree*$_1$ *and target at degree*$_2$ *then the enemy does some action A*. To accelerate the calculus we only offer 4 possible actions (enough for our action game) to be executed by the enemies. These actions are shown below:

- shot the player vehicle;
- attack the player vehicle (i.e., advance directly to destroy it by a collision);
- move nearer to the vehicle destination (in order to protect it) and,
- move randomly (i.e., move up, down, left or right). This action is added to make the enemy behavior less predictable and introduce intentional enemy errors in order to increase the user satisfaction (see Section 5).

A chromosome represents one of the possible actions to be executed by the enemy and is a value of type $action = \{shot, attack, move\_destination, random\_move\}$. Observe that actions can be coded using just 2 bits as there are only four possibilities. The simplified evolutive code that guides each enemy is shown in Figure 7 where $enemy : chromosome$ is initialized to a random move and, to the end of the algorithm execution, *enemy* will contain the action to be executed by the

```
enemy ← random_move;
InitializePopulation(pop);
(degree_v, degree_d) ← CalculateDistance(α_e, α_v, α_d, world_size);
For generation ← 1 to max_generations
    For i ← 1 to pop_size
      Evaluate(pop[i],enemy, degree_v, degree_d, res);
      if fitness(pop[i], degree_v, degree_d, res) < fitness(enemy, degree_v, degree_d, res)
              enemy ← pop[i];
      .....Apply the genetics operators.....
```

Fig. 7. On-Line evolutive algorithm to control each opponent

enemy; $\alpha_e, \alpha_v, \alpha_d \in position$ represent respectively, the position of the enemy, the vehicle and the destination; *pop* contains the chromosome population and the function *CalculateDistance* obtains the degree information of the enemy position wrt. the player and destination positions as explained above in the form of a tuple $(degree_v : degree, degree_d : degree)$ where $degree = \{near, medium, far\}$.

The function `Evaluate` receives an individual and uses the distance degree information and the resource amount of the vehicle (stored in variable $res : resources$ where *resources* is defined as in Section 3) to compute the best action to be executed by the enemy. The function `fitness/1` calculates the quality of a chromosome by considering all the possible situations considering again both the distance-degrees and the remaining resource amount. This is done by given some weights to the best actions and penalizing the worst situations for the enemy. The lowest fitness, the best chromosome.

We use a classical one-point crossover on the actions and the swap operator for mutation. The parameters used were the following: pop_size=5, max_generations=10, crossover_rate=0.9, mutation_rate=0.01.

Surprisingly the game works fine in real time[8]!!. We observe two key points for the success: the low number of generations and the small size of the population. Figures 8 and 9 visualize two different states of the game execution `in real time`. The game was implemented using the standard library OpenGL that provides advanced 2D/3D graphics capabilities.

5 Conclusions and Future Work

This paper highlights the power of evolutive techniques to be used to program action games that also demand a computationally expensive (graphical)

[8] Note that now, it is not necessary to register the sequence of movements, as done in our offline EAs, as the game visualization is done online together with the execution of our EA!.

Fig. 8. The enemy shots the vehicle

Fig. 9. Game over with vehicle destruction

simulation. We have shown, by proposing and comparing different EAs specifically coded to control autonomous agents in the simulation of a war game, that EAs are adequate for offline use. This is an expected result as EAs are time-consuming to develop and resource intensive when they are in execution. More surprisingly, we have also demonstrated, on reverse to the general belief, that EAs (designed to control the opponent intelligence) can be successfully executed online (in real time!!) during game simulation: the key for this online use consists of evolving much quicker than offline evolution. To do this, we have simplified the set of actions to be executed by the opponents in order to

reduce drastically the search space and to accelerate the computation process. In this line, [17] proposes different methods to do online coevolution of agents.

Now, the question to answer is: should we use evolutive/genetic algorithms for game programming?. In the following we briefly discuss this question.

`No, we should not.` Many game programmers suggest that EAs require too much CPU power and are very slow to produce useful results.

`Yes, we should.` Genetic (and evolutive) algorithms appear as one of the most promising game AI techniques [9]. As it is well known, evolutive/genetic algorithms are very good to find a solution in complex or poorly understood search spaces, and require a considerable computational effort that slows down the optimization process. Even although this is true, we defend the offline evolutive learning that is not too dependant on the execution time.

Moreover, as it has been shown in the paper, evolutive algorithms can be employed successfully on real-time. To do so, the evolution must happen very much quicker than offline evolution. We have shown how, by minimizing the set of actions to be executed by the opponent, this is possible. Of course, one may argue that, if there are only four actions, why should there be an EA to calculate the following action to be executed by each enemy if an easy enumeration algorithm would find the optimal solution in more real time? This is true although another issues have to be considered as for example the random component and the intentional errors of the opponent that make the game more attractive to the user. As argued in [2], it is important to provide human intelligence to the opponent to obtain a better simulation of the reality, but this does not correspond to make opponent necessarily smarter as, after all, the player is supposed to win.

Game development demands more and more research in a number of very "hot" areas such as: (1) *Multiplayer computer games* [21], (2) *Mobile game development* (i.e., games developed to run on mobile or cellular phones) and (3) *Simulation of army forces combat strategies*. With respect to (1), EAs could be applied to solve optimization problems focused in the reduction of networking resources requirements in distributed interactive real-time applications. With respect to (2), the proliferation and popularization of mobiles phones have produced the interest of game industry. This field demands professional game developers and active research on both hardware and software. Although the interest maybe in the local embedding of games into the phones, the area of downloadable games from the mobile phone seems to be particularly interesting [22]. With respect to (3), we think that EAs may be applied offline to simulate the emergent behavior of individuals (e.g., soldiers) in a combat between different army forces. Some AI techniques have already been applied in this sense in Hollywood films.

In any case, we claim that EAs are a unexplored AI technique in game programming and that they deserve particularly a wider study to be applied as a game AI technique. Moreover, we also assure that the future of game development is promising, and this paper encourages the cooperation between

the communities of the academic computer science researchers and the game industry programmers.

References

1. M. DeLoura (editor): Game Programming Gems. Editorial Charles River Media, INC., Rockland, Massachussetts (2000)
2. Lidén, L. In: Artificial Stupidity: The Art of Intentional Mistakes. Charles River Media, INC., Rockland, Massachussetts (2004) 41–48
3. Johnson, D., Wiles, J.: Computer games with intelligence. Australian Journal of Intelligent Information Processing Systems **7** (2001) 61–68
4. James, G.: Using genetic algorithms for game AI. GIGnews.com (2004) `http://www.gignews.com/gregjames1.htm`.
5. Wong, T., Wong, H.: Genetic algorithms. Internet (1996) `http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol1/tcw2/article1.html`.
6. Buckland, M.: AI Techniques for Game Programming. Premier Press (2002)
7. Sweetser, P.: AI in games: A review. Available at `http://www.itee.uq.edu.au/~penny/publications.htm` (2002)
8. Woodcock, S.: Game AI: The state of the industry. (August, 1999, `http://www.gamasutra.com/features/19990820/game_ai.htm`)
9. Rabin, S. In: Promising Game AI Techniques. Charles River Media, INC., Rockland, Massachussetts (2004) 15–28
10. Buckland, M. In: Building better genetic algorithms. Charles River Media, INC., Rockland, Massachussetts (2004) 649–660
11. Sweetser, P. In: How to build evolutionary algorithms for games. Charles River Media, INC., Rockland, Massachussetts (2003) 627–638
12. Spronck, P., Sprinkhuizen-Kuyper, I., Postma, E.: Improving opponent intelligence through offline evolutionary learning. International Journal of Intelligent Games & Simulation **2** (2003) 20–27
13. Dalgaard, J., Holm, J.: Genetic programming applied to a real time game domain. Master thesis, Aalborg University - Institute of Computer Science, Denmark (2002)
14. Alife Games: (`http://alifegames.sourceforge.net/bSerene/index.html`)
15. Sweetser, P.: (`http://www.itee.uq.edu.au/~penny/commercial_AI.htm`)
16. van Waveren, J., Rothkrantz, J.: Artificial player for Quake III arena. International Journal of Intelligent Games & Simulation **1** (2003) 25–32
17. Demasi, P., de O Cruz, A.: Online coevolution for action games. International Journal of Intelligent Games & Simulation **2** (2003) 80–88
18. Game, A.C.: (`http://tracer.lcc.uma.es/problems/index.html`, 2002)
19. Eiben, A., Schoenauer, M.: Evolutionary computing. Information Processing Letters **82** (2002) 1–6
20. Cotta, C., Troya, J.: Using a hybrid evolutionary-a* approach for learning reactive behaviours. In et al., S.C., ed.: Evo Workshops. Number 1803 in LNCS, Edinburgh, Scotland, Springer (2000) 347–356
21. Smed, J., kaukoranta, T., Hakonen, H.: Networking and multiplayer computer games - the story so far. International Journal of Intelligent Games & Simulation **2** (2003) 101–110
22. Mencher, M.: The future of game development: new skilss and new attitudes. GIGnews.com (2004) `http://www.gignews.com/gregjames1.htm`.

Co-evolving Multilayer Perceptrons Along Training Sets

M.G. Arenas[1], P.A. Castillo[2], G. Romero[2], F. Rateb[2] and J.J. Merelo[2]

[1] Department of Computer Science
University of Jaén. EPS Avda. de Madrid, 35 Ed. 50. E. 23071 Jaén (Spain)
[2] Department of Architecture and Computer Technology
University of Granada. Campus de Fuentenueva. E. 18071 Granada (Spain)
maribel@mgarenas.ujaen.es

Abstract. When designing artificial neural network (ANN) it is important to optimise the network architecture and the learning coefficients of the training algorithm, as well as the time the network training phase takes, since this is the more time-consuming phase. In this paper an approach to cooperative co-evolutionary optimisation of multilayer perceptrons (MLP) is presented. The cooperative co-evolution is performed on the MLP and training set at the same time. Results show that this co-evolutionary model reaches an optimal MLP with generalization error comparable to those presented by other authors but using a smaller training set, co-evolved with the system.

1 Introduction

To use fully the potential of Artificial Neural Networks (ANN), we have to search for the structure, connectivity and initial weights of the network, and then use a training algorithm (such as QuickProp (QP) [10] or RPROP [28]). The drawback of gradient based algorithms is that reaching the optimal solution is slow and it is not guaranteed. Moreover, it depends on the number of training epochs and learning parameters.

Evolutionary Algorithms (EAs) are capable to design ANNs because they are efficient searching methods. This kind of algorithm maintains a population of solutions (coded in the individuals). Furthermore, the problem of how to construct the training set is open and it must be done using some heuristic rule.

Usually, ANN practitioners do not search for a network that memorises the training patterns, but a network with generalisation abilities to classify those patterns not included in the training set. For this reason, the initial set is split into three sets for training, validating and testing respectively. There are different methods to construct these sets. The most used are: Partition into

M.G. Arenas et al.: *Co-evolving Multilayer Perceptrons along Training Sets*, Advances in Soft Computing **2**, 503–513 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

three separate sets [4], cross-validation [14, 9] and bootstrapping [3, 31, 21]. The best method for our purpose is the first one, because we want to reduce the training time using a smaller set (cross-validation and bootstrapping would take more time).

A "Species" is a kind of individual that represents the same kind of solution for one problem. Hence, two individuals with binary genome that represents a solution for the traditional OneMax [12, 16] problem for 100 bits are included into the same "specie". Co-evolutionary algorithms, employing more than one interacting "species" evolving under different evaluation functions, can be used to solve hard optimisation problems in a more efficient way than single species EA [18]. Moreover, the cooperative model is suitable when the problem solution is decomposable into subcomponents and there are strong interdependencies among them [26].

This paper describes a co-evolutionary system where two populations are evolved. The first one is a population of MLPs and the second one is a population of training sets, so the co-evolutionary algorithm evolves not only the weights and structure of the MLP individual but also the training set it uses for training.

We have used JEO [1] (Java Evolutionary Objects) for implementing the co-evolutionary algorithm. JEO is a *Java* framework for Evolutionary Computation experiments. JEO is layered on a virtual distributed resources machine named DRM [20][19]. DRM makes easier JEO experiments distribution using Internet. Both tools, DRM and JEO are included in DREAM (Distributed Resources Evolutionary Algorithms Machine) project [24][2].

This paper is organised as follows: section 2 presents the state of the art of co-evolutionary algorithms; section 3 describes the co-evolutionary model proposed in this paper. Section 4 explains the experiments and the obtained results and section 5 presents a brief conclusion and work in progress.

2 Co-evolutionary Algorithms: State of the art

EAs [22] are global optimisation methods, based on the theory of natural evolution. The EAs perform a multidirectional search keeping a population of potential solutions, creating information, exchanging it and using it for exploring the search space in any direction.

The population evolves: in each generation the best solutions mate, while the worst, disappear. In order to distinguish between good and bad solutions, an objective function (evaluation or "fitness" function) that measures the quality of the solution is used. The quality of the solution is measured using the distance between the individual and the optimal solution.

Most EAs involve a single "species", that is, a single genetic encoding aimed at finding solutions to one problem. Most of the authors include the optimising parameters in the individual genome, so the search has to be applied to a bigger search space. To avoid this problem, others make a different

proposal developing co-evolutionary systems because they are able to use very large search spaces [18, 32].

Co-evolutionary algorithms [25] involve more than one "specie" (populations) interacting among them. Each population evolves separately in an EA, and to obtain the fitness of an individual of a population, some individuals of other populations are taken into account.

According to the dependency between species (interactions between populations), the following classification could be done:

- competitive co-evolutionary algorithms [29], where the fitness of an individual depends on competition with other individuals from other species (each species competes with the remainder).
- cooperative co-evolutionary algorithms [26], where the goal is to find individuals from which better systems can be constructed. The fitness of an individual depends on its ability to cooperate in solving target problem with individuals from other species.

Many authors have combined successfully co-evolution with different methods. However, we will briefly describe some proposed models for evolving artificial neural networks.

In a first work on co-evolution, Hillis [17] addressed the problem of finding optimal sorting networks (regarding the number of comparators). He used a simulated evolution approach and got impressive results, given the difficulty of the problem. However, he introduced an important bias in his search algorithm by initializing the population with the first 32 comparators of the best known construction. The search was limited around this local optimum and, thus, the final solution still used these initial 32 comparators.

Then, some authors proposed to use cooperative models to design only the network architecture, like Moriarty and Miikkulainen [23] developed a method for designing ANNs based on two EAs: a population of nodes and another of networks (different ways of combining nodes of the first population). The nodes are coded using floating point vectors that represent weights.

Zhao [32] suggests to decompose a pattern recognition problem in several functions (one per class); then assign a module-network to each function. Thus, the whole classifier (network) consists of N sub-systems, and these subsystems (module-networks) are searched by evolution using several EAs. In [33], the focus of the discussion is on the evaluation of the different population individuals and applying the co-evolutionary model to evolutionary learning of radial basis function ANNs.

García-Pedrajas et al. [11] developed SYMBIONT, a model that tries to develop subnetworks (modules) instead of whole networks. These modules, that must cooperate, are combined forming ensembles that make up a network. The fitness assignment is based on competition within species and cooperation among species.

Hallinan and Jackway [15] propose a cooperative feature selection algorithm which utilises a genetic algorithm to select a feature subset in con-

junction with the weights of an ANN. Each network was encoded as a single binary string (160 bits long), with each eight bits representing a single integer in the range 0-255 (representing either a feature or a weight). Each network was encoded as a binary string. This coding might lead to a lack of precision, and good solutions could be lost due to the limitations of the representation.

Other authors present non-cooperative methods, like Smalz and Conrad [30] use two evolved separately: nodes, divided into clusters, and a population of networks that are combinations of neurons, one from each cluster. This method neither carries out competition among the neurons of the same cluster nor enforces cooperation among the different clusters of neurons.

As far as we have found, there is no approach to search for both the ANN and the minimum training set to train an ANN that solves a problem.

3 The cooperative co-evolutionary model

In this work, we are interested on optimising not only the *MLP* structure and weights, but also the training set the network will be trained with.

We propose a model where two species are evolving in a cooperative way. Both the *MLP* structure (architecture and weights) and the training patterns should be optimised. Thus, our model consists of an EA with two populations:

- **Population of TRNSET** : individuals that represent a training set are included in this population. One individual included in this population represents which patterns are selected to construct a training set.
- **Population of MLP**: this population includes a set of MLP networks for evolving the weights and the structure in layers. One individual represents a MLP with its architecture and weights.

3.1 Proposed model operation

Proposed model consists of two parts (see figure 1):

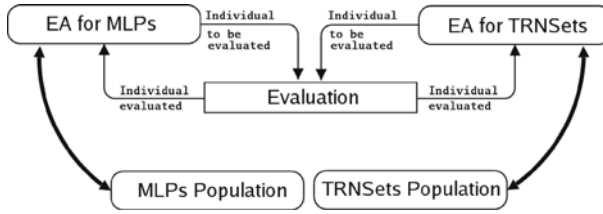


Fig. 1. Proposed model operation: to obtain the fitness of an individual, some individuals of the other population are taken into account.

EA to optimise TRNSET.
Each individual represents a training set. This training set includes patterns randomly chosen from the original training set of the problem. Each individual

represents *trnset*. The number of genes in the individual is equal to the total training patterns included, denoted by n. Gene i will be *true* if the training case i is included in the *trnset*, and $false$ if it is not.

The operators used for this EA are traditional AG operators: one-point-crossover and flip-flop mutation because the objetive is an EA as simple as possible (and it is using a binary genome).

The TRNSET population evolves following the *breeding, evaluation* and *rewarding* steps until a fixed number of generations have been completed. Each step performs the following:

Breeding step breeds a new population of children using a *tournament* selection and the mutation and crossover operators.

Evaluation step is the point of the algorithm where MLP and TRNSET populations exchange information. For fitness evaluation, one *trnset* selects randomly two *mlps* (mlp_1 and mlp_2) from the MLP population. That MLP is trained on the *trnset* and its classification ability obtained on the validation set. The *trnset* fitness is the average of classification errors on the validation set using both mlp_1 and mlp_2.

Rewarding step decides what individual survives for the next generation using generational replacement. This step includes an elitist individual.

EA to optimise MLP (architecture and weights).
The proposed co-evolutionary model is based on the *G-Prop* method (optimisation of MLPs using an evolutionary algorithm); since *G-Prop* has been described and analysed out in previous papers, thus we refer the reader to [5, 6, 7] for a full description. The evolved MLP should be coded into chromosomes to be handled by the genetic operators, however, *G-Prop* uses no codification, instead, the initial parameters of the network are evolved using specific variation operators such as mutation, multi-point crossover, addition and elimination of hidden units, and QP training applied as an operator to the individuals of the population.

The EA optimises the classification ability of the MLP and at the same time it searches for the number of hidden units (architecture) and the initial weight setting.

The main EA steps are the same as those for TRNSET population:

Breeding step breeds a new population of children using genetic operators.

Evaluation step evaluates each *mlp* using two training sets (from TRNSET population). The *mlp* is trained using the $trnset_1$ and then, its classification ability is obtained on the validation set. The process is repeat using the $trnset_2$. The *mlp* fitness is the average of both validation error percentages.

Rewarding step decides what individual survives for the next generation using generational replacement with the best *mlp* for elitism.

The collaboration steps and the rest of the algorithms may be consulted in figure 2.

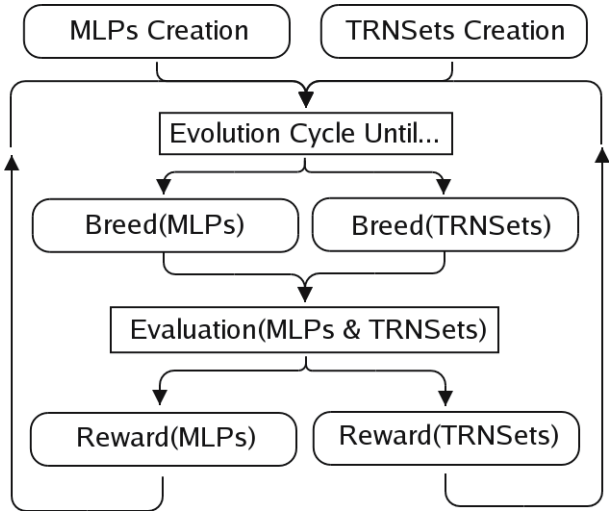


Fig. 2. Both populations evolve on different EAs but they join to check the end of the evolution and to calculate the fitnesses (evaluation phase).

4 Experiments and results

In these experiments, the **Glass** problem was used. This problem consists of the classification of glass types [27]. The results of a chemical analysis of glass splinters (percent content of 8 different elements) plus the refractive index are used to classify the sample. This dataset was created based on the glass problem dataset from the UCI repository of machine learning databases. The data set contains 214 instances. Each sample has 9 attributes plus the class attribute: refractive index, sodium, magnesium, aluminium, silicon, potassium, calcium, barium, iron, and the class attribute (types of glass).

The main data set was divided into three disjoint parts, for training (with 107 patterns), validating(with 54 patterns) and testing(with 53 cases). In order to obtain the fitness of an individual, the MLP is trained using the *trnset* coded in an individual from the TRN population; its fitness is established from the classification error with the validating set. Once the EA is finished (when it reaches the limit of generations), the classification error with the testing set is calculated: this is the result shown as *Test*.

Table 1 shows the average results for 20 executions using 15, 20, 25 and 30 generations as end-condition. The first column shows the number of generations. The following four columns show the results for MLP population. The table shows the best percentage of classification error using the validation set, column 2. The third column includes the classification error percent using the Test set for the best MLP network. The fourth column shows the size of the best MLP (measured as the number of weights or number of parameters) . The fifth column shows how many training patterns includes the *trnset* used for the best MLP network.

Table 1. Co-evolutionary algorithm results using 15, 20, 25 and 30 generations.

Generations	Best Validation	Test	Size	Patterns
15	0.28 ±0.01	0.32 ±0.02	201 ±104	54 ±5
20	0.28 ±0.01	0.31 ±0.02	242 ±114	55 ±6
25	0.27 ±0.01	0.32 ±0.03	164 ±82	56 ±3
30	0.26 ±0.01	0.30 ±0.02	188 ±82	54 ±5

Results show how we can get better MLP; by using more generation, the error classification for validation and test. Moreover, the size of the MLP is decreased with the generation number. It decrease is because if the MLP evolves during more time, the EA eliminates not useful neurons, so the final size is smaller.

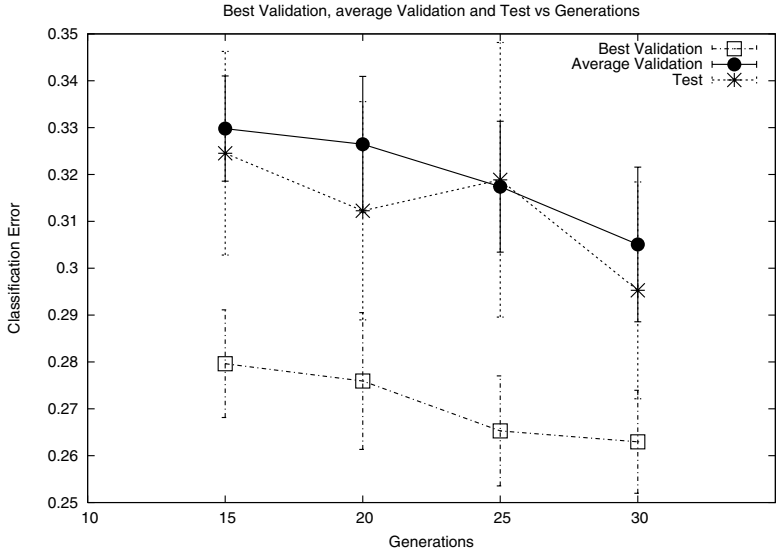
Let us compare this result with those of other authors. Table 2 shows the evaluation results of several algorithms for the same problem; G-Prop in [7], QP, those presented by Prechelt [27] and those presented by Grönroos [13]. This table shows the average classification error on the test set, the network size (measured as number of weights or number of parameters), the error and size corresponding to the individual that obtained a smaller error, and the network size of the smallest individual (and its error).

Table 2. Results for the Glass problem compared with results obtained using QP and those presented by Prechelt using RPROP, Grönroos using a GA with Kitano codification and G-Prop with default parameters. The best result is obtained by Castillo et al. using G-Prop where the classification error for test is 32% using a MLP with 252 weights.

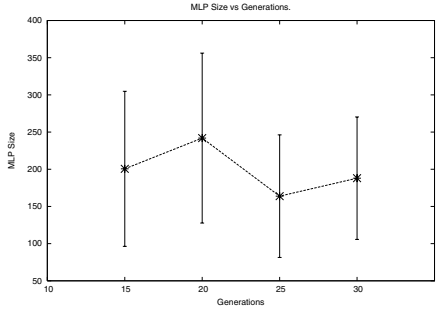
Method	Error	Size
Prechelt	33 ± 5	350
Grönroos	32 ± 5	350
QP	40 ± 5	374
G-Prop	32 ± 1	252 ± 85

If we compare the classification error for the Test set of all algorithms, we find 32 is the best result (G-Prop algorithm) getting a standard deviation of 1. co-evolutionary algorithm obtaing the best error classification for test sets using 30 generations: is 30% with standard deviation equal to 2%. That means the co-evolutionary algorithm presents better results than G-Prop using around $51,4\%$ of all training patterns. Hence, the co-evolutionary algorithm is better than G-Prop.

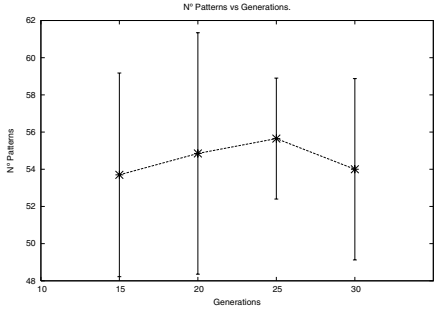
If we compare now the MLPs size we find that G-Prop finds networks bigger than the co-evolutionary algorithm. This difference is because the co-evolutionary algorithm uses less patterns so the MLP network has to learn less data than the original G-Prop. In any case, we could include multi-objetive criteria to obtain even smaller networks. This kind of *multiobjetive* goal is an objetive for the next version of the co-evolutionary algorithm.



(a) Best classification error for validation, the average classification error for validation and the classification error for testing versus the number of generations the co-evolutionary algorithms evolves. The best classification error for testing is found when we use more generations because if we have more time to find the correct combination for weights into the MLP network.



(b) Best MLP size vs the number of generations. The average MLP size is constant when we use more generations in the evolution but the standard deviation decreases with more generations.



(c) Number of training patterns the best *trnset* included are around 55 for all the generations, so, if the original training set has 107 patterns, we have reduced this number by approximately $51,4\%$

Fig. 3. Graphics comparative results for the classification error percentage with best validation, average validation and test (a), for the MLPs size (b) and for the number of training patterns (c). All results demonstrate the best MLPs are obtained when we use more generations and this results uses MLPs comparable in size than MLPs we get when evolves during more generations. (c) shows how the number of patterns included into the *trnset* does not depend on the number of generations the population undergoes.

5 Conclusions and Work in Progress

This paper presents a co-evolutionary method to optimise both the MLP architecture (number of hidden units and initial weights) and the training set used to optimise those networks.

These new algorithms present a new way for constructing training sets with less patterns than the original one, thus decreasing the training time we have to use with the MLP. The new *trnset* may be used in other algorithms to decrease the running time.

For this implementation we propose a simple algorithm that includes interactions populations only during the evaluation step. However, it is the first step for evolving not only the structure of the network but the training, validation and test sets.

However, a training set for good one ANN need not be the same for another ANN, so our purpose will be to make a new implementation getting not only the patterns that are included in the optimal training set, but the rules we can use to get these optimal paterns.

The co-evolutionary algorithm complexity may be too much for normal computers to process, so we are thinking too in a parallel version based on the Deme Model [8] including two kinds of demes, one for TRNSET populations and another one for MLP population.

6 Acknowledgements

This work has been supported in part by CICYT project TIC2003-09481-C04-04.

References

1. M. G. Arenas. *Computación Evolutiva Distribuida Asíncrona en redes heterogéneas usando una máquina virtual Java.* Tesis, Universidad de Granada, Dpto. Arquitectura y Tecnología de los Computadores., Septiembre 2003.
2. M. G. Arenas, P. Collet, A.E. Eiben, M. Jelasity, J.J. Merelo, B. Paechter, M. Preub, and M. Schoenauer. A framework for distributed evolutionary algorithms. In *Proceedings of the Seventh Conference on Parallel Problem Solving from Nature*, volume 2439 of *Lecture Notes in Computer Science*, pages 665–675, Granada, Spain, September 7-11 2002. Springer-Verlag.
3. W.G. Baxt and H. White. Bootstrapping confidence intervals for clinical input variable effects in a network trained to identify the presence of acute myocardial infarction. *Neural Computation*, 1995.
4. C. M. Bishop. *Neural Netwoks for Pattern Recognition.* Clarendon Press, Oxford, 1995.
5. P. A. Castillo, J. J. Merelo, V. Rivas, G. Romero, and A. Prieto. G-Prop: Global Optimization of Multilayer Perceptrons using GAs. *Neurocomputing, Vol.35/1-4, pp.149-163*, 2000.

6. P.A. Castillo, M.G. Arenas, J. J. Merelo, V. Rivas, and G. Romero. Optimisation of Multilayer Perceptrons Using a Distributed Evolutionary Algorithm with SOAP. *Lecture Notes in Computer Science, Vol.2439, pp.676-685, Springer-Verlag*, 2002.
7. P.A. Castillo, J.J. Merelo, G. Romero, A. Prieto, and I. Rojas. Statistical Analysis of the Parameters of a Neuro-Genetic Algorithm. *in IEEE Transactions on Neural Networks, vol.13, no.6, pp.1374-1394, ISSN:1045-9227, november*, 2002.
8. J.P. Cohoon, S. U. Hegde, W.N. Martin, and D Richards. Punctuated equilibria: a parallel algorithm. In John J. Grefenstette, editor, *Proceedings of the Second International Conference on Genetic Algorithms and their application, ICGA*. Lawrence Erlbaum Associates Publishers, 1987.
9. P. Craven and G Wahba. Smoothing noisy data with spline functions:estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische Matematik*, 31:377–403, 1979.
10. S. Fahlman. Faster-Learning Variations on Back-Propagation: An Empirical Study. *Proc. of the 1988 Connectionist Models Summer School, Morgan Kaufmann*, 1988.
11. N. García-Pedrajas, C. Hervás-Martínez, and J. Mu noz Pérez. SYMBIONT: A cooperative coevolutionary model for evolving artificial neural networks. *To appear in IEEE Transactions on Neural Networks*, 2003.
12. Josselin Garnier, Leila Kallel, and Marc Schoenauer. Rigorous hitting times for binary mutations. *Evolutionary Computation*, 7(2):173–203, 1999.
13. M.A. Grönroos. Evolutionary Design of Neural Networks. *Master of Science Thesis in Computer Science. Dept. of Mathematical Sciences. University of Turku*, 1998.
14. C. Gu, D. M. Bates, Z. Chen, and G. Wahba. The computation of generalized cross-validation functions through Householder tridiagonalization with applications to the fitting of interaction spline models. *SIAM Journal on Matrix Analysis and Applications*, 10:459–480, 1989.
15. Jennifer Hallinan and Paul Jackway. Co-operative evolution of a neural classifier and feature subset. *Lecture Notes in Computer Science*, 1585:397–404, 1999.
16. G. R. Harik, F. G. Lobo, and D. E. Goldberg. The compact genetic algorithm. *IEEE - Evolutionary Computation*, 3(4):287, November 1999.
17. Daniel Hillis. Co-evolving parasites improve simulated evolution as an optimization procedure. *Artificial Life II, Addison Wesley, pages 313-323*, 1992.
18. P. Husbands. Distributed coevolutionary genetic algorithms for multi-criteria and multi-constraint optimisation. *Evolutionary Computing, Lecture Notes in Computer Science, Vol. 865, pp.150-165, T. Fogarty (Ed.), Springer-Verlag*, 1994.
19. Mark Jelasity, Mike Preuβ, and Ben Paechter. A scalable and robust framework for distributed application. *2002 Congress on Evolutionary Computation*, pages 1540–1545, May 2002.
20. Mark Jelasity, Mike Preuβ, Maarten van Steen, and Ben Paechter. Maintaining connectivity in a scalable and robust distributed environment. *In 2nd IEEE International Symposium on Cluster Computing and the Grid (CCGrid2002)*, pages 389–394, May 2002.
21. T. Masters. *Advanced Algorithms for Neural Networks: A C++ Sourcebook*. John Wiley and Sons, ISBN 0-471-10588-0, 1995.
22. Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs , Third, Extended Edition*. Springer-Verlag, 1996.

23. D.E. Moriarty and R. Miikkulainen. Forming neural networks through efficient and adaptive coevolution. *Evolutionary Computation, vol.4, no.5*, 1998.
24. Ben Paechter, Thomas Baech, Marc Schoenauer, Michèle Sebag, A. E. Eiben, J. J. Merelo Guervós, and T. C. Fogarty. Dream distributed resource evolutionary algorithm machine. In *Proceedings of the Congress on Evolutionary Computation 2000*, volume 2, pages 951–958, 2000. Available from `http://-dr-ea-m.sourceforge.net`.
25. J. Paredis. Coevolutionary computation. *Artificial Life, 2:355-375*, 1995.
26. M.A. Potter and K.A. De Jong. Cooperative coevolution: an architecture for evolving coadapted subcomponents. *Evolutionary Computation, 8(1):1-29*, 2000.
27. L. Prechelt. PROBEN1 — A set of benchmarks and benchmarking rules for neural network training algorithms. Technical Report 21/94, Fakultät für Informatik, Universität Karlsruhe, D-76128 Karlsruhe, Germany, September 1994.
28. M. Riedmiller. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. *In IEEE International Conference on Neural Networks (San Francisco), vol. 1, pp.586-591. IEEE, New York*, 1993.
29. C.D. Rosin and R.K. Belew. New methods for competitive coevolution. *Evolutionary Computation, 5(1):1-29*, 1997.
30. R. Smalz and M. Conrad. Combining evolution with credit apportionment: A new learning algorithm for neural nets. *Neural Networks, vol.7, no.2, pp.341-351*, 1994.
31. R Tibshirani. A comparison of some error estimates for neural network models. *Neural Computation*, 1996.
32. Q. Zhao. Co-evolutionary learning of neural networks. *Journal of Intelligent and Fuzzy Systems 6, pp.83-90. ISSN 1064-1246*, 1998.
33. Q. F. Zhao, O. Hammami, K. Kuroda, and K. Saito. Cooperative Co-evolutionary Algorithm - How to Evaluate a Module. *Proc. 1st IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks, pp. 150-157, San Antonio*, 2000.

Improving Parallel GA Performances by Means of Plagues

Francisco Fernández[1] and Marco Tomassini[2]

[1] Centro Universitario de Mérida, Universidad de Extremadura
fcofdez@unex.es
[2] Institut d'Informatique, Universite de Lausanne
Marco.Tomassini@hec.unil.ch

Abstract. This paper describes a proposal for increasing performances when using Parallel Genetic Algorithms. We apply a new operator that has been recently described within Genetic Programming domain, the plague. By means of a series of experiments on a benchmark problem, we show that computational effort can be reduced when looking for solutions by means of Parallel GAs.

1 Introduction

Genetic Algorithms (GAs) are one of the Evolutionary Algorithms (EAs) more widely known and employed.

Gas are inspired by the natural evolution theory first presented and described by Darwin [2]. The aim is to endow computers with the capability for autonomously solving problems when the solution and way of solving is unknown. They are commonly applied when no algorithm capable of solving the problem at hand in a reasonable time is known. Many times, researchers apply this kind of techniques to optimization problems.

Despite the advantages described for EAs in literature, all these techniques feature a common problem: they often require large computing resources and time for attaining their goals. This is due to both the difficulty of problems that researchers face, and also the way EAs work, by iteratively evaluating a large number of candidates solutions, and making them evolve until a successful solution is found [1].

This problem can be partially alleviated by employing parallel algorithms and parallel computing architectures. Several researchers have applied these set of techniques to EAs for the last few years [9, 10, 3]. Nevertheless, if the difficulty of the problem is large, any improvement on the algorithm will be welcome to be added to the parallel model.

In this paper, we employ a Parallel Genetic Algorithm, that is improved with a new operator with the aim of improving both, quality of solutions and also reducing the time required for reaching those solutions.

F. Fernández and M. Tomassini: *Improving Parallel GA Performances by Means of Plagues*, Advances in Soft Computing **2**, 515–523 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

The paper is structured in the following way: Section 2 shows how parallelism can be added to Evolutionary Algorithm. Section 3 describes the new operator, while section 4 deals with the way of measuring results. Section 5 describe the experiments that have been performed and section 6 shows the results obtained. Finally, we present our conclusions on section 7.

2 Evolutionary Algorithms and Parallelism

Usually, EAs require large computing resources for solving real-life problems. This is due to the large number of individuals to be evaluated, and also the large number of generations -iterations- required until the convergence process leads to the solution. The time required for finding an enough useful solution may be extremely large.

During the last few years, several proposals have been described for alleviating this problem, and commonly, this solutions try to apply some degree of parallelization to the basic algorithm. There are important differences among the possible alternatives when applying parallelism, and usually depend on the EA we deal with. Among the available studies on the field, we have to mention Cantu-Paz [9] for Genetic Algorithms and Koza [10] and Fernandez [3] for Genetic Programming (GP).

When an evolutionary algorithm is parallelized, several alternatives are available:

- Parallelization at the level of the individual. Fitness values are computed on different processors simultaneously.
- Parallelization at the level of the population. The population is distributed among several smaller ones, and all of them are simultaneously run on different processors, so that the basic algorithm is performed within each of the populations, and some individuals travel among populations with a given frequency (see figure 1). This model is know as the Island Model.

There is a common opinion about the advantages of the Island Model [3, 9]. Nevertheless, any improvement for reducing the computing time required by the algorithm, without deteriorating the quality of solutions, would be very welcome.

In this research we apply the Island model. The basic algorithm is thus the following one, considering a master process that is in charge of managing communication among the subpopulations:

1. Initialize the population.
2. Evaluate all of the individuals in the population and assign a fitness value to each one.
3. Select individuals in the population using the selection algorithm.
4. Apply genetic operations to the selected individuals.
5. Insert the result of the genetic operations into the new population.

6. If the population is not fully populated go to step 3.
7. Select a number of good individuals and send them to the master process.
8. Receive individuals from the master proccess.
9. If the termination criterion is reached, then present the best individual as the output. Otherwise, replace the existing population with the new population and go to step 3.

While the master process runs the following algorithm:

1. For every population, receive individuals and send them to the next population.
2. Go to step 1.

This way of implementing the communication process allows to easily modify the communication topology. Each worker process sends a set of individuals to the master, and this resends them to a different worker process, according to the communication topology.

Although this algorithm may lack communication bottle necks, given that the aim of this research is not to study the speedup of the algorithm, but the quality of solutions and iterations required, this implementation is enough useful.

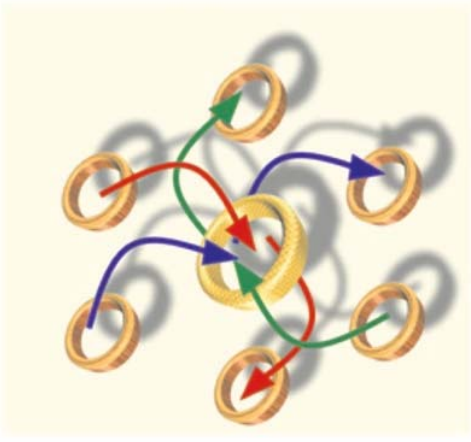


Fig. 1. Island Model

3 The Plague

A novel proposal for reducing the effort required for finding solutions by means of Genetic Programming was presented the last year: the plague [4]. The

operator has also been applied to parallel GP with successful results [5], and only very recently, the first attempt to extend the idea to GAs have been described [6].

In this paper we try to extend the study to parallel GAs.

When a difficult problem is solved by any EA, a large population of individuals is usually required, and also a large number of generations have to be computed. This means that the computational cost is frequently extremely large, even when a parallel implementation is employed.

On the other hand, when populations of individuals are randomly generated, they are usually distributed along the search space, and when generations are computed, the individuals converge to a solution (may be not optimal) [1]. The result is that after a number of generations, all the population is located on a very narrow area of the search space. In [4], researchers proposed the idea of reducing the size of the population as the generations are computed, so that we avoid to have many similar individuals in the population -individuals located in the same area of the search space. To do so, a new operator is described, the plague. It works by removing a fix number of individuals from the population every generation. Although this process cannot continue indefinitely (after a number of generations no individual is present in the population), a series of experiments have shown the usefulness of plagues for reducing computing time, while assuring the quality of solutions [4].

This paper presents a study on the advantages of plagues on Parallel Genetic Algorithms. We show that the parallel model can benefit from Plagues.

Plagues act on the algorithm's main loop -shown above. It is applied after evaluating the quality of individuals, once they are classified according to the quality. The plague removes a number of individuals, choosing them among those featuring worst fitness values.

4 Measuring effort in Genetic Algorithms

When Genetic Algorithms are applied results are usually compared employing curves that show the relationship between number of generations computed and quality of solutions obtained. Although this way of measuring results allows us to make comparisons of results when different experiments employ the same computing effort every generation, it is not valid when different numbers of individuals are employed in different experiments; this is because a generation that computes a large number of individuals will require larger time than another evaluating a smaller population. In this case, the appropriate measure is the total number of individuals evaluated for obtaining a given quality of solutions.

In the experiments presented in this paper, we have employ the latter way of measuring effort: total number of individuals evaluated from the begin-

ning of the experiment until a given generation is reached. This value is then compared with the fitness value obtained in that generation.

5 Experiments

For performing experiments, we have employed a well-known problem commonly used for comparing different GA techniques. This problem is described in [7], and basically consists of a trap function with well-known properties and its difficulty can be regulated easily. The values of the deceptive functions depend on the number, u, of bits set to one in their kbit input substring. The fitness increases with more bits set to zero until it reaches a local optimum, but the global maximum is at the opposite extreme where all the bits in the input are set to one. The order k traps are defined as:

$$f_k(u) = \begin{cases} k - u - d & \text{if } u < k, \\ k & \text{if } u = k, \end{cases} \tag{1}$$

where d is the fitness difference of the two peaks, which in our case is always set to one. The trap functions become more difficult by increasing the number of bits k in the basin of the deceptive optimum and by reducing d.

In the experiments, we varied k from 4 to 8. The fitness functions are formed by concatenating fully-deceptive trap functions and adding their individual contributions. We determined to set the length of the individuals to $l = 20 * k$ bits. For example, for the 6-bit trap problem, the individuals are $l = 120$ bits long and their fitness is calculated as $\sum_{i=0}^{20} f_6(u_{6i})$, where u_{6i} denotes the number of ones in the substring that starts at position $6i$.

The experiments were carried out using GALib [8]. Only small modifications to the source code were necessary to include the plague operation.

6 Results

We show in this Section the results that have been obtained for the trap function described above. We have set the difficulty of the problem, so that experiments with different population size are required: the more difficult the problem the larger population is required for solving it.

For each of the experiment we have performed 50 independent runs, always employing the same set of parameters. Each of the curves presented has thus been computed by averaging the results obtained along the 50 runs. Two kinds of graphs are presented: one of them shows the average quality of the best solutions in the 50 runs, when compared with the effort employed (as described in section IV). The second kind of graphs presents the number of times that the optimal solution is find in the 50 runs.

6.1 k = 4

We have begun the set of experiments by reducing the difficulty of the problem to K=4. We employ the Island Model using two populations and 5000 individuals (2500 individuals per population).

Figure 2 shows results obtained. We can see that when no plague is applied, the results are better than those obtained with a plague removing 100 individuals per population every generation.

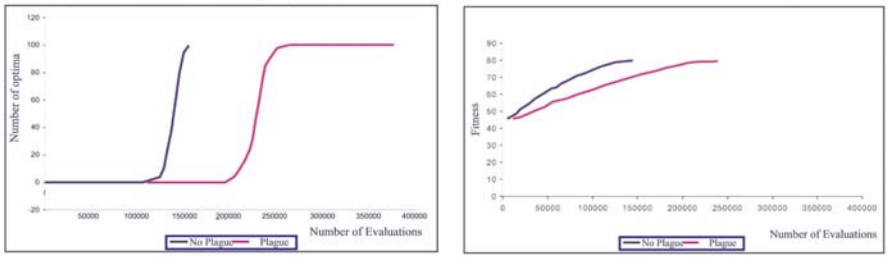


Fig. 2. Island Model with and without plagues. k=4, 2 populations with 2500 individuals each (at the beginning of the process). Plagues remove 100 individuals every generation from each of the population.

Similarly, fix-sized populations obtained the optimal value more times than populations undergoing plagues. Nevertheless, in previous research employing Genetic Programming, results proved that plagues were more effective with large populations. Therefore, we decided to increase the difficulty of the problem in order to experiment with larger populations (large populations are not justified for problems featuring small difficulty).

6.2 k = 5

Our second step was thus to increase the difficulty of the problem and also the size of the population.

Figure 3 shows results obtained with k=5. We notice that results favour now the use of plagues. Even when the improvement of quality obtained is slightly better than in the experiments that doesn't employ the plague, the number of times that the maximum value is obtained with plagues is significantly larger than when using the classical fix-sized populations in the Island Model.

6.3 k > 5

Similar experiments have been performed employing values for k such as 6 and 8, but now using larger population sizes.

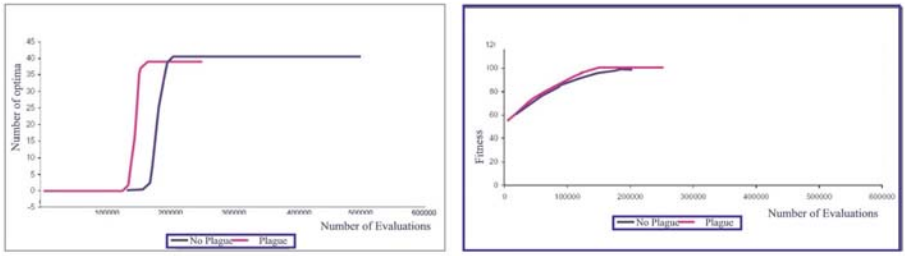


Fig. 3. Island Model with and without plagues. k=5, 2 populations with 5000 individuals each (at the beginning of the process). Plagues remove 50 individuals every generation from each of the population.

Figures 4 and 5 show results obtained for both values of k. We can notice again that experiments employing plagues reach better values. Furthermore, the difference in quality of solutions is even larger now.

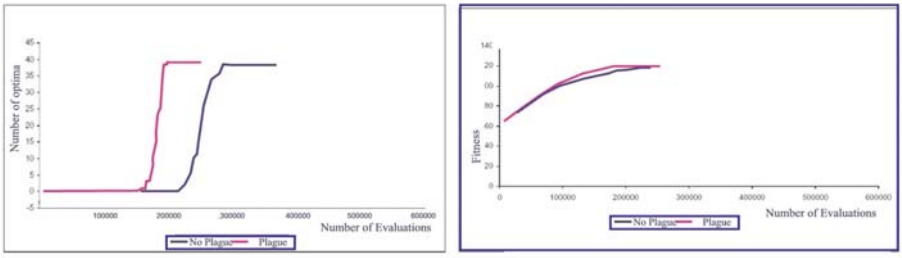


Fig. 4. Island Model with and without plagues. k=6, 2 populations with 5000 individuals each (at the beginning of the process). Plagues remove 100 individuals every generation from each of the population.

Summarising, we can say that results obtained by plagues when applied to GAs confirms the findings for Genetic Programming previously published. Plagues have proven to improve results also for GAs when large sizes of populations are required, at least for the problems we have tested.

7 Conclusions

We have presented in this work an application of a new genetic operator -called plague- to a parallel Genetic Algorithm.

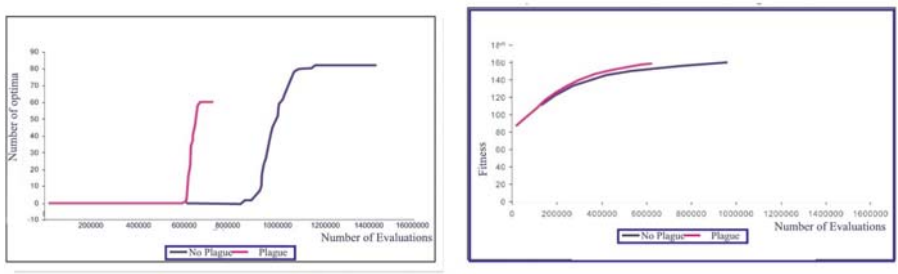


Fig. 5. Island Model with and without plagues. k=8, 2 populations with 6000 individuals each (at the beginning of the process). Plagues remove 100 individuals every generation from each of the population.

We have taken for benchmark purposes a well-know problem from the GA field, the trap function; this problem embodies some interesting features that allows to modify their difficulty by means of a parameter.

We have studied the problem with different values for the difficulty. We have specifically studied the average quality of best solutions over a set of runs, and also the number of times that the maximum value is reached within the set of runs. The general trend is that plagues are more useful when the difficulty of the problem is large -larger population sizes are required. When this is the case, both the average quality of solutions and also the number of times that the maximum is reached significantly improve. Results obtained for Parallel GAs are similar to those previously obtained for GAs.

Although the policy for removing individuals that have been employed in this research is extremely simple -a fix number of individuals per generation- we plan in the future to employ diversity measures for the population, so that plagues only will remove individuals when maximum improvement will be guaranteed.

Another problem that is related with the plagues as described in the paper, is the problem of emptying populations when all individuals have been removed. We will also study in the future different policies that allows to introduce new individuals when the size of populations is under a threshold value.

Acknowledgement

This research has been partially funded by Spanish Ministry of Science and Technology, under Project TRACER TIC2002-04498-C05-01.

References

1. Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs. Springer Verlag. Heidelberg, 1996.
2. R. Darwin "On the origin of species", 1892.
3. F. Fernández, M. Tomassini, L. Vanneschi, "An Empirical Study of Multipopulation Genetic Programming". Genetic Programming and Evolvable Machines, Vol. 4. 2003.21–51. Kluwer Academic Publishers.
4. F. Fernández, M. Tomassini, L. Vanneschi, "The effect of Plagues in Genetic Programming: A study of Variable-Size Populations", Lecture Notes in Computer Science 2610. 317–326.
5. F. Fernández, A. Martín, "Saving effort in Parallel GP by means of Plagues", LNCS 3003, 269–278. Springer Verlag.
6. F. Fernández, E. Cantú-Paz, T. Manzano, I. López, "Saving Resources in Genetic Algorithms by means of Plagues", VIII Parallel Problem Solving from Nature Conference (accepted).
7. Deb, K., Goldberg, D.E.: Analyzing deception in trap functions. In Whitley, L.D., ed.: Foundations of Genetic Algorithms 2, San Mateo, CA, Morgan Kaufmann (1993) 93–108
8. Wall, M.: Galib 2.3.2 (1995)
9. E. Cant-Paz and D. Goldberg: Predicting Speedups of Ideal Bounding Cases of Parallel Genetic Algorithms. Proceedings of the Seventh International Conference on Genetic Morgan Kaufmann.
10. David Andre and John R. Koza. "Parallel Genetic Programming: A Scalable Implementation Using The Transputer Network Architecture". P. Angeline and K. Kimea editors. Advances in Genetic Programming 2, Cambridge, MA, 1996.

Hybrid Evolutionary Algorithms for Protein Structure Prediction under the HPNX Model

Carlos Cotta

Dept. Lenguajes y Ciencias de la Computación, ETSI Informática,
University of Málaga, Campus de Teatinos, 29071 - Málaga - SPAIN
ccottap@lcc.uma.es

Abstract. The Protein Structure Prediction (PSP) problem is considered here. This problem is tackled with evolutionary algorithms (EAs) using the HPNX energy model. A classic penalty-based hybrid EAs, and two hybrid EAs are compared in this context. These hybrid EAs are based on the utilization of backtracking-based operators for mutation and recombination. They represent two different approaches for treating a constrained problem as the PSP: repairing infeasible solutions, and maintaining feasibility at all times. Both hybrid EAs perform better than the penalty-based EA, although the difference is more marked in the case of using the repairing procedure.

1 Introduction

One of the key stones of modern Biology is the so-called Central Dogma. According to it, the information stored in DNA molecules is transcribed into RNA molecules that will in turn direct the production of proteins. The latter happen to be extremely important in all cellular processes, since they serve multiple functions: carriers, catalysts, regulators, etc. This function is ultimately determined by the spatial conformation of the protein. Like pieces in a gigantic puzzle, molecules fit into one another, blocking or activating different biochemical reactions.

The spatial conformation of a protein is the outcome of different concurrent factors. First of all, a protein is a sequence of amino acids. Each of these aminoacids can be from one out of twenty different types, and it is connected to its neighbors in the sequence by a *peptide* bond. While this bond is relatively rigid, a certain amount of rotation can take place around other atomic links. Now, the protein sequence is subject to electrostatic forces among constituent atoms, van der Waals forces, interaction with the solvent, etc. As a result of these different forces, the sequence quickly folds itself, reaching a unique low-energy state (the so-called *native state*).

It is not difficult to realize how important the capability of ascertaining the native conformation of a protein from its amino acid sequence is. As mentioned

C. Cotta: *Hybrid Evolutionary Algorithms for Protein Structure Prediction under the HPNX Model*, Advances in Soft Computing **2**, 525–534 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

before, it may allow determining what role it will play in the cell. Conversely, some kind of reverse engineering can be conceived, so that proteins can be designed to serve a certain purpose, e.g., as a drug for a target disease.

Different methods can be used to determine this spatial conformation. On one hand, *structural* methods can be considered. These try to identify several well-known topological blocks within the sequence, e.g., $\alpha-$helices, $\beta-$sheets, etc. This can be typically done using Machine Learning techniques such as neural networks [1, 2]. Alternatively, *ab initio* methods are based on the assumption that the native state of a protein is the minimum energy conformation, and try to find this globally optimal conformation. To this end, analytic methods are in general out of question for finding these minima: trying to solve equations with thousands of degrees of freedom is an intractable problem as of today.

Simplified models are thus crucial elements in this context. The simplification arises from (i) abstracting the forces involved in the folding process, (ii) discretizing potential amino acid locations, and (iii) simplifying the computation of the function describing the energy of the conformation. As a consequence of these simplifications, it makes sense to talk about *Protein Structure Prediction* (PSP) rather than about *Protein Folding Problem* (PFP), since it is just the final conformation we are interested in (and not in the dynamics of the folding process).

It turns out that solving PSP instances to optimality is a very hard problem, even when these simplified models are considered [3, 4, 5]. One of the difficulties of the problem is the existence of geometrical constraints in the final conformation of the protein (i.e., self-avoidance in the chain, forbidden torsion-angles, etc.). Indeed, constraint-satisfaction approaches (based for instance on exact techniques such as Branch and Bound) have been applied to this problem, e.g., see [6, 7, 8]. Exact techniques are however inappropriate to deal with the combinatorial explosion of the space of conformations for increasing sequence lengths. Hence, the use of heuristic techniques such as Evolutionary Algorithms (EAs) is appropriate.

EAs have been applied to the PSP problem in a number of works. Typically, the constraints of the problem have been treated using a penalty function that measures to which extent these constraints are violated, e.g., see [9, 10]. Other approaches are based on repairing infeasible solutions, e.g., [11], or in maintaining feasibility at all stages of the algorithm [12]. We precisely build on this latter work, evaluating all three approaches on the basis of a more general general energy model for evaluating conformations. To be precise, we consider the HPNX model, an extension of the classical HP model of Dill [13]. In this work, those approaches not based on penalty functions rely on the embedding of a backtracking algorithm within the EA. Hence, a hybrid EA is obtained.

2 Background on Lattice Models for the PSP

As mentioned before, a protein is a sequence of amino acids. Discretizing the potential locations of these in the folded conformation is one of the typical simplifications of the problem. To do so, the sequence is assumed to be embedded in a certain regular lattice. This lattice can have different topologies. The simplest case is the 2-dimensional one, where for instance, square, triangular, and hexagonal lattices can be defined (see Figure 1). More complex (and certainly more realistic) lattices can be defined in three dimensions. For example, one can consider tetrahedral [14] or cube-octahedral [15] lattices.

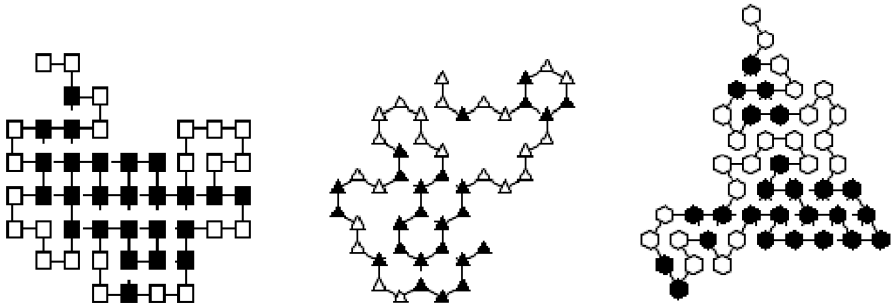


Fig. 1. Examples of protein conformations in the square lattice (left), triangular lattice (middle) and in the hexagonal lattice (right) under the HP model. Dark (resp. white) polygons represent hydrophobic (resp. hydrophilic) amino acids.

Once a certain lattice is assumed, a second simplification takes place in the energy model. In this sense, one of the most popular of such reduced models is the *Hydrophobic-Hydrophilic* model (HP model) of Dill [13]. In this model, each amino acid is classified into two classes: hydrophobic or non-polar (`H`), and hydrophilic or polar (`P`), according to their interaction with water molecules. Hydrophobic amino acids in the sequence tend to be packed together in a hydrophobic core, to avoid contact with the solvent. This is modeled in the HP model as follows: any feasible conformation (i.e., self-avoiding embedding) is assigned a free energy level; each pair of hydrophobic amino acids being topological neighbors in the conformation contributes a contact free energy $\epsilon < 0$ (say, $\epsilon = -1$), provided that these two amino acids are not adjacent in the sequence; any other topological contact does not contribute anything to the total free energy. Notice that the native state of a protein is a low-energy conformation (it is actually conjectured to be the global minimum). Thus, the number of `HH` contacts is maximized in the native state.

The HP model can serve as a first, rough approximation of the free energy of a conformation. It nevertheless exhibits some problems. One of these is degeneracy: a rather large number of different conformations can have the same minimal energy according to the model [16]. It is thus difficult to ascer-

Table 1. Energy function in the HPNX model.

	H	P	N	X
H	−4	0	0	0
P	0	1	−1	0
N	0	−1	1	0
X	0	0	0	0

tain which the "right" native conformation is (at least, any search algorithm based on this model would have no guidance in this sense). For this reason, alternative models have been defined. The HPNX model [17] is one of these. This model is an extension of the basic HP model, in which polar amino acids are split into three classes: positively charged (P), negatively charged (N), and neutral (X). This allows defining a more fine-grained energy model, described in Table 1. It has been empirically shown in [7] that the degeneracy of the HPNX model can be two orders of magnitude smaller than that of the HP model.

3 Evolutionary Algorithms for the PSP Problem

The application of EAs to the PSP problem involves determining appropriate representation and operators. We will proceed to discuss briefly these aspects in this section.

3.1 Representation of Folded Conformations

According to the description of discrete models provided in Section 2, a protein conformation is a self-avoiding embedding of the corresponding sequence in a certain lattice. Such an embedding can be expressed in three ways:

(a) external coordinates: the position of each amino acid of the sequence is expressed in terms of an external absolute reference frame.
(b) internal coordinates: positions are expressed in terms of an internal, variable reference frame. Typically, a moving reference frame located in the last amino acid placed is used.
(c) distance matrices: no direct representation of the positions for each amino acid is used; on the contrary, distances between amino acids are stored.

Option (a) has the drawback of hindering the manipulation of solutions for reproduction. For example, keeping the continuity of the sequence can be difficult during recombination. This is also true for (c), with the additional complexity of having to infer the actual embedding from the distance matrix. Of course, these problems are not unsolvable, and repair mechanisms can be

used (e.g., see [11]). Nevertheless, (b) is usually a more amenable choice for evolutionary algorithms.

If the reference frame co-moves with the last amino acid placed, it turns out that one only needs specifying which of the neighboring positions in the lattice will be used for the next amino acid. Thus, the folding is expressed as a sequence of $n-1$ *moves* with respect to the previously place amino acid, where n is the length of the sequence (the location of the first amino acid is fixed). Obviously this representation depends on the particular lattice topology considered; for example, each location has three neighbors in a triangular lattice, four neighbors in a square lattice, or six neighbors in an hexagonal lattice. This raises a second issue, i.e., the precise representation of each move.

Two major schemes for representing internal moves can be found in the literature: *absolute* moves [18] and *relative* moves [19]. In the former, the reference frame translates along the lattice as new amino acids are placed, but always keep a fixed orientation. For example, in the 2D square lattice four moves are possible: North, South, East, and West (see Figure 2–left). Thus, a conformation is expressed as a sequence $s \in \{\mathtt{N}, \mathtt{S}, \mathtt{E}, \mathtt{W}\}^{n-1}$. When relative moves are used, not only there is a translation but also a rotation. In the 2D square lattice, the North move is always aligned with the last two amino acids placed. Hence, three moves are allowed: Forward, Turn Right, and Turn Left, and conformations are expressed as sequences $s \in \{\mathtt{F}, \mathtt{T_L}, \mathtt{T_R}\}^{n-1}$ (a fixed initial orientation of the reference frame is assumed) – see Figure 2–right.

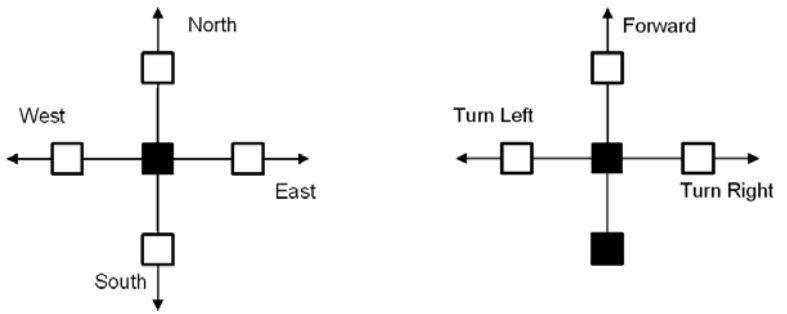


Fig. 2. (Left) Absolute moves in a 2D square lattice. The black cube represents the current location. (Right) Relative moves in a 2D square lattice. The black cubes represent the current location and the previous one.

Clearly, many sequences of moves violate the self-avoidance constraint, and hence represent infeasible solutions. This issue will be treated latter.

3.2 Reproduction as a Tree-Traversing Procedure

Let us consider the typical reproductive operations of recombination and mutation from a general point of view. Beginning with recombination, assume two solutions $\mathbf{x} = x_1 \cdots x_k$ and $\mathbf{y} = y_1 \cdots y_k$ are the parents selected. Now, the recombination itself can be regarded as a process in which information is incrementally taken from the parents to construct the descendant. This process starts from a completely unspecified solution, and subsequently gene values from any of the parents are selected and assigned to the descendant until a full solution is obtained [20]. Thus, each step involves determining from which parent a gene value (i.e., a move in our case) is transmitted. The whole process can then be assimilated to traversing a decision tree (Figure 3).

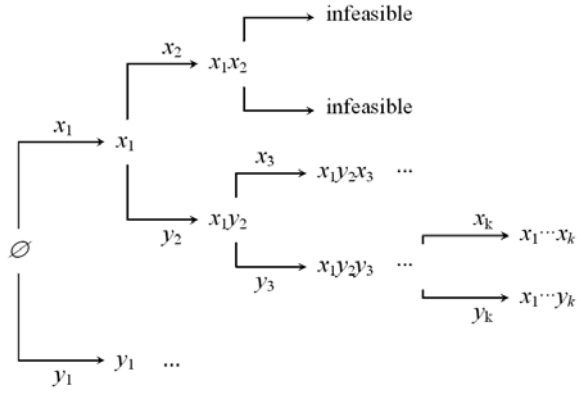


Fig. 3. Recombination can be assimilated to traversing a decision tree.

Any of the leaves of the tree constitutes a possible outcome of the recombination process. Some of them –those leaves located at depth k– are feasible solutions (the so-called dynastic span [21] of the parents). The remaining ones represent failed attempts, dead-ends in the construction of the descendant. This is generally the case, and is particularly true in the PSP problem: given two solutions being recombined, a partial solution $\mathbf{z} = z_1 \cdots z_i$ may be constructed such that no suffix $\mathbf{w} = w_{i+1} \cdots w_k$, $w_j \in \{x_j, y_j\}$ exists for which $\mathbf{zw}$ is a feasible solution.

A blind recombination procedure would simply take random decisions, and ignore whether the final outcome is feasible or not. However, *smarter* operators can be defined. More precisely, blind recombination can be upgraded as follows: random decisions are taken whenever a choice is possible, but whenever a dead-end is found, backtracking is performed, and a new decision is attempted at a previous level. This way, it is ensured that a feasible solution will be finally obtained (of course, as long as at least one of $\mathbf{x}$ and $\mathbf{y}$ is feasible; if both of them are infeasible, it cannot be ensured obtaining a feasible

solution). The process has obviously an added cost that must be taken into account though.

The situation is slightly different in the case of mutation, although some resemblance with the previous analysis can be found. A blind mutation operator would randomly change a certain position of the sequence of moves, and proceed with the resulting solution, either feasible or not. A smarter procedure can be defined whereby feasibility restoration is attempted after the random change. The restoration process reduces to traversing another decision tree in which all possible moves are allowed at any level, except at the point where the random change was introduced (this change at this position is preserved). There are two additional differences with respect to the recombination procedure: (i) the restoration process starts at the position next to the changed one (i.e., previous moves are only affected if the backtracking takes place beyond the changed position), and (ii) the first decision explored at any level is the move present in the original solution (i.e., original moves are preserved if possible).

Notice that –unlike recombination– the smart mutation process described before does not require the original solution to be feasible: it will produce a feasible solution no matter this original solution. Hence, it can be conceived not only as a mutation operator, but also as a repair mechanism.

4 Empirical Results

The experiments have been done with an elitist generational EA ($popsize = 100$, $p_c = .9$, $p_m = 0.01$) using linear ranking selection ($\eta = 2.0$). A maximum number of 10^5 evaluations has been enforced. In order to provide a fair comparison, the internal backtracking steps performed by some operators have been accounted and deducted from this computational limit (each backtracking step counts as $1/(n-1)$ evaluations). The fitness function used consists of calculating the free energy of the conformation using the function described in Table 1. In case the self-avoidance constraint is violated, those amino acids involved in an overlapping do not contribute to the free energy of the conformation, and a penalty term is added to the fitness of the solution ($2M$, where M is the number of amino acids overlapped). The test suite considered comprise five sequences with 27 up to 36 amino acids. These have been taken from [7], and are shown in Table 2. A 2D square lattice has been considered in all experiments.

Three evolutionary approaches have been tested: penalty-based EA, repair-based EA, and feasible-space EA. The first one uses single-point crossover and random mutation; the second one uses single-point crossover and backtracking-based mutation; finally, the third one uses backtracking-based recombination and mutation. The results are shown in Tables 3 through 5.

Several performance trends can be observed. First of all, the results for absolute encoding are better than those for relative encodings. This holds

Table 2. Sequences used in the experimentation.

S1	HXXNNHHHHXPHXHXHHHNHPPHHXPH
S2	HXXXHHHHNHXHHXXNHPHHPHXNXHP
S3	HPHHNXHHPNHHHHXXXHXPXHHHPXH
S4	HHXHHPHHXHHHHHHPPHHHHHXNHHHHHHH
S5	XHXNHXXHNPHXXHPXHXXHXNHPXHNXHPXHXPHX

Table 3. Results of the penalty-based EA (averaged for 50 runs).

	Absolute Encoding		Relative Encoding	
sequence	best	mean $\pm$ σ	best	mean $\pm$ σ
S1	45	39.28 $\pm$ 3.42	45	36.90 $\pm$ 3.52
S2	42	34.32 $\pm$ 3.11	37	32.14 $\pm$ 3.03
S3	45	39.76 $\pm$ 2.66	46	37.02 $\pm$ 3.22
S4	72	61.52 $\pm$ 4.73	64	58.14 $\pm$ 3.95
S5	42	37.52 $\pm$ 2.28	39	31.92 $\pm$ 3.21

Table 4. Results of the repair-based EA (averaged for 50 runs).

	Absolute Encoding		Relative Encoding	
sequence	best	mean $\pm$ σ	best	mean $\pm$ σ
S1	49	43.82 $\pm$ 2.97	49	42.70 $\pm$ 2.65
S2	42	39.78 $\pm$ 2.11	42	39.06 $\pm$ 2.12
S3	46	41.98 $\pm$ 2.12	46	42.28 $\pm$ 2.31
S4	72	67.36 $\pm$ 3.24	72	67.28 $\pm$ 2.62
S5	50	43.96 $\pm$ 2.95	47	39.26 $\pm$ 3.10

Table 5. Results of the feasible-space EA (averaged for 50 runs).

	Absolute Encoding		Relative Encoding	
sequence	best	mean $\pm$ σ	best	mean $\pm$ σ
S1	48	39.89 $\pm$ 2.78	45	40.26 $\pm$ 2.79
S2	42	36.96 $\pm$ 2.13	42	37.42 $\pm$ 1.78
S3	46	39.50 $\pm$ 2.40	45	41.02 $\pm$ 2.13
S4	72	63.92 $\pm$ 3.15	72	65.76 $\pm$ 3.01
S5	45	38.90 $\pm$ 2.29	41	36.68 $\pm$ 2.85

for the three algorithms, and is consistent with other results for the basic HP model [12]. Regarding the performance for a fixed type of encoding, it can be seen that the best results are those of the repair-based EA, and the worst results are those of the penalty-based EA. The feasible-space EA performs only slightly better than the penalty-based EA. The additional cost of backtracking recombination does not seem to be compensated by a significant quality gain. Simply using bactracking mutation appears as the best option:

infeasible solutions are repaired to a *nearby* feasible solution, and hence can be considered as a form of local improvement, as in memetic algorithms [22].

5 Conclusions

We have studied the application of evolutionary algorithms to the PSP problem under the HPNX model. It has been shown that the typical approach based on penalty functions can be outperformed if *intelligence* is deployed on the reproductive operators. However, an EA working purely in the feasible space does not provide the best results. On the contrary, temporarily crossing the feasibility barrier yields better results. These results support previous findings by Krasnogor *et al.* [10] that asserted the need for allowing the traversal of the infeasible space. The use of the repair mechanism –actually, a local improver– appears to have a major influence in the performance though. In some sense, the best results seem to be associated with the exploration of the boundary between feasible and infeasible regions, in which the repair-based EA works. This is also in full compliance with previous results of similar approaches on the simplified HP model [12].

Future work will be directed to analyze the possibilities for using additional local improvement operators. There is a very appealing line of work in connection with the advanced memetic algorithms developed in [23]

Acknowledgements

The author is partially supported by Spanish MCyT and FEDER under contract TIC2002-04498-C05-02. The author wishes to thank M. Ruiz for valuable assistance.

References

1. Muskal, S., Kim, S.: Predicting protein secondary structure content. a tandem neural network approach. Journal of Molecular Biology **225** (1992) 713–727
2. Cuff, J.A., Barton, G.J.: Application of enhanced multiple sequence alignment profiles to improve protein secondary structure prediction. PROTEINS: Structure, Function and Genetics (2000) 502–511
3. Unger, R., Moult, J.: Finding the lowest free energy conformation of a protein is an NP-hard problem: Proof and implications. Bull. Math. Biol. **55** (1993) 1183
4. Fraenkel, A.: Complexity of protein folding. Bull. Math. Biol. **55** (1993) 1199
5. Berger, B., Leight, T.: Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. Journal of Computational Biology **5** (1998) 27–40
6. Backofen, R.: Using constraint programming for lattice protein folding. In Altman, R., Keith Dunker, A., Hunter, l., Klein, T., eds.: Pacific Symposium on Biocomputing (PSB'98). Volume 3. (1998) 387–398

7. Backofen, R., Will, S., Bornberg-Bauer, E.: Application of constraint programming techniques for structure prediction of lattice proteins with extended alphabets. Bioinformatics **15** (1999) 234–242
8. Backofen, R.: The protein structure prediction problem: A constraint optimisation approach using a new lower bound. Constraints **6** (2001) 223–255
9. Khimasia, M., Coveney, P.: Protein structure prediction as a hard optimization problem: the genetic algorithm approach. Molecular Simulation **19** (1997) 205–226
10. Krasnogor, N., Hart, W., Smith, J., Pelta, D.: Protein structure prediction with evolutionary algorithms. In Banzhaf, W., et al., eds.: Proceedings of the 1999 Genetic and Evolutionary Computation Conference, San Mateo CA, Morgan Kaufmann (1999) 1596–1601
11. Piccolboni, A., Mauri, G.: Application of evolutionary algorithms to protein folding prediction. In Hao, J.K., et al., eds.: Artificial Evolution III. Volume 1363 of Lecture Notes in Computer Science. Springer-Verlag (1998) 123–136
12. Cotta, C.: Protein structure prediction using evolutionary algorithms hybridized with backtracking. In Mira, J., Álvarez, J., eds.: Artificial Neural Nets Problem Solving Methods. Volume 2687 of Lecture Notes in Computer Science., Berlin Heidelberg, Springer-Verlag (2003) 329–326
13. Dill, K.: Dominat forces in protein folding. Biochemistry **29** (1990) 7133–7155
14. Hinds, D., Levitt, M.: A lattice model for protein structure prediction at low resolution. Proceedings of the National Academy of Sciences **89** (1992) 2536–2540
15. Raghunathan, G., Jernigan, R.: Ideal architecture of residue packing and its observation in protein structures. Protein Science **6** (1997) 2072–2083
16. Dill, K., Bromberg, S., Yue, K., Fiebig, K., Yee, D., Thomas, P., Chan, H.: Principles of protein folding – a perspective of simple exact models. Protein Science **4** (1995) 561–602
17. Bornberg-Bauer, E.: Chain growth algorithms for HP-type lattice proteins. In: Proceedings of the First Annual International Conference on Computational Molecular Biology, Santa Fe NM, ACM Press (1997) 47–55
18. Unger, R., Moult, J.: Genetic algorithms for protein folding simulations. Journal of Molecular Biology **231** (1993) 75–81
19. Patton, A., Punch III, W., Goodman, E.: A standard GA approach to native protein conformation prediction. In Eshelman, L., ed.: Proceedings of the Sixth International Conference on Genetic Algorithms, San Mateo CA, Morgan Kaufmann (1995) 574–581
20. Cotta, C., Troya, J.: Information processing in transmitting recombination. Applied Mathematics Letters **16** (2003) 945–948
21. Radcliffe, N.: The algebra of genetic algorithms. Annals of Mathematics and Artificial Intelligence **10** (1994) 339–384
22. Moscato, P., Cotta, C.: A gentle introduction to memetic algorithms. In Glover, F., Kochenberger, G., eds.: Handbook of Metaheuristics. Kluwer Academic Publishers, Boston MA (2003) 105–144
23. Krasnogor, N., Blackburnem, B., Hirst, J., Burke, E.: Multimeme algorithms for protein structure prediction. In Merelo-Guervós, J., et al., eds.: Parallel Problem Solving From Nature VII. Volume 2439 of Lecture Notes in Computer Science., Berlin Heidelberg, Springer-Verlag (2002) 769–778

Invited Session

Aggregation Operators

Session Organiser:
Radko Mesiar

Quasi-Copulas on Discrete Scales

Juliana Mordelová

Department of Mathematics, Slovak University of Technology, Radlinského 11, 813 68 Bratislava, Slovak Republic,
mordel@math.sk

Abstract. In the paper the structure of quasi-copulas on discrete scales is studied. A special attention is given to the study of quasi-copulas with diagonal section. The cardinality of the set of quasi-copulas on $L_{(n)}$ determined univocally by the corresponding diagonal section is shown to be the Fibonacci number p_n.

Key words: Aggregation operators, 1-Lipschitz aggregation operators, Copulas, Quasi-copulas.

1 Introduction

In fuzzy logics with the interval $[0, 1]$ as the set of truth values the conjunctions are usually modeled by triangular norms, disjunctions by triangular conorms and negations by strong fuzzy negations. In practical situations the set of truth values is often discrete. Then as logical connectives discrete analogs of triangular norms, conorms and fuzzy negations are considered. In this contribution we will deal with a finite discrete scale $L_{(n)} = \{0, 1, \ldots, n\}$ and study quasi-copulas as a special conjunctive type of operators. As disjunctive operators, for example, dual quasi-copulas can be considered. Many results for quasi-copulas and dual quasi-copulas (as well as for copulas and dual copulas) can be studied in the framework of 1-Lipschitz aggregation operators since all these operators are special types of 1-Lipschitz aggregation operators.
Having an information on values of an operator at diagonal points $(x, x) \in L^2_{(n)}$, we can look for all quasi–copulas with given values at these points. It turns out that the set of all such quasi–copulas always possesses a greatest and a smallest element. Another problem is the problem of uniqueness of a quasi–copula with given values at diagonal points. The outlined problems are studied in Sections 4 and 5 and form the gist of the paper. In Section 2, some basic notions are introduced and in Section 3 the structure of quasi–copulas on $L_{(n)}$ is studied.

J. Mordelová: *Quasi-Copulas on Discrete Scales*, Advances in Soft Computing **2**, 537–544 (2005)
www.springerlink.com © Springer-Verlag Berlin Heidelberg 2005

2 Preliminaries

We start with recalling some basic notions. Let $n \in \mathbf{N}$ be any positive integer. Denote by $L_{(n)}$ the finite chain of $(n+1)$ elements, $L_{(n)} = \{0, 1, \ldots, n\}$ (with usual order).

Definition 1. A (binary) aggregation operator A on the set $L_{(n)}$ is a function $A : L_{(n)}^2 \to L_{(n)}$ with properties:

(A1) $A(0,0) = 0, \quad A(n,n) = n,$
(A2) A is non–decreasing, i.e., $A(x,y) \leq A(x^*,y^*)$ for all $x, x^*, y, y^* \in L_{(n)}$ such that $x \leq x^*,\ y \leq y^*$.

An aggregation operator A satisfying the Lipschitz property with constant 1, i.e.,

$$|A(x,y) - A(x^*,y^*)| \leq |x - x^*| + |y - y^*|$$

for all $x, x^*, y, y^* \in L_{(n)}$, will be called 1-Lipschitz aggregation operator. Distinguished classes of 1-Lipschitz aggregation operators are the classes of copulas and quasi-copulas.

Definition 2.
A (two-dimensional) copula on the set $L_{(n)}$ is a function $C : L_{(n)}^2 \to L_{(n)}$ with properties:

(C1) $C(x,0) = C(0,x) = 0 \quad C(x,n) = C(n,x) = x$, for each $x \in L_{(n)}$,
(C2) $C(x,y) + C(x^*,y^*) \geq C(x,y^*) + C(x^*,y)$ for all $x, x^*, y, y^* \in L_{(n)}$ with $x \leq x^*,\ y \leq y^*$.

Definition 3.
A (two-dimensional) quasi-copula on the set $L_{(n)}$ is a function $Q : L_{(n)}^2 \to L_{(n)}$ with properties

(Q1) $Q(x,0) = Q(0,x) = 0 \quad Q(x,n) = Q(n,x) = x$, for each $x \in L_{(n)}$,
(Q2) Q is non-decreasing,
(Q3) Q is 1-Lipschitz.

Quasi-copulas and copulas on $L_{(n)}$ are obviously 1-Lipschitz aggregation operators with neutral element n and annihilator 0. Evidently, each copula on $L_{(n)}$ is a quasi-copula, but not vice-versa. Recall that the dual of a quasi-copula Q on $L_{(n)}$ is an operator Q^* defined by $Q^*(x,y) = x+y-Q(x,y)$. Independently of this definition dual quasi-copulas can be introduced as functions $D : L_{(n)}^2 \to L_{(n)}$ which are non-decreasing, 1-Lipschitz and satisfy properties $D(x,0) = D(0,x) = x$ and $D(x,n) = D(n,x) = x$ for each $x \in L_{(n)}$, i.e., as 1-Lipschitz aggregation operators with neutral element 0 and annihilator n.

Finally, recall that each 1-Lipschitz aggregation operator A on $L_{(n)}$ satisfies

$$W_{(n)} \leq A \leq W_{(n)}^*, \tag{1}$$

where $W_{(n)}$ and $W^*_{(n)}$ are the Łukasiewicz t-norm and t-conorm on $L_{(n)}$, respectively, given by

$$W_{(n)}(x,y) = \max(x+y-n, 0), \qquad W^*_{(n)}(x,y) = \min(x+y, n).$$

Quasi-copulas (and also copulas) on $L_{(n)}$ satisfy:

$$W_{(n)} \le Q \le M, \tag{2}$$

where M is the minimum operator. In statistics, where copulas and quasi–copulas play an important role, these bounds are called the Fréchet–Hoeffding bounds.

3 Quasi–copulas on $L_{(n)}$

In this section we describe the structure of quasi–copulas on $L_{(n)}$. The set of all quasi-copulas on $L_{(n)}$ will be denoted by $\mathcal{Q}(L_{(n)})$. Recall that $x \in L_{(n)}$ is an idempotent element of a quasi–copula $Q \in \mathcal{Q}(L_{(n)})$ if $Q(x,x) = x$. Elements 0 and n are trivial idempotents of each quasi–copula.

Lemma 1. *Let a be an idempotent element of a quasi–copula $Q \in \mathcal{Q}(L_{(n)})$. Then*

$$Q(x,y) = x \wedge y, \qquad (x,y) \in \{a,\dots,n\} \times \{0,\dots,a\} \cup \{0,\dots,a\} \times \{a,\dots,n\}.$$

Proof. Let $y \ge a$. Then from the monotonicity of Q we have $a = Q(a,a) \le Q(a,y) \le Q(a,n) = a$, which gives $Q(a,y) = a$. Similarly $Q(x,a) = a$ for all $x \ge a$.
If $y \le a$ then from the 1–Lipschitz property of Q we obtain $Q(a,y) = Q(a,y) - Q(a,0) \le y$, and on the other hand, $a - Q(a,y) = Q(a,a) - Q(a,y) \le a - y$ gives $Q(a,y) \ge y$, i.e., $Q(a,y) = y$. Similarly, $Q(x,a) = x$ for all $x \le a$.
Finally, consider any point $(x,y) \in \{a,\dots,n\} \times \{0,\dots,a\}$. Due to the previous results and properties of Q we obtain

$$y = Q(a,y) \le Q(x,y) \le Q(n,y) = y,$$

which means that $Q(x,y) = y = \min(x,y) = x \wedge y$. The proof for points $(x,y) \in \{0,\dots,a\} \times \{a,\dots,n\}$ is analogous.

Proposition 1. *Let $Q \in \mathcal{Q}(L_{(n)})$ be a quasi–copula with idempotent elements $0 = x_0 < x_1 < \ldots < x_{k-1} < x_k = n$. Then*

$$Q(x,y) = \begin{cases} x_{i-1} + Q_i(x - x_{i-1}, y - x_{i-1}) & \text{if } (x,y) \in \{x_{i-1}+1, \dots x_i - 1\}^2, \\ & \text{for some } i \in \{1,\dots,k\} \\ x \wedge y & \text{otherwise,} \end{cases} \tag{3}$$

where Q_i is a quasi–copula on the set $\{0,\dots,x_i - x_{i-1}\}$ (i.e., on $L_{(x_i - x_{i-1})}$) for each $i \in \{1,\dots,k\}$.

Proof. By the previous claim, $Q(x,y) = x \wedge y$ at all points

$$(x,y) \in L_{(n)}^2 \setminus \bigcup_{i=1}^{n} \{x_{i-1}+1, \ldots, x_i - 1\}^2 .$$

Moreover, for each $i \in \{1, \ldots, k\}$ the function Q_i given by $Q_i(u,v) = Q(u + x_{i-1}, v + x_{i-1}) - x_{i-1}$ is a quasi–copula on the set $\{0, \ldots, x_i - x_{i-1}\}$, and for all $(x,y) \in \{x_{i-1}+1, \ldots, x_i - 1\}^2$ evidently $Q(x,y) = x_{i-1} + Q_i(x - x_{i-1}, y - x_{i-1})$.

Another interesting problem is the cardinality of the set $\mathcal{Q}(L_{(n)})$. For discrete copulas, according to Mayor and Torrens [7], it is known that $|\mathcal{C}(L_{(n)})| = n\,!$. Moreover, the number of all associative copulas on $L_{(n)}$ (i.e. divisible t–norms) is equal to 2^{n-1}. However, similarly to the case of discrete t–norms, see, e.g. [3, 5, 1], an explicit formula for the number of all quasi–copulas on $L_{(n)}$ is still unknown. We have developed a program for determining all quasi–copulas for any fixed $n \in \mathbf{N}$. The number of all quasi–copulas, copulas and associative copulas depending on n is illustrated in Table 1.

n	**Q**	**C**	**associat. C**
1	1	1	1
2	2	2	2
3	7	6	4
4	42	24	8
5	429	120	16
6	7436	720	32
7	218348	5040	64
8	10850216	40320	128

Table 1 Number of quasi–copulas, copulas and associative copulas on $L_{(n)}$.

4 Quasi-copulas with given diagonal section

Let Q be a quasi-copula on $L_{(n)}$. Its diagonal section, i.e., the function $\delta_Q : L_{(n)} \to L_{(n)}$, $\delta_Q(x) = Q(x,x)$ satisfies the following properties:

(S1) $\delta_Q(0) = 0, \quad \delta_Q(n) = n$,
(S2) $\delta_Q(x) \le \delta_Q(y)$ for all $x, y \in L_{(n)}, \quad x \le y$,
(S3) $\delta_Q(y) - \delta_Q(x) \le 2(y - x)$ for all $x, y \in L_{(n)}, \quad x \le y$,
(S4) $\delta_Q(x) \le x$ for each $x \in L_{(n)}$.

The question arises whether for each function $\delta : L_{(n)} \to L_{(n)}$ satisfying properties $(S1) - (S4)$, briefly called diagonals, there is some quasi-copula Q such that $\delta_Q = \delta$, i.e., whose diagonal section coincides with δ.

Let $\Delta_{(n)}$ be the set of all functions $\delta : L_{(n)} \to L_{(n)}$ satisfying properties $(S1) - (S4)$. It can be shown that for each $\delta \in \Delta_{(n)}$ the function $\widetilde{Q}$ defined by

$$\widetilde{Q}(x,y) = (x \wedge y) \wedge \left[\frac{\delta(x) + \delta(y)}{2}\right], \quad (x,y) \in L^2_{(n)}, \tag{4}$$

is a quasi-copula (also a copula) with $\delta_{\widetilde{Q}} = \delta$ ($[t]$ is the floor of a number t).

It turns out that the set of all quasi-copulas with diagonal section δ has a greatest and a smallest element. In [4] these problems were studied for 1-Lipschitz aggregation operators, quasi-copulas and copulas on the interval $[0,1]$. The results obtained in [4] can also be proved by the same technique for operators on the discrete scale $L_{(n)}$. Let us first state the results for 1–Lipschitz aggregation operators, in general. Note that in that case only properties (S1)-(S3) are considered.

Proposition 2.
Let $\delta : L_{(n)} \to L_{(n)}$ be a function satisfying $(S1) - (S3)$. Then

(i) The greatest 1–Lipschitz aggregation operator on $L_{(n)}$ with diagonal section δ is given by

$$\overline{A}^{\delta}(x,y) = x \vee y + \bigwedge(\delta(z) - z | z \in [x \wedge y, x \vee y]). \tag{5}$$

(ii) The smallest 1–Lipschitz aggregation operator on $L_{(n)}$ with diagonal section δ is given by

$$\underline{A}^{\delta}(x,y) = x \wedge y + \bigvee(\delta(z) - z | z \in [x \wedge y, x \vee y]). \tag{6}$$

The symbols $\wedge$, $\vee$ stand here instead of minimum and maximum, respectively.

Proposition 3.
Let $\delta : L_{(n)} \to L_{(n)}$ be a function satisfying $(S1) - (S4)$. Then

(i) The greatest quasi-copula on $L_{(n)}$ with diagonal section δ is given by

$$\overline{Q}^{\delta}(x,y) = (x \wedge y) \wedge \overline{A}^{\delta}(x,y)$$

where $\overline{A}^{\delta}$ is defined by (5).

(ii) The function $\underline{A}^{\delta}$ defined by (6) is the smallest quasi-copula on $L_{(n)}$ with diagonal section δ.

The mapping $\varphi : \mathcal{Q}(L_{(n)}) \to \Delta_{(n)} \quad \varphi(Q) = \delta_Q$ is surjective for each $n \in N$, however, for $n > 2$ it is not injective.

Example 1. (a) Let $n = 4$ and $\delta \in \Delta_{(4)}$ be given by $\delta(i) = i$ for $i \in \{0,3,4\}$ and $\delta(1) = 0, \delta(2) = 1$. Then there are 4 quasi-copulas whose diagonal section is δ, see Figure 1.

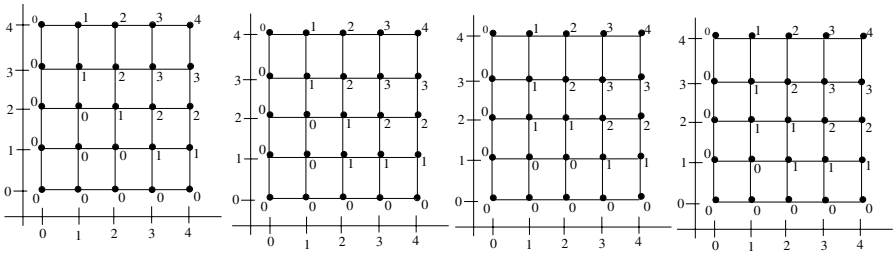


Figure 1 The values of quasi-copulas with given δ from Ex. 1(a)

(b) On the other hand, for diagonal $\delta \in \Delta_{(4)}$ defined by $\delta(i) = i$ for $i \in \{0, 1, 2, 4\}$ and $\delta(3) = 2$ there is only one quasi-copula with diagonal section δ, given in Figure 2.

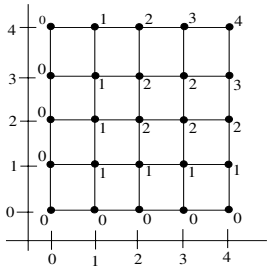


Figure 2 The values of a quasi-copula with given δ, Ex. 1(b)

Theorem 1. *Let $\delta \in \Delta_{(n)}$ be a diagonal. Then there exists a unique quasi-copula with diagonal section δ if and only if for each $x \in L_{(n)}$, such that $\delta(x) = x$ there exists an element $y \in L_{(n)}, y \neq x$ with $\delta(y) = y$ such that $|y - x| < 3$.*

Proof. Let $\delta \in \Delta_{(n)}$ and let Q be a quasi-copula with diagonal section δ, i.e., $Q(x, x) = \delta(x)$ for each $x \in L_{(n)}$. Denote the set of all idempotent elements of Q by I_Q:

$$I_Q = \{x \in L_{(n)} | Q(x, x) = x\} = \{x_0, x_1, \ldots, x_{k-1}, x_k\},$$

where $0 = x_0 < x_1 < \ldots < x_{k-1} < x_k = n$. It means, that at all points

$$(x, y) \in L_{(n)}^2 \setminus \bigcup_{i=1}^{k} \{x_{i-1} + 1, \ldots, x_i - 1\}^2$$

$Q(x, y) = x \wedge y$, and the considered set is determined by points x_i with $\delta(x_i) = x_i$ uniquely.

Sufficiency: Suppose first that $\max(x_i - x_{i-1}) < 3$. Then for each $i \in \{1, \ldots, k\}$ either $x_i - x_{i-1} = 1$ or $x_i - x_{i-1} = 2$. In the first case the set

$\{p \in L_{(n)} | x_{i-1} < p < x_i\}^2 = \emptyset$. Otherwise, this set is $\{(x_i - 1, x_i - 1)\}$ and $Q(x_i - 1, x_i - 1) = \delta(x_i - 1)$, which means that the condition $x_i - x_{i-1} < 3$ for each i, ensures the uniqueness of Q with $\delta_Q = \delta$.

Necessity: On the contrary, suppose that there exists $i \in \{1, \ldots, k\}$ such that $x_i - x_{i-1} \geq 3$. For such i, for all $(x, y) \in \{x_{i-1} + 1, \ldots, x_i - 1\}^2$ we have

$$Q(x, y) = x_{i-1} + Q_i(x - x_{i-1}, y - x_{i-1}),$$

where Q_i is a quasi-copula on the set $\{0, \ldots, x_i - x_{i-1}\}$, such that $Q_i(u, u) < u$ for each u.

Denote $\delta|_{\{x_{i-1}, \ldots, x_i\}} = \delta_i$. Without loss of generality we can suppose that $i = 1$. Put $a = \max\{x; \delta_1(x) = 0\}$. Because of 2-Lipschitz property of δ_1 either $\delta_1(a + 1) = 1$ or $\delta_1(a + 1) = 2$. For $\delta_1(a) = 0$ and $\delta_1(a + 1) = 1$ we obtain

$$\overline{Q}^{\delta}(a, a + 1) = 1 \quad \text{and} \quad \underline{Q}^{\delta}(a, a + 1) = 0,$$

which means $\overline{Q}^{\delta} \neq \underline{Q}^{\delta}$. If $\delta_1(a) = 0$ and $\delta_1(a + 1) = 2$, then $a > 1$, and a simple evaluation gives $\overline{Q}^{\delta}(a - 1, a + 1) = 1$ and $\underline{Q}^{\delta}(a - 1, a + 1) = 0$, which contradicts the uniqueness of Q.

Note that under conditions of Theorem 1, for given diagonal δ the operator $\widetilde{Q}$ given by (4) is the only quasi-copula with diagonal section δ. Moreover, this operator is also the smallest copula with diagonal section δ, and is a discrete analog of the Bertino copula on the interval $[0, 1]$.

Example 2. Let $n = 4$. Then there exist 9 different diagonals, 5 of them satisfy assumptions of Theorem 1, see Table 2.

$L_{(4)}$	0	1	2	3	4
	0	0	0	2	4
	0	0	1	2	4
	0	0	1	3	4
δ_1	**0**	0	**2**	2	**4**
δ_2	**0**	0	**2**	**3**	**4**
	0	1	1	2	4
δ_3	**0**	**1**	1	**3**	**4**
δ_4	**0**	**1**	**2**	2	**4**
δ_5	**0**	**1**	**2**	**3**	**4**

Table 2

Omitting the trivial case $n = 1$, we can claim:

Proposition 4. *Let $n \in \mathbf{N} \setminus \{1\}$. Then the number p_n of diagonals uniquely extendable to a quasi–copula on $L_{(n)}$, is*

$$p_n = p_{n-2} + p_{n-1},$$

where $p_0 = 1$ by convention.

Proof. Since $p_1 = 1$, for $n = 2$ the formula holds (recall that p_2=2). Let $n \geq 3$. Diagonals which can be extended to the only quasi–copula, are diagonals $\delta = \delta_I$ determined by sets $I \subseteq L_{(n)}$, $I = \{x_0, \ldots, x_k\}$, where $0 = x_0 < x_1 < \ldots < x_k = n$ and $x_i - x_{i-1} < 3$ for each $i = 1, \ldots, k$, and the condition $\delta_I(x) = x \Leftrightarrow x \in I$. If $\mathcal{I}_n$ is the system of all such subsets for a fixed n, then the number of desirable diagonals is $p_n = |\mathcal{I}_n|$. Note that the values of δ_I, $I \in \mathcal{I}_n$, at points $x \in L_{(n)} \setminus I$ are also determined uniquely. Consider a set $I \in \mathcal{I}_n$. Then either $x_{k-1} = n - 1$ or $x_{k-1} = n - 2$. In the first case $I \setminus \{n\} \in \mathcal{I}_{n-1}$, otherwise $I \setminus \{n\} \in \mathcal{I}_{n-2}$, which leads to the formula

$$p_n = p_{n-2} + p_{n-1}.$$

Hence for $n \in \mathbf{N} \cup \{0\}$ the number of diagonals which can be uniquely extended to a quasi–copula forms the sequence

$$1, 1, 2, 3, 5, 8, 13, 21, ...,$$

i.e., the Fibonacci sequence.

Acknowledgement

The support of the grant VEGA 1/0085/03 and action COST 274 TARSKI is kindly announced. This work was also supported by Science and Technology Assistance Agency under the contract No. APVT-20-046402.

References

1. T. Bartůšek, M. Navara (2002) Program for generating fuzzy logical operations and its use in mathematical proofs. Kybernetika 38:235–244.
2. T. Calvo, A. Kolesárová, M. Komorníková, R. Mesiar(2001) A Review of Aggregation Operators. University Press, Alcalá de Henares, Spain
3. B. De Baets, R. Mesiar (1999) Triangular norms on product lattices. Fuzzy Sets and Systems 104:61–75
4. E.P. Klement, A. Kolesárová (2004) Determination of copulas and quasi-copulas as special 1-Lipschitz aggregation operators. Preprint, submitted
5. E.P. Klement, R. Mesiar, E. Pap (2000) Triangular norms. Kluwer Academic Publishers, Dordrecht
6. G. Mayor, J. Torrens (1993) On a class of operators for expert systems. International Journal of Intelligent Systems, 8:771-778.
7. G. Mayor, J. Torrens (2004) Triangular norms on discrete settings, Preprint, submitted.
8. J. Mordelová, E.E. Muel (2001) Binary jointly strictly monotone kernel aggregation operators on finite scales. Proceedings of Uncertainty Modelling 2001, Bratislava, Slovakia, 127-131

Basic Classification of Aggregation Operators and Some Construction Methods

Radko Mesiar

Faculty of Civil Engineering, Slovak University of Technology, Slovakia
and
UTIA AV CR Prague, Czech Republic
mesiar@math.sk

Abstract. We introduce aggregation operators acting on real interval I. We discuss four basic classes of aggregation operators: conjunctive, disjunctive, averaging and mixed. Several examples are given, some construction methods are recalled. In more details we present a new method for construction of averaging aggregation operators and the general ordinal sum method for aggregation operators.

Keywords: aggregation operator, averaging operator, conjunctive operator, ordinal sum

1. Introduction

Aggregation operators are mathematical models of a fusion of several input values into a single output value. We will restrict our considerations to the most frequent situation arising from the measurement theory when all considered (input and output) values are from some continuous scale represented by a real interval $\mathbb{I} \subseteq [-\infty, +\infty]$. Observe that then the natural order and topology on $\mathbb{I}$ is herited from the standard order and topology on the real line. An operator $A: \bigcup_{n\in\mathbb{N}} \mathbb{I}^n \to \mathbb{I}$ defined for any finite input vector will be called an aggregation operator if it is non-decreasing in each coordinate, $A \mid \mathbb{I} = \mathrm{id}_{\mathbb{I}}$ and it preserves the boundaries of $\mathbb{I}$ in the following sense:

$$\text{for all } n \in \mathbb{N} \quad \sup\left\{A(x_1,\ldots,x_n) \mid (x_1,\ldots,x_n) \in \mathbb{I}^n\right\} = \sup \mathbb{I} \text{ and}$$
$$\inf\left\{A(x_1,\ldots,x_n) \mid (x_1,\ldots,x_n) \in \mathbb{I}^n\right\} = \inf \mathbb{I}.$$

Note that the above definition generalizes the standard definition of aggregation operators acting on the unit interval $\mathbb{I} = [0, 1]$, see [2, 7, 9].

R. Mesiar: *Basic Classification of Aggregation Operators and Some Construction Methods*, Advances in Soft Computing **2**, 545–553 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

Sometimes the following strengthening of the above boundary conditions (inspired by [15]) can be required:

for all $n \in N$, $a \in \mathbf{I}$, $i \in \{1, \ldots, n\}$, $\inf\{A(x_1, \ldots, x_n) \mid (x_1, \ldots, x_n) \in \mathbf{I}^n, x_i = a\} \le a \le \sup\{A(x_1, \ldots, x_n) \mid (x_1, \ldots, x_n) \in \mathbf{I}^n, x_i = a\}$ (B1)

Basic classification of aggregation operators was discussed by several authors. For a recent overview based on practical observations we recommend a recent paper of Dubois and Prade [6], where 4 types of aggregation operators are introduced: conjunctive operators, disjunctive operators, averaging operators, mixed operators. The aim of this contribution is a brief discussion of the above 4 types of aggregation operators and indication of some appropriate construction methods.

2. Conjunctive operators

Conjunctive aggregation operators are characterized by the boundedness property $A \le \mathrm{MIN}$. Consequently each conjunctive aggregation operator on the unit interval [0, 1] is a monotone extension of the classical boolean conjunction. Moreover, for symmetric operators the inequality $A \le \mathrm{MIN}$ is equivalent with the "downwards" attitude $A(x_1, \ldots, x_n, y) \le A(x_1, \ldots, x_n)$ (for all $n \in N$, $x_1, \ldots, x_n, y \in \mathbf{I}$). This equivalence is not true for nonsymmetric operators, in general. As an example recall the compensatory operator C: $\bigcup_{n \in N} [0,1]^n \to [0,1]$ given by the recursive formula

$$C(x_1, \ldots, x_n) = \min\left(C(x_1, \ldots, x_{n-1}), \sqrt{x_n}\right),$$

which was introduced in [11]. Evidently, C is a monotone extension of the boolean conjunction with "downwards" attitude, however,

$$C(0.6, 0.36) = 0.6 > 0.36 = \min(0.6, 0.36).$$

Taking into account the boundary condition (B1), we have the following result.

Proposition 1. An aggregation operator A defined on [0, 1] is a conjunctive aggregation operator satisfying (B1) if and only if for all $n \in N$, $x_1, \ldots, x_n \in [0, 1]$, card $\{i \mid x_i \ne 1\} \ge n - 1$, it holds

$$\mathbf{A}(x_1, \ldots, x_n) = \min(x_1, \ldots, x_n).$$

Observe that for the binary conjunctive aggregation operator $\mathbf{A}: [0, 1]^2 \to [0, 1]$, (B1) is equivalent to the fact that 1 is the neutral element of $\mathbf{A}$, $\mathbf{A}(x, 1) = \mathbf{A}(1, x) = x$ for all $x \in [0, 1]$. Note that this result need not be true for $n > 2$, in general.

Among typical examples of conjunctive aggregation operators recall the product on [0, 1], the sum on $[-\infty, 0]$, MIN on arbitrary interval $\mathbf{I}$. Two most typical construction methods are based on the transformation of the above mentioned conjunctive operators (multiplicative and additive generators of conjunctors, for more details see [9]) and on the ordinal sums (originally introduced by Clifford [4] for semigroups, compare also [5]).

Observe that for any interval partition $(\mathbf{I}_j)_{j\in J}$ of a given interval $\mathbf{I}$, and any corresponding system $(\mathbf{A}_j)_{j\in J}$ of conjunctive aggregation operators acting on intervals $\mathbf{I}_j$, respectively, the ordinal sum conjunctive aggregation operator $\mathbf{A}$ acting on $\mathbf{I}$ is given by $\mathbf{A}(x_1, \ldots, x_n) = \mathbf{A}_j(y_1, \ldots, y_k)$, where $\min(x_1, \ldots, x_n) \in \mathbf{I}_j$, $k = \text{card}\{i \mid x_i \in \mathbf{I}_j\}$ and $y_m = x_{i_m}$, $m = 1, \ldots, k$, $i_m = \min\{p \mid \text{card}\{i \in \{1, \ldots, p\} \mid x_i \in \mathbf{I}_j\} = m\}$.

On the other hand, let $\mathbf{A}$ be a conjunctive aggregation operator on interval $\mathbf{I}$ and let $f: \mathbf{J} \to \mathbf{I}$ be an increasing surjection from interval $\mathbf{J}$ to $\mathbf{I}$. Let $g: \mathbf{I} \to \mathbf{J}$ be any function such that $g \circ f \le \mathrm{id}_{\mathbf{J}}$ and $\sup g \circ f = \sup \mathbf{J}$. Then $\mathbf{B} = g \circ \mathbf{A} \circ f$, i. e., $\mathbf{B}(x_1, \ldots, x_n) = g(\mathbf{A}(f(x_1), \ldots, f(x_n)))$, is a conjunctive aggregation operator on $\mathbf{J}$.

Typical classes of conjunctive aggregation operators fulfilling (B1) are triangular norms [9, 20], copulas [19] and quasi-copulas, i. e., 1-Lipschitz conjunctive aggregation operators [19]. Note that all continuous t-norms can be constructed from the standard product by the above mentioned transformation and ordinal sum construction methods.

As an example of a conjunctive aggregation operator $\mathbf{A}$ on [0, 1] not fulfilling (B1) recall, e. g., $\mathbf{A}(x_1, \ldots, x_n) = \prod_{i=1}^{n} x_i^i$.

3. Disjunctive aggregation operators.

Disjunctive aggregation operators are characterized by the boundedness property $\mathbf{A} \ge \mathrm{MAX}$. Typical examples are the sum on $[0, \infty]$, the product on $[1, \infty]$, MAX on arbitrary interval $\mathbf{I}$. Observe that for a given interval $\mathbf{I}$

and a fixed decreasing bijection $\varphi: \mathbf{I} \to \mathbf{I}$, the operator $\mathbf{B} = \varphi^{-1} \circ \mathbf{A} \circ \varphi$ is a conjunctive aggregation operator whenever $\mathbf{A}$ is disjunctive (and vice-versa). Consequently, all properties and construction methods for disjunctive operators can be straightforwardly derived from the properties and construction methods for conjunctive aggregation operators as discussed in Section 2.

In particular, we have disjunctive counterparts of Proposition 1 and of the ordinal sum construction method.

For any interval partition $(\mathbf{I}_j)_{j \in J}$ of a given interval $\mathbf{I}$, and any corresponding system $(\mathbf{A}_j)_{j \in J}$ of disjunctive aggregation operators acting on intervals $\mathbf{I}_j$, respectively, the ordinal sum disjunctive aggregation operator $\mathbf{A}$ acting on $\mathbf{I}$ is given by $\mathbf{A}(x_1, \ldots, x_n) = \mathbf{A}_j (y_1, \ldots, y_k)$, where $\max (x_1, \ldots, x_n) \in \mathbf{I}_j$, $k = \text{card}\{i \mid x_i \in \mathbf{I}_j\}$ and $y_m = x_{i_m}$, $m = 1, \ldots, k$, $i_m = \min\{p \mid \text{card}\{i \in \{1, \ldots, p\} \mid x_i \in \mathbf{I}_j\} = m\}$.

Proposition 2. An aggregation operator $\mathbf{A}$ defined on [0, 1] is a disjunctive aggregation operator satisfying (B1) if and only if for all $n \in N$, $x_1, \ldots, x_n \in [0, 1]$, $\text{card}\{i \mid x_i \neq 0\} \geq n - 1$, it holds

$$\mathbf{A}(x_1, \ldots, x_n) = \max(x_1, \ldots, x_n).$$

Note that both above mentioned (and many other) results can be derived from the corresponding results for conjunctive aggregation operators by means of the standard duality of aggregation operators, $\mathbf{A}^d(x_1, \ldots, x_n) = 1 - \mathbf{A}(1 - x_1, \ldots, 1 - x_n)$, i. e., applying $\varphi(x) = 1 - x$. However, in special case of 1-Lipschitz binary aggregation operators, we can use other type of duality, namely $\mathbf{A}^*(x, y) = x + y - \mathbf{A}(x, y)$ (for more details see [13]). In both cases, $\mathbf{A}$ is a disjunctive (and 1-Lipschitz) aggregation operator if and only if $\mathbf{A}^d$ and $\mathbf{A}^*$ are conjunctive.

4. Averaging aggregation operators

Averaging operators were discussed already in 30-ties of the previous century, see e. g. [11, 15]. Many nice characterization results are due to Aczel [1], compare also [7]. An averaging aggregation operator is characterized by the boundedness properties $\text{MIN} \leq \mathbf{A} \leq \text{MAX}$ and due to the monotonicity of $\mathbf{A}$ also equivalently by the idempotency of $\mathbf{A}$, i. e., $\mathbf{A}$

$(x, \ldots, x) = x$ for any $x \in \mathbf{I}$, $n \in N$. Typical examples are the arithmetic mean on any interval $\mathbf{I}$, the geometric mean on any subinterval $\mathbf{I} \subset [0, \infty]$, weighted means, OWA operators, quasi-arithmetic means (for details see, e. g. [2]).

Among construction methods recall first the idempotization of continuous jointly strictly monotone functions [2]. We propose a new construction method based on defuzzification of fuzzy quantities. Denote

$\mathbf{K} = \{K: [-\infty, +\infty] \to [0, +\infty] \mid \mathbf{K}$ is convex and $\mathbf{K}(x) = 0$ iff $x = 0\}$,

$\mathbf{F}_{\mathbf{I}} = \{\mathbf{f} : \mathbf{I} \to [-\infty, +\infty] \mid \mathbf{f}$ is continuous and strictly monotone$\}$

$\mathbf{G} = \{\mathbf{g} : [0, \infty] \to [0, 1] \mid \mathbf{g}$ is decreasing bijection$\}$.

For any input vector $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbf{I}^n$, we define a quasi-convex fuzzy quantity on $\mathbf{I}$, $\mu_{\mathbf{x}} : \mathbf{I} \to [0, 1]$, $\mu_{\mathbf{x}}(r) = \mathbf{g}\left(\sum_{i=1}^{n} K_i\left(\mathbf{f}_i(x_i) - \mathbf{f}_i(r)\right)\right)$, with $\mathbf{g} \in \mathbf{G}$, $\mathbf{f}_i \in \mathbf{F}_{\mathbf{I}}$, $K_i \in \mathbf{K}$, $i = 1, \ldots, n$.

For an arbitrary defuzzification method DEF compatible with the fuzzy maximum, the operator $\mathbf{A}$ defined on $\mathbf{I}$ by $\mathbf{A}(x_1, \ldots, x_n) = \mathrm{DEF}(\mu_{\mathbf{x}})$ is non-decreasing.

If DEF = MOM (mean of maxima), then $\mathbf{A}(x_1, \ldots, x_n) = \mathrm{MOM}(\mu_{\mathbf{x}})$ yields an averaging aggregation operator. Note that if all $K_i = \mathbf{K}$, $\mathbf{f}_i = \mathbf{f}$, we obtain a generalized version of the penalty based method introduced in [16]. In this approach, $\mathbf{K}$ describes the penalty $\mathbf{K}(x-y)$ we have to pay when replacing an input x by another input y, i. e., it is a version of distance (metric) function. The function $\mathbf{f}$ corresponds to the scale of measurement we are dealing with. The transformation $\mathbf{g}$ is only an isomorphism transforming non-negative values to the interval [0, 1], i. e., non-negative functions into fuzzy sets membership functions. In several cases $\mathbf{g}$ is irrelevant. For example, by the MOM defuzzification the induced aggregation operator $\mathbf{A}$ depends on applied K_i and $\mathbf{f}_i$ but not on $\mathbf{g}$. Our proposed method allows to incorporate weights (importances) into the aggregation easily, even in the case of non-symmetric aggregation operators (see the last example). Indeed, it is sufficient to multiply K_i "distance" functions by the corresponding weights w_i. Moreover, from the first of subsequent examples we see that it generalizes the classical least square method.

0Examples:

- $K_i(x) = \mathbf{K}(x) = x^2$, $\mathbf{f}_i(x) = w_i\,\mathbf{f}(x)$, $w_i > 0$,

$$A(x_1, \ldots, x_n) = f^{-1}\left(\frac{\sum_{i=1}^{n} w_i f(x_i)}{\sum_{i=1}^{n} w_i}\right)$$

- $K_i(x) = K(x) = \begin{cases} cx & \text{if } x \geq 0 \\ -x & \text{if } x < 0 \end{cases}$, where $c \in]0, \infty[$, $f_i = f$ arbitrary increasing, $A = Q_\alpha$ is the α - quantile with $\alpha = \frac{c}{1+c}$; if $c = 1$, i. e., $K(x) = |x|$, $A = \text{MED}$ is the median.
- $f_i = id_I$, $w_i > 0$, $K_1(x) = |x|$, $K_2(x) = x^2$,
$$A(x_1, x_2) = \text{med}\left(x_1, x_2 - \frac{w_1}{w_2}, x_2 + \frac{w_1}{w_2}\right).$$

5. Mixed aggregation operators

In this class all remaining aggregation operators (i. e., not included into conjunctive, disjunctive and averaging operators) are contained. Typical examples are the sum on $]-\infty, \infty[$, the product on $]0, \infty[$. These operators are often extensions of a conjunctive operator A_1 acting on I_1 and a disjunctive operator A_2 acting on I_2, $I = I_1 \cup I_2$ (for example uninorms or nullnorms [8, 3], but also compensatory operators introduced in [16, 21]). In fact, any nonidempotent aggregation operator possessing a neutral element e or annihilator a in the interior of I is necessarily a mixed operator. Another big class of mixed aggregation operators is obtained in the form $A(\mathbb{x}) = M(C(\mathbb{x}), D(\mathbb{x}))$, where M is a (binary) averaging aggregation operator, C is a conjunctive and D is a disjunctive aggregation operator. Recall, e. g., gamma operator of Zimmermann and Zysno [23] or several types of compensatory operators discussed in [16].

A nice example of binary mixed aggregation operators can be derived from [12]: any 1-Lipschitz binary aggregation operator A on $[0, 1]$ with a neutral element e (an annihilator a) from $]0, 1[$ is mixed operator. Moreover, if $e \in]0, 1[$ then

$$A(x,y)=\begin{cases} e\,C\left(\dfrac{x}{e},\dfrac{y}{e}\right) & \text{if } (x,y)\in[0,e]^2,\\ e+(1-e)D\left(\dfrac{x-e}{1-e},\dfrac{y-e}{1-e}\right) & \text{if } (x,y)\in[e,1]^2,\\ x+y-e & \text{otherwise,}\end{cases}$$

where C is some quasi-copula and D is some dual quasi-copula. Similarly, if $a \in]0, 1[$,

$$A(x,y)=\begin{cases} a\,D\left(\dfrac{x}{a},\dfrac{y}{a}\right) & \text{if } (x,y)\in[0,a]^2,\\ a+(1-a)C\left(\dfrac{x-a}{1-a},\dfrac{y-a}{1-a}\right) & \text{if } (x,y)\in[a,1]^2,\\ a & \text{otherwise.}\end{cases}$$

6. Concluding remarks

We have discussed four basic classes of aggregation operators and some construction methods.

A typical general method of constructing more complex aggregation operators from simpler ones is the standard composition method $D = A(B_1, \ldots, B_k)$, i. e., $D(x_1, \ldots, x_n) = A(B_1(x_1, \ldots, x_n), \ldots, B_k(x_1, \ldots, x_n))$.

It is evident that any of the conjunctive, disjunctive or averaging class is preserved by this method (this is not true for the classes of mixed operators). Moreover, if the outer operator A is averaging, then if all $B_1, \ldots, B_k$ belong to the same of four discussed classes also the composite operator D belongs to that class.

Another general construction method is the ordinal sum of aggregation operators as proposed in [5]: for any strictly monotone sequence $0 = a_0 < a_1 < \ldots a_k = 1$ and a system $(A_i)_{i=1}^{k}$ of aggregation operators acting on $[a_{i-1}, a_i]$, respectively, their ordinal sum A: $\bigcup_{n\in N}[0,1]^n \to [0, 1]$ is given by

$$A(x_1, \ldots, x_n) = \sum_{i=1}^{k} \left(A_i\left(\varphi_i(x_1), \ldots, \varphi_i(x_n)\right) - a_{i-1}\right),$$

where $\varphi_i : [0, 1] \to [a_i, a_{i-1}]$ transforms an input $x \in [0, 1]$ to the closest element from $[a_{i-1}, a_i]$, i. e., $\varphi_i(x) = \min(a_i, \max(a_{i-1}, x))$, $i = 1, \ldots, k$.

Observe that this method preserves any of our four discussed classes. Moreover, if the operators $A_1, \ldots, A_k$ are conjunctive (disjunctive) and fulfilling (B1), then also the corresponding ordinal sum A has the same properties and, morever, it coincide with the ordinal sum of conjunctive (disjunctive) aggregation operators as described in Section 2 (in Section 3). Thus the ordinal sum of aggregation operators involves in one formula two different formulas commonly applied for t-norms (copulas, quasi-copulas) and for t-conorms (dual copulas, dual quasi-copulas). In our future research we will investigate another general construction methods extending the methods known for the specific cases.

Acknowledgment

The support of grants APVT-20-023402, GA ?R 402/04/1026 and VEGA 1/1145/04 is kindly announced.

References

1. Aczel J (1966) Lectures on Functional Equations and their applications. Acad. Press, New York
2. Calvo T, Kolesárová A, Komorníková M, Mesiar R (2002) Aggregation operators: Properties, classes and construction methods. In: Studies in Fuzziness and Soft Computing – Aggregation Operators, New Trend and Applications. Physica-Verlag, Heidelberg: 3 – 106
3. Calvo T, De Baets B, Fodor JC (2001) The functional equations of Alsina and Frank for uninorms and nullnorms. Fuzzy Sets and Systems 120: 385 – 394
4. Clifford AH (1954) Naturally totally ordered commutative semigroups. Amer. J. Math. 76: 631 – 646
5. De Baets B, Mesiar R (2002) Ordinal sums of aggregation operators. In: Bouchon – Meunier B et al. (eds) Technologies for constructing intelligent systems 2, Physica-Verlag, Heidelberg: 137 - 148
6. Dubois D, Prade H (2004) On the use aggregation operations in information fusion processes. Fuzzy Sets and Systems 142: 143 - 161
7. Fodor JC, Roubens M (1994) Fuzzy preference modelling and multicriteria decision support. Kluwer Acad. Publ., Dordrecht
8. Fodor JC, Yager RR, Rybalov A (1997): Structure of uninorms. Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems 5: 411 – 427

9. Klement EP, Mesiar R, Pap E (2000) Triangular Norms, Kluwer Acad. Publ., Dordrecht
10. Klir GJ, Folger TA (1988) Fuzzy Sets, Uncertainty and Information. Prentice Hall, Englewood Cliffs
11. Kolesárová A., Komorníková M (1999) Triangular norm-based iterative compensatory operators. Fuzzy Sets and Systems 104: 109 – 120
12. Kolesárová A (2003) 1-Lipschitz aggregation operators and quasi-copulas. Kybernetika 39, vol. 5: 615 - 629
13. Kolesárová A, Mordelová J (2001) 1-Lipschitz and kernel aggregation operators. In: Proc. AGOP'2001, Oviedo, pp 71 - 76
14. Kolmogoroff AN (1930) Sur la notion de la moyenne. Accad. Naz. Lincei Mem. Cl. Sci. Fis. Mat. Natur. Sez. 12: 388 – 391
15. Lehmke S, Thiele H, Kiseliova T (2000): Einführung in die Fuzzy-Logik. University Dortmund
16. Mesiar R (1995) Compensatory operators based on triangular norms. In: Proc. EUFIT'95, Aachen, pp 131 - 135
17. Mesiar R (1999) k-order additive measures. Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems 6: 561 – 568
18. Nagumo M (1930) Über eine Klasse der Mittelwerte. Japanese Journal of Mathematics 6: 71 – 79
19. Nelsen R (1999) Introduction to copulas. Springer Verlag, Heidelberg
20. Schweizer B, Sklar A (1983) Probabilistic metric spaces. Math. Holland. New York
21. Türksen IB (1992) Interval-valued fuzzy sets and "compensatory AND". Fuzzy Sets and Systems 51: 295 - 307
22. Yager RR, Rybalov A (1997) Noncommutative self-identity aggregation. Fuzzy Sets and Systems 85: 73 – 82
23. Zimmermann HJ, Zysno P (1980) Latent connectives in human decision making. Fuzzy Sets and Systems 4: 37 – 51

Homogeneous Aggregation Operators

Tatiana Rückschlossová

Department of Mathematics
Slovak University of Technology
Radlinského 11, 813 68 Bratislava
Slovak Republic

Abstract. Recently, the utilization of invariant aggregation operators, i.e., aggregation operators not depending on a given scale of measurement was found as a very current theme. One type of invariantness of aggregation operators is the homogeneity what means that an aggregation operator is invariant with respect to multiplication by a constant. We present here a complete characterization of homogeneous aggregation operators. We discuss a relationship between homogeneity, kernel property and shift-invariance of aggregation operators. Several examples are included.

Key words: aggregation operator, homogeneity, kernel property

1 Introduction

Aggregation operators are known as tools for aggregation (fusion) processes where from several input values one output value is required. As an example it could be mentioned the aggregation of infinitely many inputs [5, 8, 9], of inputs from ordinal scales [7] and also of complex inputs as distribution functions [14] or fuzzy sets [15]. Application of these operators in mathematics, physics, engineering, economics or social and other sciences verifies their broad utilization.

Values incoming into some aggregation process are related to a certain scale of measurement as well as output should be. Many decisions are based on the results of an appropriate aggregation and there is a need to realize the same decisions independently of a chosen scale of measurement, i.e. to apply aggregation operators reflecting this requirement. Rescaling the values we are dealing with is modeled by the transformation of an applied operator. An operator invariant under appropriate transformation is then an operator not depending on a given scale. According to the type of scales we can speak about several types of invariantnesses of aggregation operators.

Invariantness with respect to any scale which is rather restrictive has been studied for example in [12]. If we fix the unit of measurement but not the

T. Rückschlossová: *Homogeneous Aggregation Operators*, Advances in Soft Computing **2**, 555–563 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

beginning of our scale (i.e., "zero" is free), we have to deal with the shift-invariant aggregation operators (recall, e.g., the temperature measurement in degrees of Celsius and in degrees of Kelvin). These operators are completely characterized in [11], compare also [1]. Fixing the "zero" but letting free the unit (recall, e.g., the length measurement in meters and in yards), we come to the need of homogeneous aggregation operators. The aim of this paper is their complete description. The paper is organized as follows. In the next section we recall basic definitions. Section 3 is devoted to the characterization of homogeneous aggregation operators. Finally, in Section 4 we discuss the homogeneity of aggregation operators in relation to the kernel property and shift-invariantness of some related operators.

2 Preliminaries

Following [3, 9] recall the notions of an aggregation operator and a homogeneous operator.

Definition 1. *A mapping* $A : \bigcup_{n\in N}[0,1]^n \to [0,1]$ *is called an aggregation operator if it fulfils the following conditions:*

(A1) $A(x) = x$ *for each* $x \in [0,1]$,
(A2) $A(x_1,\ldots,x_n) \le A(y_1,\ldots,y_n)$ *whenever* $x_i \le y_i \ \forall\ i = 1,\ldots,n, n \in N$,
(A3) $A(0,\ldots,0) = 0$ *and* $A(1,\ldots,1) = 1$.

Definition 2. *An aggregation operator* $A : \bigcup_{n\in N}[0,1]^n \to [0,1]$ *is said to be homogeneous if* $\forall n \in N$, $\forall b \in]0,1[$, $\forall x_1,\ldots,x_n \in [0,1]$*:*

$$A(bx_1,\ldots,bx_n) = bA(x_1,\ldots,x_n).$$

By means of an arbitrary aggregation operator $C : \bigcup_{n\in N}[0,1]^n \to [0,1]$, a homogeneous operator $H^C : \bigcup_{n\in N}[0,1]^n \to [0,1]$ can be constructed as follows:

$$H^C(x_1,\ldots,x_n) = bC(\frac{x_1}{b},\ldots,\frac{x_n}{b}),$$

where $b = max(x_1,\ldots,x_n) > 0$ (and by convention we put $H^C(0,\ldots,0) = 0$). Notice, that by this construction each homogeneous operator is idempotent. Typical examples of homogeneous operators are weighted arithmetic means, including the standard arithmetic mean. However, although the homogeneity of an operator H^C is satisfied, the property of monotonicity need not be ensured. So the operator H^C need not be an aggregation operator, see example below.

Example 1. Take a product Π as an aggregation operator. Then the homogeneous operator $H^\Pi : \bigcup_{n\in N}[0,1]^n \to [0,1]$ is given by

$$H^\Pi(x_1,\ldots,x_n) = b\prod_{i=1}^{n}\frac{x_i}{b} = \frac{\prod_{i=1}^{n} x_i}{b^{n-1}}$$

with convention $\frac{0}{0} = 0$, i.e.,

$$H_{(n)}^{\Pi} = \frac{\Pi_{(n)}}{(max_{(n)})^{n-1}}.$$

Then $H_{(3)}^{\Pi}(0.5, 0.5, 0.5) = 0.5 > H_{(3)}^{\Pi}(0.5, 0.5, 1) = 0.25$, what contradicts the monotonicity of the operator $H_{(n)}^{\Pi}$.

Remark 1. In the class of quasi-arithmetic means (weighted quasi-arithmetic means), homogeneous aggregation operators form a 1-parameter subclass $(A_p)_{p\in]-\infty,\infty[}$ of so called power-root operators [6],

$$A_p(x_1, \ldots, x_n) = \left(\frac{1}{n}\sum_{n=1}^{n} {x_i}^p\right)^{\frac{1}{p}}$$

or

$$A_p(x_1, \ldots, x_n) = \left(\sum_{n=1}^{n} w_i {x_i}^p\right)^{\frac{1}{p}}$$

for $p \neq 0$ and $A_0 = G$ is the geometric mean (weighted geometric mean). For more details see [13].

Remark 2. An interesting class of homogeneous aggregation operators can be derived by minimalization of

$$\sum_{n=1}^{n} |x_i - a|^p$$

obtained in $a = B_p(x_1, \ldots, x_n)$, $p \in [1, \infty]$, see [4]. Note that $B_1 = \mathit{median}$, $B_2 = M$ (arithmetic mean), $B_\infty = \frac{\min + \max}{2}$ and $B_p(x_1, x_2) = \frac{x_1+x_2}{2}$ for all $p \in [1, \infty]$.

3 Characterization of Homogeneous Aggregation Operators

The open problem to characterize all aggregation operators C such that H^C is the aggregation operator was already stated in [3]. We give here a necessary and sufficient condition for an operator H^C to be an aggregation operator (to be monotonic).

Theorem 1. *Let $C : \bigcup_{n\in N}[0,1]^n \to [0,1]$ be an aggregation operator. Operator $H^C : \bigcup_{n\in N}[0,1]^n \to [0,1]$ is an aggregation operator if and only if for all $n \in N$, $\mathbf{x} = (x_1, \ldots, x_n), \mathbf{y} = (y_1, \ldots, y_n) \in [0,1]^n$, $\mathbf{x} \leq \mathbf{y}$, such that there is $i \in \{1, \ldots, n\}$ and $x_i = y_i = 1$, it holds (with convention $\frac{0}{0} = 1$)*

$$\frac{C(\mathbf{x})}{C(\mathbf{y})} \geq \min_k \frac{x_k}{y_k}. \tag{1}$$

Proof. By construction, $H^C(0,\dots,0)=0$, $H^C(1,\dots,1)=1$ and $H^C(x)=x$, for $x\in[0,1]$. To see the sufficiency it is enough to show the monotonicity of the operator H^C. Let $\mathbf{x}=(x_1,\dots,x_n),\mathbf{y}=(y_1,\dots,y_n)\in[0,1]^n$ and $\mathbf{x}\le\mathbf{y}$. Without loss of generality we may suppose that both $\mathbf{x}$ and $\mathbf{y}$ are non-zero vectors. Denote $\max\limits_k x_k=x_i$ and $\max\limits_k y_k=y_j$. Put $z_k=\min(x_i,y_k)$. Then evidently $\mathbf{x}\le\mathbf{z}\le\mathbf{y}$ while $\max z_k=z_i=x_i$. Let an aggregation operator C fulfil the condition (1). Following the definition of the homogeneous operator we have

$$\begin{aligned}
H^C(x_1,\dots,x_n)&= x_iC\left(\tfrac{x_1}{x_i},\dots,\tfrac{x_n}{x_i}\right)=x_iC\left(\tfrac{\mathbf{x}}{x_i}\right),\\
H^C(z_1,\dots,z_n)&= x_iC\left(\tfrac{\min(x_i,y_1)}{x_i},\dots,\tfrac{\min(x_i,y_n)}{x_i}\right)=x_iC\left(\min\left(1,\tfrac{\mathbf{y}}{x_i}\right)\right),\\
H^C(y_1,\dots,y_n)&= y_jC\left(\tfrac{y_1}{y_j},\dots,\tfrac{y_n}{y_j}\right)=y_jC\left(\tfrac{\mathbf{y}}{y_j}\right).
\end{aligned}$$

From the monotonicity of the aggregation operator C we have $H^C(\mathbf{x})\le H^C(\mathbf{z})$. Further, $\frac{y_k}{y_j}\le\min\left(1,\frac{y_k}{x_i}\right)$ for all $k=1,\dots,n$ and thus from the property (1) we have

$$\frac{C(\frac{\mathbf{y}}{y_j})}{C(\min(1,\frac{\mathbf{y}}{x_i}))}\ge\min_k\frac{\frac{y_k}{y_j}}{\min(1,\frac{y_k}{x_i})},$$

what means that

$$\frac{C(\frac{\mathbf{y}}{y_j})}{C(\min(1,\frac{\mathbf{y}}{x_i}))}\ge\min_k\max\left(\frac{y_k}{y_j},\frac{x_i}{y_j}\right)\ge\frac{x_i}{y_j},$$

and therefore

$$y_jC\left(\frac{\mathbf{y}}{y_j}\right)=H^C(\mathbf{y})\ge H^C(\mathbf{z})=x_iC\left(\min_k\left(1,\frac{y_k}{x_i}\right)\right).$$

Finally we have

$$H^C(\mathbf{x})\le H^C(\mathbf{z})\le H^C(\mathbf{y})\Rightarrow H^C(\mathbf{x})\le H^C(\mathbf{y}),$$

and the monotonicity of the operator H^C is proved. Thus H^C is an aggregation operator.

To see the necessity we prove the converse assertion. Assume that an aggregation operator C has not property (1), i.e., there is an index $i\in\{1,\dots,n\}$ and points $\mathbf{u},\ \mathbf{v}\in[0,1]^n$, $\mathbf{u}\le\mathbf{v}$, $u_i=v_i=1$ such that

$$\frac{C(\mathbf{u})}{C(\mathbf{v})}<\min_k\frac{u_k}{v_k}\le 1.$$

Create a point $\mathbf{w}$, for $j=1,\dots,n$, $w_j=v_j\min\limits_k\frac{u_k}{v_k}$. Then evidently $\mathbf{w}\le\mathbf{u}$. As the operator H^C is homogeneous we obtain that $H^C(\mathbf{u})=C(\mathbf{u})$ and

$H^C(\mathbf{w}) = \min_k \frac{u_k}{v_k} C(\mathbf{v})$, what implies $H^C(\mathbf{u}) < H^C(\mathbf{w})$, i.e., the operator H^C is not monotonic. □

Example 2. An example of a class of homogeneous aggregation operators is the class of weighted geometric means. For fixed n a weighted geometric mean G^W is given by

$$G^W(x_1, \ldots, x_n) = \prod_{i=1}^{n} x_i^{w_i},$$

where $\mathbf{w} = (w_1, \ldots, w_n) \in [0,1]^n$ denotes so called weighting vector fulfilling $\sum_{i=1}^n w_i = 1$, $i \in \{1, \ldots, n\}$, $n \in N$.

For a symmetric weighting vector $\mathbf{w} = (\frac{1}{n}, \ldots, \frac{1}{n})$ we obtain the standard geometric mean G,

$$G(x_1, \ldots, x_n) = \prod_{i=1}^{n} x_i^{\frac{1}{n}},$$

For the illustration in the case of $n = 2$ it is given by $G(x_1, x_2) = \sqrt{x_1 x_2}$, see Fig.1.

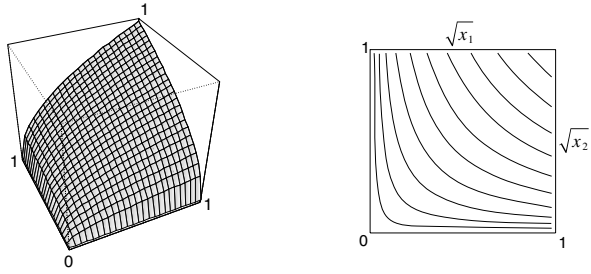


Fig. 1. 3D graph and contour plot of $G(x_1, x_2)$

For a non symmetric weighting vector $\mathbf{w} = (\frac{1}{4}, \frac{3}{4})$ we have corresponding weighted geometric mean $G^W(x_1, x_2) = {x_1}^{\frac{1}{4}} {x_2}^{\frac{3}{4}}$ illustrated by Fig.2.

Example 3. Another class of homogeneous aggregation operators is the class of ordered weighted geometric means. Before applying an ordered weighted geometric mean $\bar{G}^W$, the input n-tuple $(x_1, \ldots, x_n)$ is first rearranged into a non-decreasing permutation $(\bar{x}_1, \ldots, \bar{x}_n)$ and then we have

$$\bar{G}^W(x_1, \ldots, x_n) = \prod_{i=1}^{n} \bar{x}_i^{w_i},$$

with corresponding weighting vector $\mathbf{w} = (w_1, \ldots, w_n) \in [0,1]^n$, $\sum_{i=1}^n w_i = 1$, $i \in \{1, \ldots, n\}$, $n \in N$. Having for $n = 2$ the same weighting vector $\mathbf{w} = (\frac{1}{4}, \frac{3}{4})$ we obtain an operator $\bar{G}^W$ different from G^W, see Fig.3.

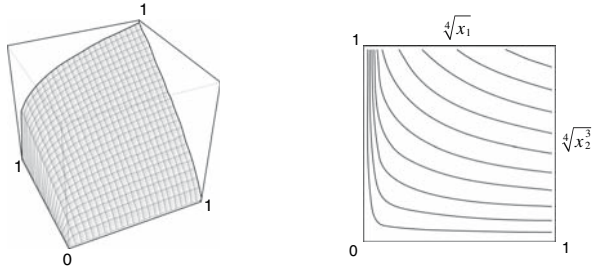


Fig. 2. 3D graph and contour plot of $G^W(x_1, x_2)$

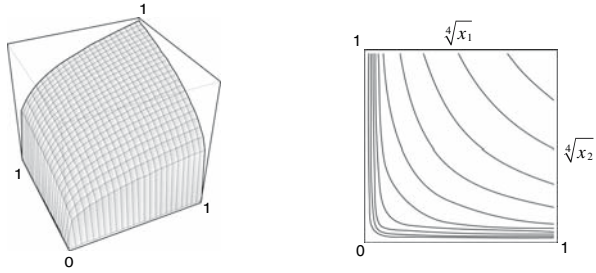


Fig. 3. 3D graph and contour plot of $\bar{G}^W(x_1, x_2)$

4 Homogeneity and Kernel Property

A similar open problem concerning the characterization of all shift-invariant aggregation operators has been solved in [11], where the close relation between shift-invariance of aggregation operators and the kernel property was shown. The kernel property has been recently introduced in [2, 10], see the next definition.

Definition 3. *An aggregation operator* $A : \bigcup_{n\in N}[0,1]^n \to [0,1]$ *is called a kernel aggregation operator if for all* $n \in N$, $(x_1, \ldots, x_n), (y_1, \ldots, y_n) \in [0,1]^n$, *it holds*

$$|A(x_1, \ldots, x_n) - A(y_1, \ldots, y_n)| \leq \max_k |x_k - y_k|. \tag{2}$$

This property of an aggregation operator A is linked to the Chebychev norm $\|A\|_\infty$ of an aggregation operator A, where

$$\|A\|_\infty = \sup \left\{ \frac{|A(x_1, \ldots, x_n) - A(y_1, \ldots, y_n)|}{\max_k |x_k - y_k|} \right\},$$

for all $(x_1, \ldots, x_n),\ (y_1, \ldots, y_n) \in [0,1]^n$, corresponding namely with the case $\|A\|_\infty = 1$.

Observe that the condition (1) for an aggregation operator C can be reformulated as

$$\frac{\min(C(\mathbf{x}), C(\mathbf{y}))}{\max(C(\mathbf{x}), C(\mathbf{y}))} \geq \min_k \frac{\min(x_k, y_k)}{\max(x_k, y_k)} \tag{3}$$

or, equivalently,

$$\left| \log \frac{C(\mathbf{x})}{C(\mathbf{y})} \right| \leq \max_k \left| \log \frac{x_k}{y_k} \right|,$$

i.e.,

$$|\log C(x) - \log C(y)| \leq \max_k |\log x_k - \log y_k|. \tag{4}$$

The property (4) of an aggregation operator C will be called log-kernel property and as far as the property (1) deals with one coordinate fixed in one, it can be denoted as one-log-kernel property. Let us define an operator $D : \bigcup_{n \in N} [0, \infty]^n \to [0, \infty]$ as follows:

$$D(u_1, \ldots, u_n) = -\log(C(\exp(-u_1), \ldots, \exp(-u_n))).$$

Then (4) is eqivalent to the kernel property of D,

$$|D(u_1, \ldots, u_n) - D(v_1, \ldots, v_n)| \leq \max_k |u_k - v_k|. \tag{5}$$

Thus we can use kernel aggregation operator D (it is sufficient to deal with so called zero-kernel property , fixing one coordinate to be zero) acting on $[0, \infty]$ to define a homogeneous aggregation operator H^C, where

$$C(x_1, \ldots, x_n) = \exp(D(-\log(x_1), \ldots, -\log(x_n))).$$

Moreover, if D is shift-invariant on $[0, \infty]$ then already C is homogeneous aggregation operator on $[0, 1]$.

Example 4.

a) If D is a weighted arithmetic mean, i.e., $D(u_1, \ldots, u_n) = \sum_{i=1}^n w_i x_i$ then

$$C(x_1, \ldots, x_n) = \exp\left(-\sum_{i=1}^n w_i - \log x_i \right) = \prod_{i=1}^n x_i^{w_i} = H^C(x_1, \ldots, x_n)$$

is the weighted geometric mean. Similarly OWA operator D leads to $C = H^C$ which is ordered weighted geometric mean.

b) Let D be any Choquet-integral aggregation operator based on fuzzy measures. For example, for $n = 2$,

$$D(u_1, u_2) = \begin{cases} \frac{1}{3}u_1 + \frac{2}{3}u_2 & if\ u_1 \leq u_2, \\ \\ \frac{3}{4}u_1 + \frac{1}{4}u_2 & otherwise. \end{cases}$$

Then

$$C(x_1, x_2) = H^C(x_1, x_2) = \begin{cases} \sqrt[3]{x_1 {x_2}^2} & if\ x_1 \geq x_2, \\ \\ \sqrt[4]{{x_1}^3 x_2} & otherwise, \end{cases}$$

is a homogeneous binary aggregation operator.

c) Finally, consider $D(u_1, u_2) = \sqrt{1 + |u_1 - u_2|} + \min(u_1, u_2) - 1$. Then $C = H^C$,

$$C(x_1, x_2) = \exp\left(1 - \min(-\log x_1, -\log x_2) - \sqrt{1 + |-\log x_1 + \log x_2|}\right)$$

$$= \max(x_1, x_2) \exp\left(1 - \sqrt{1 + \left|\log \tfrac{x_2}{x_1}\right|}\right).$$

5 Conclusion

We have investigated homogeneous aggregation operators. Especially, we have characterized all aggregation operators leading to homogeneous aggregation operators by showing the relationship between homogeneity and kernel (exactly one-log-kernel) property of an agregation operator. According to characterization of shift-invariant aggregation operators we have stressed the fact that it is sufficient to take into account shif-invariant aggregation operators acting on $[0, \infty]$ to obtain homogeneous aggregation operators. Several classes of aggregation operators as examples of classes of homogeous aggregation operators were presented.

Acknowledgments

The work on this paper was supported by grants APVT-20-046402 and VEGA 1/0273/03.

References

1. J. Aczél, D. Gronau, J. Schwaiger: *Increasing solutions of the homogeneity equation and similar equations.* Journal of Mathematical Analysis and Applications 182, 1994, pp. 436-464
2. T. Calvo, R. Mesiar: *Stability of aggregation operators.* Proc. EUSFLAT'2001, Leicester, 2001, pp. 475-478
3. T. Calvo, A. Kolesárová, M. Komorníková, R. Mesiar: *Aggregation Operators: Basic Concepts, Issues and Properties.* In: T. Calvo, G. Mayor and R. Mesiar (eds)., Aggregation Operators. New trends and applications, Physica-Verlag, Heidelberg, 2002, pp. 3-105
4. T. Calvo, R. Mesiar, R. R. Yager: *Quantitative Weights and Aggregation.* IEEE Transations on Fuzzy Systems, vol.12, No 1, February 2004
5. D. Denneberg: *Non-additive Measure and Integral.* Kluwer Academic Publishers, Dordrecht, 1994
6. J.J. Dujmovic: *Weighted Conjuctive and Disjunctive Means and Their Application in System Evaluation.* Univ. Beograd Publ. Elektrotech. Fak., 1974, pp. 147-158

7. M. Grabisch: *Symmetric and asymmetric integrals: the ordinal case.* Proceedings IIZUKA'2000, Iizuka, 2000, CD-rom
8. M. Grabisch, T. Murofushi, M. Sugeno, eds.: *Fuzzy Measures and Integrals. Theory and Applications.* Physica-Verlag, Heidelberg, 2000
9. G.J. Klir, T.A. Folger: *Fuzzy Sets, Uncertainty, and Information.* Prentice Hall, Englewood Cliffs, 1988
10. A. Kolesárová, J. Mordelová: *1-Lipschitz and kernel aggregation operators.* Proceedings of AGOP'2001, Oviedo, 2001, pp. 71-75
11. J. Lázaro, T. Rückschlossová, T. Calvo: *Shift invariant binary aggregation operators.* Fuzzy Sets and Systems, vol.142, february 2004, pp. 51-62
12. R. Mesiar, T. Rückschlossová: *Characterization of invariant aggregation operators.* Fuzzy Sets and Systems, vol.142, february 2004, pp. 63-73
13. T. Rückschlossová: *Aggregation Operators and Invariantness.* PhD Thesis, Slovak University of Technology, Bratislava, 2003
14. B. Schweizer, A. Sklar: *Probabilistic Metric Spaces.* North-Holland, New York, 1983
15. L.A. Zadeh: *Fuzzy Sets.* Inform. Control 8, 1965, pp. 338-353

1-Lipschitz Aggregation Operators, Quasi-Copulas and Copulas with Given Opposite Diagonal

Erich Peter Klement[1] and Anna Kolesárová[2]

[1] Department of Knowledge-Based Mathematical Systems, Johannes Kepler University, Linz (Austria)
ep.klemen@jku.at

[2] Department of Mathematics, Faculty of Chemical and Food Technology, Slovak University of Technology, Bratislava (Slovakia)
kolesaro@cvt.stuba.sk

Abstract. Copulas with given diagonal have been studied in [4, 10]. In [2, 5, 11] smallest and greatest (quasi-)copulas with given diagonal are constructed. Both (two-dimensional) copulas and quasi-copulas are special cases of binary 1-Lipschitz aggregation operators [3, 8], and in [7] 1-Lipschitz aggregation operators with given diagonal (and the consequences for (quasi-)copulas) are investigated. We give constructions for smallest and greatest 1-Lipschitz aggregation operators with given opposite diagonal, allowing us to obtain most results for (quasi-)copulas with given opposite diagonal as special cases.

Key words: 1-Lipschitz aggregation operator, quasi-copula, copula

1 Introduction

For the notions in this paper we refer to [3, 9], we only repeat the definitions necessary for the understanding of the results.

Definition 1.

(i) A *(binary) aggregation operator* is a function $A\colon [0,1]^2 \to [0,1]$ which is non-decreasing in each component and satisfies $A(0,0) = 0$ and $A(1,1) = 1$.

(ii) An aggregation operator A satisfying the Lipschitz condition with constant 1, i.e.,

$$|A(x,y) - A(x^*,y^*)| \leq |x - x^*| + |y - y^*| \qquad \text{for all } x, x^*, y, y^* \in [0,1],$$

will be called a 1-*Lipschitz aggregation operator*.

Many well-known binary aggregation operators, such as the arithmetic mean, the product, the minimum, the maximum, and weighted means are 1-Lipschitz aggregation operators (for more details see, e.g., [3]).

E.P. Klement and A. Kolesárová: *1-Lipschitz Aggregation Operators, Quasi-Copulas and Copulas with Given Opposite Diagonal*, Advances in Soft Computing **2**, 565–571 (2005)
www.springerlink.com © Springer-Verlag Berlin Heidelberg 2005

Also copulas [9] and quasi-copulas [1,6] are special 1-Lipschitz aggregation operators.

Definition 2.

(a) A *(two-dimensional) copula* is a function $C\colon [0,1]^2 \to [0,1]$ such that
  (i) $C(0,x) = C(x,0) = 0$ and $C(x,1) = C(1,x) = x$ for all $x \in [0,1]$,
  (ii) C is 2-alternating, i.e., $C(x,y) + C(x^*,y^*) \geq C(x^*,y) + C(x,y^*)$ for all $x, x^*, y, y^* \in [0,1]$ with $x \leq x^*$ and $y \leq y^*$.

(b) A *(two-dimensional) quasi-copula* is a function $Q\colon [0,1]^2 \to [0,1]$ such that
  (i) $Q(0,x) = Q(x,0) = 0$ and $Q(x,1) = Q(1,x) = x$ for all $x \in [0,1]$,
  (ii) Q is non-decreasing in each compoment,
  (iii) Q is 1-Lipschitz.

Obviously, each copula is a quasi-copula but not vice versa, and a 1-Lipschitz aggregation operator $A\colon [0,1]^2 \to [0,1]$ is a quasi-copula if and only if $A(0,1) = A(1,0) = 0$.

Each 1-Lipschitz aggregation operator A satisfies $W \leq A \leq W^*$, where $W(x,y) = \max(0, x+y-1)$ and $W^*(x,y) = \min(1, x+y)$. Each quasi-copula Q satisfies

$$W \leq Q \leq M \tag{1}$$

with $M(x,y) = \min(x,y)$, and the same holds for copulas. To simplify some formulas, we shall also use the infix notations $x \wedge y$ for $\min(x,y)$ and $x \vee y$ for $\max(x,y)$.

2 1-Lipschitz aggregation operators with given opposite diagonal section

In this section we show that the set of 1-Lipschitz aggregation operators with given opposite diagonal section possesses a greatest and a smallest element.

Given a 1-Lipschitz aggregation operator A, its opposite diagonal section $\omega_A\colon [0,1] \to [0,1]$ is given by $\omega_A(x) = A(x, 1-x)$. For an arbitrary 1-Lipschitz aggregation operator A we can only say that ω_A is a 1-Lipschitz function from $[0,1]$ to $[0,1]$.

It is not difficult to see that, as a consequence of its monotonicity and its 1-Lipschitz property, for each 1-Lipschitz aggregation operator A and for all $(x,y) \in [0,1]^2$ we have

$$A(x,y) \leq W(x,y) + \bigwedge\{\omega_A(z) \mid z \in [x \wedge (1-y),\, x \vee (1-y)]\}. \tag{2}$$

We start with an arbitrary 1-Lipschitz function $\omega\colon [0,1] \to [0,1]$ and look whether there is some 1-Lipschitz aggregation operator A such that for all $x \in [0,1]$ we have $\omega(x) = A(x, 1-x)$, i.e., whose opposite diagonal section

coincides with ω, and try to identify the greatest and smallest 1-Lipschitz aggregation operators with this property, provided they exist.

Motivated by (2), we obtain the following result:

Proposition 1. *For each* 1*-Lipschitz function* $\omega\colon [0,1] \to [0,1]$*, the function* $F_\omega\colon [0,1]^2 \to \mathbb{R}$ *defined by*

$$F_\omega(x,y) = W(x,y) + \bigwedge\{\omega(z) \mid z \in [x \wedge (1-y),\, x \vee (1-y)]\} \tag{3}$$

is a non-decreasing 1*-Lipschitz function with* $F_\omega(x, 1-x) = \omega(x)$ *for all* $x \in [0,1]$.

For example, for the trivial functions $\omega_0, \omega_1\colon [0,1] \to [0,1]$ given by $\omega_0(x) = 0$ and $\omega_1(x) = 1$ we obtain $F_{\omega_0} = W$ and $F_{\omega_1} = W + 1$. Note that F_{ω_1} is not an aggregation operator because of $\mathrm{Ran}(F_{\omega_1}) = [0,2]$. Indeed, in general we only know $F_\omega(0,0) \geq 0$ and $F_\omega(1,1) \geq 1$. Therefore, the function F_ω defined by (3) is a 1-Lipschitz aggregation operator if and only if it satisfies the boundary conditions for aggregation operators:

Proposition 2. *Let* $\omega\colon [0,1] \to [0,1]$ *be a* 1*-Lipschitz function and assume that* $F_\omega\colon [0,1]^2 \to \mathbb{R}$ *is as in* (3)*. Then the function* $\overline{A}_\omega\colon [0,1]^2 \to [0,1]$ *defined by*

$$\overline{A}_\omega = F_\omega \wedge W^* \tag{4}$$

is the greatest 1*-Lipschitz aggregation operator with opposite diagonal section* ω*.*

As an immediate consequence of Proposition 2 we get:

Corollary 1. *Let* $\omega\colon [0,1] \to [0,1]$ *be a* 1*-Lipschitz function. The function* $F_\omega\colon [0,1]^2 \to [0,1]$ *defined by* (3) *is the greatest* 1*-Lipschitz aggregation operator with opposite diagonal section* ω *if and only if* ω *satisfies* $\bigwedge\{\omega(z) \mid z \in [0,1]\} = 0$.

Note that a 1-Lipschitz aggregation operator A has opposite diagonal section ω_A if and only if the 1-Lipschitz aggregation operator A^* given by $A^*(x,y) = x + y - A(x,y)$ has opposite diagonal section ω_{A^*}, the latter being given by $\omega_{A^*}(x) = 1 - \omega_A(x)$.

Since the transition from A to A^* reverses the order between aggregation operators, for each 1-Lipschitz function $\omega\colon [0,1] \to [0,1]$ the smallest 1-Lipschitz aggregation $\underline{A}_\omega$ operator with opposite diagonal section is given by $\underline{A}_\omega = \left(\overline{A}_{\omega^*}\right)^*$. To be precise, in analogy to Propositions 1 and 2 and Corollary 1 we get:

Corollary 2. *Let* $\omega\colon [0,1] \to [0,1]$ *be a* 1*-Lipschitz function.*

(i) *The function* $G_\omega\colon [0,1]^2 \to \mathbb{R}$ *defined by*

$$G_\omega(x,y) = W^*(x,y) - 1 + \bigvee\{\omega(z) \mid z \in [x \wedge (1-y),\, x \vee (1-y)]\} \quad (5)$$

is a non-decreasing 1*-Lipschitz function with* $G_\omega(x, 1-x) = \omega(x)$ *for all* $x \in [0,1]$.

(ii) *The function* $\underline{A}_\omega\colon [0,1]^2 \to [0,1]$ *defined by* $\underline{A}_\omega = G_\omega \vee W$ *is the smallest* 1*-Lipschitz aggregation operator with opposite diagonal section* ω.

(iii) *The function* G_ω *is the smallest* 1*-Lipschitz aggregation operator with opposite diagonal section* ω *if and only if* $\bigvee\{\omega(z) \mid z \in [0,1]\} = 1$.

Example 1. Consider the opposite diagonal sections $\omega_W, \omega_M, \omega_\Pi\colon [0,1] \to [0,1]$ of W, M and Π given by $\omega_W(x) = 0$, $\omega_M(x) = x \wedge (1-x)$ and $\omega_\Pi(x) = x \cdot (1-x)$, respectively.

(i) W is the only 1-Lipschitz aggregation operator with opposite diagonal section ω_W.

(ii) The smallest 1-Lipschitz aggregation operator with opposite diagonal section ω_M is $(\langle 0, \frac{1}{2}, W\rangle, \langle \frac{1}{2}, 1, W\rangle)$, i.e., an ordinal sum of two copies of the Fréchet-Hoeffding lower bound W. Trivially, M is the greatest 1-Lipschitz aggregation operator with opposite diagonal section ω_M.

(iii) The greatest 1-Lipschitz operator $\overline{A}_{\omega_\Pi}$ and the smallest 1-Lipschitz operator $\underline{A}_{\omega_\Pi}$ with opposite diagonal section ω_Π are given by

$$\overline{A}_{\omega_\Pi}(x,y) = \begin{cases} (x \wedge y) \cdot (1 - x \wedge y) & \text{if } x + y \le 1, \\ W(x,y) + (x \vee y) \cdot (1 - x \vee y) & \text{otherwise,} \end{cases}$$

$$\underline{A}_{\omega_\Pi}(x,y) = \begin{cases} (x + y - \frac{3}{4}) \vee 0 & \text{if } (x,y) \in \left[0, \frac{1}{2}\right]^2, \\ (x + y - 1) \vee \frac{1}{4} & \text{if } (x,y) \in \left]\frac{1}{2}, 1\right]^2, \\ x(1-x) & \text{if } x \in \left[0, \frac{1}{2}\right] \text{ and } y \in \left]1-x, 1-x^2\right], \\ y(1-y) & \text{if } y \in \left[0, \frac{1}{2}\right] \text{ and } x \in \left]1-y, 1-y^2\right], \\ y - (1-x)^2 & \text{if } x \in \left]\frac{1}{2}, 1\right] \text{ and } y \in \left[(1-x)^2, 1-x\right], \\ x - (1-y)^2 & \text{if } y \in \left]\frac{1}{2}, 1\right] \text{ and } x \in \left[(1-y)^2, 1-y\right], \\ W(x,y) & \text{otherwise.} \end{cases}$$

Example 2. Consider the 1-Lipschitz function $\omega\colon [0,1] \to [0,1]$ defined by $\omega(x) = x \wedge (1-x) \wedge \frac{1}{3}$. The greatest 1-Lipschitz operator $\overline{A}_\omega$ and the smallest greatest 1-Lipschitz operator $\underline{A}_\omega$ with opposite diagonal section ω are given by

$$\overline{A}_\omega = (\langle \tfrac{1}{3}, \tfrac{2}{3}, W\rangle),$$
$$\underline{A}_\omega(x,y) = x \wedge y \wedge ((x + y - \tfrac{2}{3}) \vee 0) \wedge ((x + y - 1) \vee \tfrac{1}{3}).$$

3 Consequences for quasi-copulas

Turning our attention to quasi-copulas with given opposite diagonal section, note first that the opposite diagonal section ω_Q of each quasi-copula Q must be a 1-Lipschitz function satisfying $\omega_W \leq \omega_Q \leq \omega_M$ because of (1). Note also that an arbitrary 1-Lipschitz function $\omega\colon [0,1] \to [0,1]$ satisfies $0 \leq \omega(x) \leq x \wedge (1-x)$ for each $x \in [0,1]$ if and only if $\omega(0) = \omega(1) = 0$.

Proposition 3. *Let $\omega\colon [0,1] \to [0,1]$ be a 1-Lipschitz function such that $\omega(0) = \omega(1) = 0$. Then we have:*

(i) *The function $F_\omega\colon [0,1]^2 \to [0,1]$ defined by (3) is the greatest quasi-copula with opposite diagonal section ω.*
(ii) *The function $\underline{A}_\omega\colon [0,1]^2 \to [0,1]$ defined by $\underline{A}_\omega = G_\omega \vee W$, where $G_\omega\colon [0,1]^2 \to [0,1]$ is as in (5), is the smallest quasi-copula with opposite diagonal section ω.*

Example 3. As a consequence of Proposition 3, all the greatest and smallest 1-Lipschitz aggregation operators with opposite diagonal sections ω_W, ω_M, ω_Π (considered in Example 1) and ω (considered in Example 2), respectively, are also the greatest and smallest quasi-copulas with the respective opposite diagonal section.

We also mention some results for duals of quasi-copulas with given opposite diagonal section.

Proposition 4. *Let $\omega\colon [0,1] \to [0,1]$ be a 1-Lipschitz function such that $\omega(0) = \omega(1) = 1$.*

(i) *The function $\overline{A}_\omega\colon [0,1]^2 \to [0,1]$ defined in (4) has opposite diagonal section ω, and it is the greatest dual of some quasi-copula with this property.*
(ii) *The function $G_\omega\colon [0,1]^2 \to [0,1]$ defined in (5) has opposite diagonal section ω, and it is the smallest dual of some quasi-copula with this property.*

4 Consequences for copulas

The greatest quasi-copula with given opposite diagonal section (given in Proposition 3) even turns out to be a copula:

Proposition 5. *Let $\omega\colon [0,1] \to [0,1]$ be a 1-Lipschitz function such that $\omega(0) = \omega(1) = 0$. Then the function F_ω defined by (3) is the greatest copula with opposite diagonal section ω.*

Example 4.

(i) As a consequence of Propositions 3 and 5, each greatest 1-Lipschitz aggregation operator with opposite diagonal section ω_W, ω_M, ω_Π (considered in Example 1) and ω (considered in Example 2), respectively, is also the greatest copula with the respective opposite diagonal section.

(ii) The smallest 1-Lipschitz aggregation operators with opposite diagonal sections ω_W and ω_M (considered in Example 1), respectively, are also the smallest copulas with the respective opposite diagonal section.

(iii) The smallest 1-Lipschitz aggregation operator $\underline{A}_{\omega_\Pi}$ with opposite diagonal section ω_Π (considered in Example 1) is the smallest quasi-copula with this property because of Proposition 3, but not a copula because of

$$\underline{A}_{\omega_\Pi}(\tfrac{3}{8},\tfrac{7}{16}) - \underline{A}_{\omega_\Pi}(\tfrac{3}{8},\tfrac{9}{16}) + \underline{A}_{\omega_\Pi}(\tfrac{5}{8},\tfrac{9}{16}) - \underline{A}_{\omega_\Pi}(\tfrac{5}{8},\tfrac{7}{16}) < 0.$$

(iv) Similarly, the smallest 1-Lipschitz aggregation operator $\underline{A}_{\omega}$ with opposite diagonal section ω (considered in Example 2) is the smallest quasi-copula with this property, but not a copula because of

$$\underline{A}_{\omega}(\tfrac{1}{3},\tfrac{1}{3}) - \underline{A}_{\omega}(\tfrac{1}{3},\tfrac{2}{3}) + \underline{A}_{\omega}(\tfrac{2}{3},\tfrac{2}{3}) - \underline{A}_{\omega}(\tfrac{2}{3},\tfrac{1}{3}) < 0.$$

Acknowledgement

We acknowledge the support of the grant VEGA 1/0085/03 and of the COST Action 274 TARSKI . This work was also supported by the Science and Technology Assistance Agency under contract No. APVT-20-023402.

References

1. C. Alsina, R. B. Nelsen, and B. Schweizer. On the characterization of a class of binary operations on distribution functions. *Statist. Probab. Lett.*, 17:85–89, 1993.
2. S. Bertino. On dissimilarity between cyclic permutations. *Metron*, 35:53–88, 1977. (Italian).
3. T. Calvo, A. Kolesárová, M. Komorníková, and R. Mesiar. Aggregation operators: properties, classes and construction methods, pages 3–104. Physica-Verlag, Heidelberg, 2002.
4. G. A. Fredricks and R. B. Nelsen. Copulas constructed from diagonal sections. In Beneš and Štěpán, editors. *Distributions with Given Marginals and Moment Problems*. Kluwer Academic Publishers, Dordrecht, pages 129–136, 1997.
5. G. A. Fredricks and R. B. Nelsen. The Bertino family of copulas. pages 81–91. Kluwer Academic Publishers, Dordrecht, 2002.
6. C. Genest, J. J. Quesada Molina, J. A. Rodríguez Lallena, and C. Sempi. A characterization of quasi-copulas. *J. Multivariate Anal.*, 69:193–205, 1999.

7. E. P. Klement and A. Kolesárová. 1-Lipschitz aggregation operators, quasi-copulas and copulas with given diagonals. To appear in *Proceedings SMPS 2004, Oviedo*.
8. A. Kolesárová. 1-Lipschitz aggregation operators and quasi-copulas. *Kybernetika (Prague)*, 39:615–629, 2003.
9. R. B. Nelsen. *An Introduction to Copulas*, Lecture Notes in Statistics 139. Springer, New York, 1999.
10. R. B. Nelsen and G. A. Fredricks. Diagonal copulas. In Beneš and Štěpán, editors. *Distributions with Given Marginals and Moment Problems*. Kluwer Academic Publishers, Dordrecht, pages 121–127, 1997.
11. R. B. Nelsen, J. J. Quesada Molina, J. A. Rodríguez Lallena, and M. Úbeda Flores. Best-possible bounds on sets of bivariate distribution functions. *J. Multivariate Anal.*, 2004 (in press).

Fuzzy Measures and Choquet Integral on Discrete Spaces

Yasuo Narukawa [1], Vicenç Torra [2]

[1]Toho Gakuen, 3-1-10 Naka, Kunitachi, Tokyo, 186-0004 Japan
narukawa@d4.dion.ne.jp
[2] Institut d'Investigació en Intel·ligència Artificial, Campus de Bellaterra, 08193 Bellaterra, Catalonia, Spain
vtorra@iiia.csic.es

Abstract. This paper studies some relationships between fuzzy relations, fuzzy graphs and fuzzy measure. It is shown that a fundamental theorem of Discrete Convex Analysis is derived from the theory of fuzzy measures and the Choquet integral.

Key words: Fuzzy relation, Fuzzy graph, Fuzzy measure, Choquet integral, Fuzzy integrals, Matroids

1 Introduction

Graphs are used to represent relations between objects. To include fuzziness in such relations fuzzy graphs were introduced. At present a large number and variety of applications have been developed that use graphs for knowledge representation. For example, they are used in the context of inference systems (*e.g.* in probabilistic and possibilistic networks [6, 8] or fuzzy cognitive graphs [17]), matching algorithms [16], or in dictionaries (as WordNet [13]), for easing information retrieval and recommendation systems [21]. Such applications have fostered the development of new tools for graphs and fuzzy graphs, as well as the development of other graph based formalisms (as *e.g.* in [3]).

Roughly speaking, fuzzy graphs have been defined adding fuzziness either on the vertexes or on the edges. At present several alternative definitions exists for fuzzy graphs, some of them can be found in [7].

Several theoretical results have been obtained for fuzzy graphs. See *e.g.* [1, 4, 5, 9]. In this paper we propose a method to define fuzzy measures for the subsets of vertexes in a graph. This measure is based on the fuzzy memberships of the vertexes. In some sense, the proposed measures are to evaluate the connectivity that a subset of vertexes can achieve.

Y. Narukawa and V. Torra: *Fuzzy Measures and Choquet Integral on Discrete Spaces*, Advances in Soft Computing **2**, 573–581 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

Choquet integral [11] (see also [10]) is a tightly related concept with fuzzy measures. In fact, they are defined to integrate functions with respect to the fuzzy measures. In this paper we give some results for Choquet integrals. Namely, we show that they can be used to represent some functions.

In this paper, we assume that the universal set N is a finite set, that is, $N := \{1, 2, \ldots, n\}$.

The structure of this paper is as follows. In Section 2 we give some preliminaries that are needed later on in this paper. In Section 3 the fuzzy measure for fuzzy graphs is proposed and studied. Section 4 is devoted to the results about the representation in terms of Choquet integrals. The paper finishes with some conclusions.

2 Preliminaries

Definition 1. *A set function $\mu : 2^N \to [0, 1]$ is a fuzzy measure if it satisfies the following axioms:*

(i) $\mu(\emptyset) = 0$, $\mu(N) = 1$ (boundary conditions)
(ii) $A \subseteq B$ implies $\mu(A) \leq \mu(B)$ (monotonicity) for A, $B \in 2^N$

Definition 2. *[11] (see also [2]) Let μ be a fuzzy measure on $(N, 2^N)$. The Choquet integral $C_\mu(x)$ of $x : N \to R_+$ with respect to μ is defined by*

$$C_\mu(x) = \sum_{j=1}^{n} x(a_{s(j)})(\mu(A_{s(j)}) - \mu(A_{s(j+1)}))$$

where $x_{s(i)}$ indicates that the indices have been permuted so that

$$0 \leq x(a_{s(1)}) \leq \cdots \leq x(a_{s(n)}), A_{s(i)} = \{a_{s(i)}, \cdots, a_{s(n)}\}, A_{s(n+1)} = \emptyset.$$

A function $x : N \to R_+$ is regarded as $|N| = n$ -dimensional vector, that is, $x \in R_+^n$.

Definition 1 Let $x, y \in R_+^n$. We say that x and y are comonotonic if

$$x_i < x_j \Rightarrow y_i \leq y_j$$

for $i, j \in N$.

A chain of sets in 2^N is a set system $M \subset 2^N$ which is completely ordered with respect to set inclusion, i. e.

$$A, B \in M \text{ implies } A \subset B \text{ or } B \subset A.$$

Definition 2 Let $\mathcal{I}$ be a real valued functional on R_+^n. We say:

(1) $\mathcal{I}$ is comonotonic additive if and only if for comonotonic $x, y \in R_+^n$,

$$\mathcal{I}(x+y) = \mathcal{I}(x) + \mathcal{I}(y),$$

(2) $\mathcal{I}$ is comonotonic monotone if and only if for comonotonic $x, y \in R_+^n$

$$x \leq y \Rightarrow \mathcal{I}(x) \leq \mathcal{I}(y).$$

As the conditions for a functional to be the Choquet integral, we have the next theorem.

Theorem 3 *[20] Let $\mathcal{I} : [0,1]^n \to R_+$ be comonotonic monotone and comonotonic additive functional with $\mathcal{I}(1_N) = 1$. There exists a fuzzy measure μ on $(N, 2^N)$ such that*

$$\mathcal{I}(x) = C_{\mu_\mu}(x)$$

for all $x \in R_+^n$.

Next we define a pseudo-addition.

Definition 4 *A binary operation $\oplus : [0,\infty) \times [0,\infty) \to [0,\infty)$ is called a pseudo-addition if the following properties are satisfied:*

(1) *(commutativity) $a \oplus b = b \oplus a$,*
(2) *(monotonicity) $a \leq a', b \leq b'$ implies $a \oplus b \leq a' \oplus b'$,*
(3) *(associativity) $(a \oplus b) \oplus c = a \oplus (b \oplus c)$*
(4) *(continuity) $a_n \to a$ and $b_n \to b$ imply $a_n \oplus b_n \to a \oplus b$,*
(5) *(zero element) $0 \oplus a = a \oplus 0$,*

for $a, b \in [0,\infty)$.

For fixed $p > 0$, let $x \oplus y := (x^p + y^p)^{1/p}$. Then $\oplus$ is a pseudo-addition. Using pseudo-addition $\oplus$, the Choquet integral is generalized [2].

3 Fuzzy Relations and Fuzzy Measures

Let N be a finite set and $\mathcal{R}$ be a fuzzy relation on N, that is, $\mathcal{R} \subset N \times N$, where $\mu_\mathcal{R} : N \times N \to [0,1]$ is its membership function. $(N, \mathcal{R}, \mu_\mathcal{R})$ is regarded as a fuzzy graph. Thus, the set $\mathcal{R}$ is defined by pairs (x_i, x_j) and corresponds to the edges of the graph and $\mu_\mathcal{R}$, their membership, is defined from $N \times N$ into [0,1] in such a way that for the $\mathcal{R}' = N \times N - \mathcal{R}$, we have: $mu_\mathcal{R}((x,y)) = 0$ for all $(x,y) \in \mathcal{R}'$. We say that $T \subset \mathcal{R}$ is a fuzzy tree if there exists no $x_i \in N$ $(2 \leq i \leq n)$ such that $(x_1, x_2), \ldots, (x_{i-1}, x_i) \in \mathcal{R}$ and $x_1 = x_i$. This is, there is no cicle in the graph. $\mathcal{T}_\mathcal{R}$ denotes the set of all fuzzy trees of fuzzy graph $(N, \mathcal{R}, \mu_\mathcal{R})$.

Definition 5 *Let $(N, \mathcal{R}, \mu_{\mathcal{R}})$ be a fuzzy graph. We define a set function $m : 2^{\mathcal{R}} \to [0, \infty)$ by*

$$m(A) := \sup\{ \bigoplus_{(x,y)\in I} \mu(x, y) | I \subset A, I \in \mathcal{T}_{\mathcal{R}}\}.$$

The next proposition follows from the definition.

Proposition 1. *Let $(N, \mathcal{R}, \mu_{\mathcal{R}})$ be a fuzzy graph and m be a set function defined in Definition 5.*

(1) $m(\emptyset) = 0$
(2) *(monotonicity)* $A \subset B$ *implies* $m(A) \leq m(B)$
(3) *($\oplus$submodularity)* $m(A) \oplus m(B) \geq m(A \cup B) \oplus m(A \cap B)$.

Define a set function $\nu : 2^{\mathcal{R}} \to [0, 1]$ by $\nu(A) := m(A)/m(N)$. Then, conditions (1) and (2) in the proposition above imply that the set function ν on $2^{\mathcal{R}}$ is a fuzzy measure.

Example 1 *Let $(N, \mathcal{R})$ be the fuzzy graph in Figure 1. That is, $N := \{A, B, C, D\}$ and a fuzzy relation $\{a, b, c, b\}$ as indicated in the figure.*

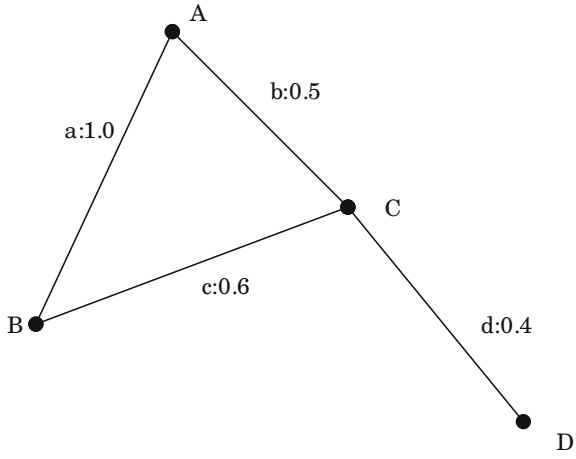


Fig. 1. Fuzzy Graph

If $\oplus := +$, then the fuzzy measure ν generated by the fuzzy graph in Figure 1 is the one in Table 1.

If $\oplus := \vee$, then the fuzzy measure ν' generated by the fuzzy graph in Figure 1 is the one in Table 2.

ν' is a possibility measure.

Let $\oplus = +$ and $\mu(x, y) = 1$ for all $(x, y) \in \mathcal{R}$, Then we have $m(A) \leq |A|$, monotonicity and submodularity. Therefore, m is a characteristic function of matroid [14, 18].

Table 1. Fuzzy measure generated by the fuzzy graph ($\oplus = +$)

set	$\emptyset$	$\{a\}$	$\{b\}$	$\{c\}$	$\{d\}$	$\{a,b\}$	$\{a,c\}$	$\{a,d\}$
ν	0	0.5	0.25	0.3	0.2	0.75	0.8	0.7
set	$\{b,c\}$	$\{b,d\}$	$\{c,d\}$	$\{a,b,c\}$	$\{a,b,d\}$	$\{a,c,d\}$	$\{b,c,d\}$	N
ν	0.55	0.45	0.5	0.8	0.95	1	0.75	1

Table 2. Fuzzy measure generated by the fuzzy graph ($\oplus = \vee$)

set	$\emptyset$	$\{a\}$	$\{b\}$	$\{c\}$	$\{d\}$	$\{a,b\}$	$\{a,c\}$	$\{a,d\}$
ν'	0	1	0.5	0.6	0.4	1	1	1
	$\{b,c\}$	$\{b,d\}$	$\{c,d\}$	$\{a,b,c\}$	$\{a,b,d\}$	$\{a,c,d\}$	$\{b,c,d\}$	N
	0.6	0.5	0.6	1	1	1	0.6	1

4 Simplex and Choquet Integral

First we define a simplex and a barycentric coordinate.

Definition 6 *Let $P_0, P_1, \ldots, P_k \in R^N$.*

(1) *We say that $\{P_0, P_1, \ldots, P_k\}$ is affinely independent if*

$$\sum_{i=0}^{k} \alpha_i = 0, \sum_{i=0}^{k} \alpha_i P_i = 0$$

and $\alpha_i \in R$ imply

$$\alpha_0 = \alpha_1 = \cdots = \alpha_k = 0.$$

(2) *Let us suppose that $\{P_0, P_1, \ldots, P_k\}$ is affinely independent, and define the subset $\sigma(P_0, P_1, \ldots, P_k) \subset R^N$ by*

$$\sigma(P_0, P_1, \ldots, P_k) := \{x | x = \sum_{i=0}^{k} \alpha_i P_i, \alpha_i \geq 0, \sum_{i=0}^{k} \alpha_i = 1\}.$$

Then, $\sigma(P_0, P_1, \ldots, P_k)$ is called a simplex.

(3) *Let $x \in \sigma(P_0, P_1, \ldots, P_k)$. In this case, if there exists unique non-negative real numbers $\alpha_0, \alpha_1, \ldots, \alpha_k$ such that $x = \sum_{i=0}^{k} \alpha_i P_i$, we say that $(\alpha_0, \alpha_1, \ldots, \alpha_k)$ is the barycentric coordinate of x.*

Example 2 *Consider Fig. 2 illustrated below. Triangles $\sigma(P_1, P_2, P_3), \sigma(P_1, P_3, P_4)$ and $\sigma(P_1, P_4, P_5)$ are simplex. $K := \{\sigma(P_1, P_2, P_3), \sigma(P_1, P_3, P_4), \sigma(P_1, P_4, P_5)\}$ is a complex and $|K| = \sigma(P_1, P_2, P_3) \cup \sigma(P_1, P_3, P_4) \cup \sigma(P_1, P_4, P_5)\}$*

We have the next proposition that follows immediately from the definition above.

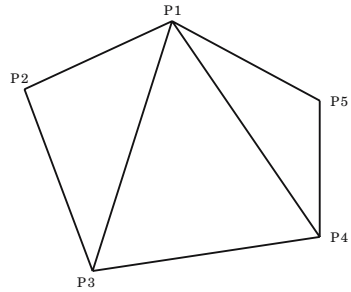


Fig. 2. Complex of $\{P_1, P_2, P_3, P_4, P_5\}$

Proposition 7 *Let $\mathcal{C}$ be a maximal chain, that is,*

$$\mathcal{C} : \emptyset = C_0 \subset C_1 \subset \ldots C_n = N,$$

and z is a point of integers, that is, $z \in Z^N$. Let $z_i := z + \chi_{C_i}$ where χ_{C_i} is a characteristic function of $C_i \in \mathcal{C}$ $(i = 0, 1, \ldots, n)$. Then, the set $\{z_0, z_1, \ldots, z_n\}$ is affinely independent. Therefore $\sigma(z_0, z_1, \ldots, z_n)$ is a simplex.

Next we define the complex and simplical subdivision.

Definition 8 *Let $\sigma(P_0, P_1, \ldots, P_k)$ be a simplex. Then, we say that subsimplexes $\sigma(P_{k_0}, P_{k_1}, \ldots, P_{k_i}), (i < k)$ are faces of $\sigma(P_0, P_1, \ldots, P_k)$ and that each P_i is a vertex.*

Let K be a set of simplex in R^N. We say that K is a complex if every face of $\sigma \in K$ belongs to K, and if for $\sigma_1, \sigma_2 \in K$, $\sigma_1 \cap \sigma_2 \neq \emptyset$ implies that $\sigma_1 \cap \sigma_2$ is a face of both σ_1 and σ_2.

We define the polyhedra $|K|$ by $|K| := \cup_{\sigma \in K} \sigma$.

Let X be a topological space and K be a complex. Then, if there exist a homeomorphic map $f : |K| \to X$, we say that (K, f) is a simplical subdivision of X.

Proposition 9 *Let K be a set of simplex $\sigma := \sigma(z_0, z_1, \ldots, z_n)$ for all $z \in Z^N$ and every maximal chain $\mathcal{C}$. Then $|K| = R^N$. Let $i : |K| \to R^N$ be an identity map. Then $(|K|, i)$ is a simplical subdivision of R^N.*

Example 3 *Let $|K| := R^2$ and $f := id$ (identity map). We illustrate the simplical subdivision of R^2.*

Proposition 2. *For every $x \in R^N$, $[x]$ denotes $[x](i) := [x(i)](i \in N)$ where $[x(i)]$ is the maximal integer less than $x(i)$.*

For $\tilde{x} = x - [x]$, $\tilde{x}(l_1) \geq \tilde{x}(l_2) \geq \cdots \geq \tilde{x}(l_n)$ and the maximal chain

$$\mathcal{C} : C_0 = \emptyset, C_i = \{l_1, l_2, \ldots, l_i\}(i = 1, 2, \ldots n),$$

denote $z'_i := [x] + \chi_{C_i}$, then we have $x \in \sigma(z'_0, z'_1, \ldots, z'_n)$ and a barycentric coordinate of x is $(1 - \tilde{x}(l_1), \tilde{x}(l_1) - \tilde{x}(l_2), \ldots, \tilde{x}(l_{n-1}) - \tilde{x}(l_n))$.

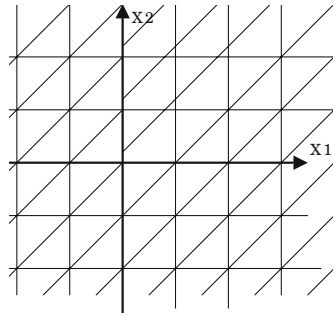


Fig. 3. Simplical subdivision of R^2

Let $f : Z^N \to R$, then, we define the piecewise linear extension (PL extension) $\hat{f}$ of f:

Definition 10 *Let $x \in R^N$. Then, there exists a simplex $\sigma(z_0, z_1, \ldots, z_n)$ such that $x \in \sigma(z_0, z_1, \ldots, z_n)$. Let $(\alpha_0, \alpha_1, \ldots, \alpha_n)$ be a barycentric coordinate of x, that is $x := \sum_{i=0}^{n} \alpha_i z_i$. Then we define the piecewise linear extension $\hat{f}$ of f by $\hat{f}(x) := \sum_{i=0}^{n} \alpha_i \hat{f}(z_i)$.*

Since $x, y \in \sigma(z_0, z_1, \ldots, z_n)$ is comonotonic and $\hat{f}$ is linear on $\sigma(z_0, z_1, \ldots, z_n)$, we have the next lemma.

Lemma 11 *$\hat{f}$ is comonotonically additive.*

It follows from the previous lemma that we can apply the representation theorem presented in [20].

Proposition 12 *The PL extension can be represented by Choquet integrals.*

Applying subadditivity theorem [11, 12], we have the next corollary, that is a fundamental theorem of discrete convex analysis [19].

Corollary 13 *[15] Set function $f : 2^N \to R$ is submodular if and only if its PL-extension $\hat{f} : [0, 1]^N \to R$ is convex.*

The corollary above can easily be extended to $\oplus$-submodular set function and extended Choquet integral.

5 Conclusion

In this paper we have studied some aspects related with fuzzy measures and fuzzy integrals. We show that a fuzzy relation induces a fuzzy measure. Conversely we can induce a fuzzy relation by some fuzzy measure. The Choquet integral is regarded as a natural extension of some set function (fuzzy measure). These fact says that theory of discrete convex analysis can be included in theory of theory of fuzzy measures and fuzzy integral.

Acknowledgements

The authors acknowledge R. Mesiar and the referees for their comments and suggestions. This work was partly funded by the Spanish MCYT (project TIC-2001-0633-C03-02) and the Generalitat de Catalunya (AGAUR, 2002XT 00111).

References

1. A. Alaoui, (1999), On fuzzification of some concepts of graphs, Fuzzy Sets and Systems, 101 363-389.
2. P. Benvenuti,R. Mesiar,D. Vivona,(2002), Monotone Set Functions-Based Integrals, in *Handbook of Measure Theory*, E. Pap, (Ed.), Elsevier.
3. M. J. Beynon, (2003), The introduction and utilization of (l, u)-graphs in the extended variable precision rough sets model, Int. J. of Intel. Systems, 18:10 1035-1055.
4. K. R. Bhutani, A. Battou, (2003), On M-strong fuzzy graphs, Information Sciences, 155 103-109.
5. K. R. Bhutani, A. Rosenfeld, (2003), Strong arcs in fuzzy graphs, Information Sciences 152 319-322.
6. R. Blanco, I. Inza, P. Larranaga, (2003), Learning Bayesian networks in the space of structures by estimation of distribution algorithms, Int. J. of Intel. Systems, 18:2 205-220.
7. M. Blue, B. Bush, J. Puckett, (2002), Unified approach to fuzzy graph problems, Fuzzy Sets and Systems 125 355-368.
8. C. Borgelt, R. Kruse, (2003), Learning Possibilistic Graphical Models From Data, IEEE Trans. on Fuzzy Systems 11:2 159-172.
9. A. Boulmakoul, (2004), Generalized path-finding algorithms on semirings and the fuzzy shortest path problem, J. of Computational and Applied Mathematics, 162:1 263-272.
10. T. Calvo, A. Kolesarova, M. Komornikova, R. Mesiar, (2002), Aggregation Operators: Propertes, Classes and Construction Methods, in T. Calvo, G. Mayor, R. Mesiar, Eds., Aggregation operators: New trends and applications, Physica-Verlag, Springer, 3-104.
11. G. Choquet, (1954), Theory of Capacities, *Ann. Inst. Fourier* 5, 131-296.
12. D. Denneberg, (1997(2nd ed)), Non-additive measure and integral, Kluwer.
13. C. Fellbaum, (1998), WordNet: An Electronic Lexical Database, The MIT Press
14. S. Fujishige,(1991), Submodular Functions and Optimization, North-Holland.
15. L. Lovász,(1983),Submodular functions and convexity,in Mathematical programming : the state of the art, Bonn 1982 / edited by A. Bachem, M. Grotschel, B. Korte, Springer-Verlag.
16. S. Medasani, R. Krishnapuram, Y. Choi, (2001),Graph Matching by Relaxation of Fuzzy Assignments, IEEE Trans. on Fuzzy Systems 9:1 173-182.
17. Y. Miao, Z.-Q. Liu, (2000), On Causal Inference in Fuzzy Cognitive Maps, IEEE Trans. on Fuzzy Systems 8:1 107-119.
18. K. Murota (2000), Matrices and Matroids for Systems Analysis. Algorithms and Combinatorics, Vol.20, Springer-Verlag.

19. K. Murota (2003), Discrete Convex Analysis. SIAM Monographs on Discrete Mathematics and Applications, Vol.10, Society for Industrial and Applied Mathematics.
20. Y. Narukawa and T. Murofushi, (2003) Choquet integral and Sugeno integral as aggregation functions, in: Information Fusion in Data Mining, V. Torra, ed., (Springer) pp. 27-39.
21. G. Pasi, (2003), Modeling users' preferences in systems for information access, Intl. J. of Intelligent Systems, 18:7 793-808.

Neural Networks

Modular Neural Network Applied to Non-Stationary Time Series*

Héctor Allende[1], Rodrigo Salas[1,2], Romina Torres[3], and Claudio Moraga[4]

[1]Universidad Tćnica Federico Santa Maria; Dept. de Informática; Casilla 110-V; Valparaiso-Chile
hallende@inf.utfsm.cl
[2]Universidad de Valparaiso; Departamento de Computacińn
rodrigo.salas@uv.cl
[3]Motorola Valparaiso
Romina.Torres@motorola.com
[4]University of Dortmund; Department of Computer Science; D-44221 Dortmund; Germany
claudio@moraga.de

Abstract. Modular artificial neural networks (MANN) have been used in the last years as clasification/forecasting machine, showing improved generalization capabilities that outperform those of single networks when the search space is stratified.

Time Series data could be generated by many unknown and different sources and Modular Neural Networks, in particular Mixture of Experts models, are suitable for this time series where each expert is more capable to model some region in the input space and a gating network makes an intelligent selection of the expert that will model the specific pattern.

Stochastical models for time series analysis are global models limited by the requirement of stationarity of the time series and normality and independence of the residuals. However, for most real world time series present behaviors such as heteroscedasticity, sudden burst of activity, or outliers. Such data are very common in finance, insurance, seismology and so on.

In this paper we propose MANN models capable of dynamically adapt their architecture to non-stationary time series when the data is generated from several sources and is affected by the presence of outliers. Simulation results based on benchmark data sets are presented to support the proposed technique.

Key words: Time Series Data Mining, Modular Neural Networks, Mixtures of Experts.

1 Introduction

In the last decade Artificial Neural Networks (ANN) have been of particular interest as a technology for data mining because they offer a means of

*This work was supported in part by Research Grant Fondecyt 1040365 and 7040051, in part by Research Grant BMBF-CHL 03/013 from the German Ministry of Education and in part by Research Grant DGIP-UTFSM and DIPUV-UV

H. Allende et al.: *Modular Neural Network Applied to Non-Stationary Time Series*, Advances in Soft Computing **2**, 585–598 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

efficiently modelling large and complex problems. ANN are data-driven models, i.e., they are capable of finding relationships (patterns) inductively by the learning algorithm based on the existing data rather than requiring the modeler to specify the functional form and interactions.

Time series analysis is fundamental for both engineering and scientific endeavors. Several models were developed for predicting and controlling processes as they evolve through time. Traditional methods such as the ARIMA method [3] are limited by the requirement of stationarity of the time series and normality and independence of the residuals. However, most real world time series present behaviors such as heteroscedasticity, sudden burst of activity, or outliers. Such data are very common in finance, insurance, seismology and so on. [9] and [2] consider the value of artificial neural networks (ANNs) in solving time series forecasting problems. They demonstrate how ANNs can improve the forecasts.

Time Series data could be generated by many unknown and different sources and the model should exploit this stratification on the data. Finite Mixture Models [15] are used as a statistical approach, but assumptions on the data are required. Modular Neural Networks, in particular Mixture of Experts models, are suitable for this time series where each expert is more capable to model some region in the input space and a gating network makes an intelligent selection of the expert that will model the specific pattern.

Most data mining applications involve data that is corrupted by outliers. We understand for outliers those observations that are substantially different to the majority of data. Outliers are of special interest in data mining because they can be the key discovery from very large datasets [14]. The identification of outliers can lead to the discovery of truly unexpected knowledge in areas such as electronic commerce exceptions, bankruptcy, credit card fraud. Since the learning process of ANN relies on the data, when data contains outliers the model is affected by these deviations, thus obtaining a poor performance.

In this paper we propose to use modular artificial neural networks (MANN) based on the Mixture of Experts (ME) framework as a predictive model when the search space is stratified [10]. In order to obtain a model that is robust to the presence of outliers we use Robust Expectation Maximization (REM) algorithm [19] introduced as a modification of the Jordan and Jabobs [12] algorithm. We address the characteristic of the model as a data mining technique and we propose an algorithm to detect outliers on multivariate data. Finally we show simulations results based on real datasets.

2 Time Series Analysis

2.1 Linear Models

The statistical approach to forecasting involves the construction of stochastic models to predict the value of an observation x_t using previous observations.

This is often accomplished using linear stochastic difference equation models, with random input. By far, the most important class of such models is the linear autoregressive integrate moving average (ARIMA) model. A more comprehensive treatment may be found for example in [3]. The seasonal ARIMA $(p, d, q) \times (P, D, Q)^s$ model for such time series is represented by

$$\Phi_P(B^S)\phi_p(B)\nabla_S^D\nabla^d x_t = \Theta_Q(B^S)\theta_q(B)\varepsilon_t \tag{1}$$

where $\phi_p(B)$ is the nonseasonal autoregressive operator of order p, $\theta_q(B)$ is the nonseasonal moving average operator of order q, $\Phi_P(B^S)$, $\Theta_Q(B^S)$ are the seasonal autoregressive and moving average operator of order P and Q and the terms x_t and ε_t are the time series and a white noise respectively. Moreover it is assumed that $E[\varepsilon_t|x_{t-1}, x_{t-2}, ...] = 0$. This condition is satisfied for example when ε_t are zero mean, independent and identically distributed and independent of past x_t´s. It is assumed throughout that ε_t has finite variance σ^2. The backshift operator B shifts the index of a time series observation backwards, e.g. $Bx_t = x_{t-1}$ and $B^k x_t = x_{t-k}$. The order of the operator is selected by Akaike's information criterion (AIC) or by Bayes information criterion (BIC) and the values of the parameters $\Phi_1, ..., \Phi_P$, $\phi_1, ..., \phi_p$, $\Theta_1, ..., \Theta_Q$ y $\theta_1, ..., \theta_q$ are selected from the time series data using optimization methods such as maximum likelihood [3] or using robust methods such as recursive generalized maximum likelihood [1]. The ARMA-model is limited by the requirement of stationarity and invertibility of the time series, i.e. the system generating the time series must be time invariant and stable. In addition, the residuals must be independent and identically distributed.

The ARMA models require a stationary time series in order to be useful for forecasting. The condition for a series to be weak stationary is that:

$$E[x_t] = \mu \quad ; \qquad V[x_t] = \sigma^2 \quad ; \quad COV[x_t, x_{t-k}] = \gamma_k \qquad \forall t \tag{2}$$

2.2 Non-Linear Models

Many types of non-linear models have been proposed in the literature, see for example bilinear models [17], classification and regression trees [4], threshold autoregressive models [18] and Projection Pursuit Regression [7]. However, on the debit side, it is generally more difficult to compute forecasts more than one step ahead [13].

An important class of non-linear models is that of non-linear ARMA models (NARMA) proposed by [5], which are generalizations of the linear ARMA models to the non-linear case. A NARMA model obeys the following equations:

$$x_t = h(x_{t-1}, x_{t-2}, ..., x_{t-p}, \varepsilon_{t-1}, ..., \varepsilon_{t-q}) + \varepsilon_t \tag{3}$$

where h is an unknown smooth function, and as in the section 2.1 it is assumed

that $E[\varepsilon_t|x_{t-1}, x_{t-2}, ...] = 0$ and that variance of ε_t is σ^2. In this case the conditional mean predictor based on the infinite past observation is

$$\hat{x}_t = E[h(x_{t-1}, x_{t-2}, ..., x_{t-p}, \varepsilon_{t-1}, ..., \varepsilon_{t-q})|x_{t-1}, x_{t-2}, ...] \tag{4}$$

Suppose that the NARMA model is invertible in the sense that there exists a function ν such that $x_t = \nu(x_{t-1}, x_{t-2}, ...) + \varepsilon_t$. Then given the infinite past of observations x_{t-1}, x_{t-2},..., one can compute the ε_{t-j} in (3) exactly $\varepsilon_{t-j} = \kappa(x_{t-j}, x_{t-j-1}, ...)$, $j = 1, .., q$. In this case the mean estimate is

$$\hat{x}_t = h(x_{t-1}, x_{t-2}, ..., x_{t-p}, \varepsilon_{t-1}, ..., \varepsilon_{t-q}) \tag{5}$$

where the ε_{t-j} are specified in terms of present and past x_u's. The predictor of (5) has a mean square error σ^2.

Since we have only a finite observation record, we cannot compute (5). It seems reasonable to approximate the conditional mean predictor (5) by the recursive algorithm

$$\hat{x}_t = h(x_{t-1}, x_{t-2}, ..., x_{t-p}, \hat{\varepsilon}_{t-1}, ..., \hat{\varepsilon}_{t-q}) \tag{6}$$

$$\hat{\varepsilon}_{t-j} = x_{t-j} - \hat{x}_{t-j}, j = 1, 2, ..., q \tag{7}$$

with the following initial conditions $\hat{x}_0 = \hat{x}_{-1} = ... = \hat{x}_{-p+1} = \hat{\varepsilon}_0 = ... = \hat{\varepsilon}_{-q+1} = 0$.

For the special case of non-linear autoregressive model (NAR), it is easy to check that (3) is given by

$$x_t = h(x_{t-1}, x_{t-2}, ..., x_{t-p}) + \varepsilon_t \tag{8}$$

In this case, the minimum mean square error (MSE) optimal predictor of x_t given x_{t-1}, x_{t-2},...., x_{t-p} is the conditional mean (for $t \geq p + 1$).

$$\hat{x}_t = E[x_t|x_{t-1}, ..., x_{t-p}] = h(x_{t-1}, ..., x_{t-p}) \tag{9}$$

This predictor has mean square error σ^2.

3 Nonlinear Time Series Analysis with Several Sources

In this section we suppose that a Time Series could be generated by many unknown and different sources S_1,...,S_K, where K is the number of data sources. In this work we assume that each source can be modelled by a NAR model given by:

$$x_t^{[i]} = h(x_{t-1}, x_{t-2}, ..., x_{t-p_i}) + \varepsilon_t^{[i]} \qquad i = 1..K \tag{10}$$

where the output $x_t^{[i]}$ of the source Si given the past observations $x_{t-1}, ..., x_{t-p_i}$ follows the probabilistic behavior given by $P[x_t^{[i]}|x_{t-1}, ..., x_{t-p_i}, S_i]$. Then, the output of each source is combined in order to generate the observed data x_t. To combined the output of the several sources a probabilistic model is used, where the sources s_i are chosen with probability $P[s_i|x_{t-1}, ..., x_{t-p}]$ given the past observations $x_{t-1}, ..., x_{t-p}$, and where $p = max_i\{p_i\}$.

For these reason the probability of the output variable x_t given the past observation $x_{t-1}, ..., x_{t-p}$ is given by

$$P[x_t|x_{t-1}, ..., x_{t-p}] = \sum_{i=1}^{K} P[s_i|x_{t-1}, ..., x_{t-p}]P[x_t^{[i]}|x_{t-1}, ..., x_{t-p_i}, S_i] \quad (11)$$

As can be noted only the data x_t is observed, but the output $x_t^{[i]}$ of each source is not detected, we say that the data is lost or hidden, and each source receive the observed data only.

4 Construction of Modular Neural Network for Nonlinear Time Series Analysis with Several Sources

An Artificial Neural Network (ANN) topology and dynamics define an approximator from input to output. The unknown function $h : \Re^m \rightarrow \Re$ produces the observed sample pattern pairs $(x_1, y_1), (x_2, y_2), ...$ The sample data modify parameters in the neural estimator and bring the neural system input-output responses closer to the input-output responses of the unknown estimate. In psychological terms, the neural system "learns from experience". This is why we call the ANN estimation model-free.

In this section first we introduce the Mixture of Experts model as a Modular Neural Network, then a robust learning algorithm as a parameter estimation process of the model is introduced. Finally the model is apply as Time Series predictor.

4.1 Construction of Mixture of Experts model

Jacobs et al. proposed a technique known as *Mixture of Experts* (ME) [10] as a generalization of the statistical mixture models. As shown in Figure 1, the ME architecture consists of K modules called *experts* and a *gating* network. The experts solve a function approximation problem over local regions of the input space, so the architecture needs a mechanism to identify for each input $\underline{x}$ which experts are more adequate to model the desire output, work that is accomplished by the gating network. The gating network implements a *soft partition* of the input space into regions corresponding to experts; the outputs of the experts are weighted as dictated by the gating networks to obtain a combined output.

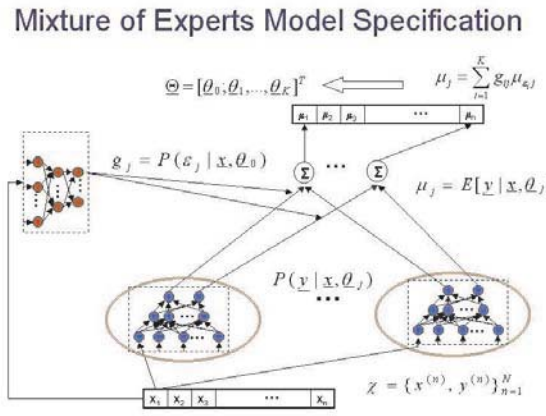


Fig. 1. Mixtures of Experts Architecture (ME): This architecture consists of a set of experts networks and a gating network. The experts compete for the learning of the problem and the gating mediates the competence.

The j^{th} expert network produces as output a parameter vector $\underline{\mu}_j = \underline{\mu}_j(\underline{x}, \underline{\theta}_j)$, $j = 1, \ldots, K$, which is the location parameter for the j^{th} probability density. The gating network partitions the input space into regions corresponding to the various expert networks by assigning a probability vector $[g_1, g_2, \ldots, g_K]^T$ to each point in the input space. The output of the gating network is given by

$$g_j = g_j(\underline{x}, \underline{\theta}_0) = \frac{e^{s_j(\underline{x}, \underline{\theta}_0)}}{\sum_{i=1}^{K} e^{s_i(\underline{x}, \underline{\theta}_0)}}, \qquad j = 1, \ldots, K, \tag{12}$$

Thus the total probability of observing $\underline{y}$ from $\underline{x}$ is given by the following finite mixture of density

$$P(\underline{y}|\underline{x}) = \sum_{j=1}^{K} P(j|\underline{x}, \underline{\theta}_0) P(\underline{y}|\underline{x}, \underline{\theta}_j, \Sigma_j) = \sum_{j=1}^{K} g_j(\underline{x}, \underline{\theta}_0) P(\underline{y}|\underline{x}, \underline{\theta}_j, \Sigma_j) \tag{13}$$

So the expected value of the output will be the ME model output and is a weighted sum of the experts outputs: $\underline{\mu} \equiv \mathrm{e}[\underline{y}|\underline{x}] = \sum_{j=1}^{K} g_j \underline{\mu}_j$. The $i-th$ output of the model is given by:

$$\mu_i = \sum_{j=1}^{K} g_j(\underline{x}, \underline{\theta}_0) \mu_{j,i}(\underline{x}, \underline{\theta}_j) \tag{14}$$

where $g_j(\underline{x}, \underline{\theta}_0)$ is the output of the gating network and $\mu_{j,i}(\underline{x}, \underline{\theta}_j)$ is the $i-th$ output of the $j-th$ expert network. Further details of this model can be found in [12].

4.2 Robust Learning Algorithm

The learning problem is treated as parameter estimation process of the ME architecture. The parameters $\underline{\Theta} = [\underline{\theta}_0^T, \underline{\theta}_1^T, ..., \underline{\theta}_K^T]^T$ are chosen in such a way to maximize the likelihood function:

$$L(\chi, \underline{\Theta}) = P(\{\underline{y}^{(n)}\}_1^N | \{\underline{x}^{(n)}\}_1^N) = \begin{array}{l} \prod_{n=1}^N P(\underline{y}^{(n)} | \underline{x}^{(n)}) \\ \prod_{n=1}^N \sum_{i=1}^K g_i(\underline{x}, \underline{\theta}_0) P(\underline{y}^{(n)} | \underline{x}^{(n)}, \underline{\theta}_i) \end{array} \quad (15)$$

The MLE consists in maximize the equation (15) or equivalently, maximize the log-likelihood function $l(\chi, \underline{\Theta}) = \log L(\chi, \underline{\Theta})$. To estimate the ME model parameters $\underline{\Theta}^*$, techniques based on gradient ascent are applied (see [10] and [11]) or based in the expectation maximization algorithm [12].

Samples with low likelihood are likely to be regarded as outliers and they have high influences in the estimation process. To avoid this problem we must bound the influence of samples with low likelihood that are likely to be regarded as outliers. For this reason we apply a robust version of the EM algorithm [19].

The Robust Expectation Maximization (REM) algorithm consists in two steps repeated iteratively until convergence criterium is met. The two steps are:

- **Expectation step:** compute the conditional expectation of the robust version of the likelihood:

$$\begin{array}{l} RQ(\underline{\Theta} | \underline{\Theta}^{(k)}) = \sum_{t=1}^N \sum_{j=1}^K h_j^{(k)}(t) \ln \left(g_j(\underline{x}^{(t)}, \underline{\theta}_0) \right) \\ \quad + \sum_{j=1}^K \sum_{t=1}^N h_j^{(k)}(t) \rho \left(\ln \left[P(\underline{y}^{(t)} | \underline{x}^{(t)}, \underline{\theta}_j) \right] \right) \end{array} \quad (16)$$

  where $\underline{\Theta}^{(k)}$ is the expected value of the parameter vector at step k. $h_j^{(k)}(t)$ is given by:

$$h_j^{(k)}(t) = \frac{g_j(\underline{x}^{(t)}, \underline{\theta}_0^{(k)}) P(\underline{y}^{(t)} | \underline{x}^{(t)}, \underline{\theta}_j^{(k)})}{\sum_{i=1}^K g_i(\underline{x}^{(t)}, \underline{\theta}_0^{(k)}) P(\underline{y}^{(t)} | \underline{x}^{(t)}, \underline{\theta}_i^{(k)})} \quad (17)$$

- **Maximization step:** compute:

$$\underline{\Theta}^{(k+1)} = \arg\max_{\underline{\Theta}} RQ(\underline{\Theta} | \underline{\Theta}^{(k)}) \quad (18)$$

This objective of the $\rho(\cdot)$ function is to introduce a bound to the influence of the outliers, and applied to each expert network after the natural logarithm $\ln(\cdot)$. The model will be robust due to the robust contribution of each expert.

To choose the $\rho(\cdot)$ function to accomplish the robustification task, in [8] some special functions for M-estimation are discussed. The goal is to weight each observation according to the magnitude of likelihood evaluated at the observation.

When the outlying observation is presented to the model, the magnitude of the likelihood that each expert and the gating network evaluate are very low (with very high absolute magnitude), so during the learning process the step that each expert must take towards the observation is of big magnitude, so the model gets away from the majority of data. To avoid this problem we must downweight the influence of samples with low likelihood that are likely to be regarded as outliers.

In particular, for the location problem, data that are far away must have a bounded impact in the estimation algorithm, we will use the Huber function given by

$$\rho_H(z) = \begin{cases} z + \frac{1}{2}\log(2\pi) & \text{if } z \geq \frac{1}{2}(-k^2 - \log(2\pi)) \\ -k\{-2z - \log(2\pi)\}^{\frac{1}{2}} - \frac{1}{2}k^2 & \text{otherwise} \end{cases} \tag{19}$$

$$\psi_H(z) = \begin{cases} 1 & \text{if } z \geq \frac{1}{2}(-k^2 - \log(2\pi)) \\ k\{-2z - \log(2\pi)\}^{-\frac{1}{2}} & \text{otherwise} \end{cases} \tag{20}$$

The function $\psi(\cdot)$ will be a type of weight function that will limit the influence of large atypical samples. In the E step, lets compute the following quantities:

$$\psi_{0,j}^{(k)}(t) = \ln(g_j(\underline{x}^{(t)}, \underline{\theta}_0)) \tag{21}$$

$$\psi_j^{(k)}(t) = \psi(\ln(P(\underline{y}^{(t)}|\underline{x}^{(t)}, \underline{\theta}_j))) \tag{22}$$

Now the Maximization (M) step computes equation (18), by numerical optimization method by solving the estimation equations:

$$\frac{\partial RQ}{\partial \underline{\theta}_0}|_{\underline{\theta}_0 = \underline{\theta}_0^{(k+1)}} = \underline{0} \quad \frac{\partial RQ}{\partial \underline{\theta}_j}|_{\underline{\theta}_j = \underline{\theta}_j^{(k+1)}} = \underline{0} \tag{23}$$

In summary, the parameters update by the REM-ME algorithm are obtained as follows [19]:

1. **The E step:** Compute the $h_j^{(k)}(t)$, $\psi_{0,j}^{(k)}(t)$ and $\psi_j^{(k)}(t)$, $j = 1..K$, by equations (17), (21) and (22) respectively.
2. **The M step:** Estimate $\underline{\theta}_j^{(k+1)}$, $j = 1, \ldots, K$ and $\underline{\theta}_0^{(k+1)}$ by solving the estimation equations (23) or solving the maximization problem given in equation (18).

5 Mixture of Experts for Time Series Forecasting

Finally we use the ME as stochastical model for Time Series analysis when data are generated by different sources. The patterns introduced into the neural network for training and prediction are obtained from the time series

$x_1, x_2, ..., x_t, ...$ in the following form: the output y of the ME model is related to the variable that we want to predict x_t, and the input $\underline{x}$ with the lags $(x_{t-1}, ..., x_{t-p})$ needed for the prediction.

In this approach, each expert $\underline{\mu}_j = \mu_j$ is considered to model the expected output of the data source S_j.

$$\hat{x}_t^{[j]} = \mu_j(x_{t-1}, ..., x_{t-p}, \underline{\theta}_j) = h(x_{t-1}, ..., x_{t-p_j}) \qquad j = 1..K \tag{24}$$

And the gating network models the probabilistic behavior of the aggregation function that combines the output of the sources, i.e.,

$$g_j(x_{t-1}, ..., x_{t-p}, \underline{\theta}_0) = P[S_j]x_{t-1}, ..., x_{t-p}] \tag{25}$$

So, in this form, the prediction of the value x_t will be given by the output of the ME model:

$$\hat{x}_t = \mu(x_{t-1}, .., x_{t-p}, \underline{\Theta}) = \sum_{j=1}^{K} g_j(x_{t-1}, ..., x_{t-p}, \underline{\theta}_0)\mu_j(x_{t-1}, ..., x_{t-p}, \underline{\theta}_j) \tag{26}$$

6 Outlier Detection

Most data mining applications involve data that is contaminated by outliers. The identification of outliers can lead to the discovery of truly unexpected knowledge in areas such as electronic commerce exceptions, bankruptcy, credit card fraud. One approach to identify outliers is to assume that the outliers have different distribution with respect to the remaining observations.

To detect outliers we measure the influence of a given observation over the parameters of each expert of the ME model by using the influence function (IF) as was defined by Hampel [8]. The IF describes the effect of an infinitesimal contamination at the point $(\underline{x}_i, y_i)$ on the estimate. The influence of the point $(\underline{x}_i, y_i)$ on each expert is given by

$$IF_j = IF(\underline{x}_i, y_i, \underline{\theta}_j) = \psi(r_i)M_j^{-1}D\mu_j^T \tag{27}$$

where $r_i = y_i - \mu_j(\underline{x}_i, \underline{\theta}_j)$ is the residual of the prediction, $D\mu_j = \frac{\partial \mu_j}{\partial \underline{\theta}_j}$ is the gradient of the expert network j, and under some regular condition M_j can be estimated by $M_j = \sum_{i=1}^{N} \psi'(r_i)D\mu_j D\mu_j^T$. Now, we obtain the influence of the data into the model by

$$IF(\underline{x}_i, y_i, \underline{\theta}_0, \underline{\theta}_1, ..., \underline{\theta}_K) = [(g_1 IF_1)^T, ..., (g_K IF_K)^T]^T \tag{28}$$

We measure the influence of a particular data into to the non-robust $Q(\underline{\theta}|\underline{\theta}^{(k)})$ functional with respect to the $\underline{\theta}_j$ parameters of each expert (i.e.

we assume $\rho(x) = id(x)$ in (16)). We assume that the experts networks have a probabilistic behavior given by the gaussian distribution:

$$P(y|\underline{x}, \underline{\theta}_j, \Sigma_j) = \frac{1}{(2\pi)^{\frac{m}{2}} |\ \ _j|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}[\underline{y} - \mu_j(\underline{x}, \underline{\theta}_j)]^T \Sigma_j^{-1}[y - \mu_j(\underline{x}, \underline{\theta}_j)]\right\} \tag{29}$$

The influence of the point $(\underline{x}_i, y_i)$ to each expert parameter is given by

$$IF_j = IF(\underline{x}, y, \underline{\theta}_j) = \left(\Sigma_j^{-1}(y - \underline{\mu}_j) D\underline{\mu}_j\right)^T \tag{30}$$

where $D\underline{\mu}_j = \frac{\partial \mu_j}{\partial \underline{\theta}_j}$ is the gradient of the expert net. Then the standardized influence function is applied as a measure of the impact of the observation to the model:

$$SIF = SIF(\underline{x}_i, y_i, \underline{\theta}_1, ..., \underline{\theta}_K) = \sqrt{IF^T V(\underline{\theta}_j)^{-1} IF} \tag{31}$$

where the variance of the estimator is given by, $V(\underline{\theta}_j) = \int IF \cdot IF^T dP(\underline{x}, y)$. After evaluating the SIF function into the data, if $SIF > \delta$ for some fixed $\delta > 0$ will indicate that the data can be regarded as outlier. Further analysis is needed to understand its cause, and possibly extract useful knowledge.

7 Experimental Results with a Real data set

In this section we analyze our method applied to sets obtained from a Benchmark datasets. The first set in the *Balloon Time Series* [6] and the second is the *Wind speed variation Time Series to measure the El Niño effect* [16].

To model this Time Series we apply two different architectures: a feedforward neural network (FANN) [2] and the Mixture of Experts model [19]. The Feedforward Neural Network model has three layers: input, hidden and output with $(p-\lambda-1)$ neurons respectively. The ME model consists on linear experts, i.e., $\underline{\mu}_j = \underline{x}^T \underline{\theta}_j$, and linear gating network, $s_j(\underline{x}, \underline{\theta}_0) = \underline{x}^T \underline{\theta}_0$, before applying the softmax function (non-linear) given by equation (12). In the ME model we vary the number of experts K for the simulations. For the learning process of the ME models four differents algorithms were used: the gradient descent algorithm (Grad), the Expectation Maximization Algorithm (EM), the Robust gradient descent (RGrad) and the Robust Expectation Maximization Algorithm (REM).

Furthermore, each of the datasets were partitioned in three subsets: Training (Tr), Validation (Val) and Test (T). The Training set was used for the learning process of the Neural Networks models while the Validation set was used as stopping criterium, i.e., when the mean square error of the adjustment of the model to the data in the validation set increase, the learning process is stopped. The Test set, not yet seen by the neural models is used to compare the generalization and prediction performance.

7.1 Balloon Time Series

We now apply our method to a set for observed data, the Balloon Time Series dataset, obtained from the StatLib archive

http://lib.stat.cmu.edu/datasets/balloon

It has long served as a benchmark and been well studied in the previous literature [6]. The data set consists of 2001 observations of radiation, taken from a balloon about 30 kilometers above the surface of the earth. In the section of the flight shown here the balloon increases in height. As radiation increases with height there is a non-decreasing trend in the data. The outliers are caused by the fact that the balloon slowly rotates, causing the ropes from which the measuring instrument is suspended to cut off the direct radiation from the sun.

To model the Balloon Time Series we use a lag of $p = 12$, so the neural networks models have as input $\underline{x}_{t-1} = (x_{t-1}, ..., x_{t-12})$ and output x_t. The data set was partitioned in (1001-250-250) data for the training, validation and test set respectively. Due to the lag twelve data points were lost in the training set. We vary the number of hidden units, λ, of the Feedforward Neural Network and the best results were obtained for 5 neurons displayed in Table 1 as FANN-5. Then we apply a Mixture of Experts model with different learning algorithms and we vary the number of experts K. In Table 1 the best results for each learning algorithm are displayed.

Table 1. Summary results about the mean square error performance of several learning methods using the Balloon Data Set

Algorithm	**Training**	**Validation**	**Test**
FANN-5	0.026794	0.028379	0.014731
ME-Grad-7	0.028261	0.024989	0.013562
ME-RGrad-7	0.028105	0.025108	0.012763
ME-EM-5	0.022022	0.019024	0.008107
ME-REM-2	0.023108	0.018474	0.007680

From this partial results we can infer that the REM obtained better performance. By evaluating the SIF function on the test set consisting on the last 250 data of the time series, 4 observations were detected as outliers and could be identified for further study by the meteorologists (see figure 2).

7.2 Wind speed variation Time Series to measure the El Niño effect

El Niño is a disruption of the ocean atmosphere system in the tropical Pacific which has important consequences for the weather around the globe. Even

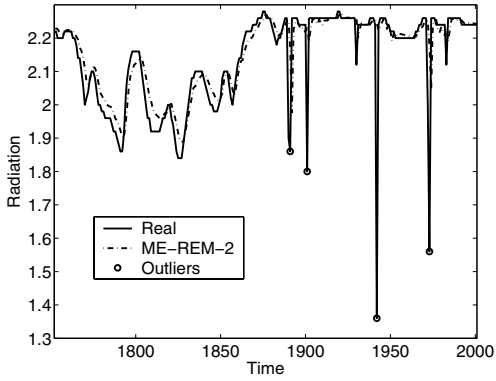


Fig. 2. Experimetal Result: Balloon Test set modelled by the ME-REM-2 model. With circle are marked the outliers found by the algorithm.

though the effect of El Niño is not avoidable, research on its forecast and its impacts allows specialists to attenuate or prevent its harmful consequences (see [16] for a detailed overview). The effect of the equatorial Pacific meridional reheating may be measured by the deviation of the wind speed on the ocean surface from its average. It is worth mentioning that this effect is produced by conduction, and thus we expect the wind speed variation to be smooth. In this section, we study the wind speed anomaly index, i.e. its standardized deviation from the mean, in a specific region of the Pacific (12-2N, 160E-70W). Modelling this anomaly helps to understand the effect of El Niño in that region. The time series composed of $T = 1704$ monthly observations is available at

http://tao.atmos.washington.edu/data_sets/eqpacmeridwindts.

The time series has missing data, and we discard them from the training, validation and test set. We partitioned the final 1511 data in (1511-250-250) data for the training, validation and test set respectively. To model the Balloon Time Series we use a lag of $p = 2$, so the neural networks models have as input $\underline{x}_{t-1} = (x_{t-1}, x_{t-2})$ and output x_t. Due to the lag two data points were lost in the training set. We vary the number of hidden units, λ, of the Feedforward Neural Network and the best results were obtained for 5 neurons displayed in Table 2 as FANN-2. Then we apply a Mixture of Experts model with different learning algorithms and we vary the number of experts K. In Table 1 the best results for each learning algorithm are displayed.

From this partial results we can infer that the REM obtained better performance. By evaluating the SIF function on the test set consisting on the last 250 data of the time series, 23 observations were detected as outliers and could be identified for further study by the meteorologists (see figure 3).

Table 2. Summary results about the mean square error performance of several learning methods using the Wind speed variation Time Series to measure the El Niño effect

Algorithm	Training	Validation	Test
FANN-2	0.7494	0.1146	0.1608
ME-Grad-2	0.7839	0.1201	0.1609
ME-RGrad-2	0.7743	0.1379	0.1981
ME-EM-7	0.7543	0.1173	0.1566
ME-REM-6	0.7568	0.1211	0.1546

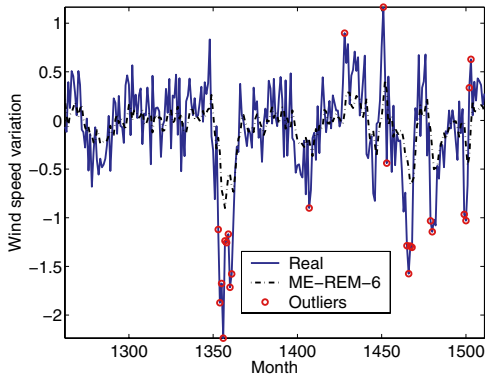


Fig. 3. Experimetal Result: Wind Test set modelled by the ME-REM-6 model. With circle are marked the outliers found by the algorithm.

8 Concluding Remarks

In this paper, we develop a forecasting methodology by considering a Mixture of Experts model to forecast nonstationary time series. The idea of applying the modular neural network model is to deal with the problem of time-changing mean and autocovariance function because the data are supposed to be generated by different sources and the combined by some means to generated an unique output. We show that the ME model is suitable for this type of time series.

In addition we treat the problem of presence of outliers in the data as another type of nonstationary aspect of real data. We introduced an M-estimator to downweight the influence of outliers data during the training process. We modify the EM algorithm with the REM that is less sensitive to this type of deviations.

Finally we introduce a methodology to detect outliers based on the robust model. This data should be considered for further analysis.

Our algorithm was tested in a meteorological data set. The results obtained compare favorable to the REM learning algorithm showing significant

improvements when the data set comes from several data sources and contains outliers.

In the future, we intend to derive some theoretical properties of the Robust version of the Mixture of Experts model as a predictive model. In addition other types of experts different than linear should be used in order to increase the Hypothesis space and obtain better fitting quality.

References

1. H. Allende and S. Heiler, *Recursive generalized m-estimates for autoregressive moving average models*, Journal of Time Series Analysis (1992), no. 13, 1–18.
2. H. Allende, C. Moraga, and R. Salas, *Artificial neural networks in time series forescasting: A comparative analysis*, Kybernetika **38** (2002), no. 6, 685–707.
3. G. E. P. Box, G.M. Jenkins, and G.C. Reinsel, *Time series analysis, forecasting and control*, 3 ed., Ed. Englewood Cliffs: Prentice Hall, 1994.
4. L. Breiman, J. Friedman, R. Olshen, and C.J. Stone, *Classification and regression trees*, Tech. report, Belmont, C. A. Wadsworth, 1984.
5. J.T. Connor and R.D. Martin, *Recurrent neural networks and robust time series prediction*, IEEE Transactions of Neural Networks **2** (1994), no. 5, 240–253.
6. L. Davies and U. Gather, *The identification of multiple outliers*, Journal of the American Statistical Association **88** (1993), 782–801.
7. J.H. Friedman, *Multivariate adaptive regression spline*, The Annals of Statistics (1991), no. 19, 1–141.
8. F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, and W.A. Stahel, *Robust statistics*, Wiley Series in Probability and Mathematical Statistics, 1986.
9. J. Hansen and R. Nelson, *Neural networks and traditional time series methods*, IEEE Trans. on Neural Networks **8** (1997), no. 4, 863–873.
10. R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, *Adaptive mixtures of local experts*, Neural Computation **3** (1991), no. 1, 79–87.
11. M. Jordan and R. Jacobs, *Modular and hierarchical learning systems, the handbook of brain theory and neural networks, cambridge, ma*, vol. 1, MIT Press, 1999.
12. M.I. Jordan and R.A. Jacobs, *Hierarchical mixtures of experts and the EM algorithm*, Neural Computation **6** (1994), no. 2, 181–214.
13. J.L. Lin and C.W. Granger, *Forecasting from non-linear models in practice*, Int. Journal of Forecasting **13** (1994), 1–9.
14. X. Liu, G. Cheng, and J. Wu, *Analyzing outliers cautiosly*, IEEE Transactions on Knowledge and Data Engineering **14** (2002), no. 2, 432–437.
15. G. McLachlan and D. Peel, *Finite mixture models*, Wiley series in probability and statistics, 2001.
16. S. Philander, *El Niño, la Niña and the southern oscillation*, San Diego: Academic Press, 1990.
17. T. Subba Rao, *On the theory of bilinear models*, J. Roy. Statist. Soc. B (1981), no. 43, 244–255.
18. H. Tong, *Non-linear time series*, Ed. Oxford University Press, 1990.
19. R. Torres, R. Salas, H. Allende, and C. Moraga, *Robust expectation maximization learning algorithm for mixture of experts*, IWANN. LNCS **2686** (2003), 238–245.

A Feedforward Neural Network based on Multi-Valued Neurons

Igor Aizenberg[1], Claudio Moraga[2], and Dmitriy Paliy[3]

[1] University of Dortmund, Department of Computer Science 1,
for correspondence: Mapu 18, ap.3, Tel Aviv, 63434, Israel,
igora@netvision.net.il
[2] University of Dortmund, Department of Computer Science 1,
44221 Dortmund, Germany,
claudio.moraga@udo.edu
[3] Tampere University of Technology, Institute for Signal Processing
TICSP, Tampere University of Technology P.O. Box 553
FIN-33101 Tampere Finland
dmitriy.paliy@tut.fi

Abstract. A feedforward neural network based on multi-valued neurons is considered in the paper. It is shown that using a traditional feedforward architecture and a high functionality multi-valued neuron, it is possible to obtain a new powerful neural network. Its learning does not require a derivative of the activation function and its functionality is higher than the functionality of traditional feedforward networks containing the same number of layers and neurons. These advantages of MLMVN are confirmed by testing using Parity *n*, two spirals and "sonar" benchmarks, and the Mackey-Glass time-series prediction.

1 Introduction

A multi-layered feedforward neural network (MLF, it is also often refereed as MLP) and a backpropagation learning algorithm for it are well studied from all point of views. It is possible to say that this is a classical example of a neural network. We can refer in this context to the hundreds of the papers and books. Let us refer, for example, to the book [1]. A multi-layer architecture of the network with a feedforward dataflow through nodes that requires full connection between consecutive layers and an idea of a backpropagation learning algorithm was proposed in [2]. It is well known [1] that MLF can be used as a universal interpolator. It is also well known that MLF is traditionally based on the neurons with a sigmoid activation function. MLF learning is based on the backpropagation learning algorithm, when the error is being sequentially backpropagated form the "right hand" layers to the "left hand" ones. It is important that the error of each neuron of the network is proportional to the derivative of the activation function.

I. Aizenberg et al.: *A Feedforward Neural Network based on Multi-Valued Neurons*, Advances in Soft Computing **2**, 599–612 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

On the other hand, it is possible to use different neurons as the basic ones for a network with a feedforward architecture. We will consider in this paper a *multi-layered neural network based on multi-valued neurons* (MLMVN).

A multi-valued neuron (MVN) is based on the principles of the multiple-valued threshold logic over the field of the complex numbers. A comprehensive observation of MVN and its learning is presented in [3]. Different applications of MVN have been considered during the last years: MVN has been successfully used, for example, as a basic neuron in cellular neural networks [3], as a basic neuron of neural-based associative memories [3], [4]-[7] and as the basic neuron of pattern recognition systems [7]-[8].

The mentioned successful applications of MVN make further extensions very attractive. Taking into account that a single MVN has a higher functionality than a single neuron with a sigmoid activation function and that learning of a single MVN is based on the simple linear error correction rule, it would be interesting to consider a neural network with a traditional feedforward architecture, but with MVN as a basic neuron. This network and its training are the main subjects that will be considered in this paper.

We will consider a generalization of a discrete MVN to the continuous-valued case. It will be shown that since MVN learning is reduced to the movement along a unit circle, it does not require a differentiability of the activation function. We will also consider a backpropagation learning algorithm for MLMVN, which also does not require a differentiability of the activation function. Finally, to show the advantages of MLMVN, simulation results will be presented. These advantages in comparison with the different network models are confirmed experimentally using Parity *n*, two spirals and "sonar" benchmarks and the Mackey-Glass time-series prediction.

2 Multi-Valued Neuron and its Training

2.1 Discrete-valued MVN

Let us remind some basic ideas related to MVN and its training. MVN was introduced in [9] as a neural element based on the principles of multiple-valued threshold logic over the field of the complex numbers deeply considered in [10]. A comprehensive observation of MVN theory, its basic properties and learning is presented in [3]. A single MVN performs a mapping between *n* inputs and a single output ([3], [9]). This mapping is described by a multiple-valued (*k*-valued) function of *n* variables $f(x_1,\dots,x_n)$ with *n*+1 complex-valued weights as parameters:

$$f(x_1,\dots,x_n) = P(w_0 + w_1x_1 + \dots + w_nx_n) \qquad (1)$$

where $x_1,\dots,x_n$ are the variables, on which the performed function depends and $w_0, w_1, \dots, w_n$ are the weights. The values of the function and of the variables are

complex. They are the k^{th} roots of unity: $\varepsilon^j = \exp(i2\pi j/k)$, $j \in \{0, k\text{-}1\}$, i is an imaginary unity. P is the activation function of the neuron:

$$P(z) = \exp(i2\pi j/k), \; if \; 2\pi j/k \le \arg z < 2\pi\,(j+1)/k \qquad (2)$$

where j=0, 1, ..., k-1 are values of the k-valued logic, $z = w_0 + w_1 x_1 + \ldots + w_n x_n$ is the weighted sum , arg z is the argument of the complex number z. Equation (2) is illustrated in Fig. 1. Function (2) divides a complex plane onto k equal sectors and maps the whole complex plane into a subset of points belonging to the unit circle. This is exactly a set of k^{th} roots of unity.

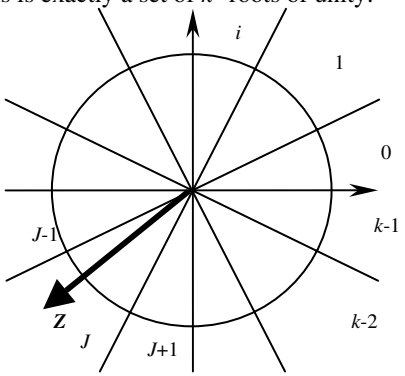


Fig. 1. Geometrical interpretation of the MVN activation function

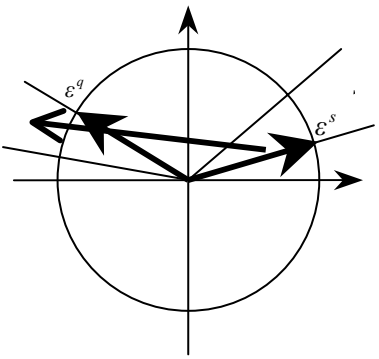


Fig. 2. Geometrical interpretation of the MVN learning rule

MVN learning is reduced to the movement along the unit circle. This movement does not require a derivative of the activation function, because it is impossible to move in the incorrect direction. Any direction of movement along the circle will lead to the target. The shortest way of this movement is completely determined by the error that is a difference between the "target" and the "current point", i.e. between the desired and actual output, respectively. This MVN property is very important for the further development of the learning algorithm for a multi-layered network. Let us consider how it works. Let $\varepsilon^q = T$ be a desired output of the neuron (see Fig. 2). Let $\varepsilon^s = Y = P(z)$ be an actual output of the neuron. The MVN learning algorithm based on the error correction learning rule is defined as follows [3]:

$$W_{m+1} = W_m + \frac{C_m}{(n+1)}(\varepsilon^q - \varepsilon^s)\,\overline{X}, \qquad (3)$$

where X is an input vector, n is the number of neuron inputs, $\overline{X}$ is a vector with the components complex conjugated to the components of vector X, m is the number of the learning iteration, W_m is a current weighting vector (to be corrected), W_{m+1} is the following weighting vector (after correction), C_m is a learning rate. The convergence of the learning process based on the rule (3) is proven in [3]. The rule (3) ensures such

a correction of the weights that a weighted sum is moving from the sector s to the sector q (see Fig. 2). The direction of this movement is completely defined by the difference $\delta = \varepsilon^q - \varepsilon^s$. Thus δ determines the MVN error. According to (3) a correcting item $\Delta w_i = \frac{C_m}{(n+1)}(\varepsilon^q - \varepsilon^s)\,\overline{x}_i, i = 0,1,...,n$, which is added to the corresponding weight in order to correct it, is proportional to δ. The correction of the weights according to (3) changes the value of the weighted sum exactly on δ.

2.2 Continuous-valued MVN

The activation function (2) is discrete. More exactly, it is piece-wise discontinuous. It has discontinuities on the borders of the sectors. Let us modify the function (2) in order to generalize it for the continuous case in the following way. Let us consider, what will happen, when $k \to \infty$ in (2). It means that the angle value of the sector (see Fig. 1) will go to zero. It is easy to see that the function (2) is transformed in this case as follows:

$$P(z) = \exp(i\,(\arg z)) = e^{iArg\,z} = \frac{z}{|z|}, \qquad \textbf{(4)}$$

where z is the weighted sum, Arg z is a main value of its argument and $|z|$ is a modulo of the complex number z.

The function (4) maps the complex plane into a whole unit circle, while the function (2) maps a complex plane just on a discrete subset of the points belonging to the unit circle. The function (2) is discrete, while the function (4) is continuous. We will use here exactly the function (4) as the activation function for the MVN. Both functions (2) and (4) are not differentiable as functions of a complex variable, but this is not important, because their differentiability is not required for MVN learning. The learning rule (3) will be modified for the continuous-valued case in the following way:

$$W_{m+1} = W_m + \frac{C_m}{(n+1)}(\varepsilon^q - e^{iArg\,z})\,\overline{X} = W_m + \frac{C_m}{(n+1)}\left(\varepsilon^q - \frac{z}{|z|}\right)\overline{X}. \qquad \textbf{(5)}$$

It is absolutely clear that convergence of the learning algorithm based on the learning rule (5) may be proven in the same way as it was done in [3] for the rule (3). It is also interesting to consider the following modification of (5):

$$W_{m+1} = W_m + \frac{C_m}{(n+1)}\,\widetilde{\delta}\,\overline{X}, \qquad \textbf{(6)}$$

where $\widetilde{\delta}$ is obtained from $\delta = \varepsilon^q - \frac{z}{|z|} = T - \frac{z}{|z|}$ using a normalization by the factor $\frac{1}{|z|}$:

$$\tilde{\delta} = \frac{1}{|z|}\delta = \frac{1}{|z|}\left(T - \frac{z}{|z|}\right), \quad (7)$$

Learning according to the rule (6)-(7) makes it possible to squeeze a space for the possible values of the weighted sum. Using (6)-(7) instead of (5), we can reduce this space to the respectively narrow ring, which will include the unit circle inside. Indeed, if $|z|<1$ and we correct the weights according to (6)-(7), then $|\tilde{z}|>|z|$ and $\tilde{z}$ will be closer to the unit circle than z, approaching to the unit circle form "inside". If $|z|>1$ and we correct the weights according to (6)-(7), then $|\tilde{z}|<|z|$ and $\tilde{z}$ will be closer to the unit circle than z, approaching to the unit circle form "outside". This approach can be useful in order to make z more smooth as a function of the weights and to exclude a situation, when a small change either of the weights or the inputs will lead to a significant change of z. At the same time, the choice of the learning rule depends on a particular mapping that we want to implement. For example, if it is described by the smooth function, there is no reason to use (6)-(7) adding more calculations.

3 Multi-Layered MVN-based Neural Network and a Backpropagation Learning Algorithm

3.1 General remarks

Let us consider a multi-layered neural network with traditional feedforward architecture, when the outputs of neurons of the input and hidden layers are connected with the corresponding inputs of the neurons from the following layer. Let us suppose that the network contains one input layer, m-1 hidden layers and one output layer. We will use here the following notations. Let

T_{km} - be a desired output of the k^{th} neuron from the m^{th} (output) layer;

Y_{km} - be an actual output of the k^{th} neuron from the m^{th} (output) layer.

Then a global error of the network for the k^{th} neuron of the m^{th} (output) layer can be calculated as follows:

$$\delta^*_{km} = T_{km} - Y_{km} \text{ - the error for the } k^{\text{th}} \text{ neuron from the output layer.} \quad (8)$$

δ^*_{km} will denote here and further a global error of the network. We have to distinguish it from the local errors δ_{km} of the particular neurons.

The learning algorithm for the classical feedforward network is derived from the consideration that a global error of the network expressed in the terms of *square error* (SE) must be minimized. The squared error is defined as follows:

$$\mathrm{E} = \sum_{k} (\delta^{*}_{km})^{2}(W) \tag{9}$$

where δ^{*}_{km} is a global error of the k^{th} neuron of the m^{th} (output) layer and W are the weighting vectors of all the neurons of the network (it is fundamental that the error depends not only on the weights of the neurons from the output layer, but on all neurons of the network).

The functional of error may be defined as follows:

$$E_{ms} = \frac{1}{N} \sum_{s=1}^{N} E_{s} \,, \tag{10}$$

where E_{ms} is the *mean square error*, N is the total number of patterns in the training set, E_{s} is the square error of the network for the pattern number s.

The minimization of the functional (10) is reduced to the search for those weights for all the neurons that ensure a minimal error. Let us remind briefly how it works for MLF. The most important problem for network learning is to express the error of the each neuron through the global errors of the network. This problem is solved using the backpropagation of the global errors through the network: from the output layer through all the hidden layers (from the m^{th} layer to the m-1^{st} one, from the m-1^{st} one to the m-2^{nd} one, …, from the 2^{nd} one to the 1^{st} one). When the error is propagated from the layer j+1 to the layer j, the error of each neuron belonging to the j+1^{st} layer is being multiplied by the weight connecting the corresponding input of this neuron from the j+1^{st} layer with the corresponding output of the neuron from the j^{th} layer. For example, the error δ_{kj+1} is propagated from the neuron kj+1 (the k^{th} neuron from the j+1^{th} layer) to the lj^{th} neuron (the l^{th} neuron from the j^{th} layer) as follows: δ_{kj+1} is multiplied by the weight w_{l}^{kj+1}, namely by the weight corresponding to the l^{th} input of the neuron kj+1.

It is also well known for the traditional MLF that the correction of the weights for all neurons is organized in such a way that each weight w_i has to be corrected by a correcting item Δw_i, which must be proportional to the gradient $\frac{\partial E}{\partial w_i}$ of the error function $E(w)$ with respect to the weights [1].

3.2 A backpropagation learning algorithm for the MLMVN

As it was mentioned from the beginning, the MVN activation function (4) is not differentiable. It means that the MLF backpropagation learning algorithm cannot be applied for the case of MLMVN because it strongly depends on the derivative of the activation function.

However, a backpropagation learning algorithm for the MVN-based network can be derived. As it was shown above for the single neuron, the differentiability of the MVN activation function is not required for learning. Since MVN learning is reduced to the movement along the unit circle, the correction of the weights is completely determined by the neuron's error. The same property is true not only for the single MVN, but for the network (MLMVN). The errors of all the neurons from MLMVN are completely determined by the global errors of the network (8). As well as MLF learning, MLMVN learning is based on the minimization of the error functional (10).

Let us use the following notations. Let w_i^{kj} be the weight corresponding to the i^{th} input of the neuron kj (k^{th} neuron of the j^{th} level). Let Y_{ij} be the actual output of the i^{th} neuron from the j^{th} layer (j=1,…,m). Let N_j be the number of the neurons in the j^{th} layer. By the way, it means that the neurons from the j+1st layer have exactly N_j inputs. Let $x_1,...,x_n$ be the network inputs (they are also the inputs of the neurons in the 1st layer, respectively).

A backpropagation learning algorithm for the MLMVN is described as follows. The global errors of the whole network are determined by (8).
For the errors of the m^{th} (output) layer neurons:

$$\delta_{km} = \frac{1}{s_m}\delta^*_{km}, \qquad \textbf{(11)}$$

where km is a k^{th} neuron of the m^{th} layer; $s_m = N_{m-1}+1$ (the number of all neurons on the previous layer (m-1, to which the error is backpropagated) incremented by 1).
For the errors of the hidden layers neurons:

$$\delta_{kj} = \frac{1}{s_j}\sum_{i=1}^{N_{j+1}}\frac{1}{\left|w_k^{ij+1}\right|^2}\delta_{ij+1}(\overline{w}_k^{ij+1}) = \frac{1}{s_j}\sum_{i=1}^{N_{j+1}}\delta_{ij+1}(w_k^{ij+1})^{-1}, \qquad \textbf{(12)}$$

where kj specifies the k^{th} neuron of the j^{th} layer (j=1,…,m-1); $s_j = N_{j-1}+1,\ j = 2,...,m;\ s_1 = 1$ (the number of all neurons on the layer j-1 (the previous layer j, to which the error is backpropagated) incremented by 1). It should be mentioned that the backpropagation rule (12) is based on the same heuristic assumption as for the classical backpropagation. According to this assumption we suppose that the error of each neuron from the previous (j^{th}) layer depends on the errors of all neurons from the following (j+1st) layer.

Let us clarify a role of the factor $\frac{1}{s_j}$ in (11) and (12). The learning rule (5) for a single MVN determines that $\Delta W = \frac{C_m}{(n+1)}\left(\varepsilon^q - \frac{z}{|z|}\right)\overline{X}$. This expression contains a factor $\frac{1}{(n+1)}$ in order to share a contribution of the correction uniformly among all

n+1 weights $w_0, w_1, ..., w_n$. Since all the inputs are equitable, it is natural that during the correction procedure ΔW has to be shared among all the weights uniformly. If we have not a single neuron, but a feedforward network, we have to take into account the same property. It has to be used, in order to implement properly a backpropagation of the error through the network. It means, that if the error of a neuron on the layer j is equal to $\tilde{\delta}$ then this $\tilde{\delta}$ must contain a factor equal to $\frac{1}{s_j}$, where $s_j = N_{j-1} + 1$ is the number of neurons whose outputs are connected to the inputs of the considered neuron (let us remind that N_i is the number of neurons on the layer i, and all of this neurons are connected to the considered neuron) incremented by 1 (the considered neuron itself). This ensures sharing of the error among all the neurons on which the error of the considered neuron depends. In other words, the error of each neuron is uniformly distributed among the neurons connected to it and itself. It should be mentioned that for the 1st hidden layer $s_1 = 1$ because there is no previous hidden layer, and there are no neurons, with which the error may be shared.

The weights for all neurons of the network must be corrected after calculation of the errors. To do it, we can use the learning rule (5) applying it sequentially to all layers of the network from the first hidden layer to the output one.

Correction rule for the neurons from the m^{th} (output) layer (k^{th} neuron of m^{th} layer):

$$\begin{aligned} \tilde{w}_i^{kj} &= w_i^{km} + \frac{C_{km}}{(n+1)} \delta_{km} \bar{\bar{Y}}_{im-1}, \quad i = 1,...,n \\ \tilde{w}_0^{km} &= w_0^{km} + \frac{C_{km}}{(n+1)} \delta_{km} \end{aligned} \tag{13}$$

Correction rule for the neurons from the 2nd till m-1st layer (k^{th} neuron of the j^{th} layer (j=2, …, m-1):

$$\begin{aligned} \tilde{w}_i^{kj} &= w_i^{kj} + \frac{C_{kj}}{(n+1)\,|\,z_{kj}\,|} \delta_{kj} \bar{x}_i, \quad i = 1,...,n \\ \tilde{w}_0^{kj} &= w_0^{kj} + \frac{C_{kj}}{(n+1)\,|\,z_{kj}\,|} \delta_{kj} \end{aligned} \tag{14}$$

Correction rule for the neurons from the 1st hidden layer:

$$\begin{aligned} \tilde{w}_i^{k1} &= w_i^{k1} + \frac{C_{k1}}{(n+1)\,|\,z_{k1}\,|} \delta_{k1} \bar{x}_i, \quad i = 1,...,n \\ \tilde{w}_0^{k1} &= w_0^{k1} + \frac{C_{k1}}{(n+1)\,|\,z_{k1}\,|} \delta_{k1} \end{aligned} \tag{15}$$

Let us clarify the presence of the factor $1/z_{kj}$ in (14) and (15) and its absence in (13). For the output layer neurons, we have the exact errors calculated according to (8), while for all the hidden neurons the errors are obtained according to the heuristic rule. This may cause a situation, where either the weighted sum for the hidden neurons (more exactly, the absolute value of the weighted sum) may become a not-smooth function with dramatically high jumps or the hidden neuron output will be close to some constant with very small variations around it. In both cases, thousands and even the hundreds of thousands of additional steps for the weights adjustment will be required. To avoid this situation, we can use the learning rule (6)-(7) instead of the rule (5) for the hidden neurons and therefore to normalize ΔW for the hidden neurons by $|z|$ every time, when the weights are being corrected. This will make the absolute value of the weighted sum for the hidden neurons (considered as a function of the weights) more smooth. This also can avoid the concentration of the hidden neurons output in some very narrow interval. On the other hand, the factor $1/|z|$ in (7) can be considered as a *variable part* of the learning rate. While used, it provides the adaptation of ΔW on each step of learning. At the same time, it is not reasonable to use the rule (6)-(7) for the output layer. The exact errors and the exact desired outputs for the output neurons are known. On the other hand, since these errors are shared among all neurons of the network according to (11)-(12), there is no reason to normalize by $1/|z|$ the errors of the output neurons. The absolute value of the output neurons weighted sums belongs to the narrow ring, which includes the unit circle. All the considerations above lead us to the conclusion that the errors of the output layer neurons and the global errors of the network will descent after correction of the weights according to the rules (13)-(15). It means that the square error (9) and the error functional (10) will also descent step by step. In general, the learning process should continue until the following condition is satisfied:

$$\frac{1}{N}\sum_{s=1}^{N}\sum_{k}(\delta^{*}_{km_s})^2(W)=\frac{1}{N}\sum_{s=1}^{N}E_s\le\varepsilon, \qquad (\mathbf{16})$$

where ε determines the precision of learning. In particular, it should be the case $\varepsilon=0$, and (16) will be transformed to $\forall k,\forall s\quad \delta^{*}_{km_s}=0$.

4 Simulation Results

The simulation results were obtained using the simplest network structure $n\rightarrow S\rightarrow 1$ (n inputs, S neurons on a single hidden layer and 1 neuron on the output layer). The efficiency of the backpropagation algorithm (11)-(12) and of the learning algorithm based on the rules (13)-(15) has been tested by experiments with the four standard and popular benchmarks: parity n, two spirals, "sonar" and the Mackey-Glass time series.

For two spirals and "sonar" benchmarks we downloaded data from the CMU benchmark collection[1].

The neural network was implemented using a software simulator developed on the Borland/Inprise Delphi 5.0 platform that was run on the PC with a Pentium III 600 MHz processor. The real and imaginary parts of the starting weights were taken as random numbers from the interval [0, 1] for all experiments. For the continuous-valued benchmarks (inputs for two spirals and sonar benchmarks and inputs/outputs for the Mackey Glass time series) the corresponding data were rescaled to the range $[0, 2\pi]$ in order to represent all inputs/outputs as points on the unit circle.

The parity n functions $(3 \le n \le 9)$ were trained completely (see Table 1) using the network $n \rightarrow S \rightarrow 1$, where $S < n$. It should be mentioned that parity 9 and parity 8 functions were implemented using only 7 and 6 hidden neurons, respectively.

Table 1. Implementation of the Parity n function ($n = 2,3,...,9$)

Function	Configuration of the network	Number of epochs (the median value of 15 independent runs is taken)	Processing time on P-III 600 MHz
Parity 3	3→2→1	57	2 seconds
Parity 4	4→2→1	109	3 seconds
Parity 5	5→3→1	2536	15 seconds
Parity 6	6→4→1	7235	30 seconds
Parity 7	7→4→1	26243	2 min. 50 sec.
Parity 8	8→7→1	21412	5 min. 20 sec.
Parity 8	8→6→1	112397	25 min.
Parity 9	9→7→1	24234	20 min.

These results show advantages of MLMVN in comparison with the traditional solutions. Indeed, it was claimed in [12] that the most optimistic estimation for the number of the hidden neurons for the implementation of the n bit parity function using one hidden layer is $\sqrt{n}$, while the realistic estimation is Ό(n). In [13] it was shown up to $n = 4$, that the minimum size of the hidden layer required to solve the N-bit parity is n. Using a special learning algorithm the parity 7 function was implemented using a 7-4-1 MLF in [14]. Using a modular network architecture, the parity 8 function was implemented in [15]. We did use neither some special architecture nor some specific learning strategy. Moreover, no adaptation of the constant part of the learning rate was used in all our experiments not only with the parity functions, but with all the benchmarks, i.e. all $C_{kj} = 1$ in (13)-(15).

The two spirals problem is a well known classification problem, where the two spirals points must be classified as belonging to the 1st or to the 2nd spiral. The two spirals data also was trained completely without the errors using the networks 2→40→1 (800123 epochs is a median of 11 experiments) and 2→30→1 (1590005

[1] http://www-2.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/neural/bench/cmu (a number of references to the different experimental results and the summary of these results can be also found at the same directory)

epochs is a median of 11 experiments). For example, for MLF with an adapted learning algorithm there is the result with about 4% errors for the network 2→40→1 and with about 14% errors for the network 2→30→1 after about 150000 learning epochs [14]. It should be mentioned that after the same number of learning epochs the MLMVN shows not more than 2% of errors for the network 2→40→1 and not more than 6% errors for the network 2→30→1.

On the other hand, the two spirals data set containing 194 points was separated into training (98 points) and testing (96 points) subsets (the first two points were assigned to the training subset, while the next two points were assigned to the testing subset, etc). After training using the first subset, the prediction capability was tested using the second one. For this experiment we used the networks containing 26, 28, …, 40 hidden neurons on the single hidden layer. The networks with the larger number of the hidden neurons were trained much faster, but the prediction results were approximately the same for all the networks. The average number of training epochs decreases from 1298406 for the network with 26 hidden neurons to 19781 for the network with 40 hidden neurons. The prediction rate obtained as a result of nine independent runs is 68-72% for each network. These results that were obtained using a simpler network are comparative with the best known results (70-74.5%) [11].

The same experiment was performed for the "sonar" data set, but using the simplest possible network 60→2→1 (the "sonar" problem initially depends on 60 input parameters) with only two neurons in the hidden layer. The "sonar" data set contains 208 samples. 104 of them are recommended to be used for training and another 104 for testing, respectively. The training process requires 400-2400 epochs and a few seconds, respectively. This statistics is based on 50 independent runs. The prediction results are also very stable. The predictions rate from the same experiments is 88-93%. This result is comparative with the best known result for the fuzzy kernel perceptron (FKP) (94%) [11] and SVM (89.5%) [11]. However, our result is obtained using the smallest possible network containing only 3 neurons, while for solving the sonar problem FKP and SVM requires in average 167 and 82.6 supporting vectors, respectively [11]. On the other hand the whole "sonar" data set was trained completely without the errors using the same simplest network 60→2→1. This training process requires from 817 till 3700 epochs according to the results of 50 independent runs.

To test the MLMVN capabilities in time series prediction, we used the well known Mackey-Glass time series. This time series is generated by the chaotic Mackey-Glass differential delay equation defined as follows [16]:

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) + n(t), \qquad (\mathbf{17})$$

where $n(t)$ is a uniform noise (it is possible that $n(t)=0$). $x(t)$ is quasi-periodic, and chosing $\tau = 17$ it becomes chaotic [16]-[18]. This means that only short term forecasts are feasible. Exactly $\tau = 17$ was used in our experiment. To integrate the equation (17) and to generate the data, we used an initial condition $x(0) = 1.2$ and a time step $\Delta t = 1$. The Runge-Kutta method was used for the integration of the equation (17). The data was sampled every 6 points, as it is usually recommended for

the Mackey Glass time-series (see e.g., [17]-[18]). Thus, we use the Mackey-Glass time series generated with the same parameters and in the same way as in the recently published papers [17]-[18]. The task of prediction is to predict $x(t+6)$ from $x(t), x(t-6), x(t-12), x(t-18)$. We generated a 1000 points data set. The first 500 points were used for training and the next 500 points were used for testing. The true values of $x(t+6)$ were used as the target values during training.

Since a *root mean square error* (RMSE) is a usual estimation of the quality for the Mackey-Glass time series prediction [17]-[18], we also use it here. We did not require a convergence of the training algorithm to the zero error. Since RMSE is a usual estimator for the prediction quality, it also was used for the training control. Thus instead of the MSE criterion (16), we used the following RMSE criterion for the convergence of the training algorithm:

$$\sqrt{\frac{1}{N}\sum_{s=1}^{N}\sum_{k}(\delta^{*}_{kms})^{2}(W)} = \sqrt{\frac{1}{N}\sum_{s=1}^{N}E_{s}} \le \varepsilon\,, \qquad (\mathbf{18})$$

where ε determines a maximum possible RMSE for the training data.

The results of our experiments are summarized in Table 2. For each of the three series of experiments we made 30 independent runs of training and prediction, like it was done in [18]. Our experiments show that choosing a smaller ε in (18) it is possible to decrease the RMSE for the testing data significantly. To estimate a training time, one can base on the following data for the networks containing 50 and 40 hidden neurons, respectively. 100000 epochs require 50 minutes for the first network and 40 minutes for the second one on the PC with the P-III 600 MHz CPU.

Comparing the results of Mackey-Glass time series prediction using MLMVN to the results obtained using different solutions during last years, we have to conclude that the MLMVN outperforms all of them. Results comparative with the ones obtained using MLMVN are obtained only using GEFREX [17]. It is reported in [17] that using GEFREX it is possible to get RMSE equal to 0.0061 on the testing set. On the other hand, it is not mentioned whether the reported RMSE is the result of averaging over the series of experiments or it is the best result of this series. However, the implementation of MLMVN is strongly simpler than one of the GEFREX (for example, referring to the difficulty of the GEFREX implementation, a different model of the several neural networks ensemble was proposed in [18], but the average RMSE obtained using it is equal 0.009, while the corresponding ensemble contains several neural networks with 56 hidden neurons against the 50 ones for MLMVN).

Table 2. The results of Mackey-Glass time series prediction using MLMVN

# of neurons on the hidden layer		**50**	**50**	**40**
ε - a maximum possible **RMSE** in (18)		0.0035	0.0056	0.0056
Actual RMSE for the training set (min - max)		0.0032 - 0.0035	0.0053 – 0.0056	0.0053 – 0.0056
RMSE for the **testing set**	Min	**0.0056**	0.0083	0.0086
	Max	**0.0083**	0.0101	0.0125
	Median	**0.0063**	0.0089	0.0097
	Average	**0.0066**	0.0089	0.0098
	SD	0.0009	0.0005	0.0011
Number of training epochs (median of 30 runs)		145137	56295	62056

5 Conclusions

In this paper, a multi-layered neural network based on multi-valued neurons (MLMVN) was proposed. This is a network with a traditional feedforward architecture and a multi-valued neuron (MVN) as a basic one. A single MVN has a higher functionality than the traditional neurons. These properties make MLMVN more powerful than traditional feedforward networks. The backpropagation learning algorithm for MLMVN that was developed in the paper also does not require the differentiability of the activation function. The proposed neural network outperforms the traditional ones in solving the traditional testing problems like parity *N*, two spirals, sonar and Mackey-Glass time series prediction.

Acknowledgement

The first author gladly acknowledges the support of the Collaborative Research Center 531 of the Computer Science Department, University of Dortmund, where he was an Invited Researcher in January-April 2004.

References

1. S. Haykin "Neural Networks: A Comprehensive Foundation (2nd edn.)", Prentice Hall, New York (1998).
2. D. E. Rumelhart and J. L. McClelland Parallel Distributed Processing: Explorations in the Microstructure of Cognition. MIT Press, Cambridge (1986).
3. I. Aizenberg, N. Aizenberg and and J.Vandewalle “Multi-valued and universal binary neurons: theory, learning, applications”, Kluwer Academic Publishers, Boston/Dordrecht/London (2000).

4. S. Jankowski, A. Lozowski and J.M. Zurada “Complex-Valued Multistate Neural Associative Memory”, IEEE Trans. on Neural Networks, Vol. 7 (1996) 1491-1496.
5. H. Aoki and Y. Kosugi “An Image Storage System Using Complex-Valued Associative Memory”, Proc. of the 15th International Conference on Pattern Recognition, Barcelona, 2000, IEEE Computer Society Press, Vol. 2 (2000) 626-629.
6. M. K. Muezzinoglu, C. Guzelis and J. M. Zurada, "A New Design Method for the Complex-Valued Multistate Hopfield Associative Memory", IEEE Trans. on Neural Networks, Vol. 14, No 4 (2003) 891-899.
7. H. Aoki, E. Watanabe, A.Nagata and Y. Kosugi "Rotation-Invariant Image Association for Endoscopic Positional Identification Using Complex-Valued Associative Memories", In: J. Mira, A. Prieto (eds.) Bio-inspired Applications of Connectionism. Lecture Notes in Computer Science, Vol. 2085 Springer-Verlag, Berlin Heidelberg New York (2001) 369-374.
8. I. Aizenberg, N. Aizenberg, and C. Butakoff. “Neural Network Based on Multi-Valued Neurons: Application in Image Recognition, Type of Blur and Blur Parameters Identification”, In: J. Mira, A. Prieto (eds.) Bio-inspired Applications of Connectionism. Lecture Notes in Computer Science, Vol. 2085 Springer-Verlag, Berlin Heidelberg New York (2001) 254-261.
9. N.N. Aizenberg and I.N. Aizenberg "CNN Based on Multi-Valued Neuron as a Model of Associative Memory for Gray-Scale Images", Proceedings of the Second IEEE Int. Workshop on Cellular Neural Networks and their Applications, Technical University Munich, Germany, (1992) 36-41.
10. N. N. Aizenberg and Yu. L. Ivaskiv Multiple-Valued Threshold Logic, Naukova Dumka Publisher House, Kiev (1977) (in Russian).
11. J.-H.Chen and CHu-Song Chen "Fuzzy Kernel Perceptron", IEEE Trans. on Neural Networks, Vol. 13, No 6 (2002) 1364-1373.
12. R. Impagliazzo, R. Paturi, and M. E. Saks “Size-depth tradeoffs for threshold circuits”, SIAM J. Comput., Vol. 26, No. 3, (1997), 693–707.
13. H. Fung and L. K. Li “Minimal feedforward parity networks using threshold gates”, Neural Comput., vol. 13, (2001), 319–326, 2001.
14. E. Mizutani and S.E. Dreyfus "MLP's hidden-node saturations and insensitivity to initial weights in two classification benchmark problems: parity and two-spirals", Proc. of the 2002 International Joint Conference on Neural Networks (IJCNN'02), (2002), 2831-2836.
15. L. Franco and S.A. Cannas "Generalization Properties of Modular Networks: Implementing the Parity Function", IEEE Trans. Neural Networks, vol. 12, (2001), 1306-1313.
16. M. C.Mackey and L. Glass "Oscillation and chaos in physiological control systems", *Science*, Vol. 197 (1977), 287-289.
17. M. Russo, "Genetic Fuzzy Learning", IEEE Trans. Evolutionary Computation, Vol. 4, No 3, (2000), 259-273.
18. M. M.Islam, Xin Yao and K. Murase "A Constructive Algorithm for Training Cooperative Neural Networks Ensembles", IEEE Trans. Neural Networks, Vol. 14, No 4, July 2003, pp. 820-834.

Least-Squares Support Vector Machines for Scheduling Transmission in Wireless Networks

Jerzy Martyna

Institute of Computer Science, Jagiellonian University, Nawojki 11, Cracow, Poland,
martyna@ii.uj.edu.pl

Abstract For the scheduling transmission over a fading channel in wireless networks, the performance increases significantly if a specialized packet scheduler is used. The properties of this scheduler demand a learning mechanism. For this purpose, a least squares support vector machine (LS-SVM) is proposed as the learning mechanism. In the SVM methodology the number of the unknown can be infinitely dimensional. The given method is illustrated by some numerical examples.

Key words Support vector machines (LSV), neural networks, power and rate control in wireless networks.

1 Introduction

The support vector machine (SVM) is a new universal learning machine proposed by Vapnik [15, 3], which at first was applied for both regression [10, 8] and pattern recognition [9, 15]. In this concept the data are mapped into a higher dimensional input space and an optimal hyperplane in this space is constructed. The data points corresponding to the non-zero weights, which are the solution of the data point obtained by quadratic programming, are called *support vector machine*. While the classical artificial neural network tolerate the existence of many local minima [19], the SVM solutions are obtained from quadratic programming concepts possessing a global minimum. Moreover, the quality and complexity of the SVM method does not depend directly on the dimensionality of the input space [15, 17, 18].

The main goal of this paper is to build a comprehensive model of scheduling transmission in wireless networks in order to solve the problem of how to serve the transmitted data when the delay exceeds the power cost. We

J. Martyna: *Least Squares Support Vector Machines for Scheduling Transmission in Wireless Networks*, Advances in Soft Computing **2**, 613–619 (2005)
www.springerlink.com © Springer-Verlag Berlin Heidelberg 2005

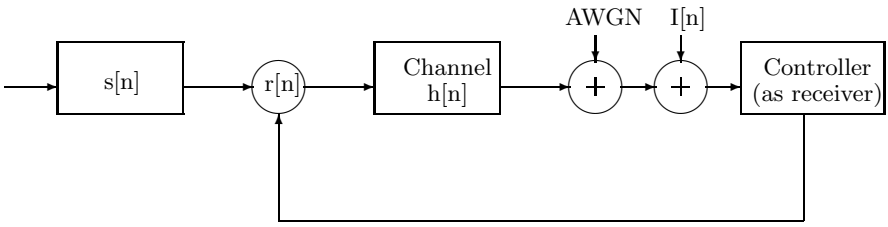


Fig. 1 System model

obtained a dependence on the lower bound required to transmit reliably at a given rate.

We recall that firstly the Vapnik's SVM method was used for a superior approximation of the nonlinear function estimation problems [4, 10]. A suggestion to use a SVM method in the learning feed-forward control was given in the paper of [7]. The same method was used for solving the optimal control problem [11], where the state vector sequence of the considered time horizon has to be obtained as a solution for the constrained nonlinear optimization theory.

This paper is organized as follows. In Section 2, we formulate the scheduling problems over the fading channels in wireless networks. In Section 3 we review the work on support vector machines. In Section 4 we discuss the optimal scheduling algorithm by least squares support vector machines. In Section 5 we present some numerical and simulation results as a solution for our problem.

2 System Model and Problem Formulation

In this section we consider a system model adopted for our work from [5, 1].

We assume a slotted system where the higher layer sends the data that arrives over a slot to the link layer at the end of each slot (see Fig. 1). The link layer has an infinite capacity buffer to hold the data. The buffer length information is transmitted to the receiver/controller. The receiver estimates the *signal to interference ratio* (SIR) on the pilot channel. In our analysis we assume that the estimates are perfect. Additionally, the receiver evaluates the optimal transmission rate and power level in dependence on the SIR estimation. It is possible with the help of the feedback loop. In practice, there are some restrictions on how much these controls can work.

We assume that the time slot length is equal to τ time units and the nth slot is the interval $[n\tau, (n+1)\tau]$, $n \geq 0$. The channel power gain process $H[n]$ is assumed to remain fixed over a slot. The receiver can correctly estimate the signal to interference ratio (SIR) γ. During the nth slot, the SIR $\gamma[n]$ can be given in terms of the current channel gain $h[n]$, the receiver noise power, σ^2, and the current other interference, $i[n]$, as

$$\gamma[n] = \frac{\sigma^2 * h[n]}{\sigma^2 + i[n]} \tag{1}$$

If in a slot n the user transmits a signal $y_s[n]$, then the receiver gets

$$y_r[n] = \sqrt{h[n]} * y_s[n] + \zeta[n], \tag{2}$$

where $\zeta[n]$ constitutes the *Additive White Gaussian Noise* (AWGN) and the others' interference signal. Obviously, we assume the external interference to be independent of the system being modelled.
In our approach, the cost of scheduling r data units in a slot is the total amount of energy required for transmission. When the SIR as defined above is γ, by means Shannon's formula we obtain the following equation

$$r = \frac{1}{\theta} ln(1 + \frac{\gamma P}{\sigma^2}) \tag{3}$$

where $\theta = \frac{2ln(2)}{N}$, N is channel symbols (it can be also related to as channel bandwidth via Nyquist's theorem).

Thus, the power required to transmit r data units is given by equation

$$p(r, x) = \frac{\sigma^2}{\gamma}(e^{\theta r} - 1) \tag{4}$$

where x is the state of system. Given $x[0] = x$ the scheduler's problem is thus to obtain the optimal $r(.)$. We use a window size of length N. Thus, the training data are described by the input $y_s[k] = y_s[k] \mid \ldots \mid y_s[k + N - 1]$ and the targets $y_r[k] = (y_r[k] \mid \ldots \mid y_r[k + N - 1])$. In our approach we implemented SVM algorithms with the window size of 300.

3 The Support Vector Method of Function Estimation

In this section we present basic ideas of the support vector method of the function estimation. More details on SVM of function estimation are available in [15, 16, 17, 18, 8, 2].

We consider a training set of N data points $\{x_k, y_k\}_{k=1}^{N}$, where $x_k \in \Re^n$ is the input data, $y_k \in \Re$ is the k-th output data, the SVM method can be given by means of the equation:

$$y(x) = w^T \phi(x) + b \tag{5}$$

where the nonlinear mapping $\phi(.)$ maps the input data into a higher dimensional feature space. The dimension of w is not specified. It means that it can be infinitely dimensional. In least squares support vector machines for the function estimation the following optimization problem can be given

$$\min_{w,e} \mathcal{I}(w,e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^{N} e_k^2 \tag{6}$$

subject to the equality constraints

$$y_k = w^T \phi(x_k) + b + e_k, \qquad k = 1, 2, \ldots, N. \tag{7}$$

One defines the Lagrangian

$$\mathcal{L}(w,b,e;a) = \mathcal{I}(w,e) - \sum_{k=1}^{N} \alpha_k \{ w_k^T \phi(x_k) + b + e_k - y_k \} \tag{8}$$

with Lagrange multipliers α_k. The conditions for optimality are

$$\begin{cases} \dfrac{\partial \mathcal{L}}{\partial w} = 0 \ \rightarrow w = \sum_k^N \alpha_k \phi(x_k) \\ \dfrac{\partial \mathcal{L}}{\partial b} = 0 \ \rightarrow \sum_{k=1}^N \alpha_k = 0 \\ \dfrac{\partial \mathcal{L}}{\partial e_k} = 0 \ \rightarrow \alpha_k = \gamma * e_k \\ \dfrac{\partial \mathcal{L}}{\partial \alpha_k} = 0 \rightarrow w^T \phi(x_k) + b + e_k - y_k = 0 \end{cases} \tag{9}$$

for $k = 1, 2, \ldots, N$. After the elimination of e_k and w the solution is presented by the following set of linear equations

$$\left[\begin{array}{c|c} 0 & 1^T \\ \hline 1_v & \Omega + \frac{1}{\gamma} I \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline y \end{array}\right] \tag{10}$$

The parameters of LS-SVM algorithm at time k are presented by the threshold value $b[k]$ and the Langrangian multiplexers $\alpha[k] = (\alpha_k, \ldots, \alpha_{k+N-1})^T$. Then, we obtain the matrix equation described by

$$\left[\begin{array}{c|c} 0 & y[k]^T \\ \hline y[k] & U[k] \end{array}\right] \left[\begin{array}{c} b[k] \\ \hline \alpha[k] \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline 1_N \end{array}\right] \tag{11}$$

For large values of N this matrix equation cannot be stored in the memory, therefore an iterative solution method for solving (11) is needed. We used a large scale algorithm for LS-SVM's given by Suykens in [11].

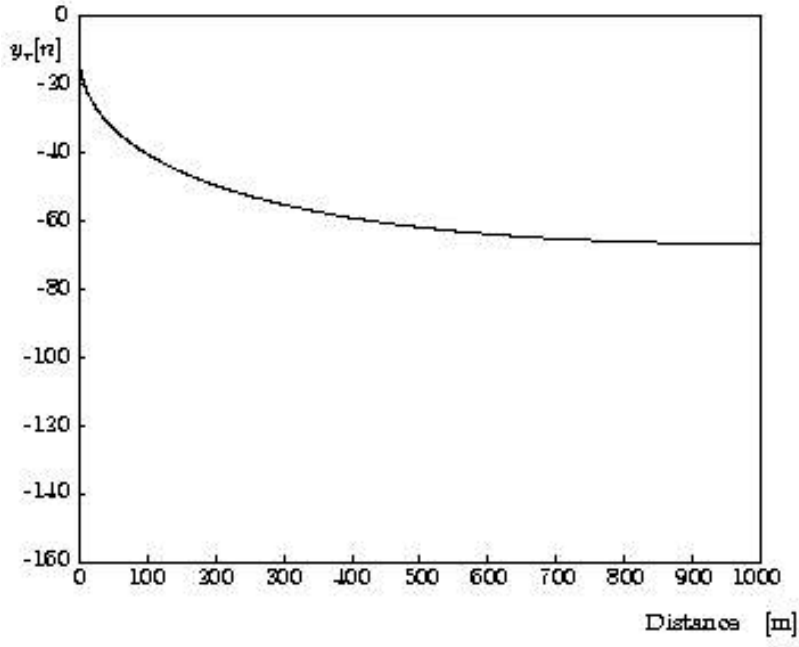


Fig. 2 Received power in dependence of distance when the channel quality is poor

4 Simulation study

We implemented the SVM method to recover a signal $y_r[n]$ by the receiver in transmission over a fading channel. We used the LS-SVMlab Toolbox [14] to simulate several tests of scheduling transmission. We considered different situations where the power of the receiver signal depends on the distance. Fig. 2. shows the power of the SIR received in [db] in dependence on the distance when the channel quality is poor. Under poor conditions, the emitted power may be entered to the minimum level. Fig. 3 gives the power of the SIR received in [dB] in dependence on the distance when the channel is noised. Hence, the interference situation becomes favorable. The receiver noise at the base station is set to -50 dBm. It can be seen that the received signal is the sum of many statistical properties of the Gaussian processes.

5 Conclusions

The least squares support vector machine is a very promising method. In general, the solutions obtained by this method can be implemented via adaptive on-line algorithms. With the help of the second-order statistics they can solve many parametric optimization problems in the unknown interconnections weights, in details of the wireless communications.

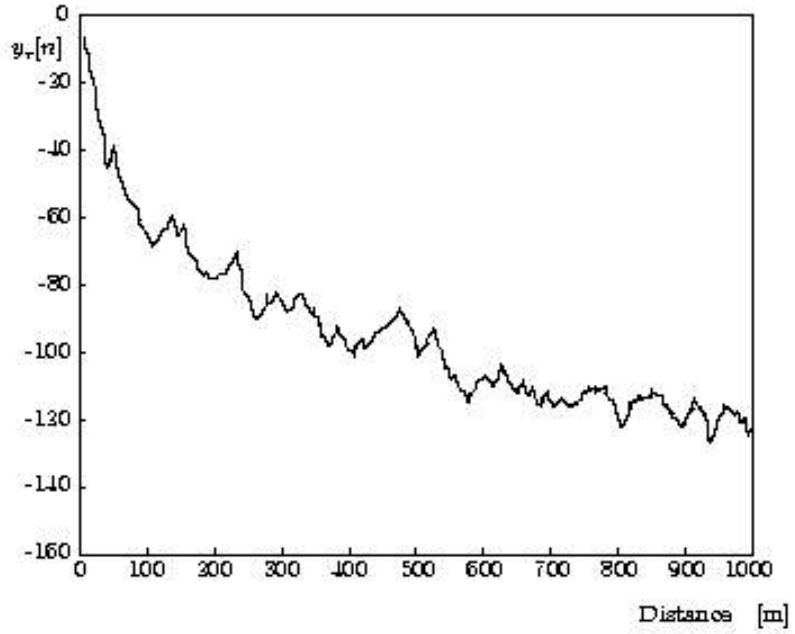


Fig. 3 Received power in dependence of distance when the channel is noised

In our future research we will conduct the SIR dependence on the bandwidth of transmission as well as on the varying number of receivers. By way of tuning some parameters we can obtain the optimality of the LS-SVM algorithm for unequal energy users.

References

[1] R.A. Berry, R.G. Gallager, *"Communication over Fading Channels with Delay Constraints"*, IEEE Trans. on Information Theory, Vol. 48, No. 5, 2002, pp. 1135 - 1149.

[2] V. Cherkassky, F. Mulier, *Learning from Data: Concepts, Theory and Methods"*, John Wiley and Sons, New York, 1998.

[3] C. Cortes, V.N. Vapnik, *"Support Vector Networks"* , Machine Learning, Vol. 20, 1995, pp. 273 - 297.

[4] N. Cristianini, J. Shawee-Taylor, *"An Introduction to Support Vector Machines"*, Cambridge University Press, Cambridge, 2000.

[5] M. Goyal, A. Kumar, V. Sharma, *"Power Constrained and Delay Optimal Policies for Scheduling Transmission over a Fading Channel"*, IEEE INFOCOM 2003, pp. 311 - 320.

[6] T. Joachims, *"Estimating the Generalization Performance of SVM Efficiently"* , in: *Proc. 17th Conf. Machine Learning*, Morgan Kaufman, San Francisco, CA, 2000, [URL http://www.kernel-machines.org/]

[7] B.J. de Kruif, T.J.A. de Vries, *"On Using a Support Vector Machine in Learning Feed-Forward Control"*, IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronic, Vol. 1, 2001, pp. 272 - 277.

[8] B. Schölkopf, C.J.C. Burges, A.J. Smola, *"Advances in Kernel Methods - Support Vector Learning"*, MIT Press, Cambridge, MA, 1999.
[9] R. Schölkopf, A. J. Smola, R. Willianson, P. Bartlett, *"New Support Vector Algorithms"*, Neural Computing, Vol. 12, No. 5, 2000, pp. 1207 - 1245.
[10] A. Smola, B. Schölkopf, *"A Tutorial on Support Vector Regression"* , Royal Holloway Coll. Univ. London, UK, 1998, [URL http://www.kernel-machines.org/]
[11] J.A.K. Suykens, L. Lukas, P. Van Dooren, B. De Moor, J. Vandewalle, *"Least Square Support Vector Machine Classifiers: a Large Scale Algorithm"*, Proc. of the European Conference on Circuit Theory and Design, ECCTD '99, Vol. 2, 1999, pp. 839 - 842.
[12] J.A.K. Suykens, J. Vandewalle, B. De Moor, *"Optimal Control by Least Squares Support Vector Machines"*, Neural Networks, Vol. 14, No. 1, 2001, pp. 23 - 35.
[13] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, *"Least Squares Support Vector Machines"* , World Scientific Pub. Co., Singapore, 2002.
[14] LS-SVMLab Toolbox
[URL http://www.esat.kuleuven.ac.be/sista/lssvmlab/toolbox.html]
[15] V.N. Vapnik, *"The Nature of Statistical Learning Learning Theory"* , Springer-Verlag, New York, 1995.
[16] V.N. Vapnik, S.G. Golowich, A. Smola, *"Support Vector Method for Function Approximation, Regression and Signal Processing"*, in: *"Advances in Neural Information Processing Systems"*, Vol. 9, MIT Press, Cambridge, MA, 1997.
[17] V.N. Vapnik, *"Statistical Learning Theory"* , John Wiley and Sons, New York, 1998.
[18] V.N. Vapnik, *"The Support Vector Method of Function Estimation"* , in: J.A.K. Suykens, J. Vandewalle (Eds.), *"Nonlinear Modeling: Advanced Black-box Techniques"*, Kluwer Academic Publishers, Boston, 1998, pp. 55 - 85.
[19] J.M. Żurada, *"Introduction to Artificial Neural Systems"* , West Publishing Company, 1992.

Neural Networks for the Control of Soccer Robots

Man-Wook Han and Peter Kopacek

Institute for Handling Devices and Robotics (IHRT), Vienna University of Technology, Favoritenstr. 9-11/318, A-1040 Vienna, Austria
Tel.: +43-1-58801-31801, Fax: +43-1-58801 31899
e318@ ihrt.tuwien.ac.at

Abstract - In 1995 robot soccer was introduced with the purpose to develop intelligent, cooperative multi-robot (agents) systems. Robot soccer provides a good opportunity to test control strategies and methods of Multi-Agent-Systems. From the scientific viewpoint a soccer robot is an intelligent, autonomous agent which should carry out its task in cooperative, coordinated, and communicative way with other agents. The group behavior of agents and the behavior of a single agent should be explored. One of the single agent's behaviors is the motion control. The desired velocity of each wheel is generated and sent to the robot comparing the desired and actual position of the robot. The mostly used motion controller today is the digital PID-controller. In this paper as a "modern", intelligent control algorithm a neural network will be introduced and tested.

1. Introduction

Several years ago robot soccer was introduced with the purpose to develop the intelligent, cooperative multi-robot (agent) systems and as one of the first examples for robots in entertainment, leisure and hobby. Robot soccer offers a good opportunity to implement and test control and cooperation algorithms of Multi-Agent-Systems (MAS). From this viewpoint each soccer robot is an intelligent autonomous agent.

The whole robot soccer system (see Fig. 1.) consists of a host computer, players (three mobile micro robots), a vision system, communication modules etc. A color-CCD-camera, located 2m above the playground - size 1.50m x 1.30m - delivers picture to the host computer. The computer generates the motion commands based on the implemented game strategy and submits it to each robot by wireless communication. Based on the location of intelligence the robot system can be divided in,

- ⇨ remote-brainless systems
  The most intelligence is located in the host computer.
- ⇨ vision-based systems
  The robot is able to generate its motion behavior, like position control, collision avoidance etc.
- ⇨ robot-based systems
  Robot is moving autonomously. The host computer delivers only the position data of each object to the robots.

At the moment the most "intelligence" is located in the host computer.

M.-W. Han and P. Kopacek: *Neural Networks for the Control of Soccer Robots*, Advances in Soft Computing **2**, 621–628 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

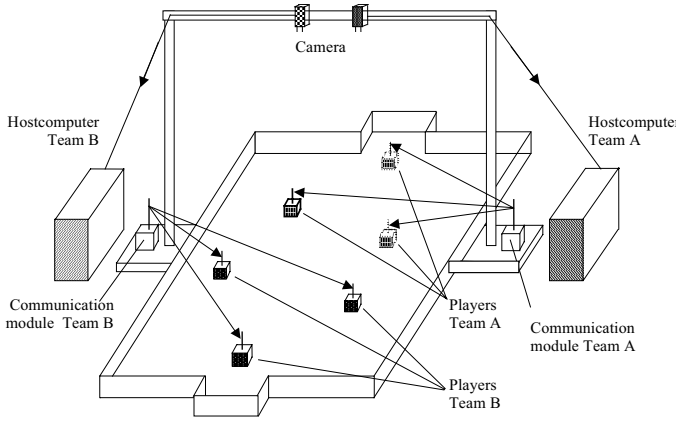


Fig. 1. Overall system

Usually a soccer robot is a two-wheel driven mobile robot, whose size may not exceed 75mm in each side of a cube. It consists of a mechanical part two wheels, two DC motors, a micro controller, a communication module and a power supply. It is a very good example for a mechatronic system. The behavior and efficiency of such a robot depends on the mechanical construction, control algorithm, and the performance and accuracy of the vision system.

2. Soccer Robot – ROBY-GO

The soccer robot at IHRT –"ROBY-GO" (see Fig. 2.) is a two-wheel driven mobile robot and is built in simple, compact and modular construction.

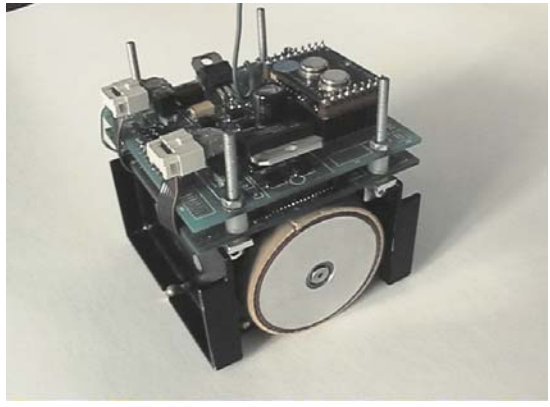


Fig. 2.Soccer robot ROBY-GO

Electronic part has a modular and open architecture and consists of two boards, one for microcontroller board and other for power electronic and communication. As a controller a C167-LM from Infineon (SIEMENS) is used. C167 is a 16-Bit CMOS microcontroller (25MHz) with on Chip CAN module. C167 contains CAN bus interface, there are possibilities to connect several microcontroller boards for different tasks, like sensors, etc.. This microcontroller has 4 channel PWM units.

ROBY-GO can reach maximum speed upto 2.5m/s. Each of two wheels is connected by a gear with a DC motor. Each DC motor receives a command value – desired speed - as input by the microcontroller generated PWM (Pulse-Width-Modulation) signal.

The behavior of a robot depends on the accuracy and dynamics of the vision system as well as the robot itself. From the communication module each robot receives reference velocity and reference angle velocity. New values are transmitted to the robots in constant time intervals of 33 ms. Values are calculated from known actual position and the goal position and goal velocity of the robot. So on host computer implemented algorithm for planning reference velocity curve serves also as a position controller. Very short time period in which the robot has to reach reference velocity enables another simplification. It can be supposed, that the robot reaches reference velocity in the moment or equivalently, that the robot is moving with constant velocity in the 33 ms time intervals, that is between points when it gets a new velocities from host computer. That assumption significantly simplifies path planning and also tasks of another modules implemented on host computer. The lowest sampling time that can be used for robot's velocity controller is 1 ms and is limited by encoder resolution. In the case of very complex control algorithm, also the computational power of the on-board processor could be limiting factor. Reference velocity curve is also calculated with the defined acceleration, separately for each of the wheels. Acceleration is set according to the actual and reference velocities of the wheels and available torque of the motors. In that way the actuator saturation and consequently increasing velocity errors are avoided.

3. A simplified Dynamical Model of the Soccer robot

The simulation study of a two wheel driven mobile robot is based on a dynamical model of the robot. Deriving the complete dynamical model faces many problems, specially determining complicated parameters of the model. One of them is inertia tensor matrix, especially when the parts are non-uniform and have a complicated shape. Additionally finding an appropriate friction model is also difficult because the friction can not be exactly measured. Therefore it is necessary to derive the model how the real robot will behave.

Simplified dynamical model of the robot :

$$m \cdot \ddot{x} = \frac{1}{r} \cdot \left(M_R + M_L \right) \tag{1}$$

$$J \cdot \ddot{\varphi} = \frac{a}{2 \cdot r} \cdot \left(M_R - M_L \right) \tag{2}$$

$$J = \frac{m \cdot a^2}{6} \tag{3}$$

$$M_R = M_{Rm} \cdot N \ ; \ M_L = M_{Lm} \cdot N \tag{4}$$

m is the mass of the robot and $\ddot{x}$ is the acceleration of the robot. $\ddot{\varphi}$ is the angular acceleration. The torque on left and right wheel (M_L, M_R) can be calculated with the torque provided by left and right motor (M_{Lm}, M_{Rm}) and the gear ratio (N). J is the amount of inertia under the assumption the robot is a homogeneous cube. r is the radius of wheel and a is the length of a robot's edge.

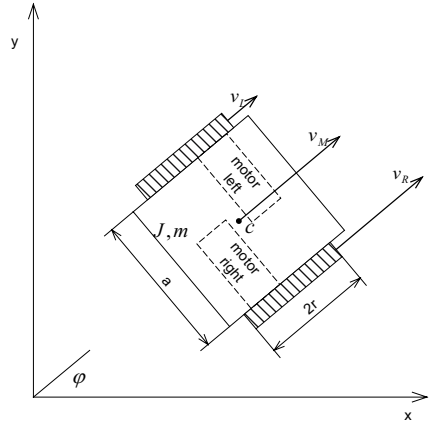


Fig. 3. Soccer robot

However getting an appropriate dynamic model of a robot near to real robot dynamics, some disturbance torque were added to the basic model. Disturbance torque should mainly compensate the absence of the friction in the simplified model, so it consists of a constant part referring to the Coulomb's friction and a velocity dependent part referring to the viscous friction Eq. (5).

$$M_d = 2 \cdot M_{dc} + k_{transl} \cdot \dot{x} + k_{rotat} \cdot \dot{\varphi} \tag{5}$$

Velocities of both wheels (V_L, V_R) which are necessary for implementation of velocity controller, can be calculated from following:

$$V_L = V_M - \frac{\dot{\varphi} \cdot a}{2};\ V_R = V_M + \frac{\dot{\varphi} \cdot a}{2} \tag{6}$$

$$V_M = (V_R + V_L)/2 \tag{7}$$

For the simulations also a model of the DC motor (for example, Minimotor Type 2224 006SR) is necessary. A model with following linear equations is used:

$$M_m = K_M \cdot (I - I_0) \tag{8}$$

$$U = I \cdot R + K_E \cdot n \tag{9}$$

$$n = (60 * N * V)/(2 * pi * r) \tag{10}$$

M_m is the motor torque and is calculating with a torque constant K_M (6.92 mNm/A) and the current Eq. (8). I_0 is no load current (0.029 A). The terminal voltage U is calculated with back electro magnetic force constant, K_E (0.725 mV/rpm) multiplied with rotation speed n and with current and resistance R (0.94 Ohm). Left and right motor's input voltages U_L and U_R are the controller's outputs Eq. (9).

4. Model based Controller with Neural Network Disturbance Estimation

Presented control algorithm is based on a simplified dynamical model of robot and motors, stated earlier in the paper. However disturbances caused by unmodelled dynamics such as friction, time delays, and differences between actual and modeled inertia would be a reason for high velocity errors and performance that does not meet our demands. Problems are also frequent collisions between robots. For compensation of those disturbances a neural network is implemented in the control scheme. Combination (sum) of voltages calculated from the model and disturbance voltage estimated by neural network makes two reference voltages one for each motor of the robot. In the control scheme the actual acceleration and actual angular acceleration are necessary. Those actual values were replaced by estimated values ($\ddot{x}^C$, $\ddot{\varphi}^C$), calculated from reference acceleration ($\ddot{x}_r$, $\ddot{\varphi}_r$) and velocity error:

$$\ddot{x}^C = \ddot{x}_r + K_x \cdot \left(\dot{x}_r - \dot{x}_a\right) \tag{11}$$

$$\ddot{\varphi}^C = \ddot{\varphi}_r + K_{fi} \cdot \left(\dot{\varphi}_r - \dot{\varphi}_a\right) \tag{12}$$

$\dot{x}_a$: actual velocity, $\dot{\varphi}_a$: actual angular velocity

Implemented neural network is constructed with one hidden layer with a non-linear transfer function and an output layer with linear transfer function. 25 neurons in hidden layer were used. NN inputs are reference and actual velocity, reference angular and actual angular velocity of the robot and velocity errors of the wheels. Error vector of the output layer is calculated as linear combination of velocity error and acceleration error of the wheels. Learning algorithm is the same as the traditional backpropagation learning rule.

Robot's initial translational and angular velocity in the beginning of the simulation are set zero ($\dot{x} = 0$ m/s, $\dot{\varphi} = 0$ rad/s). The desired end velocity is 1 m/s and desired end angular velocity 5 rad/s. It is reached in 0.32 s. Equivalently the desired end velocity of right wheel is 1.19 m/s and end velocity of left wheel 0.81 m/s. Disturbance torque (M_d) were added to the model with the Coulomb's friction (M_{dc} = 0.0025) and the velocity dependent part to the viscous friction (k_{transl}=k_{rotat}=0.005) Eq. (5).

For simulated movement 30% of available torque reach for the reference trajectory. The gains for estimation of calculated accelerations were K_x = 10 and K_{fi} = 10.

Fig. 4 presents desired and actual velocity and desired and actual angular velocity of the robot. Corresponding errors are shown on Fig. 5.

Fig. 6 and Fig. 7 show results for each wheel of the robot. Maximal dynamical velocity error of robot is 0.009 m/s and maximal dynamical angular velocity error of robot is 0.66 rad/s. There is no steady state error.

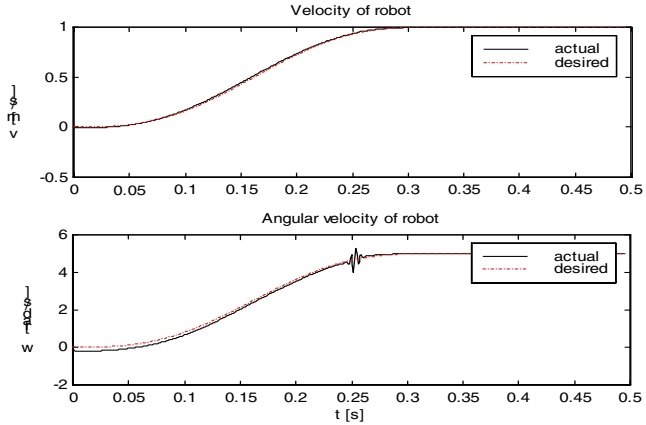


Fig. 4 Desired and actual velocity and angular velocity of robot

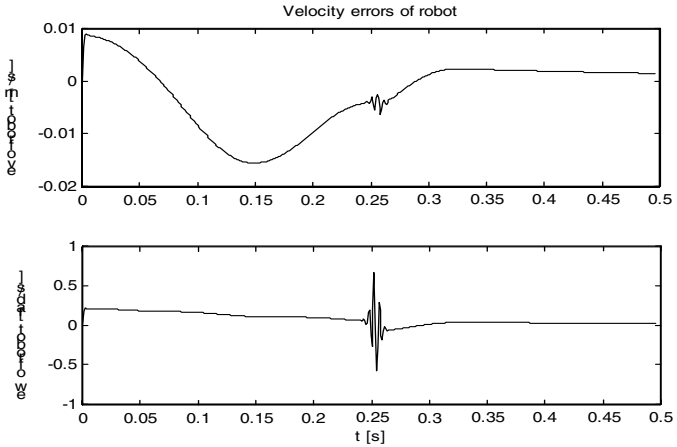


Fig. 5 Velocity and angular velocity error of robot

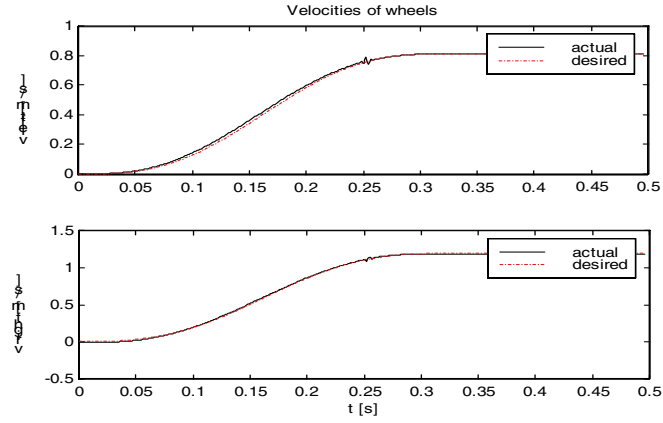


Fig. 6 Desired and actual velocities of wheels

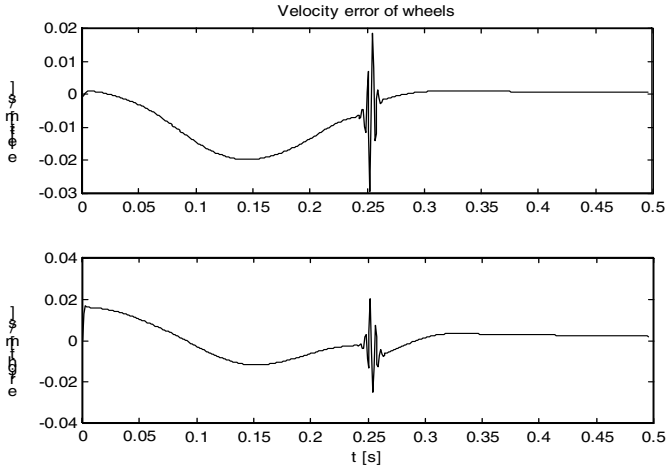


Fig. 7 Velocity errors of wheels

5. Conclusion

The research on robot soccer has various aspects, like

- The robot soccer is an interdisciplinary research theme including robotics, image processing, communication, cooperation, intelligent control and others.
- Robot soccer is a good tool for the entertainment and leisure as well as education.
- Robot soccer is a good test bed to implement and test the algorithms for Multi-Agent-Systems.

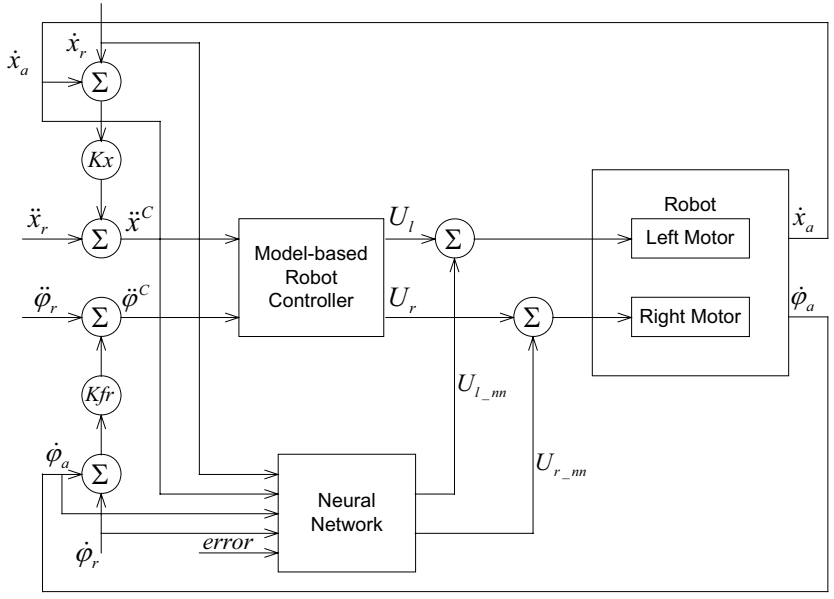


Fig. 8 Model based controller with neural network disturbance estimation

An adaptive velocity control algorithm for the control of soccer robots was presented in this paper. Controller is based on a derived dynamical model of the robot, which is upgraded with a neural network, that compensate unmodelled dynamics as well as other disturbances, that arise from frequent collisions between robots. Based on simulation the controller result in fast error convergence and no steady state error.

A robot can move with maximum speed approximately 2.5 m/s. The vision system can detect objects at constant frame rate (30 frames per second). The exact position of objects can not be calculated. It is necessary to predict future locations of the ball. For this purpose an extended Kalman filter (EKF) is implemented.

6. References

1. M.-W. Han,, P. Kopacek, G. Kronreif : "MAS at IHRT", *Survey paper by the 1st IFAC Workshop on Multi-Agent-Systems in Production MAS'99*, Vienna, Austria,
2. M.-W. Han, P. Kopacek, G. Novak and A. Rojko : "Adaptive Velocity Control of Mobile Soccer Robots", the 9th Int. Workshop on Robotics in Alp-Adria-Danube-Region RAAD '00
3. M.-W. Han, P. Kopacek, G. Novak : "Roboterbussball", 45th International Scientific Colloquium, October 4 - 6, 2000, Ilmenau, Germany (will be published in German.)
4. J.-H. Kim (Editor). First Micro-Robot World Cup Soccer Tournament, MIROST, *Robotics and Autonomous Systems* Vol. 2 No. 2,10. September 1997
5. T. Lüth: *Technische Multi-Agenten-Systeme*, Hanser Verlag, 1998 (in German)

Neuro-Fuzzy Systems

Universal Approximator Employing Neo-Fuzzy Neurons

Vitaliy Kolodyazhniy[1], Yevgeniy Bodyanskiy[1], and Peter Otto[2]

[1]Control Systems Research Laboratory,
Kharkiv National University of Radioelectronics,
14, Lenin Av., Kharkiv, 61166, Ukraine
bodya@kture.kharkov.ua.kolodyazhniy@ukr.net

[2]Department of Informatics and Automation,
Technical University of Ilmenau,
PF 100565, 98684 Ilmenau, Germany
peter.otto@systemtechnik.tu-ilmenau.de

Abstract. A novel fuzzy neural network, called *Fuzzy Kolmogorov's Network* (FKN), is considered. The network consists of two layers of neo-fuzzy neurons (NFNs) and is linear in both the hidden and output layer parameters, so it can be trained with very fast and computationally efficient procedures. Two-level structure of the rule base helps the FKN avoid the combinatorial explosion in the number of rules, while the antecedent fuzzy sets completely cover the input hyperbox. The number of rules in the FKN depends linearly on the dimensionality of input space. The validity of theoretical results and the advantages of the FKN are confirmed by a comparison with other techniques in benchmark problems and a real-world problem of electrical load forecasting.

1 Introduction

According to the Kolmogorov's superposition theorem (KST) [1], *any* continuous function of d variables can be *exactly* represented by superposition of continuous functions of one variable and addition:

$$f(x) = \sum_{l=1}^{2d+1} g_l \left[\sum_{i=1}^{d} \psi_{l,i}(x_i) \right], \tag{1}$$

where $x \in [x_1^{min}, x_1^{max}] \times \ldots \times [x_d^{min}, x_d^{max}]$, $g_l(\bullet)$ and $\psi_{l,i}(\bullet)$ are some continuous univariate functions, and $\psi_{l,i}(\bullet)$ are independent of f. Aside from the exact representation, the KST can be used for the construction of parsimonious universal ap-

V. Kolodyazhniy et al.: *Universal Approximator Employing Neo-Fuzzy Neurons*, Advances in Soft Computing **2**, 631–640 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

proximators, and has thus attracted the attention of many researchers in the field of soft computing.

Hecht-Nielsen was the first to propose a neural network approximation of KST [2], but did not consider how such a network can be constructed. Computational aspects of approximate version of KST were studied by Sprecher [3], [4]. Igelnik and Parikh [5] proposed the use of spline functions for the construction of Kolmogorov's approximation. Yam *et al* [6] proposed the multi-resolution approach to fuzzy control, based on the KST, and proved that the KST representation can be realized by a two-stage rule base. They demonstrated that the exponential growth in number of rules can be avoided via the two-stage fuzzy inference, but did not show how such a rule base could be created from data. Lopez-Gomez and Hirota developed the Fuzzy Functional Link Network (FFLN) [7] based on the fuzzy extension of the Kolmogorov's theorem. The FFLN is trained via fuzzy delta rule, whose convergence can be quite slow.

We propose a novel KST-based universal approximator called Fuzzy Kolmogorov's Network (FKN) with simple structure and optimal linear learning procedures with high rate of convergence.

2 Network Architecture

The FKN is comprised of two layers of neo-fuzzy neurons (NFNs) [8] and is described by the following equations:

$$\hat{f}(x_1,\ldots,x_d)=\sum_{l=1}^{n} f_l^{[2]}(o^{[1,l]}),\quad o^{[1,l]}=\sum_{i=1}^{d} f_i^{[1,l]}(x_i),\quad l=1,\ldots,n, \tag{2}$$

where n is the number of hidden layer neurons, $f_l^{[2]}(o^{[1,l]})$ is the l-th nonlinear synapse in the output layer, $o^{[1,l]}$ is the output of the l-th NFN in the hidden layer, $f_i^{[1,l]}(x_i)$ is the i-th nonlinear synapse of the l-th NFN in the hidden layer.

The equations for the hidden and output layer synapses are

$$f_i^{[1,l]}(x_i)=\sum_{h=1}^{m_1}\mu_{i,h}^{[1]}(x_i)w_{i,h}^{[1,l]},\quad f_l^{[2]}(o^{[1,l]})=\sum_{j=1}^{m_2}\mu_{l,j}^{[2]}(o^{[1,l]})w_{l,j}^{[2]}, \tag{3}$$
$$l=1,\ldots,n,\quad i=1,\ldots,d,$$

where m_1 and m_2 is the number of membership functions (MFs) per input in the hidden and output layers respectively, $\mu_{i,h}^{[1]}(x_i)$ and $\mu_{l,j}^{[2]}(o^{[1,l]})$ are the MFs, $w_{i,h}^{[1,l]}$ and $w_{l,j}^{[2]}$ are the tunable weights.

Nonlinear synapse is a single input-single output fuzzy inference system with crisp consequents, and is thus a universal approximator [9] of univariate functions. It can provide a piecewise-linear approximation of any functions $g_l(\bullet)$ and

$\psi_{l,i}(\bullet)$ in (1). So the FKN, in turn, can approximate any function $f(x_1,\ldots,x_d)$ on the input space hyperbox as in (1).

The output of the FKN is computed as the result of two-stage fuzzy inference:

$$\hat{y} = \sum_{l=1}^{n}\sum_{j=1}^{m_2}\mu_{l,j}^{[2]}\left[\sum_{i=1}^{d}\sum_{h=1}^{m_1}\mu_{i,h}^{[1]}(x_i)w_{i,h}^{[1,l]}\right]w_{l,j}^{[2]}. \tag{4}$$

The description (4) corresponds to the following two-level fuzzy rule base:

$$\text{IF } x_i \text{ IS } X_{i,h} \text{ THEN } o^{[1,1]} = w_{i,h}^{[1,1]}d \text{ AND}\ldots\text{AND } o^{[1,n]} = w_{i,h}^{[1,n]}d, \quad i=1,\ldots,d, \quad h=1,\ldots,m_1, \tag{5}$$

$$\text{IF } o^{[1,l]} \text{ IS } O_{l,j} \text{ THEN } \hat{y} = w_{l,j}^{[2]}n, \quad l=1,\ldots,n, \quad j=1,\ldots,m_2, \tag{6}$$

where $X_{i,h}$ and $O_{l,j}$ are the antecedent fuzzy sets in the first and second level rules, respectively. Each first level rule contains n consequent terms $w_{i,h}^{[1,1]}d,\ldots,w_{i,h}^{[1,n]}d$, corresponding to n hidden layer neurons.

Total number of rules is

$$N_R^{FKN} = d\cdot m_1 + n\cdot m_2, \tag{7}$$

i.e., it depends *linearly* on the number of inputs d.

The rule base is complete, as the fuzzy sets $X_{i,h}$ in (5) completely cover the input hyperbox with m_1 membership functions per input variable. Due to the linear dependence (7), this approach is feasible for input spaces with high dimensionality d without the need for clustering techniques for the construction of the rule base.

Straightforward grid-partitioning approach with m_1 membership functions per input requires $(m_1)^d$ fuzzy rules, which results in combinatorial explosion and is practically not feasible for $d > 4$.

3 Learning Algorithm

The weights of the FKN are determined by means of a batch-training algorithm as described below. A training set containing N samples is used. The minimized error function is

$$E(t) = \sum_{k=1}^{N}\left[y(k)-\hat{y}(t,k)\right]^2 = \left[Y-\hat{Y}(t)\right]^T\left[Y-\hat{Y}(t)\right], \tag{8}$$

where $Y = [y(1),\ldots,y(N)]^T$ is the vector of target values, and $\hat{Y}(t) = [\hat{y}(t,1),\ldots,\hat{y}(t,N)]^T$ is the vector of network outputs at epoch *t*.

Yamakawa *et al* [8] proposed the use of gradient descent-based learning for the NFN. Although this method can be directly applied to the output layer, it would also require the use of the back-propagation technique [10] for the hidden layer. Besides that, the gradient descent-based learning procedure converges slowly.

However, since the nonlinear synapses (3) are linear in parameters, we can employ direct linear least squares (LS) optimization instead of derivative-based methods. To formulate the LS problem for the output layer, re-write (4) as

$$\hat{y} = W^{[2]^T}\varphi^{[2]}(o^{[1]}), \quad W^{[2]} = \left[w_{1,1}^{[2]}, w_{1,2}^{[2]},\ldots, w_{n,m_2}^{[2]}\right]^T, \tag{9}$$
$$\varphi^{[2]}(o^{[1]}) = \left[\mu_{1,1}^{[2]}(o^{[1,1]}), \mu_{1,2}^{[2]}(o^{[1,1]}),\ldots, \mu_{n,m_2}^{[2]}(o^{[1,n]})\right]^T.$$

The error function (8) is minimized when $Y - \hat{Y} = 0$. Thus, we can determine the values of the output layer weights solving the following equation:

$$Y - W^{[2]^T}\Phi^{[2]} = 0, \quad \Phi^{[2]} = \left[\varphi^{[2]}(o^{[1]}(1)),\ldots,\varphi^{[2]}(o^{[1]}(N))\right]^T. \tag{10}$$

The LS solution will be

$$W^{[2]} = \left(\Phi^{[2]^T}\Phi^{[2]}\right)^{-1}\Phi^{[2]^T}Y^{[2]}, \quad Y^{[2]} = Y. \tag{11}$$

Now we have to determine the hidden layer weights. The use of triangular MFs enables the linearization of the second layer around $o^{[1,l]}$:

$$f_l^{[2]}(o^{[1,l]}) = a_l^{[2]}(o^{[1,l]})o^{[1,l]} + b_l^{[2]}(o^{[1,l]}), \tag{12}$$

$$a_l^{[2]}(o^{[1,l]}) = \frac{w_{l,p+1}^{[2]} - w_{l,p}^{[2]}}{c_{l,p+1}^{[2]} - c_{l,p}^{[2]}}, \quad b_l^{[2]}(o^{[1,l]}) = \frac{c_{l,p+1}^{[2]}w_{l,p}^{[2]} - c_{l,p}^{[2]}w_{l,p+1}^{[2]}}{c_{l,p+1}^{[2]} - c_{l,p}^{[2]}}, \tag{13}$$

where $w_{l,p}^{[2]}$ and $c_{l,p}^{[2]}$ are the weight and center of the *p*-th MF in the *l*-th synapse of the output layer, respectively. The MFs in an NFN are chosen such that only two adjacent MFs *p* and *p*+1 fire at a time [8].

With respect to (2), (4), and (12), we obtain the expression for the linearized FKN:

$$\hat{y} = \sum_{l=1}^{n}\sum_{i=1}^{d}\sum_{h=1}^{m_1} a_l^{[2]}(o^{[1,l]})\mu_{i,h}^{[1]}(x_i)w_{i,h}^{[1,l]} + \sum_{l=1}^{n} b_l^{[2]}(o^{[1,l]}). \tag{14}$$

Re-write the previous equation as follows:

$$\hat{y} = W^{[1]T}\varphi^{[1]}(x) + \theta^{[1]}(x), \quad W^{[1]} = \left[w_{1,1}^{[1,1]}, w_{1,2}^{[1,1]}, \ldots, w_{d,m_1}^{[1,1]}, \ldots, w_{d,m_1}^{[1,n]}\right]^T, \tag{15}$$

$$\varphi^{[1]}(x) = \left[\varphi_{1,1}^{[1,1]}(x_1), \varphi_{1,2}^{[1,1]}(x_1), \ldots, \varphi_{d,m_1}^{[1,1]}(x_d), \ldots, \varphi_{d,m_1}^{[1,n]}(x_d)\right]^T,$$

$$\varphi_{i,h}^{[1,l]}(x_i) = a_l^{[2]}(o^{[1,l]})\mu_{i,h}^{[1,l]}(x_i), \quad \theta^{[1]}(x) = \sum_{l=1}^{n} b_l^{[2]}(o^{[1,l]}).$$

Introducing vector $\Theta^{[1]} = \left[\theta^{[1]}(x(1)), \ldots, \theta^{[1]}(x(N))\right]^T$ and matrix $\Phi^{[1]} = \left[\varphi^{[1]}(x(1)), \ldots, \varphi^{[1]}(x(N))\right]^T$ and noticing that $\hat{Y} = W^{[1]T}\Phi^{[1]} + \Theta^{[1]}$, we can formulate the LS problem for the hidden layer weights:

$$Y - (W^{[1]T}\Phi^{[1]} + \Theta^{[1]}) = (Y - \Theta^{[1]}) - W^{[1]T}\Phi^{[1]} = 0. \tag{16}$$

The solution of the LS problem is:

$$W^{[1]} = \left(\Phi^{[1]T}\Phi\right)^{-1}\Phi^{[1]T}Y^{[1]}, \quad Y^{[1]} = Y - \Theta^{[1]}. \tag{17}$$

The solutions (11) and (17) are not unique when matrices $\Phi^{[q]T}\Phi^{[q]}$ are singular ($q = 1,2$ is the layer number). To avoid this, instead of (11) and (17) we find

$$W^{[q]}(t) = \left(\Phi^{[q]T}(t)\Phi^{[q]}(t) + \eta I\right)^{-1}\Phi^{[q]T}(t)Y^{[q]}(t), \tag{18}$$

where η is the regularization parameter with typical value $\eta = 10^{-5}$.

The FKN is trained via a two-stage derivative-free optimization procedure without any nonlinear operations. In the forward pass, the output layer weights are calculated. In the backward pass, calculated are the hidden layer weights. The number of tuned parameters in the hidden layer is $S_1 = d \cdot m_1 \cdot n$, in the output layer $S_2 = n \cdot m_2$, and total $S = S_1 + S_2 = n \cdot (d \cdot m_1 + m_2)$. Thus, in the forward pass, a matrix $S_2 \times S_2$ is inverted, and in the backward pass inverted is a matrix $S_1 \times S_1$. For comparison, the nonlinear LS methods, such as the Gauss-Newton and Levenberg-Marquardt procedures, require the inversion of a matrix $S \times S$. Since the number of calculations in matrix inversion is proportional to S^3 and it will always hold that $S^3 > S_1^3 + S_2^3$, the proposed training method is much faster.

Hidden layer weights are initialized deterministically using the formula

$$w_{h,i}^{[1,l]} = \exp\left\{-\frac{i\left[m_1(l-1) + h - 1\right]}{d(m_1 n - 1)}\right\}, h = 1, \ldots, m_1, i = 1, \ldots, d, l = 1, \ldots, n, \tag{19}$$

broadly similar to the parameter initialization technique proposed in [5] for the KSN based on rationally independent random numbers. The output layer weights are initialized with zeros.

4 Experiments

To verify the theoretical results and compare the performance of the proposed network to the known approaches, we have carried three experiments. The first two were the well-known benchmarks: the XOR problem and Mackey-Glass time series prediction [11]. The third experiment was the real-world problem of electric load forecasting for a region in Germany [12, 13].

4.1 The XOR problem

The XOR problem was solved by the FKN with 1 neuron in the hidden layer with 2 MFs per input, 1 synapse in the output layer with 3 MFs (7 parameters altogether). The FKN was trained for 1 epoch, after which it realized the XOR function with no errors. For comparison, a one-hidden layer perceptron with 2 neurons in the hidden layer, 1 neuron in the output layer (9 parameters), trained for 10 epochs with the Levenberg-Marquardt procedure gave 0.36 errors on average after 10 runs. The results of the first experiment are summarized in Table 1, and visualized in Fig. 1.

Table 1. Results of the XOR problem solution

Network	Param.	Epochs	Runs	Errors min	Errors max	Errors average
FKN	7	1	1	0	0	0
MLP	9	10	10	0	2	0.36

4.2 Mackey-Glass time series prediction

The time series, used in the second experiment, was generated by the chaotic Mackey-Glass time-delay differential equation [11]:

$$\frac{dy(t)}{dt} = \frac{0.2\, y(t-\tau)}{1+y^{10}(t-\tau)} - 0.1\, y(t)\,. \tag{20}$$

The values of the time series (20) at each integer point were obtained by means of the fourth-order Runge-Kutta method. The time step used in the method was 0.1, initial condition $y(0)=1.2$, delay $\tau=17$, and $y(t)$ was derived for $t=0,...,1200$. The values $y(t-18), y(t-12), y(t-6)$, and $y(t)$ were used to predict $y(t+6)$.

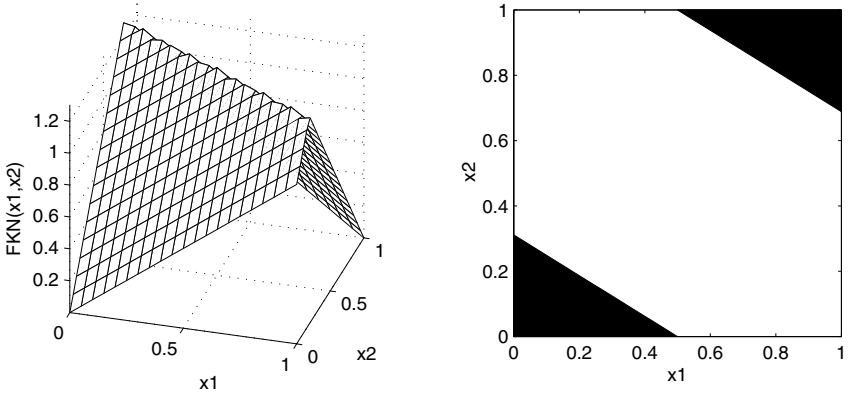


Fig. 1. The XOR problem: learned surface (*left*) and decision regions (*right*)

From the generated data, 500 values for $t = 118,...,617$ were used as the training data set, and the next 500 for $t = 618,...,1117$ as the checking data set. The FKN used for prediction had 4 inputs, 9 neurons in the hidden layer with 3 MFs per input, and 1 neuron in the output layer with 5 MFs per synapse (153 adjustable parameters altogether). It demonstrated similar performance as a two-hidden layer perceptron with 145 parameters trained with the Levenberg-Marquardt procedure. Both networks were trained for 50 epochs.

Root mean squared error on the training and checking sets ($RMSE_{TRN}$ and $RMSE_{CHK}$) was used to estimate the accuracy of predictions. The results are listed in Table 2. Actual time series, the prediction provided by the FKN, and prediction error are shown in Fig. 2.

Table 2. Results of Mackey-Glass time series prediction

Network	Param.	Epochs	$RMSE_{TRN}$	$RMSE_{CHK}$
FKN	153	50	0.0028291	0.004645
MLP	145	50	0.002637	0.003987

4.3 Eletric load forecasting

The electric load data were provided by a local supplier from Thuringia, Germany [13], and describe hourly electric load in that region in the years of 1996 and 1997 (8784 and 8760 samples respectively). Sampling time of these data is 1 hour. The data for the year of 1996 were used for training, and for the year of 1997 for testing. The forecast made by the FKN was 1 hour ahead.

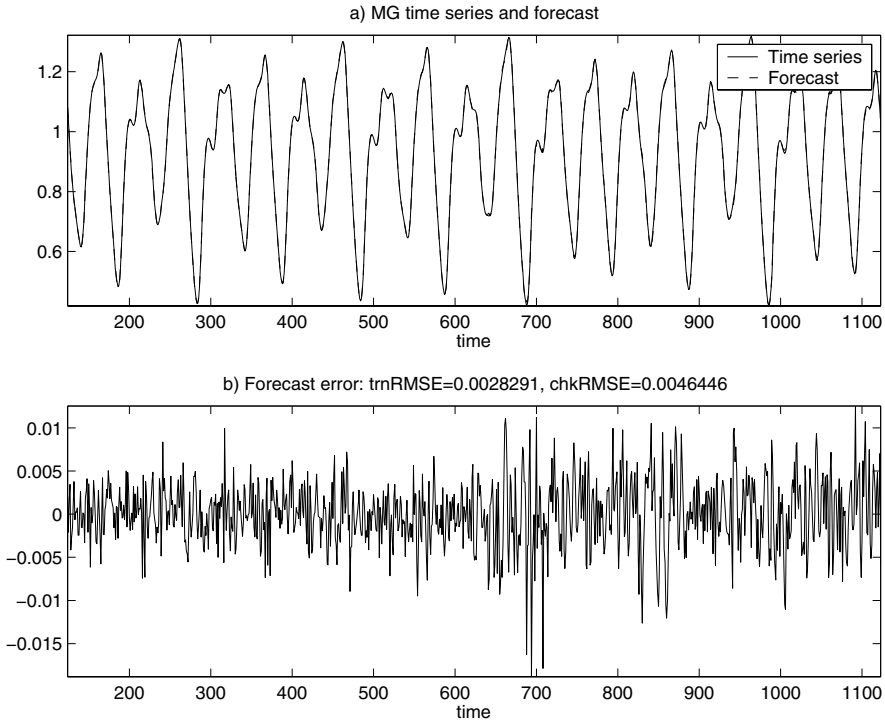


Fig. 2. Mackey-Glass time series prediction

The input variables to the forecasting FKN were the load a week ago and yesterday at the same hour as predicted, the load an hour ago, the current load, the change in load from the previous to the current hour, the number of the current hour within the current year, current week, and current day (8 inputs altogether).

The FKN used for prediction contained 17 neurons in the hidden layer with 4 MFs per input, and 1 neuron in the output layer with 7 MFs per synapse (663 adjustable parameters).

After training for 50 epochs, the network gave 1.41% mean average percentage error (MAPE) of prediction for the year of 1997. In Fig. 3, the forecast for three weeks at the end of March and the beginning of April 1997 is shown. This period includes Easter holidays and the changeover from the winter to summer time, so it is characterized by less regularity of electricity consumption than most other weeks throughout a year. For this period alone, the FKN provided a prediction with MAPE=2.26%. For comparison, for July 1997 MAPE=1.29%, because there were no holidays during this month.

These results are about 40% more accurate than those previously obtained for the same time series by the methods proposed in [12, 13]. However, they can be further improved by taking more input variables into account, such as the dates of holidays, air temperature, etc.

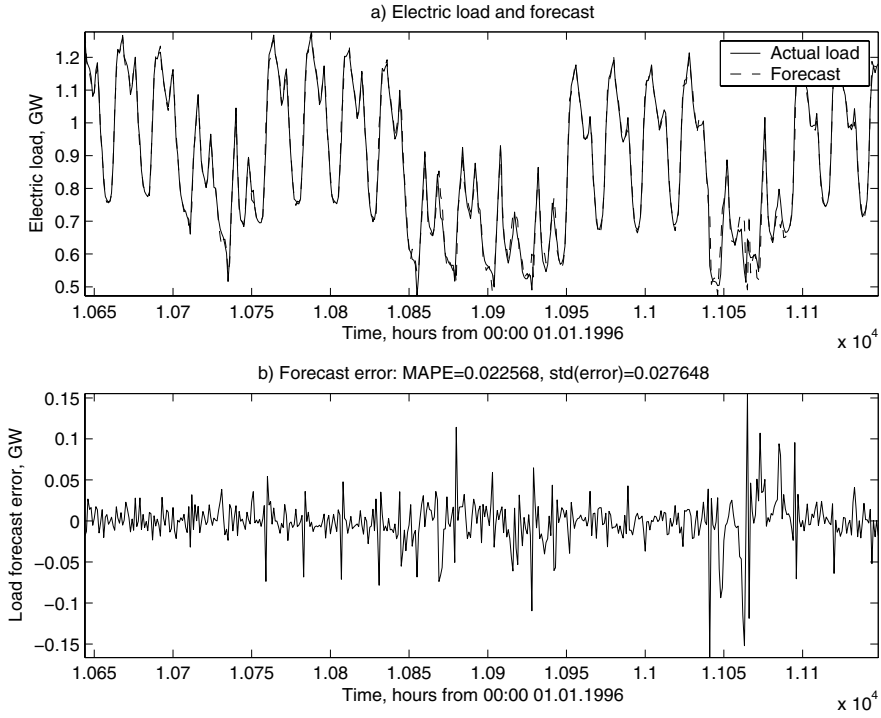


Fig. 3. Electric load forecast for 20.03.1997 – 09.04.1997

5 Conclusion

In the paper, a new practical and feasible approach to the construction of KST-based universal approximators was proposed. The FKN is the first multilayer network, containing the neo-fuzzy neurons in both the hidden and output layers, and it is not affected by the curse of dimensionality because of its two-level structure. The use of the neo-fuzzy neurons enabled us to develop linear training procedures for all the parameters in the FKN. High-performance *analog* hardware implementation of the pre-trained FKN is straightforward [14]. The output layer of the hardware-based FKN can be adapted online via the gradient descent method using the onboard learning circuits. The universality of internal functions in the KST implies that the once-trained hidden layer can be fixed for all the problems with the dimensionality d.

Although the FKN demonstrated excellent results in the experiments described above, its performance can be further improved via the tuning of antecedent parameters of fuzzy rules, and the use of membership functions other than triangular. This would require the development of improved learning algorithms.

References

1. Kolmogorov, A.N.: On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. Dokl. Akad. Nauk SSSR **114** (1957) 953-956
2. Hecht-Nielsen, R: Kolmogorov's mapping neural network existence theorem. Proc. IEEE Int. Conf. on Neural Networks, San Diego, CA, Vol. 3 (1987) 11-14
3. Sprecher, D.A.: A numerical implementation of Kolmogorov's superpositions. Neural Networks **9** (1996) 765-772
4. Sprecher, D.A.: A numerical implementation of Kolmogorov's superpositions II. Neural Networks **10** (1997) 447–457
5. Igelnik, B., and Parikh, N.: Kolmogorov's spline network. IEEE Transactions on Neural Networks **14** (2003) 725-733
6. Yam, Y., Nguyen, H. T., and Kreinovich, V.: Multi-resolution techniques in the rules-based intelligent control systems: a universal approximation result. Proc. 14th IEEE Int. Symp. on Intelligent Control/Intelligent Systems and Semiotics ISIC/ISAS'99, Cambridge, Massachusetts, September 15-17 (1999) 213-218
7. Lopez-Gomez, A., Yoshida, S., Hirota, K.: Fuzzy functional link network and its application to the representation of the extended Kolmogorov theorem. International Journal of Fuzzy Systems **4** (2002) 690-695
8. Yamakawa, T., Uchino, E., Miki, T., and Kusanagi, H.: A neo fuzzy neuron and its applications to system identification and prediction of the system behavior. Proc. 2nd Int. Conf. on Fuzzy Logic and Neural Networks “IIZUKA-92”, Iizuka, Japan (1992) 477-483
9. Kosko, B.: Fuzzy systems as universal approximators. Proc. 1st IEEE Int. Conf. on Fuzzy Systems, San Diego, CA (1992) 1153-1162
10. Rumelhart, D.E., Hinton, G.R., Williams, R.J.: Learning Internal Representation by Error Propagation. In: Rumelhart, D.E., McClelland, J.L. (Eds.): Parallel Distributed Processing, Vol. 1. MIT Press, Cambridge, MA (1986) 318–364
11. Mackey, M. C., and Glass, L.: Oscillation and chaos in physiological control systems. Science **197** (1977) 287-289
12. Otto, P. and Schunk, T.: Fuzzy based time series forecasting of electric load. Proc. European Control Conference (ECC’99), Karlsruhe, Germany (1999)
13. Otto, P., Bodyanskiy, Ye., Kolodyazhniy, V.: A new learning algorithm for a forecasting neuro-fuzzy network. Integrated Computer-Aided Engineering **10** (2003) 399-409
14. Miki, T., and Yamakawa, T.: Analog implementation of neo-fuzzy neuron and its on-board learning. In: Mastorakis, N.E. (Ed.): Computational Intelligence and Applications, WSES Press, Piraeus (1999) 144–149

Combined Learning Algorithm for a Self-Organizing Map with Fuzzy Inference

Yevgeniy Bodyanskiy[1], Yevgen Gorshkov[1], Vitaliy Kolodyazhniy[1], and Andreas Stephan[2]

[1] Control Systems Research Laboratory,
Kharkiv National University of Radioelectronics,
14, Lenin Av., Kharkiv, 61166, Ukraine,
bodya@kture.kharkov.ua, ye_gorshkov@ukr.net, kolodyazhniy@ukr.net
[2] PSI-Tec GmbH, Grenzhammer, 8, D-98693, Ilmenau, Germany,
Stephan@psi-tec.de

Abstract. A combined learning algorithm for a self-organizing map (SOM) is proposed. The algorithm accelerates information processing due to the rational choice of the learning rate parameter, and can work when the number of clusters is unknown, as well as when the clusters are overlapping. This is achieved via the introduction of fuzzy inference that determines the level of membership of the classified pattern to each of the available classes. For neighborhood and membership functions, raised cosine is used. This function provides more flexibility and some new properties for the self-learning and clustering procedures.

1 Introduction

SOMs, proposed by T. Kohonen [1], are widely used in the problems of data mining and intelligent data processing, such as clustering, heteroassociation, diagnostics, information compression, etc.

The properties of self-organization in SOMs are due to the tuning of synaptic weights without an external training signal, i.e. in the unsupervised mode when each of the incoming patterns causes the tuning of certain parameters.

Decision about the membership of the incoming pattern to one of the clusters according to the "winner-take-all" rule in the case of overlapping non-convex clusters can result in inaccurate clustering because one pattern can belong to several clusters at the same time with some degrees of membership. That's why it is reasonable to provide the SOM with the properties of fuzzy clustering [2].

In [3, 4], a fuzzy SOM was proposed in which the neurons of the original SOM are replaced by fuzzy sets and fuzzy rules. This network proved to be quite effective in pattern recognition, though its learning properties deteriorated. In [5, 6], a fuzzy Kohonen clustering network was proposed which is

Y. Bodyanskiy et al.: *Combined Learning Algorithm for a Self-Organizing Map with Fuzzy Inference*, Advances in Soft Computing **2**, 641–650 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

essentially an implementation of the fuzzy c-means clustering algorithm quite different from the classical SOM.

The goal of this work is the synthesis of a combined self-organization algorithm, that would accelerate the information processing and improve clustering accuracy in the case of overlapping clusters via the use of a non-standard neighborhood function and a fuzzy inference procedure. It is also desired to provide an ability of operation when there is no prior information about the number of clusters is available.

The self-organization procedure described below is based on the principles of competitive learning. The conventional self-organization procedure is implemented in three basic stages [7]: competition, co-operation and synaptic adaptation.

2 Competition and Synaptic Adaptation

Consider a SOM, containing n receptive fields and m neurons in the Kohonen layer. Each has its own n-dimensional vector of synaptic weights w_j, $j = 1, 2, \ldots, m$, and all the neurons during the unsupervised learning or clustering receive the incoming patterns $x(k)$, $k = 1, 2, \ldots$ at their inputs and generate the signals $y_j(k) = w_j^T(k)x(k)$, $j = 1, 2, \ldots, m$ at their outputs. The incoming patterns close in the sense of the adopted metrics can activate either the same neuron w_j or a couple of neighboring neurons.

The competition process starts with the analysis of the pattern $x(k)$ coming from the receptive field layer to all the neurons of the Kohonen layer. The distance

$$\mathcal{D}(w_j(k), x(k)) = \|x(k) - w_j(k)\|, \tag{1}$$

is calculated for all the neurons. If the inputs and the synaptic weights are normalized as

$$\|x(k)\| = \|w_j(k)\| = 1, \tag{2}$$

and the distance measure is the Euclidean metrics then the dot product $w_j^T(k)x(k) = y_j(k) = \cos(w_j(k), x(k)) = \cos\theta_j(k)$ can be used as the neighborhood measure for the vectors $w_j(k)$ and $x(k)$. In this case the distance (1) assumes the form

$$\mathcal{D}_{\cos}(w_j(k), x(k)) = \mathcal{D}_{\cos}(y_j(k)) = \sqrt{2(1 - y_j(k))},$$

and is shown in figure 1.

Then the winning neuron closest to the input pattern is determined, such that

$$\mathcal{D}(w^*(k), x(k)) = \min_j \mathcal{D}(w_j(k), x(k)).$$

After that (skipping the co-operation stage for the moment) synaptic weights can be adjusted according to the elementary "winner-take-all" learning rule

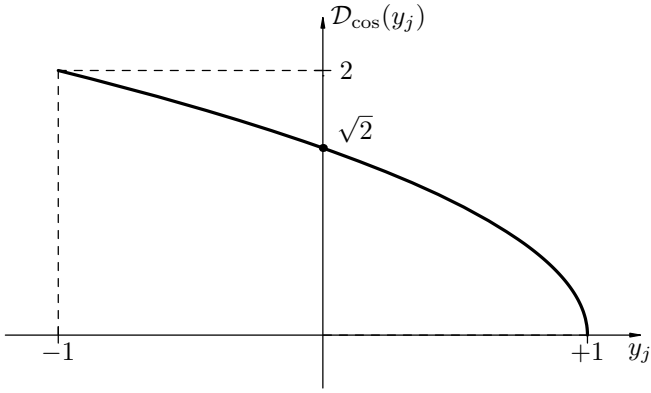


Fig. 1. Dependence of distance $\mathcal{D}(y_j(k))$ on network output

$$w^*(k+1) = w^*(k) + \eta(k)(x(k) - w^*(k)), \tag{3}$$

where w^* is the winning neuron, and the parameter $\eta(k)$ determines the step length and is usually chosen empirically, such that it would decrease monotonically in time.

It can be readily seen that such a learning rule minimizes the criterion

$$E_j^K = \sum_k E_j^K(k) = \frac{1}{2}\sum_{p=1}^{k} \|x(p) - w_j\|^2, \tag{4}$$

i.e. the arithmetic mean

$$w_j(k) = \frac{1}{k}\sum_{p=1}^{k} x(p)$$

is used as the estimate of the synaptic weights.

The considered learning rule is thus a stochastic approximation procedure, and the coefficient $\eta(k)$ should be chosen according to the Dvoretzky's condition [8], though the choice $\eta(k) = 1/k$ results in obvious deterioration of the learning rate.

The requirement of the monotonic decay of the step parameter is met in the procedure

$$\eta(k) = r^{-1}(k), \quad r(k) = \alpha r(k-1) + \|x(k)\|^2, \quad 0 \le \alpha \le 1, \tag{5}$$

proposed in [9]. This procedure coincides with that proposed in [10] for the adaptive algorithm with improved rate of convergence, when the input signals are normalized and $\alpha = 1$.

Varying the forgetting factor α, a wide range of the step length

$$\frac{1}{k} \le \eta(k) \le 1$$

can be obtained.

3 Co-operation and Synaptic Adaptation

One of the specific features of SOMs is the possible co-operation stage during the self-organization when the winning neuron determines a local area of topological neighborhood in which not only the winner, but also his closest neighbors are updated. This topological area is defined by the neighborhood function $\phi(l,k)$, dependent on the distance $\mathcal{D}(w^*(k), w_l(k))$ between the winner $w^*(k)$ and a neuron $w_l(k)$, $l = 1, 2, \ldots, m$, and on a width parameter $\sigma(k)$.

Usually, $\phi(l,k)$ is a kernel function symmetric with respect to the maximum at $w^*(k) = w_{l^*}(k)$ $(\phi(l^*,k) = 1)$ and decreasing with the distance $\mathcal{D}(w^*(k), w_l(k))$. The gaussian, Mexican hat, cosine, and alike functions are used for the neighborhood function.

The use of a neighborhood function leads to the learning rule

$$w_l(k+1) = w_l(k) + \eta(k)\phi(l,k)(x(k) - w_l(k)), \quad l = 1, 2, \ldots, m, \tag{6}$$

minimizing the criterion

$$E_l^K = \sum_{p=1}^{k} E_l^K(p) = \frac{1}{2}\sum_{p=1}^{k} \phi(l,p)\|x(p) - w_l\|^2 \tag{7}$$

according to the principle "winner-take-more".

Convergence analysis of the self-organization process [11–13] shows that during synaptic adaptation phase both learning rate $\eta(k)$ and neighborhood width parameter must decrease in time. H. Ritter and K. Schulten [12, 13] proposed to adapt the σ parameter of the widely used Gaussian function

$$\phi(l,k) = \exp\left(-\frac{\|w^*(k) - w_l(k)\|^2}{\sigma(k)^2}\right)$$

with the following procedure

$$\sigma(k) = \beta\sigma(k-1), \quad 0 < \beta < 1, \tag{8}$$

where β is a scalar parameter which determines the neighborhood decrease rate.

Learning in a SOM can be realized with no competition stage at all. In this case, the synaptic weights can be tuned according to their proximity to the current pattern $x(k)$. The neighborhood function $\phi(l)$ will depend not on the winning neuron $w^*(k)$, but on the vector $x(k)$, and it will hold $\phi(j) = 1$ for $x(k) = w_j(k)$. The functions $\phi(l)$ will also decrease with the distance $\mathcal{D}(w_l(k), x(k))$ as a usual kernel function.

Consider the raised cosine radial basis function [14]

$$\phi(\theta) = \begin{cases} \dfrac{1+\cos\theta}{2}, & \text{for } \theta \in [-\pi, \pi], \\ 0, & \text{otherwise} \end{cases}$$

This leads to the neighborhood function

$$\phi_l(k) = \frac{1 + \cos(w_l(k), x(k))}{2} = \frac{1 + \cos\theta_l(k)}{2} = \frac{1}{2} + \frac{1}{2} y_l(k). \tag{9}$$

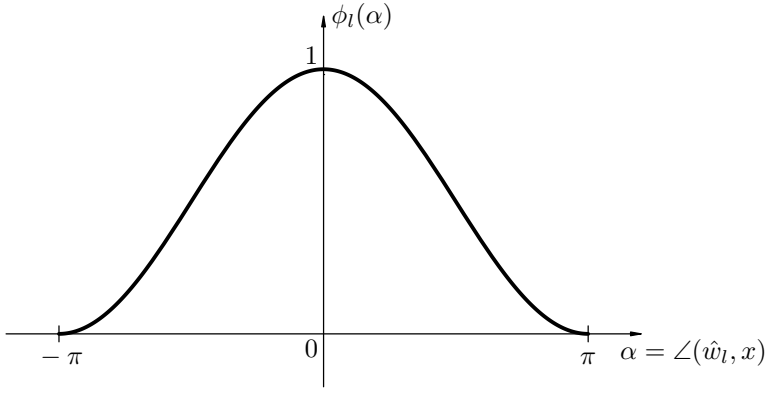


Fig. 2. Raised cosine neighborhood function

The learning algorithm in this case is

$$\begin{aligned} w_l(k+1) &= w_l(k) + \eta(k)\frac{1 + y_l(k)}{2}(x(k) - w_l(k)) = \\ &= w_l(k) + \frac{1}{2}\eta(k)(x(k) - w_l(k)) + \frac{1}{2}\eta(k) y_l(k)(x(k) - w_l(k)) \end{aligned} \tag{10}$$

and consists of two parts, the first $\eta(k)(x(k) - w_l(k))$ corresponding to the Kohonen learning rule, and the second — $\eta(k) y_l(k)(x(k) - w_l(k))$ — to the Instar rule proposed by S. Grossberg [15]. The Instar rule minimizes the criterion

$$E_l^G = \sum_k E_l^G(k) = \frac{1}{2}\sum_{p=1}^{k}(w_l^T x(p))^2 = \frac{1}{2}\sum_{p=1}^{k} y_l^2(p), \tag{11}$$

when the normalization conditions are met. Thus, the learning algorithm (10) optimizes the additive goal function

$$E_l = \frac{1}{2}E_l^K + \frac{1}{2}E_l^G. \tag{12}$$

The capabilities of the SOM can be extended by using a more general than (12) goal function

$$E_l = \gamma E_l^K + (1 - \gamma)E_l^G, \quad 0 \le \gamma \le 1, \tag{13}$$

resulting in the following learning algorithm

$$w_l(k+1) = w_l(k) + \eta(k)(\gamma(x(k) - w_l(k)) + (1-\gamma)y_l(k)(x(k) - w_l(k))), \quad (14)$$

and the neighborhood function

$$y(l) = \gamma + (1-\gamma)\cos\theta_l(k). \quad (15)$$

Varying the parameter γ, we can realize a particular compromise between the Kohonen and Grossberg learning.

To guarantee the convergence of the procedure (14), we have to provide for a possibility of decreasing the width of the neighborhood function $y_l(k)$ during the learning. Introducing a decreasing width parameter $\sigma(k)$ into (14), we obtain the final combined learning algorithm

$$\begin{cases} w_l(k+1) = w_l(k) + \eta(k)\delta_{[-1,1]}(\theta_l(k)\sigma^{-1}(k))(x(k) - w_l(k)) \\ \qquad\qquad \cdot (\gamma + (1-\gamma)\cos(\pi\theta_l(k)\sigma^{-1}(k))), \\ \eta(k) = r^{-1}(k), \quad r(k) = \alpha r(k-1) + \|x(k)\|^2, \quad 0 \le \alpha \le 1, \\ \sigma(k) = \beta\sigma(k-1), \quad 0 < \beta < 1, \end{cases} \quad (16)$$

where $\delta_{[a,b]}(x)$ is defined as $\delta_{[a,b]}(x) = 1$, if $x \in [a,b]; 0$, otherwise.

If a data set $x(1), x(2), \ldots, x(N)$ is given (with or without the classification of patterns), the result of the learning process will be the synaptic weights $w_l(N+1) = \hat{w}_l$, $l = 1, 2, \ldots, m$ that determine the centroids of the clusters.

4 Fuzzy Inference

A trained SOM can be used for the classification of the newly incoming patterns $x(p)$, $p = N{+}1, N{+}2, \ldots$. When the clusters are overlapping, a "winner-take-all" decision may be incorrect. To cope with this situation, it is reasonable to introduce an estimate of the pattern membership to each of the available clusters using the membership function

$$\mu_{\hat{w}_l}(x(p)) = \frac{1 + \cos(\hat{w}_l, x(p))}{2}, \quad (17)$$

which has the range of variation $0 \le \mu_{\hat{w}_l}(x) \le 1$ thus satisfying all conditions for a fuzzy membership function.

To determine the membership level of the input pattern $x(p)$ to all clusters a normalized membership value is used:

$$\mu_{\hat{w}_l}(x(p)) = \frac{y_l(x(p))}{\sum_{l=1}^{m} y_l(x(p))}. \quad (18)$$

The membership function $\mu_{\hat{w}_l}(x)$ is a special case of the quadratic radial-basis function [16, 17]

$$\phi_l(x) = \max\left\{0,\ 1 - \frac{\|x - \hat{w}_l\|^2}{\sigma_l^2}\right\} = \max\left\{0,\ 1 - 2\frac{1 - \hat{w}_l^T x}{\sigma_l^2}\right\}, \tag{19}$$

which assumes the form

$$\phi_l(x) = \frac{1 + \hat{w}_l^T x}{2}. \tag{20}$$

with the width of receptive field $\sigma_l^2 = 4$ when the normalization conditions are met. Thus, the receptive field of the introduced membership function (17) is a unit hypersphere as shown on figure 3.

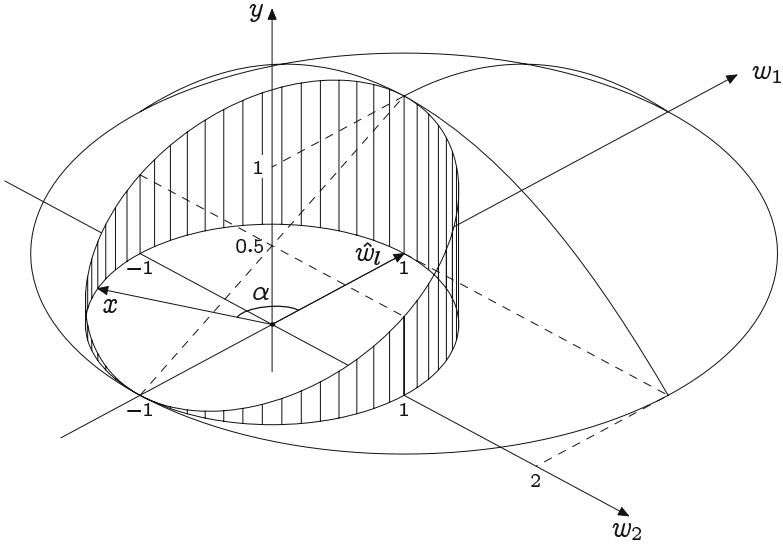


Fig. 3. Equivalence of the raised cosine and quadratic radial-basis functions

Furthermore, if prior classification of patterns is given in the training data set, a simple gradient descent based procedure can be used to determine cluster widths more accurately, e.g. using the methods introduced in [16–18]. Adaptation of the cluster widths would result in a more accurate fuzzy classification of the patterns.

5 Experiments

We tested the proposed algorithm on a set of standard benchmarks from the UCI repository (Iris, Wine, WBC, and Thyroid disease) [19]. The Iris data set consists of 150 patterns with 4 attributes, which are divided into 3 classes. The Wine data set consists of 179 patterns of 3 classes with 13 attributes. The Wisconsin Breast Cancer (WBC) data set consists of 699 samples of 2

classes with 10 attributes. We used only 683 of them, because 16 had missing values. The Thyroid disease data set consists of 215 patterns with 5 attributes, divided into 3 classes.

Our experiments show the possibility of the algorithm to determine number of clusters in the data set by finding cluster centroids. This is especially useful when no prior classification is given.

It should be noted that the standard "winner-take-all" learning rule (3) will not converge to the centroids of the clusters if the number of neurons (which is set constant for the network) exceeds the number of clusters because of undesirable competition between neighboring neurons.

For all the tests we initialized the network with 10% of patterns extracted randomly from the learning data set. To avoid redundant computation during the self-organization phase, we also introduced a simple network pruning procedure by replacing two of closely situated neurons with one neuron placed in the middle.

The following initial parameters values were set for all tests: $\eta(0) = 0.95$, $r(0) = 0$, $\sigma(0) = 1.4$, $\beta \approx 0.0015$, $\gamma = 0.45$.

For test purposes all the data sets were randomly divided into the learning and checking data sets respectively with 70% to 30% ratio.

We performed two runs of benchmarks with and without *a priori* classification known to the algorithm. In the latter case the resulting class labels for checking purposes were chosen by the nearest centroid of the full data set.

Some of the data sets (e.g. Thyroid disease) cannot be learned when no prior classification is given, because the degree of cluster overlap is high, and no evident cluster boundaries can be seen. We do not list these results as the algorithm finds incorrect number of classes.

Each self-organization process lasts for $30-50$ epoch, which is significantly faster than it is needed for conventional SOMs.

We performed the tests 100 times each and the classification errors (minimum, average, and maximum values) on a checking data sets for all the benchmarks are shown in table 1. Number of misclassified patterns is shown in brackets.

Table 1. Classification error on checking data sets

Dataset	Classification error		
	Min.	Avg.	Max.
Iris	2.2% (1)	10.0% (4)	15.6% (7)
Wine	0% (0)	3.7% (2)	9.2% (5)
WBC	0% (0)	1.5% (3.2)	3.4% (7)
Iris (supervised)	0% (0)	8.2% (3.7)	15.6% (7)
Wine (supervised)	0% (0)	3.7% (2)	9.3% (6)
WBC (supervised)	0% (0)	0.9% (2)	1.9% (4)
Thyroid (supervised)	0% (0)	4.9% (3)	10.8% (7)

It can be readily seen from the test results that the classification performance on the data sets with and without prior classification are almost equal for our tests. This confirms high performance of the proposed algorithm in finding cluster centroids in the case of self-organization without prior classification available.

6 Conclusion

A combined approach to the learning in a self-organizing network is proposed. The proposed algorithm have an improved rate of convergence, and can be used for the training of a classifier when the clusters are overlapping. Another advantage is the ability to determine the number of clusters in data with no prior information available (when $\gamma = 0.45 \div 0.5$) in both cases of separate or overlapping clusters. The algorithm was tested on a set of standard benchmarks (Iris, Wine, WBC, and Thyroid) with and without the information about classes. These experiments confirm high performance of the algorithm.

References

1. T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, Berlin, 1995.
2. J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, N.Y., 1981.
3. P. Vuorimaa. Fuzzy self-organizing maps. *Fuzzy Sets and Systems*, 66:223–231, 1994.
4. P. Vuorimaa. Use of the fuzzy self-organizing maps in pattern recognition. In *Proc. 3-rd IEEE Int. Conf. Fuzzy Systems "FUZZ–IEEE'94"*, pages 798–801, Orlando, USA, 1994.
5. E. C.-K. Tsao, J. C. Bezdek, and N. R. Pal. Fuzzy kohonen clustering networks. *Pattern Recognition*, 27:757–764, 1994.
6. R. D. Pascual-Marqui, A. D. Pascual-Montano, K. Kochi, and J. M. Caraso. Smoothly distributed fuzzy c-means: a new self-organizing map. *Pattern Recognition*, 34:2395–2402, 2001.
7. S. Haykin. *Neural Networks. A Comprehensive Foundation*. Prentice Hall, Inc., Upper Saddle River, N.Y., 1999.
8. A. Dvoretzky. On stochastic approximation. In *Proc. 3-rd Berkeley Symp. Math. Statistics and Probability*, volume 1, pages 39–55. University of California Press, 1956.
9. Ye. V. Bodyanskiy, I. P. Pliss, and T. V. Solovyova. Multistep optimal predictors for multi-variate non-stationary stochastic processes. *Dokl. AN USSR*, (12):47–49, 1986.
10. G. C. Goodwin, P. J. Ramadge, and P. E. Caines. A globally convergent adaptive predictor. *Automatica*, 17(1):135–140, 1981.
11. M. Cottrel and J. Fort. A stochastic model of retinotopy: a self-organizing process. *Biological Cybernetics*, 53:405–411, 1986.

12. H. Ritter and K. Shulten. On the stationary state of the Kohonen self-organizing sensory mapping. *Biological Cybernetics*, 54:234–249, 1986.
13. H. Ritter and K. Shulten. Covergence properties of Kohonen's topology conserving maps: fluctuations, stability, and climension selection. *Biological Cybernetics*, 60:59–71, 1988.
14. R. J. Schilling, J. J. Carroll, and A. F. Al-Ajlouni. Approximation of nonlinear systems with radial basis function neural networks. *IEEE Trans. on Neural Networks*, 12(1):1–15, 2001.
15. S. Grossberg. Classical and instrumental learning by neural networks. In *Proc. "Progress in Theoretical Biology"*, volume 3, pages 57–141, N.Y., 1974. Academic Press.
16. Ye. Bodyanskiy, O. Chaplanov, V. Kolodyazhniy, and P. Otto. Adaptive quadratic radial basis function network for time series forecasting. In *Proc. East West Fuzzy Coll. 2002, 10-th Zittau Fuzzy Coll.*, pages 164–172, Zittau/Goerlitz, 2002. HS.
17. Ye. Bodyanskiy, Ye. Gorshkov, V. Kolodyazhniy, and J. Wernstedt. Probabilistic neuro-fuzzy network with nonconventional activation functions. *Lecture Notes in Artificial Intelligence*, v.2773:973–979, 2003.
18. Ye. Bodyanskiy, Ye. Gorshkov, and V. Kolodyazhniy. Resource-allocating probabilistic neuro-fuzzy network. In *Proc. Third Conf. of the European Society for Fuzzy Logic and Technology (EUSFLAT–2003), 10–12 September, 2003*, pages 392–395, Zittau, Germany, 2003.
19. P. M. Murphy and D. W. Aha. *UCI Repository of machine learning databases*. University of California, Department of Information and Computer Science, CA, 1994.

Fuzzy/Neural Connection Admission Controller for Multimedia Traffic in Wireless ATM Networks

Jerzy Martyna

Institute of Computer Science, Jagiellonian University, Nawojki 11, Cracow, Poland,
martyna@ii.uj.edu.pl

Abstract In this paper, we propose fuzzy neural controller for a dynamic connection admission control (CAC) that supports the cell loss requirement and QoS parameters for multimedia traffic in Wireless ATM networks. Our CAC algorithms explicitly compute the bandwidth required for each class of connections based on the observed traffic statistics and the declared parameters.

Key words Connection admission control, neuro-fuzzy controller, WATM, QoS maintenance, unsupervised learning.

1 Introduction

Wireless Asynchronous Tranfer Mode (WATM) networks have become one of the most promising technologies for supporting broadband and mobile multimedia services [5, 6, 15]. We recall that WATM technology has the possibility to diversity traffic. Therefore, these networks have been designed to support a mixture of multimedia traffic (e.g. audio, video, data, images) with different traffic parameters (among others, peak bit rate, burst ratio, cell delay variation, cell loss ratio) and several Quality of Services (QoS) requirements. This concept of networks includes various traffic classes, such as constant bit rate (CBR), real-time variable rate (rt-VBR), nonreal-time variable bit rate (nrt-VBR), available bit rate (ABR) and unspecified bit rate (UBR).

We recall that in the traditional approach to call admission control (CAC) we must characterize the offered traffic a priori in terms of the parameters of stochastic or deterministic model. To these approaches belong, among others, the model of Guerin [4] and [3]. However, none of the models

J. Martyna: *Fuzzy/Neural Connection Admission Controller for Multimedia Traffic in Wireless ATM Networks*, Advances in Soft Computing **2**, 651–666 (2005)
www.springerlink.com © Springer-Verlag Berlin Heidelberg 2005

is suitable in many not fully incompletely characterised traffic sources and some worst case scenarios can practically appear in some situations. For instance, if more bandwidth is allocated than needed, the ATM network is utilized by network utilization.

To overcome the drawback of the model-based CAC schemes, some measurement-based neural fuzzy controllers have been proposed. Among others, Pittsillides et al. [12] have applied fuzzy explicit rate working mechanism to effectively control the traffic flow in ATM networks. A fuzzy connection admission controller used for a possible distribution of the cell ratio was proposed by [16]. In the paper by Youssef et al. [17] a controller for bandwidth allocation in ATM networks was proposed.

A dynamic connection admission control algorithm based on the traffic parameters, on-line traffic measurements and a fuzzy logic controller was proposed in the paper by Ren and Ramamurthy [14]. A new architecture of a fuzzy controller with a feedback loop for the traffic rate prediction and regulation has been given by Douligeris et al. [2]. However, the given solutions cannot be directly applicable to wireless ATM networks, where there are additional situations possible such as the handoff failure, call dropping, etc.

In this paper, we introduce a connection admission control (CAC) mechanism based on neuro-fuzzy network that supports ATM cell loss requirement and QoS parameters for multimedia traffic in Wireless ATM networks. Additionally, our CAC controller estimates the bandwidth required for each class of multimedia traffic, so as able to recognize of the movement direction of the Mobile Station (MS) in a cellular environment.

The structure of the paper is as follows. In section 2, we define the connection admission control scheme used in Wireless ATM networks. In section 3, we introduce our fuzzy neural network controller for providing QoS support in a variable wireless network environment in which QoS is adaptive within the range requested by the application. Section 3 is devoted to the learning algorithm used by our fuzzy neural network. Numerical results are presented in section 4. Conclusions are given in section 5.

2 Connection Admission Control Scheme in Wireless ATM Networks

In this section, we give the basic principles for bandwidth allocation scheme, providing the QoS support in a variable wireless network environment.

For simplicity we assume that to CAC mechanism belong at least two procedures: *"connecting_a_call"* and *"rejecting_a_call"*. The first procedure must identify call's class of service, such as CBR, rt-VBR, nrt-VBR, ABR and UBR. In the dependence of service class and the actual channel link capacity the bandwidth C_k $(k = 1, \ldots, K)$ is admitted, where K is the total number od class. For all call's class k the following condition must be satisfied:

$$\sum_{k=1}^{K} C_k \leq C_{link} \tag{1}$$

where C_{link} is the bandwidth of the link. A pool of free bandwidth C_f is defined as

$$C_f = C_{link} - \sum_{k=1}^{K} C_k \tag{2}$$

In the case of lack of free bandwidth each call borrows the bandwidth from the neighbouring cells. Additionally, thanks to the movement prediction the needed bandwidth in all neighbouring cells is reserved in the direction of movement. In the opposite direction of the movement the bandwidth is released and returned to the pool of free bandwidth for ech cells. Only in the case of lack of free bandwidth in the neighbouring cells the connection is rejected. Hence, the *"rejecting_a_call"* procedure is invoked. The *"connecting_a_call"* procedure estimates QoS parameters (such as *cell loss ratio* (CLR), *cell delay variation* (CDV), etc.) and compares them with the target QoS. If the estimated QoS parameters are smaller than the target QoS parameters, then the procedure improves the connection.

3 Fuzzy/Neural Connection Admission Controller for WATM Networks

In this section, we introduce a fuzzy neural network CAC controller for WATM networks that forces traffic measurements for determining the equivalent bandwidth, the reserved and released bandwidth, the change of QoS parameters, etc.

The architecture of our fuzzy neural CAC controller mechanism contains the following functional blocks: *Fuzzy Channel Allocator* (FCA), *Fuzzy Cell Selector* (FCS), *Neural Fuzzy CAC Mechanism* (NFCACM) *and QoS Parameters Controller* (QoSPC). The block diagram is shown in Fig. 1.

In our fuzzy/neural CAC controller mechanism, we used as input parameters: $\overline{v}$ - the speed of MS, P_H - the handoff (handover) probability [11], $\overline{M^{(j)}}$ - the mean ATM cell rate of jth call, $(\sigma^{(j)})^2$ - the variance of the ATM cell rate and two Quality of Service parameters for jth call, namely $QoS_1^{(j)}$, $QoS_2^{(j)}$. In our approach in the observed periods some measurement parameters such as the mean cell rate $\overline{M^{(j)}}$ and variance $(\sigma^{(j)})^2$ of jth call are given by

$$\overline{M^{(j)}} = \frac{1}{N_s} \sum_{n=1}^{N_s} R_n^{(j)} \tag{3}$$

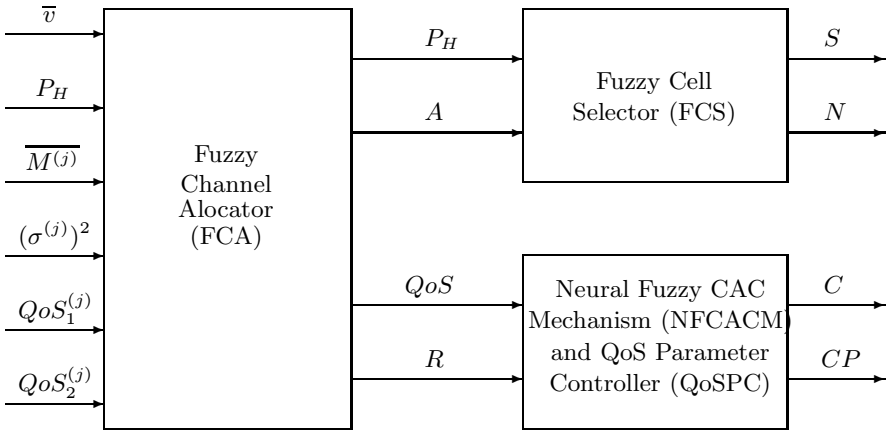


Fig. 1 The block diagram of Fuzzy/Neural Connection Admission Controler (FNCAC) for WATM networks

$$(\sigma^{(j)})^2 = \frac{1}{N_s - 1} \sum_{n=1}^{N_s} (R_n^{(j)} - \overline{M^{(j)}})^2 \tag{4}$$

where $R_n^{(j)}$ is a measured ATM cell rate for call j in the observed period n.

It allows us to estimate the autocovariance function $Cov_R(\tau)$ at sampled points as

$$Cov_{R^{(j)}}(kt_m) = \frac{1}{N_s - k} \sum_{n=1}^{N_s - k} (R_n^{(j)} - \overline{M^{(j)}})(R_{n+k}^{(j)} - \overline{M^{(j)}}), \;\; k = 0, 1, \ldots, N_s - 1, \tag{5}$$

which is unbiased for $k < N_s$. We assume that the statistics $\overline{M^{(j)}}$, ${\sigma^{(j)}}^2$ and $Cov_{R^{(j)}}(\tau)$ are updated after each update period.

The required bandwidth, C, to support all the existing connections with the specified CLR can be given [4, 14] by means of the expression:

$$C = \overline{M} + \xi\sigma \tag{6}$$

where $\overline{M} = \sum_j \overline{M^{(j)}}$, $\sigma^2 = \sum_j (\sigma^{(j)})^2$, the parameter $\xi \cong 1.8 - 0.46 \log_{10}(\eta)$, $\eta = \frac{\overline{M}\sqrt{2\pi}}{\sigma} * \epsilon_{tar}$, ϵ_{tar} is the target CLR.

Now, we introduced the *Fuzzy/Neural Connection Admission Controller* (FNCAC) for WATM networks that consists of three elements. The first of them, *Fuzzy Channel Alocator* (FCA) chooses six input linguistic variables for FCA (see Fig. 1): $\overline{v}$ - the mobile speed, P_H - the handoff probability,

$\overline{M^{(j)}}$ - the mean cell rate of jth call, $(\sigma^{(j)})^2$ - the variance of the cell rate of jth call, and two Quality of Service parameters $QoS_1^{(j)}$, $QoS_2^{(j)}$. In our approach vector $\overline{v} = (v_x, v_y)$ shows the direction of movement of the MT station. We can indicate the cell which will be visited by the MT station. The parameter P_H allows for the switching beetwen base stations (BSs) during the movement of the MT station in the cellular environment. We recall that the handoff probability P_H depends on the distance from the central point of each cell in which a BS station is located.

In our approach the first QoS parameter for jth call is the *Cell Loss Ratio* (CLR), which is usually very small for CBR and VBR traffic class (e.g. in the range of 10^{-7} and 10^{-10}) [McDY 98]. The second QoS parameter for call j is the *Cell Delay Variation* (CDV). This parameter changes for the CBR and VBR traffic classes in the range of 10^{-4} and 10^{-2} [ms].

Now, we introduce the term set of $\overline{v}$ as

$T(v_x) = \{Small, Large\} = \{Sm, La\}$,
$T(v_y) = \{Slow, Fast\} = \{Sl, Fa\}$,
$T(P_H) = \{Low, High\} = \{Lo, Hi\}$,
$T(\overline{M^{(j)}}) = \{Little, Big\} = \{Li, Bi\}$,
$T(\sigma^{(j)})^2) = \{Slightly, Substantial\} = \{Sli, Sub\}$.

The term sets for both QoS parameters, CLR and CDV, are respectively:

$T(CLR) = T(CDV) = \{More\ Enough, Slightly\ Enough, Not\ Enough\}$
$= \{ME, SL, NE\}$.

The membership functions for

$T(v_x)$, $T(v_y)$, $T(P_H)$, $T(\overline{M^{(j)}})$,
$T((\sigma^{(j)})^2)$, $T(CLR)$, $T(CDV)$

are defined as:

$M(v_x) = \{\mu_{Sm}, \mu_{La}\}$,
$M(v_y) = \{\mu_{Slo}, \mu_{Fas}\}$,
$M(P_H) = \{\mu_{Lo}, \mu_{Hi}\}$,
$M(\overline{M^{(j)}}) = \{\mu_{Li}, \mu_{Bi}\}$,
$M((\sigma^{(j)})^2) = \{\mu_{Sli}, \mu_{Sub}\}$,
$M(CLR) = \{\mu_{ME}^{(CLR)}, \mu_{SE}^{(CLR)}, \mu_{NE}^{(CLR)}\}$,
$M(CDV) = \{\mu_{ME}^{(CDV)}, \mu_{SE}^{(CDV)}, \mu_{NE}^{(CDV)}\}$,

where all the membership functions as constructed in the following way, for instance

$\mu_{Sm}(v_x) = g(v_x; v_{x,min}, v_{x,max}, 0, v_{x,bound})$, etc.

The function $g(.)$ is a trapezoidal function defined as

$g(x; x_0, x_1, a_0, a_1)$

$$g(x; x_0, x_1, a_0, a_1) = \begin{cases} \frac{x - x_0}{a_0} + 1, & \text{for } x_0 - a_0 < x \le x_0 \\ 1, & \text{for } x_0 < x \le x_1 \\ \frac{x_0 - x}{a_1} + 1, & \text{for } x_1 < x \le x_1 + a_1 \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

where x_0 is the left edge of the trapezoidal function, x_1 is the right edge of the trapezoidal function; a_0 (a_1) is the left (right) width of the trapezoidal function.

For the MTs that are in movement, the *Fuzzy Cell Selector* (FCS) decides to choose the BS in the same cell or overtake the call to the nearest BS. The term sets used to describe the handoff failure probability and the available resource are defined as

$T(P_H) = \{Few, Much\} = \{Fe, Mu\}$,

$T(A) = \{Less, Loose\} = \{Le, Lo\}$.

The term set of the output linguistic variable S (the same cell) is defined as

$T(S) = \{Reject, Weakly\ Accept, Accept\} = \{R, WA, A\}$.

Similarly, the term set of the output linguistic variable N (nearest BS) is defined as

$T(N) = \{BS, Nearest\ BS\} = \{BS, NBS\}$.

The NBS variable indicates the switching between the BS stations.

For the correction of the QoS parameters for all calls that are in use, we applied the *Neural Fuzzy CAC Mechanism* (NFCACM) and *QoS Parameter Controller* (QoSPC). The term sets used to describe the QoS change and eventually reject the call are defined as:

$T(QoS) = \{Wrong, Neutral, Good\} = \{Wr, Ne, Go\}$,

$T(R) = \{Non\text{-}Reject, Reject\} = \{NR, R\}$.

The term set of the output linguistic variables (bandwidth and call permission) are as follows:

$T(C) = \{To\ Small, Acceptable, To\ Large\} = \{TS, ACC, TL\}$

and

$T(CP) = \{Reject\ call, Admit\ call\} = \{RC, AC\}$.

Currently, we introduce the *Neural Fuzzy CAC Mechanism* (NFCACM).

3.1 Neural Fuzzy CAC Mechanism (NFCACM)

Our Neural Fuzzy CAC Mechanism (NFCACM) possesses a fuzzy inference system implemented within the framework of a the neural network. The idea of the design of a neuro-fuzzy system is to exploit the advantages of the fuzzy logic and a neural network's ability to learn.

The connectionist structure of the NFCACM is based on the four-layer Wang-Mendel neural fuzzy network [7, 8]. All the elements of the first layer realized membership functions A_i^k, $i = 1, \ldots, n$, $k = 1, \ldots, N$, where n is the number of controller inputs, N is the number of rules. To this layer are provided input data $\overline{x}_i$, and the membership function to fuzzy sets, $\mu_{A_i^k}(\overline{x}_i)$. We used the membership function given by Gauss function in the form

$$\mu_{A_i^k}(\overline{x}_i) = \exp[-(\frac{x_i - a_i^k}{b_i^k})^2] \tag{8}$$

where a_i^k and b_i^k are the centre and the width of Gauss function. All the parameters a_i^k, b_i^k are modified in the learning process. The number of elements in the first layer is equal to the number of all the fuzzy sets in the predecessing rules.

The second layer consists of the elements corresponding to the rules from the database of our NFCACM controller. According to the generalized modus ponens as the output of this inference layer we obtain the fuzzy sets $\overline{B}^k$, $k = 1, 2, \ldots, N$, with the membership function

$$\mu_{B'}(y) = \sup_{\mathbf{x} \in X} \{\mu_{A'}(\mathbf{x})^T_* \mu_{A^k \to B^k}(\mathbf{x}, y)\} \tag{9}$$

For the fuzziness implication $A^k \to B^k$ (Larsen implication), namely

$$\mu_{A^k \to B^k}(\mathbf{x}, y) = \mu_{A^k}(\mathbf{x}) \mu_{B^k}(y) \tag{10}$$

and the definition of the Cartesian product of fuzzy sets

$$\mu_{A_1 \times A_2 \times \ldots \times A_n(x_1, x_2, \ldots, x_n)} = \mu_{A_1}(x_1) \mu_{A_2}(x_2) \ldots \mu_{A_n}(x_n) \quad \text{for each } x_1 \in X_1, \ldots, x_n \in X_n \tag{11}$$

we obtain the membership function for the set $\overline{B}^k$, namely

$$\mu_{\overline{B}^k}(y) = \mu_{B^k}(y) \prod_{i=1}^{n} \mu_{A_i^k}(\overline{x}_i) \tag{12}$$

In the defuzzification block we used the center-average defuzzication method in which

$$\overline{y} = \frac{\sum_{k=1}^{N} \overline{y}^k \mu_{\overline{B}^k}(\overline{y}^k)}{\sum_{k=1}^{N} \mu_{\overline{B}^k}(\overline{y}^k)} \tag{13}$$

$\overline{y}^k$ is the point in which the membership function $\mu_{B^k}(y)$ of B^k set has the maximum value, namely

$$\mu_{B^k}(\overline{y}^k) = \max_y \{\mu_{B^k}(y)\} \tag{14}$$

By placing in Eq. (13) the dependence from Eq. (12) and by the use of condition (14) and assuming that the fuzzy sets B^k are normal, we obtain

$$\overline{y} = \frac{\sum_k^N \overline{y}^k \prod_{i=1}^n \mu_{A_i^k}(\overline{x}_i)}{\sum_{k=1}^N \prod_{i=1}^n \mu_{A_i^k}(\overline{x}_i)} \tag{15}$$

We can see that the whole structure of this fuzzy-neural system (see Fig. 2) reflects the above equation.

Now, we shall present the learning scheme for the proposed NFCACM controller. Our controller is learnt by means of a especially prepared two-phase learning algorithm. The first phase is devoted to the construction

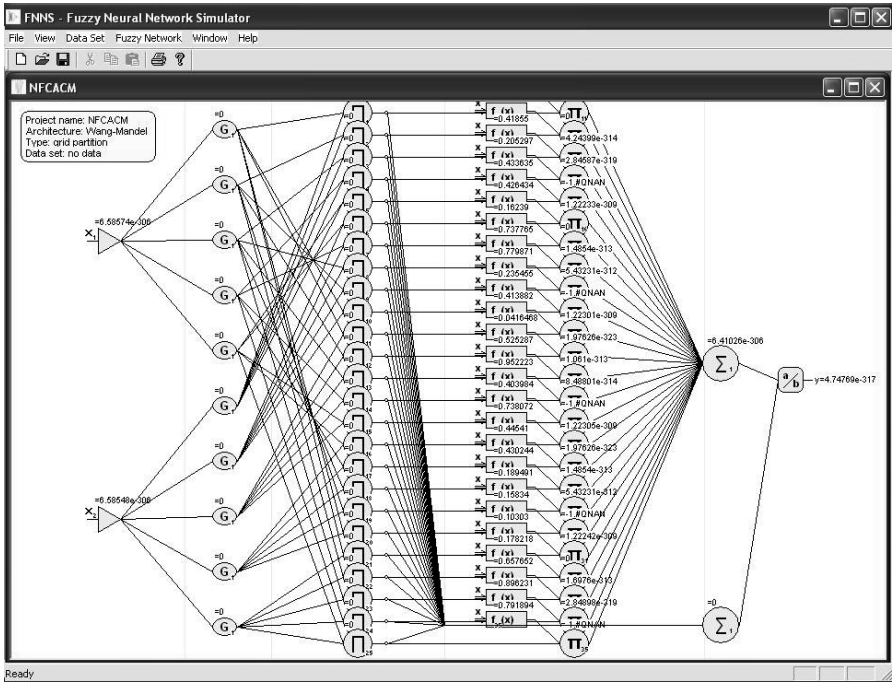


Fig. 2 The connectionist structure of NFCACM controller

of rules and to the location the initial membership functions. In the second phase (applied in the feedback loop), we used a reinforcement learning method to optimally adjust the membership functions for the needed outputs.

In phase one, we find the optimal value of parameters $\overline{x}_i^k$, σ_i^k, $\overline{y}^k$, σ^k which determine the centers and widths of the membership functions, respectively. The backpropagation scheme was used as the method of this supervised learning. The goal is to minimize the error function

$$E(t) = \frac{1}{2}(d(t) - \overline{y}(t))^2 \tag{16}$$

where $d(t)$ is the desired output, $\overline{y}(t)$ is the current output. For each training data set, starting at the input nodes, a forward pass is used to compute the activity levels of all the nodes in the network. Then, starting at the output nodes, a backward pass is used to compute $\frac{\partial E(t)}{\partial d(t)}$ for all the hidden nodes. Assuming that w is an adjustable parameter in a node (e.g. the center of a membership function), the general learning rule is

$$\Delta w(t) = \eta \frac{\partial E(t)}{\partial w(t)} = \eta(d(t) - \overline{y}(t)) \cdot \mathcal{C} \cdot y'(t) \tag{17}$$

where η is the learning rate, $\mathcal{C}$ is the predefined constant. The difference $d(t) - y(t)$ is substituted here by a reinforcement factor R. We define R

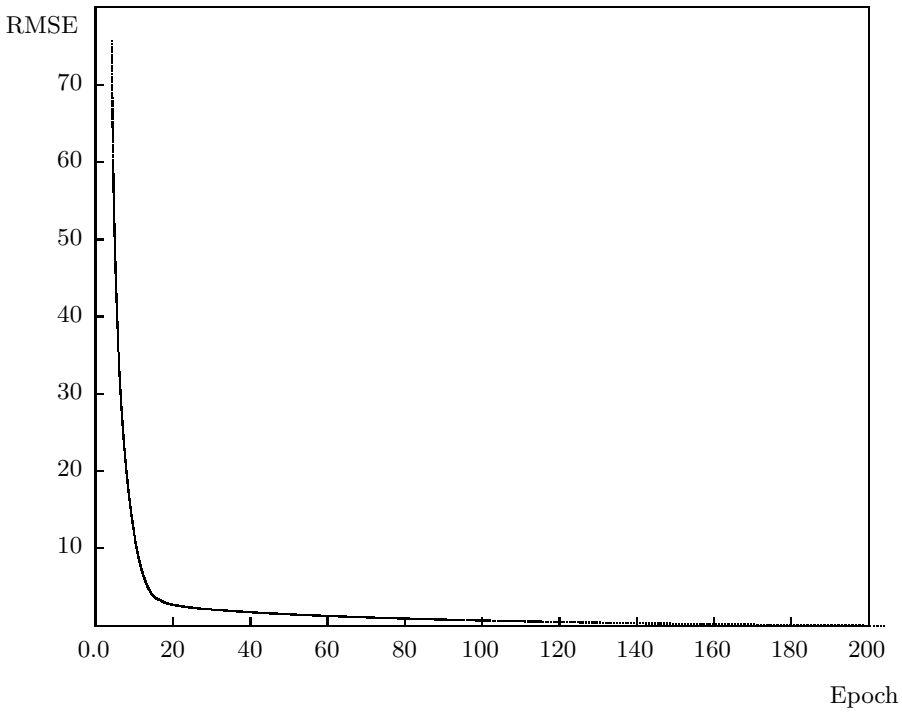


Fig. 3 Relative error of learning for NFCACM controller

as equal to $R = \widehat{R_C} - R_C$ ($R = \widehat{I} - I$) if $I_s \leq \widehat{I}$ ($I_s > \widehat{I}$), where $\widehat{R_C}$ is the required corruption ratio of the whole controller, $\widehat{R_C}$ is the required bandwidth corruption ratio of the whole NFCACM controller when the corruption ratio of bandwidth is equal 10^2 and R_C is the current bandwidth corruption ratio. The interference threshold value, I_s, is given as 10^{-3}. Thus,

$$\Delta w(t) = \begin{cases} \eta(\widehat{R_C} - R_C)\mathcal{C} \cdot \overline{y}(t) \text{ if } I_s \leq \widehat{I_s} \\ \eta(\widehat{I_s} - I_s)\mathcal{C} \cdot \overline{y}(t) \quad \text{if } I_s > \widehat{I_s} \end{cases} \tag{18}$$

Then, we can define weight $w(t+1)$ as equal to

$$w(t+1) = w(t) + \Delta w(t) \tag{19}$$

where t is a step in the backpropagation algorithm, $t = 0, 1, \ldots,$. For $t = 0$ all the parameters are given in a random way. Relative error of learning for our NFCACM controller is given in Fig. 3.

4 Simulation Results

In the simulation study, we constructed a cellular cluster with 7 cells. We assumed that a cell edge had 500 m. The probability of a handoff call was prepared by a matrix with given a priori values. Each element of this matrix,

$\mathbf{P}_H^{(i,j)}$, $(i \neq j)$, represented the probability of a handoff behaviour of call originated in cell i and directed to cell j. The cell with number zero was in the centre of our cellular system. This matrix is defined as follows

$$\mathbf{P}_H = \begin{bmatrix} P_H^{(0,0)} & P_H^{(0,1)} & \dots & P_H^{(0,6)} \\ P_H^{(1,0)} & P_H^{(1,1)} & \dots & P_H^{(1,6)} \\ \vdots & \vdots & \ddots & \vdots \\ P_H^{(6,0)} & P_H^{(6,1)} & \dots & P_H^{(6,6)} \end{bmatrix} = \begin{bmatrix} 0.0 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.0 & 0.1 & 0.0 & 0.0 & 0.0 & 0.1 \\ 0.1 & 0.1 & 0.0 & 0.1 & 0.0 & 0.0 & 0.0 \\ 0.1 & 0.0 & 0.1 & 0.0 & 0.1 & 0.0 & 0.0 \\ 0.1 & 0.0 & 0.0 & 0.1 & 0.0 & 0.1 & 0.0 \\ 0.1 & 0.0 & 0.0 & 0.0 & 0.1 & 0.0 & 0.1 \\ 0.1 & 0.1 & 0.0 & 0.0 & 0.0 & 0.1 & 0.0 \end{bmatrix}$$

The interarrival times of the MTs receiving a call in each cell are exponentially distributed with parameter λ_{new} [calls/s/cell]. Each call has a defined cell rate $R^{(j)}$ which depends on the class of the ATM traffic. The probability of a cell belonging to the traffic classes CBR, rt-VBR and others are 0.1, 0.2 and 0.7, respectively. The call holding times are assumed to be exponentially distributed with the average connection times: 10 min, 5 min, 3 min, for traffic classes CBR, rt-VBR and others, respectively. We also assumed that each MT travels towards a random virtual cell chosen from the 7 cells in the cellular cluster with equal probability. Its direction may change with the given probability 0.2 at every position update time interval Δt. The speeds are constants 10 $[\frac{km}{h}]$ in the first case (for low mobility users) and 100 $[\frac{km}{h}]$ in the second case (for high mobility users).

The rule structure for the NFCACM controller was arranged according to the experience and knowledge. The RA and RC parameters played a dominant role in the rule structure. Some rules was omitted. All initial values of the membership functions of term sets $\overline{v}$, P_H, $QoS^{(j)}$ parameters are chosen according to CBR and rt-VBR traffic classes and properly adjusted via learning algorithm. The obtained relative error of learning for our FNCACM controller is given in Fig. 3. As a defuzzification strategy the *center of area* method was used.

Four performance measures such as the mean number of the realized switchings, the mean number of rejected calls, the handoff rate, and the system utilization were studied. The mean number of realized switchings between the BS stations, $MS(t)$, is defined as

$$MS_N(t) = \frac{\sum_{i=1}^{N} A_i(t)}{\sum_{i=1}^{N} R_i(t) + \sum_{i=1}^{N} A_i(t)} \tag{20}$$

where $A_i(t)$ is the number of the accepted switchings between BSs, and $R_i(t)$ is the number of the rejected switchings between the adjacent BS stations for the cell i, $i = 1, \dots, N$ in the observed time. The mean number of rejected calls is given by

$$RC_N(t) = \frac{1}{N} \sum_{i=1}^{N} RC_i(t) \tag{21}$$

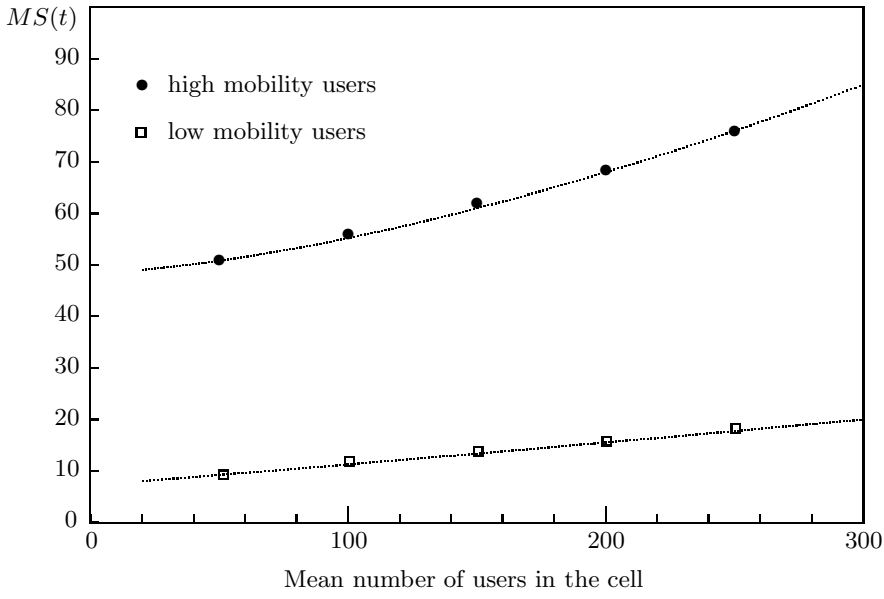


Fig. 4 Mean number of BS switchings between adjacent cells versus the mean number of users in the cell in the observed time

where $RC_i(t)$ is the mean number of rejected calls in cell i $(i = 1, \ldots, N)$ in the observed horizon time. The handoff rate is defined here as

$$HR_N(t) = \frac{\sum_{i=1}^{N} AC_i(t)}{\sum_{i=1}^{N} AC_i(t) + \sum_{i=1}^{N} RC_i(t)} \tag{22}$$

where $AC_i(t)$ is the mean number of accepted calls in cell i $(i = 1, \ldots, N)$ in the observed time. The mean system utilization is defined as

$$U(t) = \frac{\sum_{i=1}^{N} BC_i(t)}{\sum_{i=1}^{N} BC_i(t) + \sum_{i=1}^{N} CC_i(t)} \times 100\% \tag{23}$$

where $BC_i(t)$ is the average number of busy channels in cell i $(i = 1, \ldots, N)$, $CC_i(t)$ is the average number of empty channels in cell i in the observed time.

Fig. 4 shows the mean number of BS switchings between adjacent cells versus the mean number of users in the cell for the low and high mobility users in time t = 10^5 time units. It shows a much greater value of the mean number of switchings between BSs for the higher mobile users.

Fig. 5 shows the mean number of rejected calls versus the mean number of users in the cell for the low and high mobility users in the same time of observation. Two cases was studied: with the NFCACM controller and with the NFCACM without learning. It can be seen that the performance

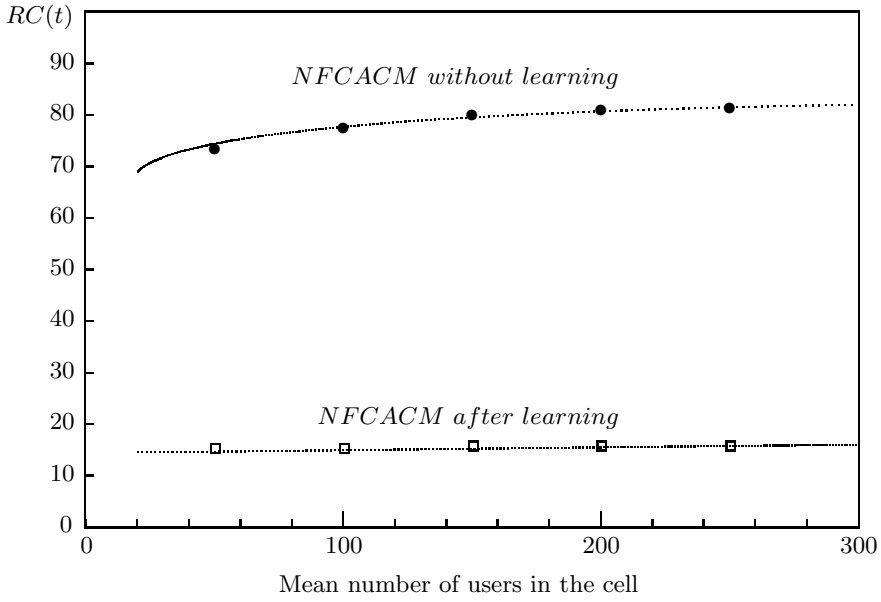


Fig. 5 Mean number of rejected calls for the low mobility users versus the mean number of users in the cell in the observed time

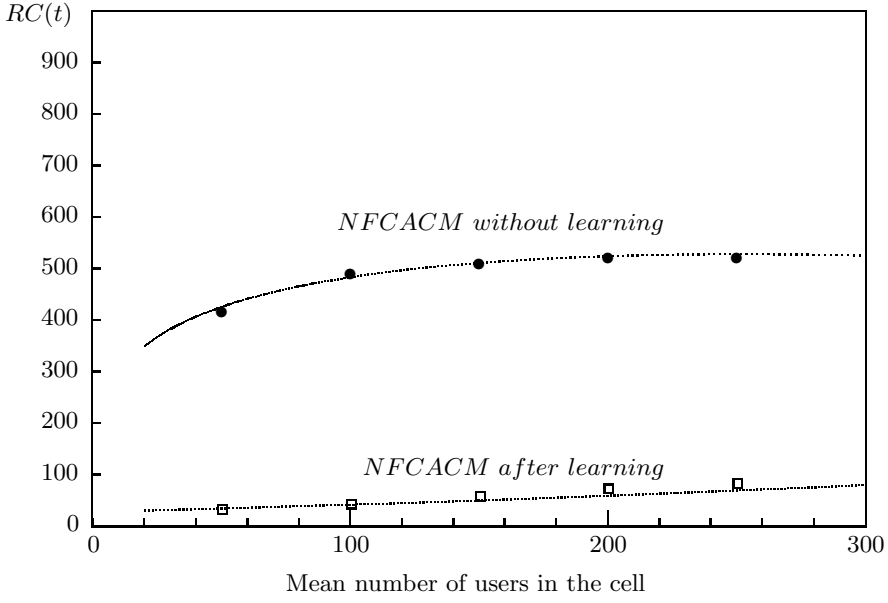


Fig. 6 Mean number of rejected calls for the high mobility users versus the mean number of users in the cell in the observed time

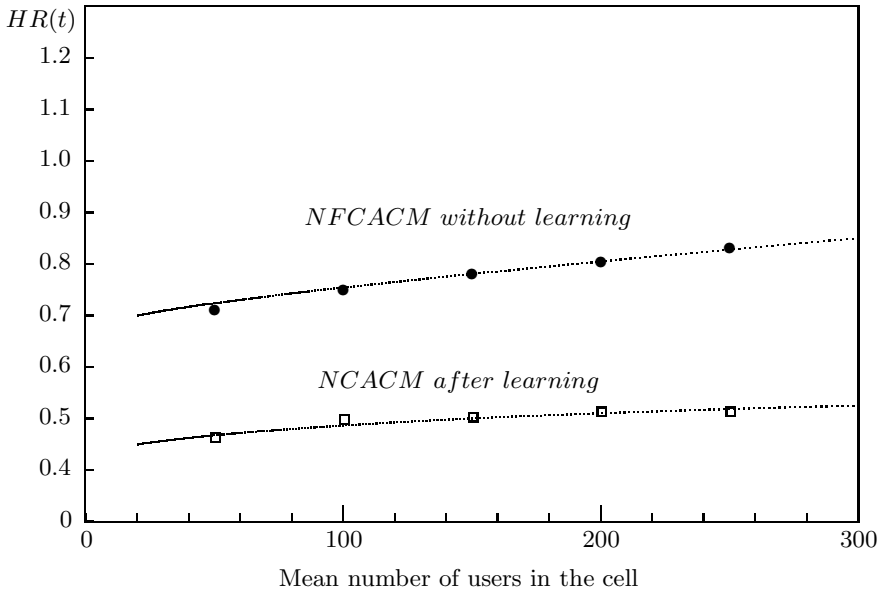


Fig. 7 Handoff rate for the low mobility users versus the mean number of users in the cell in the observed time

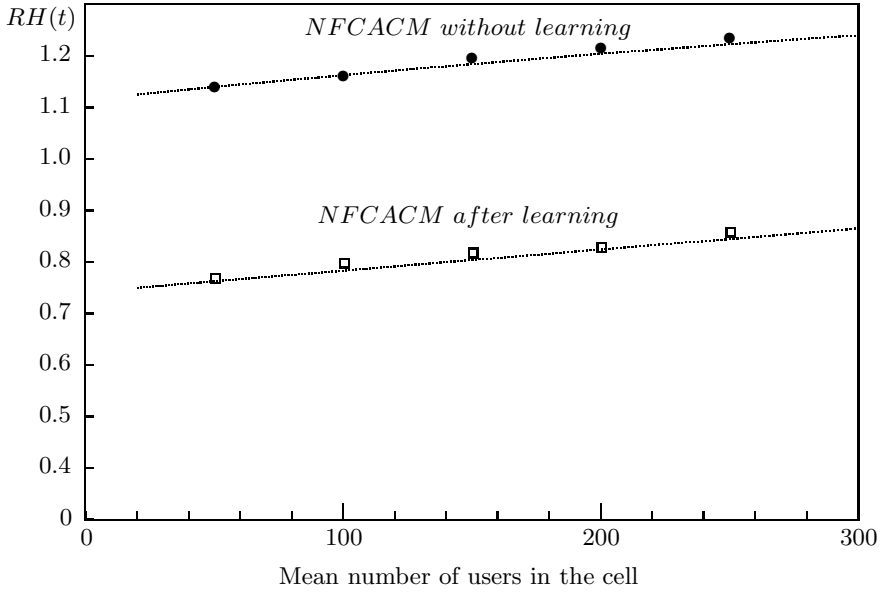


Fig. 8 Handoff rate for the high mobility users versus the mean number of users in the cell in the observed time

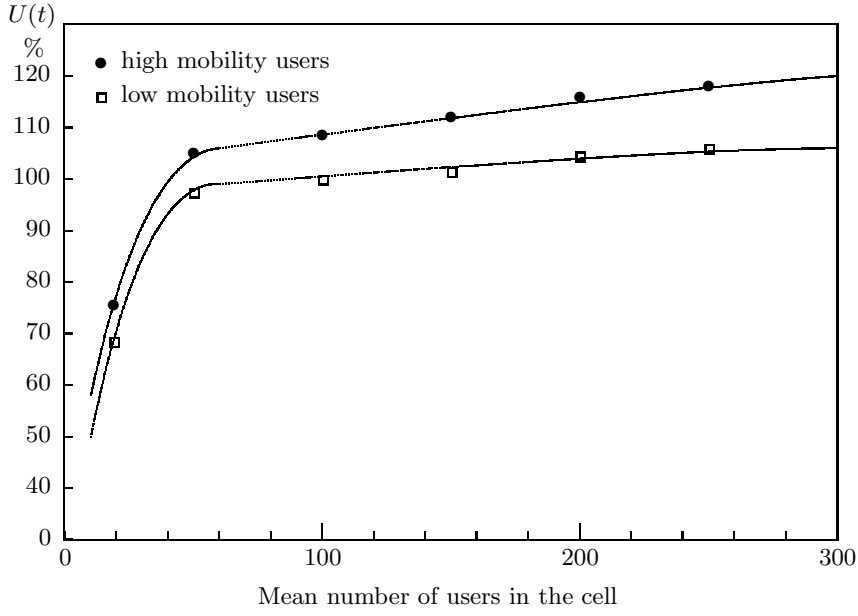


Fig. 9 Mean system utilization for the low and high mobility users versus the mean number of users in the cell in the observed time

of NFCACM controller with learning is significantly better than the performance measure of the system without the NFCACM controller. Fig. 6 shows the same dependence for the high mobility users. It has been found that the number of rejected calls is the smaller for the system with the NFCACM controller with learning than with the same controller without learning. Fig. 7 shows the handoff rate versus the number of users in the cell for low and high mobility users with the NFCACM controller after learning and without it, respectively. The Fig. 8 gives the same dependecies for the high mobility users. The system with the learned neural fuzzy controller has a smaller handoff rate than the conventional system (ca. 15% and 25% for low and high mobility users, respectively). Fig. 9 shows the mean system utilization versus the number of calls in the cell for the low and high mobility users, respectively. It is shown that both systems with the NFCACM controllers have similarly characteristics. It means that the system speed of mobile stations (MSs) makes no difference whatsoever for the performance of our controller.

5 Conclusions

We have developed a neuro-fuzzy connection admission controller for multimedia traffic in WATM networks. Our mechanism is based on the declared traffic parameters, on-line traffic statistics and movement parameters. With

the help of the simulation experiment, we have shown how to apply a neuro-fuzzy logic controller to measurement-based more robust CAC.

For further extension of our study, we will include the varying number of MTs stations in the cellular environment. Thus, we will observe the varying load of system. We will also include a very large speed of the MTs which allows for a better adjustment of all the FNCAC parameters.

References

[1] Ch.-J. Chang, B.-W. Chen, T.-Y. Liu, F.-Ch. Ren, *Fuzzy/Neural Congestion Control for Integrated Voice and Data DS-CDMA/FRMA Cellular Networks"*, IEEE Journal on Selected Areas in Communications, Vol. 18, No. 2, 2000, pp. 283 - 293.

[2] Ch. Douligeris, Y.-Ch. Liu, *"Fuzzy Rate Regulation with Feedback Controllers in ATM Networks"*, in: W. Pedrycz, A. Vasilakos (Eds.), *"Computational Intelligence in Telecommunications Networks"*, CRC Press, Boca Raton, 2001, pp. 177 - 196.

[3] A. Elwalid, D. Mitra, R. Wentworth, *"A New Approach for Allocating Buffers and Bandwidth to Heterogeneous Regulated Traffic in an ATM Node"*, IEEE Journal on Selected Areas in Communications, Vol. 13, No. 8, 1995, pp. 1115 - 1127.

[4] R. Guerin, H. Ahmadi, M. Nagshineh, *"Equivalent Capacity and Its Application in High Speed Network"*, IEEE Journal on Selected Areas in Communications, Vol. 9, 1991, pp. 968 - 981.

[5] A. Hać, *"Multimedia Applications Support for Wireless ATM Networks"*, Prentice-Hall, Inc., Upper Saddle River, 2000.

[6] A. Hać, *"Mobile Telecommunications Protocols for Data Networks"*, John Wiley and Sons, The Atrium, Southern Gate, Chichester, 2003.

[7] J.-S. R. Jang, C.-T. Sun, E. Mizutani, *"Neuro-Fuzzy and Soft Computing"*, Prentice-Hall, Englewood Cliffs, 1997.

[8] Y. Jin, *"Advanced Fuzzy Systems Design and Applications"*, Springer-Verlag, Berlin, Heidelberg, New York, 2003.

[9] Ch.-T. Lin, C.S.G. Lee, *"Neural-Network-Based Fuzzy Logic Control and Decision Systems"*, IEEE Trans. on Computers, Vol. 40, No. 12, 1991, pp. 1320 - 1336.

[10] D.E. McDysan, *"ATM: Theory and Application"*, McGraw-Hill, New York, 1998.

[11] H. Mitts, et al. *"Microcellular Handover for WATM Release 1.0: Proposal for Scope and Terms of Reference"*, ATM Forum 97-0338, April 1997.

[12] A. Pitsillides, Y. Sekercioglu, G. Ramamurthy, *"Effective Control of Traffic Flow in ATM Networks Using Fuzzy Explicit Rate Marking (FERM)"*, IEEE Journal on Selected Areas in Communications, Vol. 15, No. 2, 1997, pp. 209 - 225.

[13] D. Raychandhuri, *"Wireless ATM Networks: Technology Status and Future Directions"*, Proceedings of the IEEE, Vol. 87, No. 10, 1999, pp. 1790 - 1806.

[14] Q. Ren, G. Ramamurthy, *"A Real-Time Dynamic Connection Admission Controller Based on Traffic Modeling, Measurement, and Fuzzy Logic Control"*, IEEE Journal on Selected Areas in Communications, Vol. 18, No. 2, 2000, pp. 184 - 196.

[15] Special Issue on Wireless ATM, IEEE Journal on Selected Areas in Communications, January 1997.

[16] K. Uehara, K. Hirota, *"Fuzzy Connection Admission Control for ATM Networks Based on Possibility Distribution of Cell Los Ratio"*, IEEE Journal on Selected Areas in Communications, Vol. 15, No. 2, 1997, pp. 179 - 190.

[17] S. Youssef, I. Habib, T. Saadawi, *A Neurocomputing Controller for Bandwidth Allocation in ATM Networks"*, IEEE Journal on Selected Areas in Communications, Vol. 15, No. 2, 1997, pp. 191 - 199.

Fuzzy Mathematics

Limits of Functional Sequences in the Concept of Nearness Relations

Martin Kalina

Dept. of Mathematics, Slovak University of Technology, Radlinského 11
813 68 Bratislava, Slovakia

Abstract. This paper is a continuation of the paper [22]. It goes on investigating a kind of 'fuzzy limits' based on the notion of nearness relation. We will study limits of functional sequences.

1 Introduction

'Fuzzy analysis' and the concept of fuzzy continuity, fuzzy limits and similar topics, were already studied by many authors, to mention just few of them: M. Burgin, [3, 4], M. Burgin - M. Kalina [5], M. Burgin - A. Šostak [6], M. Demirci [8, 9], P. Diamond - P. Kloeden [10], D. Dubois - H. Prade [14, 15, 16], V. Janiš [17, 18, 19].

Limits in the concept of nearness relations were for the first time defined in [22]. That paper was devoted to limits of functions, having both their domain and range in the real line. It was shown there how the basic calculus works. A generalisation of nearness relations to Banach spaces was done in [25]. Further generalisation to linear spaces one can find in [23]. The aim of this paper is to go on developing 'fuzzy mathematics' by means of nearness relations and within this concept to define limits of functions having their domain in linear spaces and to show how the limits work for functional sequences.

1.1 Nearness Relation

X will denote a fixed linear space.

Definition 1 *We say that $\mathcal{N} : X \times X \to [0;1]$ is a nearness relation if and only if the following hold*

1. *$\mathcal{N}(x,x) = 1$ for any $x \in X$*
2. *$\mathcal{N}(x,y) = \mathcal{N}(y,x)$ for all $x, y \in X$*
3. *for all $x, y \in X$, $y \neq 0$, and all $t_1, t_2, t_3, t_4 \in R$ such that $t_1 \leq t_2 \leq t_3 \leq t_4$ the following holds*

$$\mathcal{N}(x + t_1 y, x + t_4 y) \leq \mathcal{N}(x + t_2 y, x + t_3 y)$$

M. Kalina: *Limits of Functional Sequences in the Concept of Nearness Relations*, Advances in Soft Computing **2**, 669–676 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

4. for all $x, y \in X$, $y \neq 0$, the following holds

$$\lim_{t\to\infty} \mathcal{N}(x, x+ty) = 0$$

In the whole paper we will have the following additional assumption:

5. For each nearness relation $\mathcal{N}$ and each x there is an α_0 such that

$$(\forall \alpha > \alpha_0)(\exists x_\alpha)\,(\mathcal{N}(x, x_\alpha) = \alpha)\,.$$

The concept of T-equivalence (fuzzy equality) is well-known, first introduced by L. Zadeh in [27]. Relevant results can be found also in [1, 2, 26]. In fact, nearness relation (introduced in [21]) is its generalisation. More on nearness relations, their topological properties and relationship to T-equivalences one can find in [11, 12, 13, 24, 25]. Recently, the paper [7] appeared, introducing a notion of *resemblance*. The relationship of this notion to the above defined nearness relation is given in [20].

We give some examples of nearness relations.

Example 1 a. Assume that the linear space X is equipped with a metric ρ. Then for any $x, y \in X$

$$\mathcal{N}_1(x, y) = \max\{0, 1 - \rho(x, y)\}$$
$$\mathcal{N}_2(x, y) = \mathrm{e}^{-\rho(x,y)}$$
$$\mathcal{N}_3(x, y) = \begin{cases} 1, & \text{if } \rho(x, y) \leq 1 \\ \max\{0, 2 - \rho(x, y)\} & \text{otherwise} \end{cases}$$

are nearness relations.

b. Consider the following one-dimensional nearness relation

$$\mathcal{N}(x, y) = \max\{0, 1 - |x - y|\}$$

For any sequence of positive weights $\{w_j\}_j$ with $\sum_{j=1} w_j = 1$ and any elements $\{x_j\}_j, \{y_j\}_j \in X$ denote

$$\mathcal{H}(\{x_j\}_j, \{y_j\}_j) = \frac{1}{\sum_j \frac{w_j}{\mathcal{N}_1(x_j, y_j)}}$$

Then $\mathcal{H}$ is a nearness relation (cf. [24]). □

In the following $\mathcal{D} : X \times X \to [0, 1]$ and $\mathcal{H} : R \times R \to [0, 1]$ will always denote nearness relations connected with the domain (the linear space X) and the range of considered functions, respectively.

For nearness relations $\mathcal{H}$ we will assume

($\mathcal{H}$ 1.) $\mathcal{H}(x, z) = \mathcal{H}(x + t, z + t)$ for each $x, z, t \in R$
($\mathcal{H}$ 2.) for each x and each $\alpha \in]0, 1[$ $\{z; \mathcal{H}(x, z) \geq \alpha\}$ will be always a closed interval.
($\mathcal{H}$ 3.) $\mathcal{H}(x, y) = 1 \;\Leftrightarrow\; x = y$.

2 Limits in the Context of Nearness Relation

2.1 Basic Definitions

Definition 2 *We say that $f : X \to R$ has an $(\mathcal{D}, \mathcal{H}, \alpha)$-limit at x if and only if for any couple $x_1, x_2 \in X$ such that $\mathcal{D}(x, x_1) = \mathcal{D}(x, x_2) = \alpha$ the following holds*

$$\mathcal{H}\left(f(x_1), f(x_2)\right) \ \geq \ \alpha.$$

The corresponding limit will be denoted by ${}^{(\mathcal{D},\mathcal{H})_\alpha}\lim_{z\to x} f(z)$ and defined by

$${}^{(\mathcal{D},\mathcal{H})_\alpha}\lim_{z\to x} f(z) = \left[\inf\{f(z); \mathcal{D}(z,x) = \alpha\}; \sup\{f(z); \mathcal{D}(z,x) = \alpha\}\right].$$

Roughly speaking, a function f has an $(\mathcal{D}, \mathcal{H}, \alpha)$-limit at x, if the values of f at points, which are near to x at the level α, are 'near enough to each other'. In fact, the value of the $(\mathcal{D}, \mathcal{H}, \alpha)$-limit (if it exists) does not depend on the choice of the nearness relation $\mathcal{H}$.

$f(x)_{-\alpha}$ and $f(x)_\alpha$ will always denote the left and right endpoints, respectively, of the $(\mathcal{D}, \mathcal{H}, \alpha)$-limit of f at x.

By Definition 2 we get the following lemma:

Lemma 1 *Let $f : X \to R$ be any function and $\mathcal{D} : X \times X \to [0,1], \mathcal{H}_1 : R \times R \to [0,1], \mathcal{H}_2 : R \times R \to [0,1]$ some nearness relations such that for all $s, t \in R$*

$$\mathcal{H}_1(s,t) \leq \mathcal{H}_2(s,t).$$

Then the existence of the $(\mathcal{D}, \mathcal{H}_1, \alpha)$-limit of f at some point $x \in X$ implies the existence of $(\mathcal{D}, \mathcal{H}_2, \alpha)$-limit of f at $x \in X$ and

$${}^{(\mathcal{D},\mathcal{H}_1)_\alpha}\lim_{z\to x} f(z) = {}^{(\mathcal{D},\mathcal{H}_2)_\alpha}\lim_{z\to x} f(z)$$

Definition 3 *We say that f has a $\mathcal{D}$-limit at x if and only if there exists a nearness relation $\mathcal{H} : R \times R \to [0,1]$ and an α_0 such that, for all $\alpha > \alpha_0$, f has the $(\mathcal{D}, \mathcal{H}, \alpha)$-limit at x. The corresponding limit will be defined by*

$${}^{\mathcal{D}}\lim_{z\to x} f(z) = \left[\liminf_{\alpha\nearrow 1} f(x)_{-\alpha}; \limsup_{\alpha\nearrow 1} f(x)_\alpha\right].$$

I.e. fuzzy limit is an interval (possibly one-point). For the completness-sake we present the folowing theorem concerning the basic calculus. For the proof see [22]. The multiplication and addition of intervals, is defined by

$$[a_1; b_1] * [a_2; b_2] = \tag{1}$$
$$[\min\{a_1 * a_2; a_1 * b_2; b_1 * a_2; b_1 * b_2\}; \max\{a_1 * a_2; a_1 * b_2; b_1 * a_2; b_1 * b_2\}]$$

where $*$ means the multiplication and addition, respectively.

Theorem 1 *Let $f : X \to R$ and $g : X \to R$ be any functions having their $\mathcal{D}$-limits at a point $x_0 \in X$ such that*

$$\pm\infty \notin {}^{\mathcal{D}}\lim_{z\to x_0} f(z), \quad \pm\infty \notin {}^{\mathcal{D}}\lim_{z\to x_0} g(z).$$

Then also fg, $f+g$ and kf have fuzzy limits at x_0, where $k \in R$ and there holds

a. ${}^{\mathcal{D}}\lim_{z\to x_0}(f(z)g(z)) \subseteq \left({}^{\mathcal{D}}\lim_{z\to x_0} f(z)\right)\left({}^{\mathcal{D}}\lim_{z\to x_0} g(z)\right)$
b. ${}^{\mathcal{D}}\lim_{z\to x_0}(f(z)+g(z)) \subseteq {}^{\mathcal{D}}\lim_{z\to x_0} f(z) + {}^{\mathcal{D}}\lim_{z\to x_0} g(z)$
c. ${}^{\mathcal{D}}\lim_{z\to x_0} kf(z) = k\,{}^{\mathcal{D}}\lim_{z\to x_0} f(z)$.

2.2 Main Results

We will focus our consideration to the limits of functional sequences.

Theorem 2 *Let $f_n : X \to R$ be a sequence of functions pointwise convergent to $f : X \to R$. Let for an $x \in X$, an $\alpha < 1$ and a couple of nearness relation $(\mathcal{D}, \mathcal{H})$, there exist $(\mathcal{D}, \mathcal{H}, \alpha)$-limits of f_n at x for each n. Then there exists a $(\mathcal{D}, \mathcal{H}, \alpha)$-limit of f at x and there holds*

$${}^{(\mathcal{D},\mathcal{H})_\alpha}\lim_{z\to x} f(z) = \left[\lim_{n\to\infty} f_n(x)_{-\alpha}, \lim_{n\to\infty} f_n(x)_\alpha\right].$$

Proof. Fix some point x. Denote

$$a = \inf\{z \in X; z < x \ \&\ \mathcal{D}(z,x) \geq \alpha\} \quad b = \sup\{z \in X; z > x \ \&\ \mathcal{D}(z,x) \geq \alpha\}$$

From the pointwise convergence of $\{f_n\}$ to f we get $f(a) = \lim_{n\to\infty} f_n(a)$ and $f(b) = \lim_{n\to\infty} f_n(b)$. Since $f(x)_{-\alpha} = \min\{f(a), f(b)\}, f(x)_\alpha = \max\{f(a), f(b)\}$, from assumption ($\mathcal{H}$ 2.) we get the assertion in question. □

Theorem 3 *Let $f_n : X \to R$ be a sequence of functions pointwise convergent to $f : X \to R$. Let for an $x \in X$ there exist ${}^{\mathcal{D}}\lim_{z\to x} f_n(z)$ for each n. Assume that there exists an $\alpha_0 < 1$ and a nearness relation $\mathcal{H}$ such that for $\alpha > \alpha_0$ there exist $(\mathcal{D}, \mathcal{H}, \alpha)$-limits of f_n at x for each n. Then there exists a $\mathcal{D}$-limit of f at x.*

Proof. Theorem 2 implies that for each $\alpha > \alpha_0$ there exists the $(\mathcal{D}, \mathcal{H})_\alpha$-limit of f at the point x. By Definition 3 we get the existence of the $\mathcal{D}$-limit of f at x. □

Example 2 Let $X = R^2$, equipped with the usual Euclidean metric ρ. Consider the following sequence of functions:

$$f_n(x,y) = \begin{cases} \cos\left(\frac{1}{\sqrt{x^2+y^2}}\right) & \text{if } \rho\left((x,y);(0,0)\right) > \frac{1}{n} \\ 0 & \text{otherwise} \end{cases}$$

$$f(x,y) = \begin{cases} \cos\left(\frac{1}{\sqrt{x^2+y^2}}\right) & \text{if } (x,y) \neq (0,0) \\ 0 & \text{otherwise} \end{cases}$$

Then the sequence $\{f_n\}_n$ converges pointwise to f. Put $\mathcal{H}$ the one-dimensional nearness relation $\mathcal{N}$ from Example 1b.

a. Let $\mathcal{D}$ be the nearness relation $\mathcal{N}_1$ from Example 1a. Then for each n we get

$$0 = {}^{\mathcal{D}}\lim_{(x,y)\to(0,0)} f_n(x,y) \neq {}^{\mathcal{D}}\lim_{(x,y)\to(0,0)} f(x,y) = [-1,1]$$

however

$${}^{(\mathcal{D},\mathcal{H})_\alpha}\lim_{(x,y)\to(0,0)} f(x,y) = \lim_{n\to\infty}\left({}^{(\mathcal{D},\mathcal{H})_\alpha}\lim_{(x,y)\to(0,0)} f_n(x,y)\right) = \cos\left(\frac{1}{1-\alpha}\right)$$

since $\mathcal{N}_1((x,y);(0,0)) = \alpha$ if and only if $\sqrt{x^2+y^2} = 1-\alpha$ (for $\alpha < 1$).

b. Let $\mathcal{D} = \mathcal{N}_3$ from Example 1a. Then we get the following results

$${}^{(\mathcal{D},\mathcal{H})_\alpha}\lim_{(x,y)\to(0,0)} f(x,y) = \lim_{n\to\infty}\left({}^{(\mathcal{D},\mathcal{H})_\alpha}\lim_{(x,y)\to(0,0)} f_n(x,y)\right) = \cos\left(\frac{1}{2-\alpha}\right)$$

since $\mathcal{N}_3((x,y);(0,0)) = \alpha$ if and only if $\sqrt{x^2+y^2} = 2-\alpha$ and

$$\lim_{n\to\infty}\left({}^{\mathcal{D}}\lim_{(x,y)\to(0,0)} f_n(x,y)\right) = {}^{\mathcal{D}}\lim_{(x,y)\to(0,0)} f(x,y) = \cos 1$$

□

Definition 4 *Functions f and g are called α-comonotone at x with respect to a nearness relation $\mathcal{D}$ if and only if the following holds*

$$(\mathcal{D}(z_1,x) = \alpha \;\&\; \mathcal{D}(z_2,x) = \alpha) \quad \Rightarrow \quad (f(z_1) - f(z_2))\,(g(z_1) - g(z_2)) \geq 0.$$

Lemma 2 *Let $\mathcal{D}$ be a nearness relation, $x \in X$ some fixed point, $\alpha_0 \in]0,1[$, and f and g be any functions such that they are α-comonotone at x for any $\alpha > \alpha_0$ Then for a one-dimensional nearness relation $\mathcal{H}$ such that there exist ${}^{(\mathcal{D},\mathcal{H})_\alpha}\lim_{z\to x} f(z)$, ${}^{(\mathcal{D},\mathcal{H})_\alpha}\lim_{z\to x} g(z)$ the following holds:*

- *There exists some one-dimensional nearnes relation $\mathcal{H}_1$ such that*

$${}^{(\mathcal{D},\mathcal{H})_\alpha}\lim_{z\to x} f(z) + {}^{(\mathcal{D},\mathcal{H})_\alpha}\lim_{z\to x} g(z) = {}^{(\mathcal{D},\mathcal{H}_1)_\alpha}\lim_{z\to x}(f(z)+g(z))$$

- $${}^{\mathcal{D}}\lim_{z\to x} f(z) + {}^{\mathcal{D}}\lim_{z\to x} g(z) = {}^{\mathcal{D}}\lim_{z\to x}(f(z)+g(z))$$

Proof. First, we construct the one-dimensional nearness relation $\mathcal{H}_1$. We put

$$\mathcal{H}_1(0,z) = \sup\{\alpha \in [0,1]; (\exists x,y, x+y=z)(\mathcal{H}(x,0) \geq \alpha \;\&\; \mathcal{H}(y,0) \geq \alpha)\}$$
$$\mathcal{H}_1(s,t) = \mathcal{H}_1(0,t-s)$$

The rest of the proof is just an easy excercise with the property of α-comonotonicity of functions f and g, since

$${}^{(\mathcal{D},\mathcal{H}_1)_\alpha}\lim_{z\to x}(f(z)+g(z)) =$$
$$[\inf\{f(z)+g(z); \mathcal{D}(z,x)=\alpha\}, \sup\{f(z)+g(z); \mathcal{D}(z,x)=\alpha\}]$$

The α-comonotonicity of the functions f and g implies

$$\inf\{f(z)+g(z); \mathcal{D}(z,x)=\alpha\} = \inf\{f(z); \mathcal{D}(z,x)=\alpha\} + \inf\{g(z); \mathcal{D}(z,x)=\alpha\}$$

□

Theorem 4 *Let $\{f_n\}_n$ be a sequence of pairwise α-comonotone functions at some point x with respect to a nearness relation $\mathcal{D}$. Denote $F_n = \sum_{i=1}^{n} f_n$. Assume that F_n pointwise converge to F. Let the sequence $\{F_n\}_n$ and the function F fulfil the assumptions of Theorem 2 at the point x for the given α, given nearness relation $\mathcal{D}$ and some nearness relation $\mathcal{H}$. Then*

$$^{(\mathcal{D},\mathcal{H})_\alpha}\lim_{z\to x} F(z) = \sum_{n=1}^{\infty} {}^{(\mathcal{D},\mathcal{H})_\alpha}\lim_{z\to x} f_n(z) \tag{2}$$

provided the $^{(\mathcal{D},\mathcal{H})_\alpha}\lim_{z\to x} f_n(z)$ exist for all n.

Proof. The assertion of this theorem is in fact implied by Theorem 2 and Lemma 2. Really, Lemma 2 implies that there exist $^{(\mathcal{D},\mathcal{H}_1)_\alpha}\lim_{z\to x} F_n(z)$ for all n and some one-dimensional nearness relation $\mathcal{H}_1$. Under the assumptions of this theorem $\mathcal{H}_1 = \mathcal{H}$. The existence of the $(\mathcal{D},\mathcal{H},\alpha)$-limit of F at x and formula (2) is implied by Theorem 2. □

Lemma 2 and Theorem 4 can be generalized into the following:

Theorem 5 *Assume that $\{f_n\}_{n=1}^{\infty}$ is a sequence of functions which are pairwise comonotone at some fixed point $x_0 \in X$. Denote $F_n(x) = \sum_{i=1}^{n} f_i(x)$ and $F(x) = \sum_{i=1}^{\infty} f_i(x)$. Assume that the function F is well defined by the sum. Let us fix some nearness relation $\mathcal{D} : X \times X \to [0,1]$ and a sequence of nearness relations $\mathcal{H}_n : R \times R \to [0,1]$ such that for the fixed $x_0 \in X$ and some $\alpha \in]0,1[$ there exist $(\mathcal{D},\mathcal{H}_n,\alpha)$-limits of f_n at the point x_0. Then there exist some nearness relations $\tilde{\mathcal{H}}_n : R \times R \to [0,1]$ such that the $(\mathcal{D},\tilde{\mathcal{H}}_n,\alpha)$-limits of F_n at the point x_0 do exist and the following holds*

$$^{(\mathcal{D},\tilde{\mathcal{H}}_n)_\alpha}\lim_{z\to x_0} F_n(z) = \sum_{i=1}^{n} \left({}^{(\mathcal{D},\mathcal{H}_i)_\alpha}\lim_{z\to x_0} f_i(z) \right). \tag{3}$$

If moreover

$$\mathcal{H}(x,y) = \sup_n \left(\tilde{\mathcal{H}}_n(x,y)\right) \tag{4}$$

is a nearness relation fulfilling the properties $\mathcal{H}$1-3, then the $(\mathcal{D},\mathcal{H},\alpha)$-limit of F at the point x_0 does exist and the following holds

$$^{(\mathcal{D},\mathcal{H})_\alpha}\lim_{z\to x_0} F(z) = \sum_{i=1}^{\infty} {}^{(\mathcal{D},\mathcal{H}_i)_\alpha}\lim_{z\to x_0} f_i(z) \tag{5}$$

Proof. To construct the nearness relations $\tilde{\mathcal{H}}_n$, it is enough to put

$$\tilde{\mathcal{H}}_n(0,z) = \sup\left\{ \alpha \in [0,1]; \left(\exists \{x_i\}_{i=1}^{n}, \sum_{i=1}^{n} x_i = z \right) (\forall i \le n) \left(\mathcal{H}(x_i,0) \ge \alpha\right) \right\}$$

$$\tilde{\mathcal{H}}_n(s,t) = \tilde{\mathcal{H}}_n(0,t-s)$$

Formula (3) is implied by the comonotonicity of the functions f_i and by Lemmas 1 and 2.

If Formula (4) holds, then due to Lemma 1

$$^{(\mathcal{D},\tilde{\mathcal{H}}_n)_\alpha}\lim_{z\to x_0} F_n(z) = {}^{(\mathcal{D},\mathcal{H})_\alpha}\lim_{z\to x_0} F_n(z).$$

This and Theorem 2 imply Formula (5). □

A direct consequence of Theorem 4 is the following:

Theorem 6 *Assume that there exists an $\alpha_0 < 1$ such that, for all $\alpha > \alpha_0$, $\{f_n\}_n$ is a sequence of pairwise α-comonotone functions at some point x, with respect to a nearness relation $\mathcal{D}$. Denote $F_n = \sum_{i=1}^{n} f_n$. Assume that F_n pointwise converge to F. Let the sequence $\{F_n\}_n$ and the function F fulfil the assumptions of Theorem 3 at the point x for the given nearness relation $\mathcal{D}$ and some nearness relation $\mathcal{H}$. Then there exists $^{\mathcal{D}}\lim_{z\to x} F(z)$.*

In general, nothing can be said on the relationship between the sum of $\mathcal{D}$-limits of a sequence of functions $\{f_n\}_{n=1}^{\infty}$ and the $\mathcal{D}$-limit of the sum of $\{f_n\}_{n=1}^{\infty}$. This is illustrated in the following example:

Example 3 Take for simplicity sequences of real functions $\{f_n\}_n$ and $\{g_n\}_n$ defined by

$$f_n(x) = \begin{cases} |x|, & \text{iff } x \in \left[-\frac{1}{n}, \frac{1}{n}\right] \\ 0, & \text{otherwise} \end{cases}, \quad g_n(x) = \begin{cases} \frac{1}{2^n}\left|\sin\frac{1}{x}\right|, & \text{iff } x \neq 0 \\ 0, & \text{otherwise} \end{cases}$$

and put $\mathcal{D} = \mathcal{N}$, the one-dimensional nearness relation from Example 1b. Then the following hold

$$^{\mathcal{D}}\lim_{x\to 0} f_n(x) = 0, \qquad {}^{\mathcal{D}}\lim_{x\to 0} g_n(x) = \left[0, \frac{1}{2^n}\right]$$

$$^{\mathcal{D}}\lim_{x\to 0}\left(\sum_{n=1}^{\infty} f_n(x)\right) = \left[\tfrac{1}{2}, 1\right] \neq \sum_{n=1}^{\infty}\left({}^{\mathcal{D}}\lim_{x\to 0} f_n(x)\right) = 0$$

$$^{\mathcal{D}}\lim_{x\to 0}\left(\sum_{n=1}^{\infty} g_n(x)\right) = [0,1] = \sum_{n=1}^{\infty}\left({}^{\mathcal{D}}\lim_{x\to 0} f_n(x)\right)$$

□

Acknowledgement. This work was supported by Science and Technology Assistance Agency under the contract No. APVT-20-023402 and by VEGA grants 1/0273/03, 1/0085/03.

References

1. De Baets, B., Mesiar, R., (1997) Pseudometrics and $\mathcal{T}$-equivalences, Jour. Fuzzy Mathematics 5: 471-481.
2. De Baets, B., Mesiar, R., (2002) Metrics and $\mathcal{T}$-equalities, J. Math. Anal. Appl. 267: 531-547.

3. Burgin, M., (1999) General approach to continuity measures, Fuzzy sets and Systems 105: 225-231.
4. Burgin, M., (2000) Theory of fuzzy limits, Fuzzy Sets and Systems 115: 133-443.
5. Burgin, M., Kalina, M., Fuzzy conditional convergence and nearness relation, Fuzzy Sets and Systems, in press.
6. Burgin, M., Šostak, A., (1994) Fuzzification of the theory of continuous functions, Fuzzy Sets and Systems 62: 71-81.
7. De Cock, M., Kerre, E., (2003) On (un)suitable fuzzy relations to model approximite equality, Fuzzy Sets and Systems 133: 137-153.
8. Demirci, M., (1999) Fuzzy functions and their fundamental properties, Fuzzy Sets and Systems 106: 239-246.
9. Demirci, M., (2001) Gradation of being fuzzy function, Fuzzy Sets and Systems 119: 386-392.
10. Diamond, P., Kloeden, P., (1994) Metric Spaces of Fuzzy Sets, Theory and Applications, World Scientific, Singapore, New Jersey, London, Hong Kong.
11. Dobrakovová, J., (1999) Nearness, convergence and topology, Busefal 80: 17-23.
12. Dobrakovová, J., (2000) Indistinguishability and nearness, Busefal 83: 46-52.
13. Dobrakovová, J., (2001) Nearness based topology, Tatra Mount. Math. Publ. 21: 163-170.
14. Dubois, D., Prade, H., (1982) Towards fuzzy differential calculus, part 1; Integration of fuzzy mappings, Fuzzy Sets and Systems 8: 1-17.
15. Dubois, D., Prade, H., (1982) Towards fuzzy differential calculus, part 2; Integration on fuzzy intervals, Fuzzy Sets and Systems 8: 105-116.
16. Dubois, D., Prade, H., (1982) Towards fuzzy differential calculus, part 3; Differentiation, Fuzzy Sets and Systems 8: 225-233.
17. Janiš, V., (1997) Fixed points of fuzzy functions, Tatra Mountains Math. Publ. 12: 13-19.
18. Janiš, V., (1998) Fuzzy uniformly continuous functions, Tatra Mountains Math. Publ. 14: 177–180.
19. Janiš, V., (1998) Fuzzy mappings and fuzzy methods for crisp mappings, Acta Univ. M. Belii 6: 31-47.
20. Janiš, V., (2003) Resemblance is a nearness, Fuzzy Sets and Systems 133: 171-173.
21. Kalina, M., (1997) Derivatives of fuzzy functions and fuzzy derivatives, Tatra Mount. Math. Publ. 12: 27-34.
22. Kalina, M., (2001) Fuzzy limits and fuzzy nearness relation, in: B. Reusch (Ed.): 7-th Fuzzy Days, LNCS 2206, Springer-Verlag, Berlin, Heidelberg: 755-761.
23. Kalina, M., (2003) Coordinatewise aggregation of nearness relations, in: Proc. AGOP'2003, Univ. de Alcalá: 89-96.
24. Kalina, M.: Nearness relations in linear spaces, Kybernetika, in press.
25. Kalina, M., Dobrakovová, J., (2002) Relation of fuzzy nearness in Banach spaces, in: Proc. East West Fuzzy Colloquium, HS Zittau/Görlitz: 26-32.
26. Klement, E.P., Mesiar, R., Pap, E., (2000) Triangular norms, Kluwer Acad. Publ., Dordrecht.
27. Zadeh, L.A., (1971) Similarity relations and fuzzy orderings, Information Sciences 3: 177-200.

On the Law of Large Numbers on IFS Events

Beloslav Riečan

Faculty of Natural Science, Matej Bel University, Department of Mathematics, Tajovského 40, 974 01 Banská Bystrica, and Math. Institut Slovak Academy of Sciences, Štefánikova 49 Bratislava, SLOVAKIA.
riecan@fpv.umb.sk
riecan@mat.savba.sk

Abstract. A probability theory on IFS events has been constructed in [2], and axiomatically in [4]. Here we use a general system of axioms for probabilities and observables. Using analogous results of probability theory on MV-algebras (see [6]) a version of the weak law of large numbers can be proved.

Keywords: IFS-events, MV-algebras, probability, observables

1 Introduction

We assume that a Lukasiewicz tribe is given, i.e. a non-empty family $\mathcal{T}$ of functions $f : \Omega \to [0,1]$ satisfying the following conditions:

$$(i) f \in \mathcal{T} \Longrightarrow 1 - f \in \mathcal{T};$$
$$(ii) f, g \in \mathcal{T} \Longrightarrow f \oplus g = min(f + g, 1) \in \mathcal{T};$$
$$(iii) f_n \in \mathcal{T} (n = 1, 2, ...), f_n \nearrow f \Longrightarrow f \in \mathcal{T}.$$

Evidently, if $f, g \in \mathcal{T}$, then also

$$f \odot g = max(f + g - 1, 0) \in \mathcal{T}.$$

By an IFS-event ([2], [3], [4], [5])we understand any element of the family

$$\mathcal{F} = \{(f, g); f, g \in \mathcal{T}, f + g \leq 1\}.$$

If $A = (\mu_A, \nu_A), B = (\mu_B, \nu_B) \in \mathcal{F}$, then

$$A \oplus B = (\mu_A \oplus \mu_B, \nu_A \odot \nu_B),$$
$$A \odot B = (\mu_A \odot \mu_B, \nu_A \oplus \nu_B).$$

If $A_n = (\mu_{A_n}, \nu_{A_n})$, then

$$A_n \nearrow A \Leftrightarrow \mu_{A_n} \nearrow \mu_A, \nu_{A_n} \searrow \nu_A.$$

Finally denote by $\mathcal{J}$ the family of all closed intervals $[a, b]$ such that $0 \leq a \leq b$. Recall that

$$[a, b] + [c, d] = [a + c, b + d],$$
$$[a_n, b_n] \nearrow [a, b] \Leftrightarrow a_n \nearrow a, b_n \nearrow b.$$

B. Riečan: *On the Law of Large Numbers on IFS Events*, Advances in Soft Computing **2**, 677–680 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

The basic notions of the probability theory are probability and random variable. In the classical case the probability is a mapping from a σ-algebra $\mathcal{S}$ of sets to [0,1], in IFS case it is defined axiomatically as a mapping from $\mathcal{F}$ to [0,1]. Random variable is usually defined as a measurable mapping. Of course, to any random variable $\omega \mapsto \xi(\omega)$ a mapping $A \mapsto \xi^{-1}(A)$ can be assigned from the σ-algebra $\mathcal{B}(R)$ of Borel sets to the given σ-algebra $\mathcal{S}$. As an analogy we shall consider an observable as a mapping from $\mathcal{B}(R)$ to $\mathcal{F}$.

In Section 2 we shall define probability and observables in the IFS theory and independency of a sequence of observables. Using two representations theorems taken from [3] and [5] we can decompose problems from IFS probability theory into two components working in MV-algebra probability theory. Therefore some known results from the MV-algebra theory ([6]) may be used. As an illustration the weak law of large numbers will be presented in Section 3.

2. Probability and Observation

Recall that the probability on $\mathcal{T}$ ([6]) is a function $p : \mathcal{T} \to [0,1]$ such that

$$(i) p(1_\Omega) = 1, p(0_\Omega) = 0;$$

$$(ii) p(f) + p(g) = p(f \oplus g) + p(f \odot g), f, g \in \mathcal{T};$$

$$(iii) f_n \nearrow f \Longrightarrow p(f_n) \nearrow p(f).$$

Observable is a mapping $y : \mathcal{B}(R) \to \mathcal{T}$ satisfying the following conditions:

$$(i) y(R) = 1_\Omega;$$

$$A \cap B = \emptyset \Longrightarrow y(A) \odot y(B) = 0_\Omega, y(A \cup B) = y(A) + y(B);$$

$$A_n \nearrow A \Longrightarrow y(A_n) \nearrow y(A).$$

An observable y is integrable, if there exists

$$E(y) = \int_R t dp_y(t),$$

where $p_y : \mathcal{B}(R) \to [0,1]$ is defined by $p_y(A) = p(y(A)), A \in \mathcal{B}(R)$.

Definition 1. By an IFS-probability on $\mathcal{F}$ we understand any function $\mathcal{P} : \mathcal{F} \to \mathcal{J}$ satisfying the following properties:

$$(i) \mathcal{P}((1_\Omega, 0_\Omega)) = [1,1], \mathcal{P}((0_\Omega, 1_\Omega)) = [0,0];$$

$$(ii) \mathcal{P}(A \oplus B) + \mathcal{P}(A \odot B) = \mathcal{P}(A) + \mathcal{P}(B);$$

$$(iii) A_n \nearrow A \Longrightarrow \mathcal{P}(A_n) \nearrow \mathcal{P}(A).$$

$\mathcal{P}$ is called separating, if $\mathcal{P}((f,g)) = [p(f), 1 - q(g)]$ for some $p, q : \mathcal{T} \to [0,1]$.

Proposition 1. Let $\mathcal{P} : \mathcal{F} \to \mathcal{J}$ be a separating IFS-probability. For any $(f,g) \in \mathcal{F}$ denote $\mathcal{P}((f,g)) = [p(f), 1 - q(g)]$ Then $p, q : \mathcal{T} \to [0,1]$ are probabilities (in the sense of [6]).

Proof. [5], Section 3.

Definition 2. A mapping $x : \mathcal{B}(R) \to \mathcal{F}$ is called an IFS-observable, if it satisfies the following conditions:

$$(i) x(R) = (1_\Omega, 0_\Omega);$$

$$(ii) A \cap B = \emptyset \Longrightarrow x(A) \odot x(B) = (0_\Omega, 1_\Omega), x(A \cup B) = x(A) \oplus x(B);$$

$$(iii) A_n \nearrow A \Longrightarrow x(A_n) \nearrow x(A).$$

Proposition 2. Let $x : \mathcal{B}(R) \to \mathcal{F}$ be an IFS-observable. For any $A \in \mathcal{B}(R)$ denote $x(A) = (x^{\flat}(A), 1 - x^{\sharp}(A))$. Then $x^{\flat}, x^{\sharp} : \mathcal{B}(R) \to \mathcal{T}$ are observables in the sense of [6].

Proof. [5], Section 4.

Definition 3. IFS-observables $x_1, ..., x_n : \mathcal{B}(R) \to \mathcal{F}$ are called independent, if there exists n-dimensional IFS-observable $h : \mathcal{B}R^n) \to \mathcal{F}$ such that

$$\mathcal{P}(h(A_1 \times ... \times A_n)) = \mathcal{P}(x_1(A_1)) \otimes ... \otimes \mathcal{P}(x_n(A_n))$$

for any $A_1, ..., A_n \in \mathcal{B}(R)$. Here $[a_1, b_1] \otimes ... \otimes [a_n, b_n] = [a_1...a_n, b_1...b_n]$ for any $[a_i, b_i] \in \mathcal{J}$.

The mapping h is called the joint observable of the observables $x_1, ..., x_n$.

Proposition 3. If $x_1, ..., x_n$ are independent, then $x_1^{\flat}, ..., x_n^{\flat} : \mathcal{B}(R) \to \mathcal{T}$ are independent as well as $x_1^{\sharp}, ..., x_n^{\sharp} : \mathcal{B}(R) \to \mathcal{T}$.

Proof. [3], Theorem 4.3.

3. Weak Law of Large Numbers

Definition 4. Let $x_1, ..., x_n$ be independent IFS-observables, $h_n : R^n \to \mathcal{F}$ their joint observable, $g_n : R^n \to R, g_n(u_1, ..., u_n) = \frac{1}{n}\sum_{i=1}^{n} u_i$. Then $\frac{1}{n}\sum_{i=1}^{n} x_i : \mathcal{B}(R) \to \mathcal{F}$ is defined by the formula $\frac{1}{n}\sum_{i=1}^{n} x_i = h_n \circ g_n^{-1}$.

Theorem. Let $(x_n)_{n=1}^{\infty}$ be a sequence of independent, equally distributed observables ($i.e. x_n(A) = x_1(A)$ for any $n \in N$ and any $A \in \mathcal{B}(R)$), $E(x_1^{\flat}) = E(x_1^{\sharp}) = 0$. Then

$$lim_{n\to\infty}\mathcal{P}((\frac{1}{n}\sum_{i=1}^{n} x_i)(-\epsilon, \epsilon)) = [0, 0].$$

Proof. Consider the sequences $(x_n^{\flat})_{n=1}^{\infty}, (x_n^{\sharp})_{n=1}^{\infty}$ of observables from $\mathcal{B}(R)$ to R. They are sequences of independent, equally distributed observables from $\mathcal{B}(R)$ to $\mathcal{T}$, $E(x_n^{\flat}) = 0, E(x_n^{\sharp}) = 0 (n = 1, 2, ...)$. Therefore by [6] Theorem 2.15

$$lim_{n\to\infty} p((\frac{1}{n}\sum_{i=1}^{n} x_i^{\flat})((-\epsilon, \epsilon))) = 0,$$

$$lim_{n\to\infty} q((\frac{1}{n}\sum_{i=1}^{n} x_i^{\sharp})((-\epsilon, \epsilon))) = 0$$

for any $\varepsilon > 0$. Therefore

$$lim_{n\to\infty}\mathcal{P}((\frac{1}{n}\sum_{i=1}^{n} x_i)((-\epsilon, \epsilon)))$$

$$= lim_{n\to\infty}[p((\frac{1}{n}\sum_{i=1}^{n} x_i)^{\flat}((-\epsilon, \epsilon))), 1 - q(1 - (\frac{1}{n}\sum_{i=1}^{n} x_i)^{\sharp}((-\epsilon, \epsilon)))]$$

$$= lim_{n\to\infty}[p((\frac{1}{n}\sum_{i=1}^{n} x_i^{\flat})((-\epsilon, \epsilon))), q((\frac{1}{n}\sum_{i=1}^{n} x_i^{\sharp})((-\epsilon, \epsilon)))]$$

$$= [0, 0]$$

.

4. Concluding Remarks

The problem of constructing probability theory on IFS events has been posed by K. Atannasov ([1]). As a solution of the problem we suggest in the paper a decomposition of a probability as well as observables into two components. There is well known how to work with these components: probability theory on tribes or more general on MV-algebras ([6]). As an illustration of the method we have presented here a variant of the law of large numbers. While any observable can be decomposed by the described method, only a type of probabilities is decomposable. In the general case this method does not seem to be useful and another approach must be developed. It will be realized in another paper.

Acknowledgement. This paper has been supported by grant VEGA 1/9056/02.

References

[1] Atannasov K.: Intuitionistic Fuzzy Sets: Theory and Applications, Physica Verlag, New York, 1999.

[2] Grzegorzewski, P. - Mrowka, E.: Probability on intuitionistic fuzzy events. In: Soft Methods in Probability, Statistics, and data Analysis (P.Grzegorzewski et al. eds.), Physica Verlag, New York 2002, 105 - 115.

[3] Petrovičová, J. - Riečan, B.: On the central limit theorem on IFS-events. Soft Computing, and Mathware (submitted).

[4] Riečan, B.A descriptive definition of the probability on intuitionistic fuzzy sets. In: EUSFLAT 2003 (Wagenecht, M., Hampel, R. eds.) Zittau-Goerlitz Univ. Appl. Sci., 2003, 263 - 266.

[5] Riečan, B.: representation of probabilities on IFS events. Proc. SMPS'2004, Oviedo (to appear).

[6] Riečan, B. - Mundici, D.: Probability on MV-algebras. In: Handbook of Measure theory (E.Pap ed.) Elsevier, Amsterdam 2002, 869 - 999.

An Axiomatic Approach to Cardinalities of IF Sets

Pavol Kráĺ

Faculty of Economics, Matej Bel University,
Department of Applied Informatics,
Tajovského 10,
975 90 Banská Bystrica, Slovakia
pavol.kral@umb.sk

Abstract. There are several approaches to the cardinality of fuzzy sets. One group of them are constructive approaches. Following these approaches we get single numbers (scalar cardinalities) or convex fuzzy sets (fuzzy cardinalities) as cardinalities of fuzzy sets. Wygralak in [8] has presented an axiomatic theory of the scalar cardinality of finite fuzzy sets which contains as particular cases all standard concepts of the scalar cardinality. In our contribution we will present possible extensions to the case of IF sets.

Key words: cardinality, IF sets

1 Introduction

Cardinality is a very important characteristic of a crisp set. We would like to get similar characteristic also for fuzzy sets. There is rather theoretical motivation for dealing with the notion of the cardinality of fuzzy sets. Nevertheless, measuring the cardinality of fuzzy sets has also many applications, especially in the case of finite fuzzy sets. For instance in communication with databases we mean the problem of satisfactory and adequate answer to queries of the form: "How many elements are p? " or "Are there more elements which are p than elements which are q?", where p, q are arbitrary properties. Those queries are about cardinalities or comparisons of cardinalities of fuzzy sets. There are several approaches to the cardinality of fuzzy sets. One group of them are constructive approaches. Following these approaches we get a single number alternatively fuzzy set as a cardinality of fuzzy sets. In many applications one prefers a simple scalar approximation of cardinality of fuzzy sets. The complete axiomatic theory of scalar cardinality of fuzzy sets can be found in Wygralak's book [8]. The axiomatic theory of fuzzy cardinality was introduced by Casasnovas and Torrens in [3]. We introduce here the scalar car-

P. Král: *An Axiomatic Approach to Cardinalities of IF Sets*, Advances in Soft Computing **2**, 681–691 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

dinality for IF sets as a generalization of cardinality of fuzzy sets and a special case of cardinality of L-fuzzy sets. We use the following basic definitions:

Definition 1. *(Zadeh 1965) A fuzzy set (FS) A on a universe X is a function* $A : X \to [0, 1]$.

Definition 2. *(Atanassov 1983) An IF set on a universe X is an object of the form*

$$A = \{(x, \mu_A(x), \nu_A(x))| \quad x \in X\},$$

where μ_A *and* ν_A *satisfy the following condition*

$$(\forall x \in X)(\mu_A(x) + \nu_A(x) \leq 1).$$

$\mu_A(x) \in [0, 1]$ *and* $\nu_A(x) \in [0, 1]$ *are called the membership degree and non-membership degree, respectively, of* $x \in A$.

Definition 3. *(Goguen 1967) An L-fuzzy set (LFS) A on a universe X is a function* $A : X \to L$, *where L is a complete distributive lattice equipped with standard operations* $\vee, \wedge$, *bottom element* $\mathbf{0}$, *top element* $\mathbf{1}$ *and unary, involutive, order-reversing operator* $\mathbf{N}$.

The union, intersection and complement for fuzzy sets, IF sets and LF sets are defined as follows:

Definition 4. *Let A, B* $\in$ *FS. Then*
$A \cup B(x) = max(A(x)B(x)), \forall x \in X$
$A \cap B(x) = min(A(x)B(x)), \forall x \in X$
$coA(x) = 1 - A(x), \forall x \in X$

Definition 5. *Let A, B* $\in$ *IFS. Then*
$A \cup B = \{x, \max(\mu_A(x), \mu_B(x)), \min(\nu_A(x)\nu_B(x))|x \in X\}$
$A \cap B = \{x, \min(\mu_A(x), \mu_B(x)), \max(\nu_A(x)\nu_B(x))|x \in X\}$
$coA = \{(x, \nu_A(x)\mu_A(x))|x \in X\}$

Definition 6. *Let A, B* $\in$ *LFS. Then*
$A \cup B(x) = \sup(A(x)B(x)), \forall x \in X$
$A \cap B(x) = \inf(A(x)B(x)), \forall x \in X$
$coA(x) = N(A(x)), \forall x \in X$

We use the following notations. The cardinality of a crisp set B will be denoted by $|\mathbf{B}|$. The cardinality of a fuzzy set A will be denoted by $\mathbf{card}(\mathbf{A})$. The cardinality of an IF set A will be denoted by $\mathbf{card_I}(\mathbf{A})$. The cardinality of a LF set A will be denoted by $\mathbf{card_L}(\mathbf{A})$.

2 Cardinality of L-fuzzy sets

The concept of LFS (introduced by Goguen [4]) is a generalization of the concept of fuzzy sets and includes the latter as a special case when $L = [0, 1]$. There are several different kinds of understanding of the concept of an LF set distinguished by how to specify the lattice L.

We use the following basic definitions:

$$A_\alpha = \{x \in X : A(x) \geq \alpha\}$$

with $\alpha \in L\backslash\{0\}$ (α-cut set of LFS A),

$$A^\alpha = \{x \in X : A(x) > \alpha\}$$

with $\alpha \in L\backslash\{1\}$ (sharp α-cut set of LFS A),

$$core(A) = A_\alpha \text{ (core of } A),$$

$$supp(A) = A^0 \text{ (support of } A \text{).}$$

If $supp(A)$ is finite, we shall say that A is a finite LF set (LFFS). Following [8] we define the cardinality of LFF sets:

Definition 7. *An element a of the lattice L is irreducible if $a = b \vee c$ implies $b = a$ or $c = a$. An irreducible element a of the lattice L is maximal irreducible if for each b irreducible $a \leq b$ implies $a = b$.*

Definition 8. *Let n be a number of maximal irreducible elements of a lattice L. A function $card_L : LFFS \to [0, \infty)^n$ will be called a scalar cardinality if the following postulates are satisfied for each a, $b \in L$, x, $y \in X$ and A, $B \in LFFS$:*

1. *coincidence*
$$card_L(\mathbf{1}/x) = (\underbrace{1, ..., 1}_{n})$$
2. *monotonicity*
$$a \leq b \Rightarrow card_L(a/x) \leq card_L(b/y)$$
3. *additivity*
$$supp(A) \cap supp(B) = \emptyset \Rightarrow$$
$$card_L(A \cup B) = card_L(A) + card_L(B)$$

Proposition 1. *Let A, $B \in LFFS$ and $A_i \in LFFS$ for each index $i \in J$ (J-finite). The following properties hold for each scalar cardinality:*

1. **additivity**
$$card_L(\bigcup_{i \in J} A_i) = \sum_{i \in J} card_L(A_i)$$
whenever $supp(A_i) \cap supp(A_j) = \emptyset$ for each $i \neq j$,

2. **coincidence** $card_L(A) = |core(A)|$ *if* $A \in LFCS$,
3. **monotonicity** $card_L(A) \leq card_L(B)$ *if* $A \subset B$,
4. **boundedness** $|supp(A)| \leq card_L(A) \leq |core(A)|$,
5. **shiftability** $card_L(A) = card_L(B)$ *iff there exists a bijection* $b : supp(A) \to supp(B)$ *such that* $A(x) = B(b(x))$ *for each* $x \in supp(A)$.

Proof.

It is easy to see that this properties are immediate consequences of the axioms from definition 8. □

The following theorem brings a useful characterization of scalar cardinality of LF sets:

Theorem 1. *Let* n *be a number of maximal irreducible elements of a lattice* L. *A mapping* $card_L : LFFS \to [0, \infty)^n$ *is a scalar cardinality iff there exists a function* $f_L : L \to [0, 1]^n$ *fulfilling the conditions:*

1. $f_L(\mathbf{0}) = (\underbrace{0, ..., 0}_{n})$, $f_L(\mathbf{1}) = (\underbrace{1, ..., 1}_{n})$
2. $f_L(a) \leq f_L(b)$ *whenever* $a \leq b$ *and such that*

$$card_L(A) = \sum_{x \in supp(A)} f_L(A(x))$$

for each $A \in LFFS$.

Proof.

" $\Rightarrow$ " Assume a scalar cardinality $card_L : LFFS \to [0, \infty)^n$. By additivity we obtain

$$card_L(A) = \sum_{x \in supp(A)} card_L(A(x)/x)$$

for each $A \in LFFS$. Let $f_L : L \to [0, 1]^n$ be a function such that $f_L(a) = card_L(a/x)$ for $a \in L$. It is easy to see that f_L fulfils required conditions.

" $\Leftarrow$ " Assume that f_L satisfies conditions (1), (2) and $card_L : LFFS \to [0, \infty)^n$ is of the form

$$card_L(A) = \sum_{x \in supp(A)} f_L(A(x))$$

for each $A \in LFFS$.

Conditions (1), (2) lead to the fulfilment of coincidence and monotonicity for $card_L$. Let $supp(A_i) \cap supp(B_j) = \emptyset$. Hence

$$card_L(A \cup B) = \sum_{x \in supp(A \cup B)} f_L((A \cup B)(x)) =$$

$$= \sum_{x \in supp(A)} f_L(A(x)) + \sum_{x \in supp(B)} f_L(B(x)) = card_L(A) + card_L(B),$$

which means that additivity is satisfied. □

Each function f_L satisfying the conditions from the previous theorem will be called a L-cardinality pattern as it expresses understanding of the scalar cardinality of a singleton.

Example 1 *Let L be a lattice (see figure 1).*

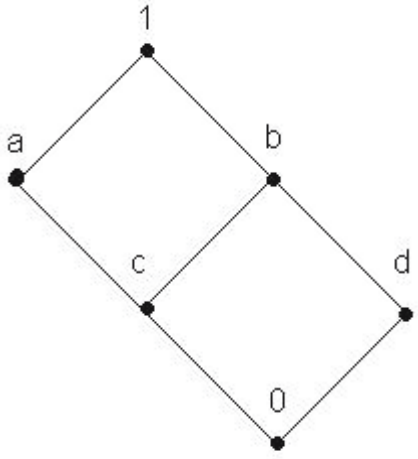


Fig. 1.

Let $A : \{x_1, x_2, x_3, x_4, x_5\} \to L$ *be a LF set:*

$$A = \mathbf{1}/x_1 + \mathbf{0}/x_2 + a/x_3 + b/x_4 + d/x_5.$$

Irreducible elements of lattice L are a, c, d. *Maximal irreducible elements are* a, d. *For each element in* $L\backslash\{\mathbf{0}, \mathbf{1}\}$ *we can find a minimal partition on irreducible elements* $(b = c \vee d)$. *We can assume following L - cardinality pattern:*

$$f_L(y) = \begin{cases} (1,0) & if\ y \geq c\ and\ y\ is\ irreducible, \\ (0,1) & if\ y \geq d\ and\ y\ is\ irreducible, \\ (0,0) & if\ (y < c) \wedge (y < d)\ and\ y\ is\ irreducible, \\ \sum_i f_L(y_i) & if\ y = \vee_i y_i\ and\ y_i\ are\ irreducible. \end{cases}$$

The cardinality $card_L(A) = (3, 3)$.

Important features of sets and their cardinalities are well-known valuation property, subadditivity and complementarity rule:

Valuation property For each A,$B \in$ LFFS

$$card_L(A \cap B) + card_L(A \cup B) = card_L(A) + card_L(B).$$

Subadditivity property For each $(A_i)_{i \in J} \in$ LFFS

$$card_L(\bigcup_{i \in J} A_i) \leq \sum_{i \in J} card_L(A_i).$$

Complementarity rule For each $A \in LFFS$

$$card_L(A) + card_L(coA) = |X|$$

with an involution N.

Cartesian product rule For each A,$B \in$ LFFS

$$card_L(A \times_\wedge B) = card_L(A).card_L(B)$$

Necessary and sufficient conditions for cardinality pattern to satisfy valuation property, subadditivity property, complementarity rule can be found in [5]:

Proposition 2. *The subadditivity property is satisfied by a cardinality pattern f_L iff for each a, $b \in L$, a, b-incomparable:*

$$f_L(a \vee b) \leq f_L(a) + f_L(b).$$

Proposition 3. *The valuation property holds true iff a cardinality pattern f_L is such that for each a, $b \in L$, a, b-incomparable:*

$$f_L(a \vee b) + f_L(a \wedge b) = f_L(a) + f_L(b).$$

Proposition 4. *The complementarity rule holds true for a L-cardinality pattern f_L and an order-reversing involution $\mathbf{N}$ iff*

$$f_L(a) + f_L(\mathbf{N}(a)) = (\underbrace{1, ..., 1}_{n}) \text{ for each } a \in L,$$

where n is a number of maximal irreducible elements of a lattice L.

3 Cardinality of IF sets

The concept of IF sets (Atanassov in [1]) is a generalization of the concept of fuzzy sets and a special case of LF sets.

Remark 1 *An IF set can be understood as an L^*FS for a complete lattice $(L^*, \leq_L)$ with $\mathbf{N}$ defined by $L^* = \{(\mu, \nu) \in [0;1]^2 | \ \mu \leq 1 - \nu\}$, $(\mu_1, \nu_1) \leq (\mu_2, \nu_2) \Longleftrightarrow \mu_1 \leq \mu_2 \wedge \nu_1 \geq \nu_2$, $\mathbf{N}(\mu, \nu) = (\nu, \mu)$.*

Deschrijver, Cornelis and Kerre in [2] have extended triangular norm, triangular conorm a negator to the lattice L^*.

Definition 9. *[2] A negator on L^* is any decreasing mapping $\mathbf{N} : L^* \to L^*$ satisfying $\mathbf{N}(\mathbf{0}) = \mathbf{1}$ and $\mathbf{N}(\mathbf{1}) = \mathbf{0}$. A triangular norm (t-norm) on L^* is a mapping $\mathbf{T} : (L^*)^2 \to L^*$ satisfying the following conditions:*

- $(\forall x \in L^*)(T(x, \mathbf{1}) = x)$,
- $(\forall (x, y)) \in (L^*)^2 (T(x, y) = T(y, x))$,
- $(\forall (x, y, z) \in (L^*)^3 (T(x, T(y, z)) = T(T(x, y), z)))$,
- $(\forall (x, x', y, y') \in (L^*)^4)(x \leq_{L^*} x' \text{ and} y \leq_{L^*} y' \Rightarrow T(x, y) \leq_{L^*} T(x', y'))$.

A triangular conorm (t-conorm) on L^* is a mapping $\mathbf{S} : (L^*)^2 \to L^*$ satisfying the following conditions:

- $(\forall x \in L^*)(S(x, \mathbf{0}) = x)$,
- $(\forall (x, y)) \in (L^*)^2 (S(x, y) = S(y, x))$,
- $(\forall (x, y, z) \in (L^*)^3 (S(x, S(y, z)) = S(S(x, y), z)))$,
- $(\forall (x, x', y, y') \in (L^*)^4)(x \leq_{L^*} x' \text{ and} y \leq_{L^*} y' \Rightarrow S(x, y) \leq_{L^*} S(x', y'))$.

In IF set theory negators are used to model the complement of an IF set, t-norms are used to model the intersection and t-conorms are used to model union of two IF sets. Let A, $B \in$ IFS:

- $A \cup_T B(x) = T(A(x)B(x)), \forall x \in X$
- $A \cap_S B(x) = S(A(x)B(x)), \forall x \in X$
- $coA(x) = N(A(x)), \forall x \in X$.

Definition 10. *[2] A t-norm $\mathbf{T}$ on L^* is called t-representable iff there exist a t-norm T and a t-conorm S on [0,1] such that, for all $x, y \in L^*$,*

$$\mathbf{T}(x, y) = (T(x_1, y_1), S(x_2, y_2)).$$

A t-conorm $\mathbf{S}$ on L^* is called t-representable iff there exist a t-norm T and a t-conorm S on [0,1] such that, for all $x, y \in L^*$,

$$\mathbf{S}(x, y) = (S(x_1, y_1), T(x_2, y_2)).$$

Remark 2 *[2] The t-norm*

$$\mathbf{T_W}(x, y) = (\max(0, x_1 + y_1 - 1), \min(1, x_2 + 1 - y_1, y_2 + 1 - x_1)$$

is an example of not t-representable t-norm on L^*. The dual t-conorm

$$\mathbf{S_W}(x, y) = (\min(1, x_2 + 1 - y_1, y_2 + 1 - x_1), \max(0, x_2 + y_2 - 1))$$

is an example of not t-representable t-conorm on L^*.

For each IF set in X, we will call $\pi_A(x) = 1 - \mu_A(x) - \nu_A(x)$, the degree of hesitation of x in A. The following cardinalities of IF set are widely used:

Definition 11. *[6] Let A be an IF set in X. The least cardinality of A is equal to the so-called sigma-count and is called the minCount:*

$$minCount(A) = \Sigma_{i=1}^{n} \mu_A(x_i).$$

The biggest cardinality of A is called the maxCount:

$$maxCount(A) = \Sigma_{i=1}^{n} \{\mu_A(x_i) + \pi_A(x_i)\}.$$

Let A be an IF set. For each element (μ, ν) of a lattice L^* we have a partition $(\mu, \nu) = (\mu, 1-\mu) \vee (0, \nu)$. Maximal irreducible elements of a lattice L^* are $(1, 0)$ and $(0, 0)$. The cardinality of IF sets can be defined quite naturally following way:

Definition 12. *Let A be an IF set. A function $card_I : X \to [0, \infty)^2$ is a scalar cardinality of IF set A and $card_I(A) = (card(\mu), card(1-\nu))$, where $card(\mu), card(1-\nu)$ are cardinalities of fuzzy sets μ and $1-\nu$ respectively.*

Some instances of cardinality patterns of IF sets we present in the next example.

Example 2 *a) Let $a \in L^*$,*

$$f_p(a) = \begin{cases} \mathbf{1} \ if \ a \geq p, \\ 0 \ otherwise \end{cases}$$

for $p \in L^*$.
We obtain the smallest cardinality pattern for $p = \mathbf{1}$

$$f_*(a) = \begin{cases} \mathbf{1} \ if \ a = \mathbf{1}, \\ 0 \ otherwise. \end{cases}$$

b) Let $a \in L^*$,

$$f^*(a) = \begin{cases} \mathbf{1} \ if \ a > p, \\ 0 \ otherwise \end{cases}$$

for $p \in L^*$.
We obtain the largest cardinality pattern for $p = \mathbf{0}$

$$f^*(a) = \begin{cases} \mathbf{1} \ if \ a > \mathbf{0}, \\ 0 \ otherwise. \end{cases}$$

We can reformulate subadditivity property, valuation property, complementarity, cartesian product rule and propositions 2 - 4 using t-norms, t-conorms and negators on lattice L^*.

Valuation property For each A,$B \in$ IFFS

$$card_I(A \cap_{\mathbf{T}} B) + card_I(A \cup_{\mathbf{S}} B) = card_I(A) + card_I(B).$$

Subadditivity property For each $(A_i)_{i \in J} \in$ LFFS

$$card_I(\bigcup_{i\in J}^{\mathbf{S}} A_i) \leq \sum_{i\in J} card_I(A_i).$$

Complementarity rule For each $A \in LFFS$

$$card_I(A) + card_I(coA) = |X|$$

with an involution $\mathbf{N}$.

Cartesian product rule For each A,$B \in$ LFFS

$$card_I(A \times_{\mathbf{T}} B) = card_I(A).card_I(B)$$

Proposition 5. *The valuation property holds true iff a cardinality pattern f_I, t-norm $\mathbf{T}$ and t-conorm $\mathbf{S}$ are such that for each a, $b \in L^*$:*

$$f_I(\mathbf{T}(a,b)) + f_I(\mathbf{S}(a,b) = f_I(a) + f_I(b).$$

Proof. " $\Leftarrow$ " Following theorem 1 we obtain

$$card_I(A\cap_{\mathbf{T}}B)+card_I(A\cup_{\mathbf{S}}B) = \sum_{x\in X}(f_I(\mathbf{T}(A(x),B(x)))+f_I(\mathbf{S}(A(x),B(x))) =$$

$$= \textstyle\sum_{x\in X}(f_I(A(x)+f_I(B(x)) = \sum_{x\in supp(A)} f_I(A(x))+\sum_{x\in supp(B)} f_I(B(x)) =$$

$$= card_I(A) + card_I(B).$$

" $\Rightarrow$ " The converse part of the proof is obvious. □

Example 3 *We present examples of triples $(f_I, \mathbf{T}, \mathbf{S})$ satisfying the valuation property:*

a) $(f_I, \wedge, \vee)$ *with any cardinality pattern*
b) $(id, \mathbf{T} = (T_{F,\lambda}, S_{F,\lambda}), \mathbf{S} = (S_{F,\lambda}, T_{F,\lambda}))$, *where* $\lambda \in [0, \infty)$
c) $(f_*, \mathbf{T} = (T, S), \mathbf{S} = (S, T))$, *where S has no zero divisors and* $T(a,b) \leq 1 - S(1-a, 1-b)$ *for* a, $b \in [0,1]$
d) $(f^*, \mathbf{T} = (T, S), \mathbf{S} = (S, T))$, *where T has no zero divisors and* $T(a,b) \leq 1 - S(1-a, 1-b)$ *for* a, $b \in [0,1]$

Proposition 6. *The subadditivity property is satisfied by a cardinality pattern f_I and t-conorm $\mathbf{S}$ iff for each a, $b \in L^*$:*

$$f_I(\mathbf{S}(a,b)) \leq f_I(a) + f_I(b).$$

Proof. " $\Leftarrow$ " Following theorem 1 and the equality $supp(A \cup B) = supp(A) \cup supp(B)$, one gets

$$card_I(A \cup_{\mathbf{S}} B) = \sum_{x\in supp(A\cup B)} f_I(\mathbf{S}(A(x), B(x))) \leq$$

$$\leq \sum_{x \in supp(A \cup B)} f_I(A(x)) + \sum_{x \in supp(A \cup B)} f_I(B(x)) =$$

$$= \sum_{x \in supp(A)} f_I(A(x)) + \sum_{x \in supp(B)} f_I(B(x)) =$$

$$= card_I(A) + card_I(B)$$

for each $A, B \in IFFS$ which implies subadditivity.

" $\Rightarrow$ " The converse part of the proof is obvious. □

The previous example contains instances of couples $(f_I, \mathbf{S})$ satisfying the subadditivity property.

Proposition 7. *The complementarity rule holds true for a cardinality pattern f_I and an order-reversing involution $\mathbf{N}$ iff*

$$f_I(a) + f_I(\mathbf{N}(a)) = (1,1) \text{ for each } a \in L^*.$$

Proof. " $\Leftarrow$ " If condition $f_I(a) + f_I(\mathbf{N}(a)) = (1,1)$ is satisfied, then

$$\sum_{x \in X} f_I(A(x)) + \sum_{x \in X} f_I(\mathbf{N}(A(x))) = card_I(A) + card_I(coA) = |X|.$$

" $\Rightarrow$ " The converse part of the proof is obvious. □

We present the example of couples $(f_I, \mathbf{N})$ satisfying the complementarity rule.

Example 4 *a)* $(f_*, \mathbf{N}(a) = \mathbf{1}$ *for each* $a < \mathbf{1})$
b) $(f^*, \mathbf{N}(a) = \mathbf{0}$ *for each* $a > \mathbf{0})$

Proposition 8. *The cartesian product rule holds true iff cardinality pattern f_I and t-norm $\mathbf{T}$ are such that for each a, $b \in L^*$*

$$f(\mathbf{T}(A(x), B(y))) = f(a).f(b)$$

Proof. " $\Leftarrow$ " If condition $f(\mathbf{T}(A(x), B(y))) = f(a).f(b)$ is satisfied , then

$$\sum_{(x,y) \in X \times X} f_I(\mathbf{T}(A(x)B(y))) = \sum_{(x,y) \in X \times X} f_I(A(x)).f_I(B(x)) =$$

$$= \sum_{(x,y) \in supp(A \times_{\mathbf{T}} B)} f_I(\mathbf{T}(A(x)B(y))) = \sum_{(x,y) \in supp(A) \times supp(B)} f_I(A(x)).f_I(B(x)) =$$

$$= card_I(A).card_I(B)$$

" $\Rightarrow$ " The converse part of the proof is obvious. □

The next example shows couples $(f_I, \mathbf{T})$ which satisfy the cartesian product rule.

Example 5 *a)* $(f_*, \wedge)$
b) $(f^*, \wedge)$

4 Conclusion

We have extended the notion of cardinality to IF sets. We have proven some important results which are a natural generalization of the axiomatic cardinality theory of fuzzy sets.

References

1. Atanassov, K., T. (1986) Intuitionistic fuzzy sets. Fuzzy Sets and Systems 20: 87-96.
2. Deschrijver, G., Cornelis, C., Kerre, E. E. (2003) Intuitionistic fuzzy connectives revisited, In: Proceedings of the 9th Iternational Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, 1839 - 1844.
3. Casasnovas, M., Torrens, J. (2003) An axiomatic approach to fuzzy cardinalities of finite fuzzy sets. Fuzzy Sets and Systems 133: 193-209.
4. Goguen, J., A. (1967) L-fuzzy sets, J. Math. Anal. Appl. 18: 145-174.
5. Král', P. (2004) On the Scalar Cardinality of IF Sets, submitted to Journal of Electrical Engineering
6. Szmidt E., Kacprzyk J. (2001) Entropy for intuitionistic fuzzy sets. Fuzzy Sets and Systems 110: 467-477
7. Wygralak, M. (2000) An axiomatic approach to scalar cardinalities of a fuzzy set. Fuzzy Sets and Systems 110: 175-176.
8. Wygralak, M. (2003) Cardinalities of Fuzzy Sets. Studies in Fuzziness and Soft Computing 118, Springer-Verlag Berlin Heidelberg.

Fuzzy Optimization

Sensitivity Analysis for Fuzzy Shortest Path Problem

T. Starostina[1] and S. Dempe[2]

[1] Taganrog State University of Radio Engineering, Taganrog, Russia
tanya@pbox.ttn.ru
[2] Technical University Bergakademie Freiberg, Department of Mathematics and Computer Science, Freiberg, Germany
dempe@math.tu-freiberg.de

Abstract. The shortest path problem is an optimization problem in which the best path between two considered objects is searched for in accordance with an optimization criterion, which has to be minimized. In this paper this problem is investigated in the case when the distances between the nodes are fuzzy numbers. The problem is formulated as a linear optimization problem with fuzzy coefficients in the objective function. This problem is solved using crisp parametric two-criterial linear optimization. Special emphasis is given to the sensitivity of the solution with respect to the fuzzy objective function coefficients.

1 Introduction

Consider a directed graph $G = (V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ is a set of nodes and $E = \{(v_1, w_1), \ldots, (v_m, w_m)\}$ is a set of directed edges. Each edge $(v, w) \in E$ connects two nodes $v, w \in V$ of the graph G. There is a positive number (or a weight) $c(v, w)$ associated with each edge $(v, w) \in E$ that can represent the length of this edge, the time needed to cover it, etc. Given two nodes $q \in V$ and $s \in V$ a path from q to s is a sequence of edges $\{(u_0, u_1), (u_1, u_2), \ldots, (u_{t-2}, u_{t-1}), (u_{t-1}, u_t)\} \subseteq E$ with $u_0 = q$ and $u_t = s$, where $\{q, u_1, \ldots, u_t, s\} \subseteq V$ are all distinct. In the shortest path problem a path from q to s is searched with the minimal length $\sum_{i=0}^{t-1} c(u_i, u_{i+1})$.

Denote a set of arrows beginning in node $v \in V$ by $\Gamma_+(v) := \{(\overline{v}, \overline{w}) \in E : \overline{v} = v\}$ and $\deg^+(v) := |\Gamma_+(v)|$ is the *outdegree* of $v \in V$. Similarly, the set $\Gamma_-(w) := \{(\overline{v}, \overline{w}) \in E : \overline{w} = w\}$ describes a set of arrows which are ended in the node $w \in V$ and $\deg^-(w) := |\Gamma_-(w)|$ is the *indegree* of node $w \in V$. Then, the shortest path problem can be modelled as a linear optimization problem [6, 8] as:

T. Starostina and S. Dempe: *Sensitivity Analysis for Fuzzy Shortest Path Problem*, Advances in Soft Computing **2**, 695–702 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

$$\begin{aligned} &\sum_{(v,w)\in E} c(v,w)x(v,w) \to \min \\ &\sum_{w\in\Gamma_+(v)} x(v,w) - \sum_{u\in\Gamma_-(v)} x(u,v) = g(v),\ \forall v \in V \\ &x(v,w) \geq 0,\ \forall (v,w) \in E, \end{aligned} \tag{1}$$

where

$$g(v) = \begin{cases} 0, \text{ if } v \notin \{q,s\} \\ 1, \text{ for } v = q \\ -1, \text{ for } v = s. \end{cases} \tag{2}$$

Denote the feasible set of this problem by M. Walks in graph G correspond to integer feasible solutions of this problem. $M \neq \emptyset$ whenever there is a path from q to s. Since the coefficient matrix of the constraints of this problem is totally unimodular [7], the vertices of M have integer components and, hence, are the incidence vectors of the walks in G, i.e. sequences of edges starting in q and ending in s but possibly crossing one node multiply. In this case it is said that the walk contains cycles. If all the distances between the nodes of the graph G are positive, optimal vertex solutions of problem (1) are paths, they do not contain cycles. Hence, in what follows, $c(v,w) > 0$ for all $v, w \in V$ is assumed.

Usually it is supposed that the parameters $c(v,w)$ in the objective function of this model are exactly known. However, in many real situations these data can not be given exactly because of the influence of various factors of environment. Then the problem can be appropriately modelled using a graph with fuzzy parameters. This corresponds to a model (1) with fuzzy coefficients in the objective function.

Focus in this paper is on the situation when the membership functions of the distances are not precisely known in advance. This could be considered as a realistic situation. Then, both the dependency of the optimal (fuzzy) solution of the problem on the fuzzy distances as well as a path from q to s in the graph being more or less equally "good" for all possible distances are of special interest. For related investigations of a fuzzy linear optimization problem the interested reader is referred to the paper [4].

2 The two-criterial optimization approach

Assume now that in model (1) the weights $c(v,w)$ are fuzzy numbers of the type $L-L$ [5]:

$$\tilde{c}(v,w) = (\underline{c}(v,w); \overline{c}(v,w); \alpha(v,w); \beta(v,w))_{L-L},\ \forall\ (v,w) \in E \tag{3}$$

where $\underline{c}(v,w), \overline{c}(v,w)$ - are the left and right borders of the fuzzy number $\tilde{c}(v,w)$ corresponding to the maximal reliability level ($\lambda = 1$) and $\alpha(v,w)$ and $\beta(v,w)$ are non-negative real numbers (cf. Figure 1). To guarantee that $\tilde{c}(v,w)$ is positive it is assumed that $\underline{c}(v,w) - \alpha(v,w) > 0$ for all edges $(v,w) \in E$.

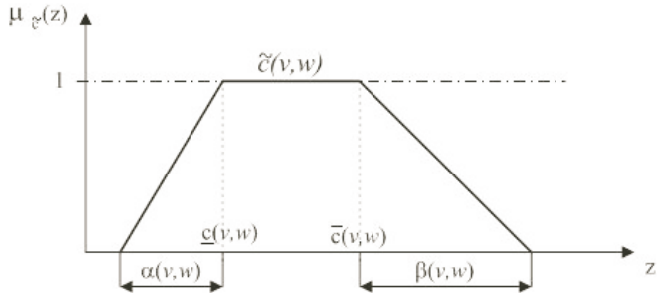


Fig. 1. Used type of fuzzy numbers

A fuzzy number $\tilde{c}(v,w)$ is defined as a fuzzy set in the space of real numbers with the following membership function [5]:

$$\mu_{\tilde{c}}(z) = \begin{cases} 1 & \text{if } \underline{c} \le z \le \overline{c}, \\ L\left(\dfrac{\underline{c}-z}{\alpha}\right) & \text{if } z \le \underline{c}, \\ L\left(\dfrac{z-\overline{c}}{\beta}\right) & \text{if } z \ge \overline{c}, \end{cases} \tag{4}$$

where L is a shape function, which satisfies to following conditions:
- L is a continuous non-increasing function on $[0,\infty)$ with $L(0)=1$;
- L is strictly decreasing on that part of $[0,\infty)$ on which it is positive.

The shortest path problem in a directed graph with fuzzy weights (or "fuzzy shortest path problem" for short) is problem (1) with $c(v,w)$ being replaced with the fuzzy weights $\tilde{c}(v,w)$ in the objective function. In analogy with [3, 2] this problem can be associated with a set of the following problems, which depend on a parameter $\theta \in (0,1)$:

$$\begin{aligned} f_1(x,\theta) &= \sum_{(v,w)\in E} (\underline{c}(v,w) - \alpha(v,w)\theta)x(v,w) \to \min \\ f_2(x,\theta) &= \sum_{(v,w)\in E} (\overline{c}(v,w) + \beta(v,w)\theta)x(v,w) \to \min \\ \sum_{w\in\Gamma_+(v)} x(v,w) &- \sum_{u\in\Gamma_-(v)} x(u,v) = g(v),\ \forall v \in V, \\ x(v,w) &\ge 0,\ \forall (v,w) \in E. \end{aligned} \tag{5}$$

This model is based on the preference relation

$$\begin{aligned} a \le b &\Longleftrightarrow \underline{a} \le \underline{b} \wedge \overline{a} \le \overline{b}, \\ a < b &\Longleftrightarrow a \le b \wedge a \ne b, \end{aligned}$$

between intervals $a = [\underline{a}, \overline{a}]$ and $b = [\underline{b}, \overline{b}]$ [2].

Thus, to find a shortest path between the nodes q and s problem (5) has to be solved for all $\theta \in (0,1)$.

Problem (5) is a two-criterial optimization problem. One solution concept for such problems is to find one (or better all) Pareto-optimal solution(s).

Definition 1. *A point $x^* \in X$ is a Pareto-optimal solution of a two-criterial optimization problem*

$$\begin{aligned} &f_1(x,\theta) \to \min \\ &f_2(x,\theta) \to \min \\ &x \in X \end{aligned}$$

at $\theta = \theta^*$ if there does not exist another point $\overline{x} \in X$ with $f_1(\overline{x},\theta^*) \leq f_1(x^*,\theta^*)$ and $f_2(\overline{x},\theta^*) \leq f_2(x^*,\theta^*)$ with at least one strict inequality.

Hence, the sets of Pareto-optimal solutions $\Psi(\theta)$ of problem (5) are searched for all $\theta \in [\underline{\theta}, \overline{\theta}]$.

As result a number of different paths in the graph G are computed and each such path is Pareto-optimal for problem (5). All these paths can now be used to compose the fuzzy optimal solution $\tilde{x}$ of the initial fuzzy shortest path problem. Let $\Psi(\theta)$ denote the set of Pareto-optimal solutions of problem (5) for fixed θ. Then the frequency of $x \in \Psi(\theta)$ for $\theta \in [0,1]$ can be used to determine a membership function for $\tilde{x}$ [3]:

$$\mu_{FS}(x) = \left| \left\{ \lambda \in [0,1] : \begin{array}{c} x \text{ is a Pareto-optimal vertex} \\ \text{of the problem (5) for } \theta = L^{-1}(\lambda) \end{array} \right\} \right|.$$

Here, $|Q|$ means the geometric measure of the set Q. Since the set of Pareto-optimal points can be computed using parametric linear programming, $\Psi(\theta)$ equals the union of faces of M. By parametric linear programming, too, Pareto-optimal solutions for one parameter value θ^0 remain Pareto-optimal for all parameter values within some interval $[\underline{\theta}, \overline{\theta}]$. This implies that

$$\mu_{FS}(x) = \sum_{i=1}^{l} (L(\theta_{2i-1}) - L(\theta_{2i})) \tag{6}$$

where $\{\theta_i\}_{i=1}^{2l}$ is such that x is Pareto-optimal for problem (5) for all $\theta \in [\theta_{2i-1}, \theta_{2i}],\ i = 1, \ldots, l$.

3 Sensitivity analysis

Usually the fuzzy numbers $\tilde{c}(v,w)$ have been determined by a group of experts. Asking other experts other fuzzy numbers will result. Hence, it is an interesting question to consider the dependency of the solutions obtained from the parameters of the fuzzy numbers $\tilde{c}(v,w)$. In the following only the special case of the question is considered in which these numbers are perturbed by an additive number $\delta(v,w), \forall\ (v,w) \in E$ (cf. Figure 2). Applying such perturbations to the problem (5) the following model arises:

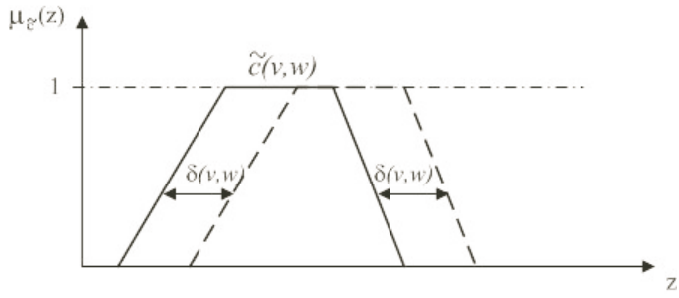


Fig. 2. Perturbed fuzzy numbers

$$\begin{aligned} f_1(x,\theta) &= \sum_{(v,w)\in E} [\underline{c}(v,w) - \alpha(v,w)\theta + \delta(v,w)]\, x(v,w) \to \min \\ f_2(x,\theta) &= \sum_{(v,w)\in E} [\overline{c}(v,w) + \beta(v,w)\theta + \delta(v,w)]\, x(v,w) \to \min \\ &\sum_{w\in\Gamma_+(v)} x(v,w) - \sum_{u\in\Gamma_-(v)} x(u,v) = g(v),\ \forall v \in V, \\ &x(v,w) \geq 0,\ \forall (v,w) \in E. \end{aligned} \tag{7}$$

The interesting point here is the determination of the range in which the $\delta(v,w)$ may vary without violating Pareto-optimality of some path in G. Let

$$\mathcal{R}(x,\theta) := \{\delta : x \in \Psi^{\delta}(\theta)\}$$

denote this set and call it *region of stability* of the path in G with incidence vector x. Here, $\Psi^{\delta}(\theta)$ denotes the set of Pareto-optimal vertices of problem (7).

Theorem 1. *For fixed θ and each feasible point x the set $cl\,\mathcal{R}(x,\theta)$ is a convex polyhedron.*

Proof. Abbreviate the coefficient matrix of the constraints in M by A such that

$$M = \{x : Ax = g,\ x \geq 0\}.$$

The matrix A is the incidence matrix of G, having exactly two nonzero entries in each column. The columns correspond to the edges $(v,w) \in E$ of G with a 1 in the row v and a -1 in row w. In this notation, the vector x is determined by $x_{(v,w)} := x(v,w)$.

Then, the normal cone to M at some incidence vector x to a path in G is

$$N_M(x) = \{z : \exists\ y,\ \exists t \geq 0 \text{ with } z = A^{\top}y + It, x^{\top}t = 0\},$$

where I denotes the unit matrix. An incidence vector x is Pareto-optimal for problem (7) iff there exists $\gamma \in (0,1)$ such that x is an optimal solution of the problem

$$\left.\begin{array}{l}\gamma f_1(x,\theta) + (1-\gamma) f_2(x,\theta) \to \min \\ Ax = g \\ x \geq 0\end{array}\right\} \tag{8}$$

cf.e.g. [10]. A necessary and sufficient optimality condition for this problem is

$$-\gamma \nabla f_1(x,\theta) - (1-\gamma)\nabla f_2(x,\theta) \in N_M(x) \tag{9}$$

by linear programming. Here,

$$\nabla f_1(x,\theta) = \left(\underline{c}(v,w) - \alpha(v,w)\theta + \delta(v,w)\right)_{(v,w)\in E}$$

and

$$\nabla f_2(x,\theta) = \left(\overline{c}(v,w) + \beta(v,w)\theta + \delta(v,w)\right)_{(v,w)\in E}$$

are independent of x. This implies that (9) is a system of linear equalities and inequalities in δ, y, t, γ. Hence, the projection of the solution set of this system onto the δ–space is a convex polyhedron.

Formula (9) can be used both to compute the bounds θ_i in (6) by setting $\delta \equiv 0$ and the dependency of θ_i from δ in a neighborhood of $\delta \equiv 0$. The θ_i are the bounds of θ for which x enters the set of Pareto-optimal solutions respectively leaves this set. Note, that this system is no longer linear if θ is not constant. This results in nonconvex regions of stability which is also reflected by the results in [3].

4 Robust optimization

In contrast to sensitivity analysis where the dependency of shortest paths on variations of the membership functions of the distances is investigated, robust optimization intends to find paths in G which are at the same time "equally good" with respect to all membership functions in a certain set [1]. For that, let P denote a set of all possible realizations of membership functions for the distances between the nodes of the graph G and assume that the membership functions in P are composed by the elements in a convex bounded polyhedron for simplicity. Hence, this polyhedron is given by

$$Q = \text{ conv } \left\{ \left(\underline{c}^k(v,w), \alpha^k(v,w), \overline{c}^k(v,w), \beta^k(v,w)\right)_{(v,w)\in E} : k = 1,\ldots,K \right\},$$

the convex hull of its K vertices

$$\left(\underline{c}^k(v,w), \alpha^k(v,w), \overline{c}^k(v,w), \beta^k(v,w)\right)_{(v,w)\in E}.$$

This results in the two-criterial optimization problem

$$\left.\begin{array}{l} z_1 \to \min \\ z_2 \to \min \\ \sum\limits_{(v,w)\in E} \left(\underline{c}(v,w) - \alpha(v,w)\theta\right) x(v,w) \le z_1 \;\forall\, (\underline{c},\alpha,\overline{c},\beta) \in Q \\ \sum\limits_{(v,w)\in E} \left(\overline{c}(v,w) + \beta(v,w)\theta\right) x(v,w) \le z_2 \;\forall\, (\underline{c},\alpha,\overline{c},\beta) \in Q \\ Ax = g \\ x \ge 0, \end{array}\right\} \tag{10}$$

where $(\underline{c},\alpha,\overline{c},\beta)$ is an abbreviation of the matrix

$$\Big(\underline{c}(v,w), \alpha(v,w)\overline{c}(v,w), \beta(v,w)\Big)_{(v,w)\in E}.$$

It is easy to see that the first and second group of inequalities in (10) are satisfied if and only if

$$\tilde{f}_1(x,\theta) := \max_{(\underline{c},\alpha,\overline{c},\beta)\in Q} \sum_{(v,w)\in E} \left(\underline{c}(v,w) - \alpha(v,w)\theta\right) x(v,w) \le z_1$$

and

$$\tilde{f}_2(x,\theta) := \max_{(\underline{c},\alpha,\overline{c},\beta)\in Q} \sum_{(v,w)\in E} \left(\overline{c}(v,w) + \beta(v,w)\theta\right) x(v,w) \le z_2.$$

For fixed θ linear functions are maximized over a convex bounded polyhedron which implies that the maximum is attained at a vertex of Q. Hence,

$$\tilde{f}_1(x,\theta) := \max_{k=1,\ldots,K} \sum_{(v,w)\in E} \left(\underline{c}^k(v,w) - \alpha^k(v,w)\theta\right) x(v,w) \le z_1$$

and

$$\tilde{f}_2(x,\theta) := \max_{k=1,\ldots,K} \sum_{(v,w)\in E} \left(\overline{c}^k(v,w) + \beta^k(v,w)\theta\right) x(v,w) \le z_2$$

which are convex, piecewise linear functions. Summing up, for computing a robust solution of the fuzzy linear optimization problem, the set of Pareto-optimal solutions of the following problem has to be determined:

$$\left.\begin{array}{l} \tilde{f}_1(x,\theta) = \max\limits_{k=1,\ldots,K} \sum\limits_{(v,w)\in E} \left(\underline{c}^k(v,w) - \alpha^k(v,w)\theta\right) x(v,w) \to \min \\ \tilde{f}_2(x,\theta) = \max\limits_{k=1,\ldots,K} \sum\limits_{(v,w)\in E} \left(\overline{c}^k(v,w) + \beta^k(v,w)\theta\right) x(v,w) \to \min \\ Ax = g \\ x \ge 0, \end{array}\right\} \tag{11}$$

To compute Pareto-optimal solutions for (11) the problem

$$\left.\begin{array}{l}\gamma\tilde{f}_1(x,\theta)+(1-\gamma)\tilde{f}_2(x,\theta)\to\min\\ Ax=g\\ x\geq 0,\end{array}\right\} \tag{12}$$

is solved for $\gamma\in[0,1]$. The following result is a consequence of convex (multicriterial) optimization [9, 10].

Theorem 2. *Let problem (12) has unique optimal solutions for* $\gamma=0$ *and* $\gamma=1$. *Then, an incidence vector* x^0 *of a path in* G *can have a positive membership function value for the robust fuzzy shortest path problem only if there is* $\theta\in[0,1]$ *such that*

$$0\in\gamma\partial f_1(x^0,\theta)+(1-\gamma)\partial f_2(x^0,\theta)+N_M(x^0).$$

Here, $\partial f_i(x^0,\theta)$ equals the subdifferential of the function $f_i(x^0,\theta)$.

Analogously to the proof of Theorem 1 this makes the computation of bounds $\{\theta_i\}_{i=1}^{2l}$ possible such that an incidence vector x^0 is Pareto-optimal for problem (11) for all $\theta\in\cup_{i=1}^{l}[\theta_{2i-1},\theta_{2i}]$. This implies that the membership function of such a point can be computed in a similar manner to (6).

References

1. A. Ben-Tal and A. Nemirovski. Robust solution of uncertain linear programs. *Operations Research Letters*, 25:1–13, 1999.
2. S. Chanas and D. Kuchta D. Multiobjective programming in optimization of interval objective functions - a generalized approach. *European Journal of Operations Research*, 94:594–598, 1996.
3. S. Chanas and D. Kuchta. Linear programming problem with fuzzy coefficients in the objective function. In M. Delgado, J. Kacprzyk, J.-L. Verdegay, and M. A. Vila, editors, *Fuzzy optimization*, pages 148–157. Physica Verlag, 1994.
4. S. Dempe and T. Starostina. Sensititvity analysis for linear optimization problem with fuzzy data in objective function. Technical Report Preprint 2003-09, TU Bergakademie Freiberg, Fakultät für Mathematik und Informatik, 2004. published electronically at `http://www.optimization-online.org/DB_HTML/2004/05/870.html`.
5. D. Dubois and H. Prade. Operations on fuzzy numbers. *Int. J. of Systems Science*, 6:613–62, 1978.
6. B. Korte and J. Vygen. *Combinatorial Optimization. Theory and Algorithms*. Berlin: Springer, 2000.
7. G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. New York: J. Wiley & Sons, 1988.
8. A. Orden. The transshipment problem. *Management Science*, 2:276–285, 1955-56.
9. R. T. Rockafellar. *Convex analysis*. Princeton: Princeton University Press, 1970.
10. S. Zlobec. *Stable Parametric Programming*. Dordrecht: Kluwer Academic Publishers, 2001.

Fuzzy Coloring of Fuzzy Hypergraph

L. Bershtein, A. Bozhenyuk, and I. Rozenberg

Scientific-technical center "Information technologies", Taganrog State University of Radioengineering Oktyabrskaya ploschad, 4, 347922, Taganrog Russia

1 Introduction

Practical tasks of map coloring in case of objects groups' allocation, not connected by any binary relation, come to the problem of coloring of graph [1]. This task is closely connected to the calculation of internal stable sets of graphs, calculation of chromatic number and a chromatic class of the graph.

Hypergraphs [2] are the generalization of graphs in case of set of multiarity relations. It means the expansion of graph models for the modeling complex systems.

In case of modeling systems with fuzzy binary and multiarity relations between objects, transition to fuzzy hypergraphs, which combine advantages both fuzzy and graph models, is more natural. It allows to realise formal optimisation and logical procedures. On the grounds of these, of practical interest are the tasks of various features of fuzzy hypergraphs, in particular, some tasks of coloring of fuzzy hypergraphs.

2 A Coloring of Fuzzy Hypergraph

Let a fuzzy hypergraph $\widetilde{H} = (X, \widetilde{E})$ be given, where $X = \{x_i\}$, $i \in I = \{1, 2, \ldots, n\}$ – is a finite set and $\widetilde{E} = \{\widetilde{e}_k\}$, $\widetilde{e}_k = \{< \mu_{e_k}(x)/x >\} k \in K = \{1, 2, \ldots, m\}$ is a family of fuzzy subsets in X. Thus elements of set X are the vertices of hypergraph, a family $\widetilde{E}$ is the family of fuzzy edges of hypergraph. The volue $\mu_{e_k}(x) \in [0, 1]$ is an incidence degree of a vertex x to an edge $\widetilde{e}_k$ [3].

A value $\mu(x, y) = \bigvee\limits_{e_k \in \widetilde{E}} \mu_{e_k}(x) \& \mu_{e_k}(y), x, y \in X$ is called an adjacent degree of two vertices x and y of fuzzy hypergraph $\widetilde{H}$.

There are at least two ways of coloring fuzzy hypergraphs, as they are a generalization of crisp hypergraphs (which value is $\mu_{e_k}(x) \in \{0, 1\}$) and fuzzy graph (its number of vertices x incident to one edge is no more than two).

L. Bershtein et al.: *Fuzzy Coloring of Fuzzy Hypergraph*, Advances in Soft Computing **2**, 703–711 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

In the first case, considering fuzzy hypergraph as crisp hypergraph generalization, and another one as fuzzy graph generalization, which has each edge appropriate only to two vertices, we come to the following concept of coloring hypergraph. K-coloring hypergraph H is a partition of set X into k-subsets so, that each edge of the hypergraph has crossing at least with two of these subsets [2]. In other words, all vertices of hypergraph, which incidence to one edge, should be painted not less than two colors. At such approach the task in work [3] of coloring fuzzy $\widetilde{H}$ is reduced to a determination of such a partial hypergraph H^{ri} of r_i level, which supposes k-coloring. And any hypergraph H^{rj} of r_j level, $(r_i < r_j)$ – does not suppose. In these work by hypergraph $H^{ri} = (X^{ri}, E)$ implied crisp hypergraph, for which the statement $(x \in X^{r_i}) \leftrightarrow (\exists \widetilde{e}_k \in \widetilde{E})[\mu_{e_k}(x) \geq r_i]$ is true.

Example 1. Let $\widetilde{H}$ be a fuzzy hypergraph which the incidence matrix is given by:

$$I = \begin{array}{c|ccc} & \widetilde{e}_1 & \widetilde{e}_2 & \widetilde{e}_3 \\ A & 1 & 0 & 0 \\ B & 0,9 & 0,7 & 0 \\ C & 0 & 0,8 & 0,7 \\ D & 0 & 0,4 & 0,6 \end{array}.$$

Partial hypergraphs H^{rj} are $\mathrm{H}^1 = (\{\mathrm{A}\}, \{\{\mathrm{A}\}\}), \mathrm{H}^{0,9} = (\{\mathrm{A}, \mathrm{B}\}, \{\{\mathrm{A}, \mathrm{B}\}\})$, $\mathrm{H}^{0,8} = (\{\mathrm{A}, \mathrm{B}, \mathrm{C}\}, \{\{\mathrm{A}, \mathrm{B}\}, \{\mathrm{C}\}\}), \mathrm{H}^{0,7} = (\{\mathrm{A}, \mathrm{B}, \mathrm{C}\}, \{\{\mathrm{A}, \mathrm{B}\}, \{\mathrm{B}, \mathrm{C}\}, \{\mathrm{C}\}\})$, $\mathrm{H}^{0,6} = (\{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}\}, \{\{\mathrm{A}, \mathrm{B}\}, \{\mathrm{B}, \mathrm{C}\}, \{\mathrm{C}, \mathrm{D}\}\})$ and $\mathrm{H}^{0,4} = (\{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}\}, \{\{\mathrm{A}, \mathrm{B}\}, \{\mathrm{B}, \mathrm{C}, \mathrm{D}\}, \{\mathrm{C}, \mathrm{D}\}\})$. A chromatic number of fuzzy hypergraph $\widetilde{H}$ equals 2 because all vertices suppose coloring by two colors (vertices A and D are colored by first color and B and C are colored by second color).

In this paper we consider the second case. Considering fuzzy hypergraph as a generalization of fuzzy graph, we come to a hypergraph coloring as to partition of set X on k subsets so, that all hypergraph vertices, which appropriate to one edge, should be colored in different colors. There are two approaches to these problem. At the first approach to each vertex of fuzzy hypergraph is attributed one color. Such approach termed fuzzy coloring of first order. At the second approach to each vertex of fuzzy hypergraph is attributed the maximum possible number of colors (not exceeding n -number). Such approach termed fuzzy coloring of second order.

2.1 Fuzzy Coloring of First Order

This task of fuzzy coloring is reduced to a determination of maximum possible separation degree similar to fuzzy graph at which fuzzy hypergraphs supposes k-painting. For formalization of such approach we shall give the following definitions. Let $X' \subseteq X$ - any subset vertices of fuzzy hypergraph $\widetilde{H}$.

Definition 1. *An internal stability degree of vertices subset X' of fuzzy hypergraph $\widetilde{H}$ we shall name $a_{X'} \in [0,1]$, determined as:*

$$a_{X'} = 1 - \max_{x,y \in X'} \mu(x,y) \, .$$

Definition 2. *Subset $X' \subseteq X$ is called a maximal fuzzy internally stable set with the degree of internal stability $a_{X'}$, if the statement $(\forall X'' \supseteq X')(a_{X''} < a_{X'})$ is true.*

Let's paint each vertex $x \in X$ of hypergraph $\widetilde{H}$ in one of k colors ($l \leq k \leq n$) and we shall consider a X_i, subset of vertices, colored identically.

Definition 3. *The value $L_I = \underset{i \equiv 1,k}{\&} \alpha_i = \underset{i \equiv 1,k}{\&} (1 - \underset{x,y \in X_i}{\vee} \mu(x,y))$ is called a separation degree of first order of fuzzy hypergraph $\widetilde{H}$ at its k-colorings.*

Fuzzy hypergraph $\widetilde{H}$ can be colored in any number of k colors and thus separation degree L_I depends on their number. Fuzzy hypergraph $\widetilde{H}$ we shall put in conformity family of fuzzy sets $\Re = \{\widetilde{A}_{\widetilde{H}}\}, \widetilde{A}_{\widetilde{H}} = \{< L(k)/k | k = \overline{1,n}\}$, where $L(k)$ determines a separation degree of fuzzy hypergraph $\widetilde{H}$ at its certain k - coloring.

Definition 4. *Fuzzy set $\widetilde{\gamma}_I = \{< L_{\widetilde{\gamma}}(k)/k | k = \overline{1,n}\}$ we shall name fuzzy chromatic set of the first order of hypergraph $\widetilde{H}$, if for any other set $\widetilde{A}_{\widetilde{H}} \in \Re$, it is true $\widetilde{A}_{\widetilde{H}} \subseteq \widetilde{\gamma}_I$.*

In other words, $(\forall \widetilde{A}_{\widetilde{H}} \in \Re)(\forall k = \overline{1,n})[L(k) \leq L_I(k)]$. Or, otherwise, fuzzy chromatic set of the first order of hypergraph $\widetilde{H}$ determines the greatest separation degrees at coloring its tops in one of $1, 2 \ldots$ n colors.

The following algorithm is proposed for finding fuzzy chromatic set:

1°. For fuzzy hypergraph $\widetilde{H} = (X, \widetilde{E})$ it is determined fuzzy vertex graph $\widetilde{\mathrm{G}} = (\mathrm{X}, \widetilde{\mathrm{U}})$, which set of vertices coincides with set of vertices of initial hypergraph. And the fuzzy set of nondirectional edges is determined as $\widetilde{U} = \{< \mu(x,y)/(x,y) >\}$, where value $\mu(x,y)$ is an adjacent degree of vertices x and y which is calculated as stated above.

2°. For fuzzy graph $\widetilde{G}$ we determine class of all maximal fuzzy internally stable sets $\{\Psi_1, \Psi_2, \ldots, \Psi_r\}$ with degrees of internal stability $\alpha_1, \alpha_2, \ldots, \alpha_r$.[4].

3°. From set $\{\Psi_1, \Psi_2, \ldots, \Psi_r\}$ it is determined such sample k internally stable sets $\{\Psi_1, \Psi_2, \ldots, \Psi'_k\}$, for which it would be carried out $\bigcup\limits_{\mathrm{j}=1,\mathrm{k}'} \Psi_{\mathrm{j}} = \mathrm{X}$, and a value $\min\{\alpha'_1, \alpha'_2, \ldots, \alpha'_{\mathrm{k}}\}$ accepted the maximum possible value. Thus value $L_I(k) = \min\{\alpha'_1 \alpha'_2, \ldots, \alpha'_k\}$.

4°. The step 3° is repeated for all values $\mathrm{k} = 1, \mathrm{n} - 1$.

Example 2. We consider the example of finding of first order fuzzy chromatic set. Let $\widetilde{H}$ be a fuzzy hypergraph which the incidence matrix is given by:

$$I = \begin{array}{c|ccccc} & \widetilde{e}_1 & \widetilde{e}_2 & \widetilde{e}_3 & \widetilde{e}_4 & \widetilde{e}_5 \\ x_1 & 0,8 & 0,5 & 0 & 0 & 0 \\ x_2 & 1 & 0 & 0 & 0 & 0 \\ x_3 & 0,4 & 1 & 0,3 & 0,7 & 0 \\ x_4 & 0 & 0,6 & 0,4 & 0,2 & 1 \\ x_5 & 0 & 0 & 0,7 & 1 & 0 \\ x_6 & 0 & 0 & 0 & 0 & 0,4 \end{array} .$$

We determine an adjacent degree of vertices by:

$$\mu(x_1, x_2) = \vee_{e_k \in \widetilde{E}} \mu_{e_k}(x_1) \& \mu_{e_k}(x_2) = 0,8 \ \& \ 1 = 0,8;$$
$$\mu(x_1, x_3) = \vee_{e_k \in \widetilde{E}} \mu_{e_k}(x_1) \& \mu_{e_k}(x_3) = 0,8 \& 0,4 \vee 0,5 \ \& \ 1 = 0,5; \ldots;$$
$$\mu(x_4, x_5) = \vee_{e_k \in \widetilde{E}} \mu_{e_k}(x_4) \& \mu_{e_k}(x_5) = 0,4 \ \& \ 0,7 \vee 0,2 \ \& \ 1 = 0,4 \ .$$

Corresponding fuzzy vertex graph $\widetilde{\mathrm{G}} = (\mathrm{X}, \widetilde{\mathrm{U}})$ is presented in Fig.1:

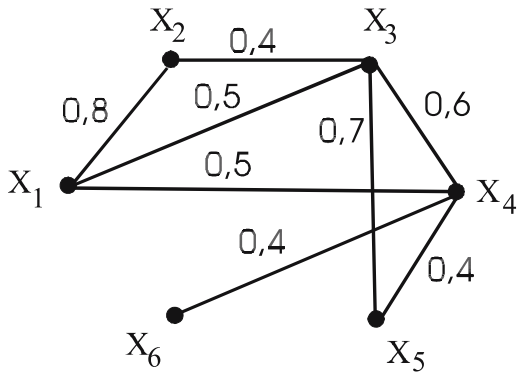


Fig. 1. Fuzzy vertex graph of fuzzy hypergraph $\widetilde{H}$

We will determine all maximal internal stable sets with the highest degree of internal stable by the method, which is considered in [4].

Let Ψ be a certain maximal internal stable set with the degree of internal stable $\alpha(\Psi)$. For arbitrary vertices $x_i, x_j \in X$, one of the following cases may be realised: a) $x_i \notin \Psi$; b) $x_j \notin \Psi$; c) $x_i \in \Psi$ and $x_j \in \Psi$. In the last case the degree $\alpha(\Psi) \leq 1 - \mu_U(x_i, x_j)$.
In other words, the following expression is true:

$$(\forall x_i, x_j \in X)[x_i \notin \Psi \vee x_j \notin \Psi \vee (\alpha(\Psi) \leq 1 - \mu_U(x_i, x_j))] \ . \qquad (1)$$

We connect a Boolean variable p_i taking 1 when $x_i \in \Psi$ and 0 when $x_i \notin \Psi$, with each vertex $x_i \in X$. We associate the expression $\alpha(\Psi) \leq 1 - \mu_U(x_i, x_j)$ with a fuzzy variable $\xi_{ij} = 1 - \mu_U(x_i, x_j)$.

Considering the expression (1) for all possible values i and j we obtain the truth of the following expression:

$$\Phi_\Psi = \underset{i}{\&}\underset{j\neq i}{\&}(\overline{p}_i \vee \overline{p}_j \vee \xi_{ij}) = 1 \ .$$

We open the parentheses and reduce the similar terms using the following rule:

$$\xi' \& \, a \vee \xi'' \& \, a \& b = \xi' \ \& \, a, \text{ for } \xi' \geq \xi'' \ . \tag{2}$$

Here, $a, b \in \{0, 1\}$ and $\xi', \xi'' \in [0,1]$.

Then for each disjunctive term, the totality of all vertices corresponding to the variables missing in the totality, gives a maximal internal stable set with the obtained degree of internal stable.
For the fuzzy vertex graph $\widetilde{\mathrm{G}}$ presented in Fig. 1, we obtain:

$$\begin{aligned}\Phi_\Psi = {} & (\overline{p}_1 \vee \overline{p}_2 \vee 0,2)\&(\overline{p}_1 \vee \overline{p}_3 \vee 0,5)\&(\overline{p}_1 \vee \overline{p}_4 \vee 0,5)\&(\overline{p}_2 \vee \overline{p}_3 \vee 0,6)\& \\ & \&(\overline{p}_3 \vee \overline{p}_4 \vee 0,4)\&(\overline{p}_3 \vee \overline{p}_5 \vee 0,3)\&(\overline{p}_4 \vee \overline{p}_5 \vee 0,6)\&(\overline{p}_4 \vee \overline{p}_6 \vee 0,6) \ .\end{aligned}$$

Completing the transformations of the fuzzy logical variables by (2), we finally have:

$$\begin{aligned}\Phi_\Psi = {} & \overline{p}_1\overline{p}_3\overline{p}_4 \vee \overline{p}_1\overline{p}_3\overline{p}_5\overline{p}_6 \vee \overline{p}_2\overline{p}_3\overline{p}_4 \vee \overline{p}_1\overline{p}_2\overline{p}_4\overline{p}_5 \vee 0,6\overline{p}_1\overline{p}_3 \vee 0,6\overline{p}_1\overline{p}_4\overline{p}_5 \vee \\ & \vee 0,5\overline{p}_2\overline{p}_3 \vee 0,5\overline{p}_2\overline{p}_4\overline{p}_5 \vee 0,4\overline{p}_1\overline{p}_5 \vee 0,4\overline{p}_2\overline{p}_5 \vee 0,3\overline{p}_1 \vee 0,3\overline{p}_2 \vee 0,2 \ .\end{aligned}$$

It follows from the last expression that the considered fuzzy vertex graph $\widetilde{G}$ has 13 maximal internal stable sets:

$-\Psi_1 = \{x_2, x_5, x_6\}, \Psi_2 = \{x_2, x_4\}, \Psi_3 = \{x_1, x_5, x_6\}, \Psi_4 = \{x_3, x_6\}$, with the degrees of internal stability $\alpha(\Psi_1) = \alpha(\Psi_2) = \alpha(\Psi_3) = \alpha(\Psi_4) = 1$;

$-\Psi_5 = \{x_2, x_4, x_5, x_6\}$ and $\Psi_6 = \{x_2, x_3, x_6\}$ with the degrees of internal stability $\alpha(\Psi_5) = \alpha(\Psi_6) = 0,6$;

$-\Psi_7 = \{x_1, x_4, x_5, x_6\}$ and $\Psi_8 = \{x_1, x_3, x_6\}$ with the degrees of internal stability $\alpha(\Psi_7) = \alpha(\Psi_8) = 0,5$;

$-\Psi_9 = \{x_2, x_3, x_4, x_6\}$ and $\Psi_{10} = \{x_1, x_3, x_4, x_6\}$ with the degrees of internal stability $\alpha(\Psi_9) = \alpha(\Psi_{10}) = 0,4$;

$-\Psi_{11} = \{x_2, x_3, x_4, x_5, x_6\}$ and $\Psi_{12} = \{x_1, x_3, x_4, x_5, x_6\}$ with the degrees of internal stability $\alpha(\Psi_{11}) = \alpha(\Psi_{13}) = 0,3$;

$-\Psi_{13} = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ (all set X) with the degree of internal stability $\alpha(\Psi_{13}) = 0,2$.

We construct the matrix $\mathrm{R} = \left\|\mathrm{r_{ij}}\right\|, \mathrm{i} = \overline{1,\mathrm{n}}, \mathrm{j} = \overline{1,\mathrm{t}}$ where the lines correspond to the vertices of graph $\widetilde{\mathrm{G}}$ and columns correspond to the maximal internal stable sets. If $\mathrm{x_i} \in \Psi_\mathrm{j}$, then the value $\mathrm{r_{ij}}$ has the value α_j, if $\mathrm{x_i} \notin \Psi_\mathrm{j}$, then the value $\mathrm{r_{ij}}$ has the value 0. For the fuzzy vertex graph $\widetilde{\mathrm{G}}$ presented in Fig.1, we obtain the table 1.

The task of finding fuzzy chromatic set $\widetilde{\gamma}$ is the task of finding the covering of all lines by k columns ($\mathrm{k} = \overline{1, \mathrm{n}-1}$) with the maximum of the volume $\min\{\alpha_{\mathrm{i}_1}, \alpha_{\mathrm{i}_2}, \ldots, \alpha_{\mathrm{i}_\mathrm{k}}\}$.

Table 1. Vertex entering to internal stable sets

	Ψ_1	Ψ_2	Ψ_3	Ψ_4	Ψ_5	Ψ_6	Ψ_7	Ψ_8	Ψ_9	Ψ_{10}	Ψ_{11}	Ψ_{12}	Ψ_{13}
x_1	0	0	1	0	0	0	0,5	0,5	0	0,4	0	0,3	0,2
x_2	1	1	0	0	0,6	0,6	0	0	0,4	0	0,3	0	0,2
x_3	0	0	0	1	0	0,6	0	0,5	0,4	0,4	0,3	0,3	0,2
x_4	0	1	0	0	0,6	0	0,5	0	0,4	0,4	0,3	0,3	0,2
x_5	1	0	1	0	0,6	0	0,5	0	0	0	0,3	0,3	0,2
x_6	1	0	1	1	0,6	0,6	0,5	0,5	0,4	0,4	0,3	0,3	0,2

We write the expression (5) for finding of such covering:

$$\Phi_C = \underset{l=1,n}{\&}(r_{l1}\&\Psi_1 \vee r_{l2}\&\Psi_2 \vee \ldots \vee r_{lt}\&\Psi_t)\ . \tag{3}$$

Here, t is number of columns in the table 1.

We open the parentheses and reduce the similar terms using the rule (2). Then we can rewrite the expression (3) as:

$$\Phi_C = \underset{l=1,m}{\vee}(\Psi_{1i}\&\Psi_{2i}\&\ldots\&\Psi_{ki}\&r_i)\ .$$

Here, $r_i \in [0,1], \Psi_{ji} \in [\Psi_1, \Psi_2, \ldots, \Psi_l], l \in \overline{1,n}$.

So, for the fuzzy vertex graph $\widetilde{\mathrm{G}}$ presented in Fig. 1, the expression Φ_C is defined as:

$$\begin{aligned}\Phi_C = {} & (\Psi_3 \vee 0,5\Psi_7 \vee 0,5\Psi_8 \vee 0,4\Psi_{10} \vee 0,3\Psi_{12} \vee 0,2\Psi_{13})\& \\ & \&(\Psi_1 \vee \Psi_2 \vee 0,6\Psi_5 \vee 0,6\Psi_6 \vee 0,4\Psi_9 \vee 0,3\Psi_{11} \vee 0,2\Psi_{13})\& \\ & \&(\Psi_4 \vee 0,6\Psi_6 \vee 0,5\Psi_8 \vee 0,4\Psi_9 \vee 0,4\Psi_{10} \vee 0,3\Psi_{11} \vee 0,3\Psi_{12} \vee 0,2\Psi_{13})\& \\ & \&(\Psi_2 \vee 0,6\Psi_5 \vee 0,5\Psi_7 \vee 0,4\Psi_9 \vee 0,4\Psi_{10} \vee 0,3\Psi_{11} \vee 0,3\Psi_{12} \vee 0,2\Psi_{13})\& \\ & \&(\Psi_1 \vee \Psi_3 \vee 0,6\Psi_5 \vee 0,5\Psi_7 \vee 0,3\Psi_{11} \vee 0,3\Psi_{12} \vee 0,2\Psi_{13})\& \\ & \&(\Psi_1 \vee \Psi_3 \vee \Psi_4 \vee 0,6\Psi_5 \vee 0,6\Psi_6 \vee 0,5\Psi_7 \vee 0,5\Psi_8 \vee 0,4\Psi_9 \vee 0,4\Psi_{10} \vee \\ & \vee 0,3\Psi_{11} \vee 0,3\Psi_{12} \vee 0,2\Psi_{13})\ .\end{aligned}$$

Using the rule (2), we finally have:

$$\Phi_C = 0,2\&\Psi_{13} \vee 0,5\&\Psi_6\&\Psi_7 \vee \Psi_2\&\Psi_3\&\Psi_4 \vee \ldots\ .$$

From the received expression follows:

If $k = 1$, then the covering defines the column Ψ_{13} by the degree $L_I(1) = 0,2$. If $k = 2$, then the covering defines the columns Ψ_6 and Ψ_7 by the degree $L_I(2) = 0,5$. If $k = 3$, then the covering defines the columns Ψ_2, Ψ_3 and Ψ_4 by the degree $L_I(3) = 1$.

So, the fuzzy chromatic set for the fuzzy hypergraph is

$$\widetilde{\gamma}_I = \{< 0,2/1 >, < 0,5/2 >, < 1/3 >\}\ .$$

Otherwise, the fuzzy hypergraph may be colored by one color with the degree of separation 0,2; by 2 colors with the degree of separation 0,5 (vertices x_2, x_3 and x_6 - first color, vertices $x_1, x_4 H x_5$ - second color); by 3 colors with the degree of separation 1 (vertices x_2, and x_4 - first color, vertices x_1, x_5 and x_6 - second color, vertex x_3 - third color).

2.2 Fuzzy coloring of second order

Fuzzy coloring of second order can be formulated as follows: to appropriate to every vertex of fuzzy hypergraph $\widetilde{H}$ the maximum possible number of colors from n possible so that they had the greatest separation degree on each color.

Let's attribute to each vertex $x \in X$ of the hypergraph k from n colors. Let $X_{K1}, X_{K2}, \ldots, X_{Kn}$ - subsets of vertices by which the first, the second,..., the n-th - colors are attributed with a degree of internal stability $\alpha_{K1}, \alpha_{K2}, \ldots, \alpha_{Kn}$ accordingly. It is obvious, that the given subsets form some covering of a subset of vertices X, and the general number of such coverings equally to number of combinations $C(n,k) = \frac{n!}{k! \times (n-k)!}$.

Definition 5. *Value* $L_K = \underset{i \overline{\equiv} 1,n}{\&} \alpha_{Ki}$ *we shall name a separation degree of the second order of fuzzy hypergraph* $\widetilde{H}$ *at assignment to its vertices* k *from* n *colors.*

Definition 6. *Fuzzy chromatic set of the second order we shall name set* $\widetilde{\gamma}_{II}, = \{< L_{II}(k)/k | k = \overline{1,n}\}$ *in which values* $L_{II}(k)$ *determine the greatest separation degrees of vertices the hypergraph at assignment of each of them* k *from* n *colors.*

Let's note some properties of fuzzy chromatic set of second order.

Property 1. A degree $L_{II}(1) = 1$.

Differently, at assignment of each vertex of fuzzy hypergraph $\widetilde{H}$ of one color, a partibility degree of the second order is equal to each vertex 1.

Property 2. A degree $L_{II}(n) \geq 0$, and

$$(L_{II}(n) = 0) \leftrightarrow (\exists x, y \in X)(\exists e_i \in \widetilde{E} | \mu_i(x) = 1 \& \mu_i(y) = 1) \ .$$

Differently, at assignment of each vertex of fuzzy hypergraph $\widetilde{H}$ it is equal n colors, a separation degree of the second order is equal 0 if there are even two vertices, incidental to the same edge with degree 1.

Property 3. The statement is true:

$$(\forall i, j = \overline{1,n})(i > j \rightarrow L_{II}(i) \leq L_{II}(j)) \ .$$

Differently, more colors is appropriated to each vertex, less separation degree of the second order.

Property 4. The statement is true:

$$(\forall i,j,k=\overline{1,n})[(i>k>j)\&(L_{II}(i)=L_{II}(j))\rightarrow(L_{II}(k)=L_{II}(i)=L_{II}(j))]\ .$$

Differently, if at "optimum" coloring of fuzzy hypergraph $\widetilde{H}$ in i and j colors a separation degree of the second order is identical, than at each vertex the of the hypergraph of any other number of colors, laying between values i or j, the separation degree will not change.

Property 5. The statement is true:

$$(L_{II}(i)=1\&i\neq 1)\longleftrightarrow(\forall k\in\overline{1,i-1})(L_{II}(k)=1)\ .$$

Differently, if at assignment of each vertex of the hypergraph of some number $i(i\neq 1)$ of colors the separation degree of second order is equal to and at assignment of any other number of colors, smaller than i, the separation degree of second order also is equal to 1.

Property 6. The statement is true:

$$(L_{II}(i)=0\&i\neq n)\leftrightarrow(\forall k\in\overline{i+1,n})(L_{II}(k)=0).$$

Differently, if at assignment of each vertex of the hypergraph of some number $i(i\neq n)$ of colors the separation degree of second order is equal to 0, more than i, also does not exist coloring with a separation degree of second order distinct from 0.

Let's consider correlation between fuzzy chromatic set of first and second order. Let's denote through $t=]\frac{n}{k}[$ - the whole from division of value n on value k, then the following property is true:

Property 7. $(\forall k=\overline{1,n})(L_{II}(k)=L_I(t))$.

Differently, the separation degree of the second order at assignment to each vertex of the hypergraph of k- colors coincides with a separation degree of first order at coloring each vertex of the hypergraph in one of $t=]\frac{n}{k}[$ colors. The given property allows calculating fuzzy chromatic set of the second order on fuzzy chromatic set of the first order.

Example 3. We consider an example of finding of second order fuzzy chromatic set. Taking into account, the first order fuzzy chromatic set, we receive $\tilde{\gamma}_{II}=\{<1/1>,<1/2>,<0,5/3>,<0,2/4>,<0,2/5>,<0,2/6>\}$.

Otherwise, the vertices of fuzzy hypergraph $\widetilde{H}$ may be appropriated 2 colors by the degree of separation 1; 3 colors by the degree of separation 0,5; 4, 5 or 6 colors by the degree of separation 0,2.

3 Conclusion

It is necessary to note that the considered statements of tasks of coloring fuzzy hypergraph H on the basis of minimax criteria of determination of fuzzy chromatic sets are not unique. Tasks of coloring fuzzy hypergraph by other criteria also are of interest, in particular, on the basis of determination of average fuzzy chromatic sets of the first or second order.

Acknowledgements

This work has been supported by the Russian Foundation for Basic Research project $\mathcal{N}_{\underline{o}}$ 03–07–90202.

References

1. Kaufmann A. (1977) Introduction a la theorie des sous-ensemles flous. Masson, Paris
2. Berge C. (1989) Hypergraphs: combinatorics of finite sets. Elsevier Science Publishers
3. Monderson J.N., Nair P.S. (2000) Fuzzy graphs and fuzzy hypergraphs. Heidelberg; New-York: Physica-Verl
4. Bershtein L.S., Bozhenuk A.V. (2001) Maghout method for determination of fuzzy independent, dominating vertex sets and fuzzy graph kernels. J. General Systems 30: 45–52

Nonlinear Optimization with Fuzzy Constraints by Multi-Objective Evolutionary Algorithms

F. Jiménez[1], G. Sánchez[1], J.M. Cadenas[1], A.F. Gómez-Skarmeta[1], and J.L. Verdegay[2]

[1] Dept. Ingeneiría de la Información y las Comunicaciones, Facultad de Informática, Universidad de Murcia, Spain
fernan,gracia,jcadenas,skarmeta@dif.um.es

[2] Dept. Ciencias de la Computación e Inteligencia Artificial, E.T.S.I.I., Universidad de Granada, Spain
verdegay@ugr.es

1 Introduction

Fuzzy constrained optimization problems have been extensively studied since the seventies. In the linear case, the first approaches to solve the so-called fuzzy linear programming problem were made in [12] and [15]. Since then, important contributions solving different linear models have been done and these models have been recipients of a great dealt of work. In the nonlinear case the situation is quite different, as there is a wide variety of specific and both practical and theoretically relevant nonlinear problems, each having a different solution method. In the following we consider a *Nonlinear Programming* problem with fuzzy constraints. From a mathematical point of view the problem can be addressed as:

$$\begin{array}{l} Min\ f(\boldsymbol{x}) \\ s.t.: g_j(\boldsymbol{x}) \lesssim b_j, \quad j = 1, \ldots, m \\ \qquad x_i \in [l_i, u_i], \quad i = 1, \ldots, n,\ l_i \geq 0 \end{array} \tag{1}$$

where $\boldsymbol{x} = (x_1, \ldots, x_n) \in \Re^n$ is a n dimensional real-valued parameter vector, $[l_i, u_i] \subset \Re$, $b_j \in \Re$, $f(\boldsymbol{x})$, $g_j(\boldsymbol{x})$ are arbitrary functions, and the symbol $\lesssim$ indicates a fuzzy constraint [15]. Here we will consider the following linear membership function related to each fuzzy constraint:

$$\mu_j(\boldsymbol{x}) = \begin{cases} 0 & \text{if } g_j(\boldsymbol{x}) \geq b_j + d_j \\ h\left(\frac{b_j + d_j - g_j(\boldsymbol{x})}{d_j}\right) & \text{if } b_j \leq g_j(\boldsymbol{x}) \leq b_j + d_j \\ 1 & \text{if } g_j(\boldsymbol{x}) \leq b_j \end{cases} \tag{2}$$

which gives the accomplishment degree of $g_j(\boldsymbol{x})$, and consequently of $\boldsymbol{x}$, with respect to the j-th constraint (the decision maker can tolerate violations of each constraint up to the value $b_j + d_j$, $j = 1, \ldots, m$). We assume that the

F. Jiménez et al.: *Nonlinear Optimization with Fuzzy Constraints by Multi-Objective Evolutionary Algorithms*, Advances in Soft Computing **2**, 713–722 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

function h is a arbitrary function which allows to represent accurately the accomplishment degree.

Then using the results obtained in [14], the problem (1) can be easily transformed into a parametric programming problem as follows:

$$\begin{array}{l} Min\ f(\boldsymbol{x}) \\ s.t.: g_j(\boldsymbol{x}) \leq b_j + d_j\left(1 - h^{-1}(\alpha)\right),\ j = 1, \ldots, m \\ \qquad x_i \in [l_i, u_i],\quad i = 1, \ldots, n,\ l_i \geq 0 \end{array} \tag{3}$$

where $\alpha \in [0, 1]$, and h^{-1} is the inverse function of h.

A fuzzy solution to (1), if there is any, may be defined as the fuzzy set of membership function:

$$\mu_s(\boldsymbol{x}) = \begin{cases} \sup\limits_{\boldsymbol{x} \in S(\alpha)} \alpha \text{ if } \boldsymbol{x} \in \bigcup\limits_{\alpha} S(\alpha) \\ 0 \qquad\qquad \text{elsewhere} \end{cases} \tag{4}$$

where:

$$S(\alpha) = \left\{ \boldsymbol{x} \in \Re^n / z(\boldsymbol{x}) = \min_{\boldsymbol{x}' \in X(\alpha)} f(\boldsymbol{x}') \right\}$$

with:

$X(\alpha) = \left\{ \boldsymbol{x} \in \Re^n / g_j(\boldsymbol{x}) \leq b_j + d_j\left(1 - h^{-1}(\alpha)\right) \right\}$
$x_i \in [l_i, u_i],\ i = 1, \ldots, n$
$d_j \geq 0,\ j = 1, \ldots, m,\ and$
$\alpha \in [0, 1]$.

Unfortunately, there are no much general-oriented solution methods to solve nonlinear parametric programming problems in the literature, although it deserves to mention the cases of linear programming problems in which data are continuously varied as a linear function of a single parameter. Therefore in order to theoretically solve (3) we shall try to find an approximate solution. It is patent that *Evolutionary Algorithms* (EA) [1, 6, 10] could be used to solve fuzzy nonlinear programming problems like the above one because of EA are solution methods potentially able of solving general nonlinear programming problems or, at least, of approaching theoretic solution ways that, each case, are to be specified according to the concrete problem to be solved. An evolutionary-parametric based approach to solve fuzzy transportation problems have been proposed in [7]. In [9] the problem (3) is solved for a finite set of values of the parameter α by means of an EA for constrained nonlinear optimization problems. Final solution is constructed with numerical approximation techniques. The main disadvantage of this approach arises in the need for run an EA for each value of the parameter α. Moreover, numerical approximation does not ensures the feasibility of solutions. In this paper we propose a multi-objective approach to solve the problem (3).

With this background, the paper have been organized as follows: in section 2 a multi-objective technique for parametric programming problems is approached, section 3 describes an ad-hoc Pareto-based multi-objective EA

to solve the multi-objective problems connected with the parametric programming problems, in section 4 a nonlinear fuzzy problem is considered as case study and results of experiments are shown in section 5. Finally section 6 indicates the main conclusions and future research.

2 A multi-objective approach for nonlinear parametric programming problems

In this section we propose a multi-objective approach to solve the problem (3). The problem (3) can be transformed into a two-objective nonlinear programming problem in which the parameter α is treated as a new decision variable in the constrains and as a second objective to maximize.

It is clear that, $\forall \alpha, \alpha' \in [0,1]$, $\alpha \geq \alpha'$, $X(\alpha) \subseteq X(\alpha')$, and then, $S(\alpha) \subseteq S(\alpha')$. Consequently, $S(0) = \{\min_{\alpha \in [0,1]} S(\alpha)\}$. By confronting the objectives $f(\boldsymbol{x})$ and α, i.e., minimizing $f(\boldsymbol{x})$ and maximizing α, we obtain a set of non-dominated solutions (*Pareto*) in which the Pareto front represents the better values of $f(\boldsymbol{x})$ for each value of the parameter α.

The multi-objective problem is stated as follows:

$$\begin{array}{l} Min\ f(\boldsymbol{x}) \\ Max\ \alpha \\ s.t.:\ g_j(\boldsymbol{x}) \leq b_j + d_j\left(1 - h^{-1}(\alpha)\right),\ j = 1, \ldots, m \\ \qquad x_i \in [l_i, u_i],\ i = 1, \ldots, n, \alpha \in [0,1] \end{array} \tag{5}$$

3 A Pareto-based multi-objective evolutionary algorithm

Multi-objective Pareto-based EA [2, 5, 8] are specially appropriated to solve multi-objective nonlinear optimization problems because they can capture a set of Pareto solutions in a single run of the algorithm. We propose an ad hoc multi-objective Pareto-based EA to solve the problem (5) with the following characteristics:

- Pareto-based multi-objective EA; it finds, in a single run, multiple non-dominated solutions.
- The EA has a real-coded representation. Each individual of a population contains $n+1$ real parameters to represent the solution $(x_1, \ldots, x_n, \alpha)$.
- The initial population is generated randomly with a uniform distribution within the boundaries of the search space $x_i \in [l_i, u_i]$, $i =, 1, \ldots, n$, $\alpha \in [0,1]$.
- The variation operators act on real numbers. It has been used two cross types, *uniform cross* and *arithmetical cross*, and three types of mutation, *uniform mutation*, *non-uniform mutation*, *minimal mutation* [8].

- Diversity among individuals is maintained by using an ad-hoc elitist generational replacement technique.
- It uses the $min - max$ formulation to handle constrains.

3.1 Constraint handling

The populations generated by the algorithm are made up of both feasible and unfeasible individuals. Guided by the multi-objective optimization Pareto concept, the feasible individuals evolve towards optimality, while the non-feasible individuals evolve towards feasibility guided by an evaluation function based on the *min-max* formulation. See below for details. Thus the resulting algorithm is weakly dependent on the problem to be optimized since it is the evolutionary heuristics itself that is used to satisfy the constrains, unlike the repair, decoding or penalty techniques which tend to be heavily dependent on the problem.

3.2 Variation operators

Bearing in mind that the EA uses a floating point representation and given the coexistence of feasible and unfeasible individuals within the EA populations, the variation operators therefore act on chains (sequences) of real numbers without any consideration regarding the feasibility of new descendants. After experimenting for real parameter optimization with different variation operators proposed in the literature and with others, it was finally decided to use two cross types, *uniform cross* and *arithmetical cross*, and three types of mutation, *uniform mutation*, *non-uniform mutation* and *minimal mutation*. The first four have been studied and described in depth by other authors [11]. Minimal mutation causes a minimal change in the descendant as compared to the father, and it is especially appropriate in fine tuning real parameters [8].

3.3 Generating a new population

The algorithm performs the following steps in the generation of a new population:

1. Two random individuals are selected.
2. Two offspring are obtained by parent crossing, mutation and repair.
3. The offspring are inserted into the population.

The insertion of the offspring is the fundamental point for maintaining diversity. We use an ad hoc technique for insertion. Objectives space is distributed into $D = N/2$ slots, where N is the population size. The insertion of an individual $X = (x_1, \ldots, x_n, \alpha)$ is performed as follows:

- Calculate the slot t the individual belongs to according with the following expression $t = \lceil \alpha D \rceil$.

- If individual is better than some individual in slot t, then replace the worse individual in slot t by the new individual.

In order to determine if an individual is better than another, the following criteria are established:

- A feasible individual is better than another unfeasible one.
- One unfeasible individual is better than another one if its function:

$$\max_{j=1,\ldots,m} \left\{ g_j(\boldsymbol{x}) - b_j - d_j \left(1 - h^{-1}(\alpha)\right) \right\}$$

  is better.
- One feasible individual is better than another one if the first dominates the second.

It should be observed that we are using the *min-max* formulation to satisfy the constrains. This method has been used in multi-objective optimization [4] to minimize the relative deviations of each objective function from its individual optimum, and the best compromise solution can be obtained when objectives of equal priority are optimized. Since constrains and objectives can be treated in a similar way, and it is assumed that all constrains have equal priority, the *min-max* formulation is appropriate for satisfying constrains and is, furthermore, a technique which is independent of the problem.

It should also be noted that insertion of the new individuals is not always carried out, but only in those cases in which the new individual is better than the individual replaced and the diversity is not worsened in any case. Thus the technique simultaneously permits optimization and conservation of the diversity. It is also an elitist technique, since an individual is only replaced by another individual which is better than itself.

4 A case study

In this section we set out a nonlinear fuzzy optimization problem as case study which describes a possible situation in a exporting company. The problem is the following:

Two products for export A and B are to be produced by utilizing three different processes (p_1, p_2 and p_3). The production of one unit of product A (B) requires 10 (6) minutes of processing time in the p_1 department, 5 (10) minutes in the p_2 department, and 7 (10) minutes in the p_3 department. The total time available for each production process is 2500 minutes for p_1, at the most 2000 minutes for p_2 (although violations up to 2064 minutes are permitted) and at most 2050 minutes for p_3 (depending on the urgency of sending), but this total p_3 time can never exceed of 2124 minutes. When sold abroad, product A (B) yields a profit of 20 (32) per unit, although it is made a discount increasing of 4 cent (3 cent) from each order. The managers want to maximize the benefit.

The fuzzy problem can be formulated as follows:

$$
\begin{aligned}
&Max\ 20x_1 + 32x_2 - 0,04x_1^2 - 0,03x_2^2\\
&s.t.:\\
&\qquad 10x_1 + 6x_2 \leq 2500\\
&\qquad 5x_1 + 10x_2 \lesssim 2000\\
&\qquad 7x_1 + 10x_2 \lesssim 2050\\
&\qquad x_j \geq 0, j = 1,2
\end{aligned}
$$

When we spoke about the violations of the restrictions we referred to the user allowing in decresing way (as far as we approch the limit) the level of violation admittion. That is to say, the behavior that we have when we transferred the 2000 minutes border (2050 minutes) is a decreasing function, decreasing quickly until the end. Therefore, the membership functions of the second and third fuzzy constraint can be defined as:

$$
\mu(x_1, x_2) = \begin{cases} 1 & 5x_1 + 10x_2 \leq 2000 \\ (\frac{2064 - 5x_1 - 10x_2}{64})^3 & 2000 \leq 5x_1 + 10x_2 \leq 2064 \\ 0 & \text{otherwise} \end{cases}
$$

$$
\mu(x_1, x_2) = \begin{cases} 1 & 7x_1 + 10x_2 \leq 2050 \\ (\frac{2124 - 7x_1 - 10x_2}{74})^3 & 2050 \leq 7x_1 + 10x_2 \leq 2124 \\ 0 & \text{otherwise} \end{cases}
$$

According with (3), the fuzzy problem is transformed into the following nonlinear parametric programming problem:

$$
\begin{aligned}
&Max\ 20x_1 + 32x_2 - 0,04x_1^2 - 0,03x_2^2\\
&s.t.:\\
&\qquad 10x_1 + 6x_2 \leq 2500\\
&\qquad 5x_1 + 10x_2 \leq 2064 - 64\sqrt[3]{\alpha}\\
&\qquad 7x_1 + 10x_2 \leq 2124 - 74\sqrt[3]{\alpha}\\
&\qquad x_j \geq 0, j = 1,2
\end{aligned}
$$

where $\alpha \in [0,1]$ is the parameter emerging in the parametric problem.

In order to solve the nonlinear parametric programming problem we consider the following multi-objective nonlinear optimization problem according to (5):

$$
\begin{aligned}
&Max\ f_1(\boldsymbol{x}) = 20x_1 + 32x_2 - 0,04x_1^2 - 0,03x_2^2\\
&Max\ f_2(\boldsymbol{x}) = x_3\\
&s.t.:\\
&\qquad 10x_1 + 6x_2 \leq 2500\\
&\qquad 5x_1 + 10x_2 \leq 2064 - 64\sqrt[3]{x_3}\\
&\qquad 7x_1 + 10x_2 \leq 2124 - 74\sqrt[3]{x_3}\\
&\qquad x_j \geq 0, j = 1,2, 0 \leq x_3 \leq 1
\end{aligned}
\tag{6}
$$

5 Experiments and Results

In order to check out our technique, the evolutionary algorithm has been executed on problem detailed in (6).

The parameters given in Table 1 were used in the executions.

Table 1. Parameters in the execution of the algorithm.

Population size	$N = 220$
Cross probability	$pCross = 0.9$
Mutation probability	$pMutate = 0.2$
Uniform cross probability	$pUniformCross = 0.3$
Uniform mutation probability	$pUniformMutate = 0.1$
Non uniform mutation probability	$pNotUniformMutate = 0.4$
Parameter c for non uniform mutation	$c = 2.0$

The results obtained with the algorithm are shown in Fig. 1. We compare the solutions obtained with our multi-objective evolutionary algorithm with solutions obtained by a gradient method for 10 constant values of the α parameter (x_3 in problem (6)) showed in Table 3 and graphically in Figure 1. It can be observed that non dominated points are obtained by the multi-objective evolutionary algorithm evenly distributed in the whole Pareto optimal front.

Various metrics for both convergence and diversity of the populations obtained have been proposed for a more exact evaluation of the effectiveness of the evolutionary algorithms. In his book, Deb [5] assembles a wide range of the metrics which figure in the literature. For this paper we propose the use of two metrics to evaluate the goodness of the algorithm. The first metric, the generational distance (Υ) proposed by Veldhuizen [13] evaluates the proximity of the population to the Pareto optimal front by calculating the average distance of the population Q from an ideal population P^* made up of solutions distributed uniformly along the Pareto front. This metric is shown in the following expression:

$$\Upsilon = \frac{\left(\sum_{i=1}^{|Q|} d_i^p\right)^{1/p}}{|Q|}$$

For $p = 2$, parameter d_i is the Euclidean distance (in the objective space) between the solution $i \in Q$ and the nearest solution in P^*:

$$d_i = \min_{k=1}^{|P^*|} \sqrt{\sum_{m=1}^{M} \left(f_m{}^{(i)} - f_m^{*\,(k)}\right)^2}$$

where $f_m^{*\,(k)}$ is the value of the m-th objective function for the k-th solution in P^*, and M is the number of objectives. For our problem, we use the 10 solutions showed in Table 3 as ideal population P^*.

To evaluate the diversity of the population we use the measurement put forward by Deb et al. [5]:

$$\Delta = \frac{\sum_{m=1}^{M} d_m^e + \sum_{i=1}^{|Q|} \left| d_i - \overline{d} \right|}{\sum_{m=1}^{M} d_m^e + |Q| \, \overline{d}}$$

where d_i may be any metric of the distance between adjacent solutions, and $\overline{d}$ is the mean value of such measurements. In our case, d_i has been calculated using the Euclidean distance. Parameter d_m^e is the distance between the extreme solutions in P^* and Q corresponding to the m-th objective function.

Table 2 shows the values for convergence and diversity metrics Υ and Δ respectively obtained with the proposed multi-objective algorithm for the problem (6).

Table 2. Convergence and diversity values.

$\Upsilon = 0.67189927954177$
$\Delta = 0.82661701735378$

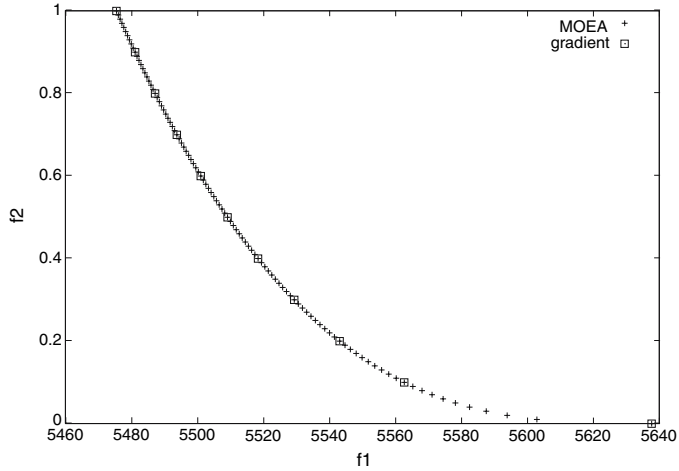


Fig. 1. Non dominated points obtained for the problem (6).

6 Conclusions and future research

Nonlinear optimization problems with fuzzy constrains are, in general, difficult to solve. Parametric programming techniques have been shown in literature

Table 3. Results obtained with a gradient method for the problem (6) with x_3 (f_2) constant.

x_1	x_2	x_3	f_1	f_2
59.605	170.676	0.0	5637.724	0.0
58.256	168.165	0.1	5562.726	0.1
57.944	167.512	0.2	5543.147	0.2
57.703	167.054	0.3	5529.391	0.3
57.512	166.689	0.4	5518.431	0.4
57.350	166.381	0.5	5509.165	0.5
57.209	166.112	0.6	5501.058	0.6
57.083	165.872	0.7	5493.806	0.7
56.968	165.653	0.8	5487.210	0.8
56.862	165.452	0.9	5481.139	0.9
56.764	165.265	1.0	5475.503	1.0

as suitable methods to approach these kinds of problems. However, parametric programming problems have been solved mainly for linear case. Multiobjective evolutionary computation provides a chance to solve nonlinear parametric programming problems. The set of points composing the parametric solutions can be capture in a single run of the algorithm beside of the power of these techniques in solving hard problems. Obtained results show a real ability of the proposed approach to solve problems arising in exporting companies from South of Spain. A complete set of test problems and a generalitation of the problem to consider fuzzy costs and coefficients is being considered as main future research.

Acknowledgements

Research supported in part by MCyT/FEDER under projects TIC2001-0245-C02-01 and TIC2002-04242-CO3-02.

References

1. J. Biethahn, V. Nissen (1995), *Evolutionary Algorithms in Management Applications*. Springer-Verlag Berlin Heidelberg.
2. C.A. Coello, D.V. Veldhuizen, G.B. Lamont (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic/Plenum publishers, New York.
3. S. Chanas(1983) *The use of parametric programming in fuzzy linear programming*. Fuzzy Sets and Systems. 11:243-251.
4. V. Chankong, Y.Y. Haimes (1983). *Multiobjective Decision Making: Theory and Methodology. North-Holland series in Systems Science and Engineering, Andrew P. Sage (Ed.)*.
5. K. Deb (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley and Sons, LTD.
6. D.E. Goldberg (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.
7. F. Jiménez, J.L. Verdegay (1999). *Solving fuzzy solid transportation problems by an evolutionary algorithm based parametric approach*. European Journal of Operational Research, 113(3):688-715.
8. F. Jiménez, A.F. Gómez-Skarmeta, G. Sánchez, K. Deb (2002) *An evolutionary algorithm for constrained multi-objective optimization*. 2002 IEEE World Congress on Evolutionary Computation.
9. F. Jiménez, J.M. Cadenas, J.L. Verdegay, G. Sánchez (2003) *Solving fuzzy optimization problems by evolutionary algorithms*. Information Science, An International Journal (to appear).
10. Z. Michalewicz (1992). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag.
11. Z. Michalewicz, M. Schoenauer (1996). Evolutionary Algorithms for constrained parameter optimization problems. *Evolutionary Computation*, vol. 4, no. 1, pp. 1-32.
12. H. Tanaka, T. Okuda, K. Asai (1974). *On fuzzy mathematical programming*. Journal of Cybernetics, 3(4):37-46.
13. Veldhuizen, D.V., Lamont, G.B. (1999). *Multiobjective evolutionary algorithms: Classifications, Analyses, and New Innovations*. Ph. D. Thesis Dayton, OH: Air Force Institute of Technology. Technical Report No. AFIT/DS/ENG/99-01.
14. J.L. Verdegay (1982). *Fuzzy mathematical programming*. Fuzzy Information and Decision Processes, M.M. Gupta and E. Sánchez (Eds.), 231-237.
15. H.J. Zimmermann (1976). *Description and optimization of fuzzy systems*. International Journal of General Systems, 2:209-215.

Poster Contributions

Comparison of Reasoning for Fuzzy Control

Bohdan Butkiewicz

Warsaw University of Technology
bb@ise.pw.edu.pl

1 Introduction

Researchers proposed many reasoning methods. However, many of the methods are suitable neither for fuzzy control nor for fuzzy modeling. In the paper some possible reasoning methods are compared from this point of view. The most popular approach to fuzzy control and modeling is based on if ... then rules. Using this approach four general problems must be solved:

- what interpretations of sentence connectives "and" "or" and negation "not" may be used for if part
- what implication or other operation may be used for conclusion (then) part
- what interpretation to use for rule aggregator "also"
- what defuzzification procedure can be applied.

The widely applied reasoning as Mamdani, Larsen [13] Tsukamoto (see ex. [14]), Takagi-Sugeno [21] uses triangular norms [12] for "and", "or", "also". However, it is possible to use more general B-operations [3] or mean operations [9]. The implication methods: Kleene-Dienes, early Zadeh, Willmott, standard sharp, standard strict, standard star, Lukasiewicz, Gaines, Mamdani, and some combination of them can be found in the literature [14]. In fuzzy control Mamdani and Larsen methods are used very often in practice. Sometimes Tsukamoto or Takagi-Sugeno methods are applied. Mamdani used minimum for interpretation of "and", maximum for "or", and minimum for conclusion. However, other triangular norms give also good results. Larsen used product for conclusion. Mizumoto [16] [17] are introduced product-sum-gravity and some other methods mixing different kinds of operations. He compared also a few reasoning methods using one practical example of fuzzy control system. Many researchers proposed different kinds of reasoning, but no many who try to compare them in practice. Schwartz [20], Butnariu et al. [1], Kerre [11], Li [15], Gupta [10] compared the reasoning methods from theoretical point of view. A practical comparison can be found in Mizumoto [17], Czogala and

B. Butkiewicz: *Comparison of Reasoning for Fuzzy Control*, Advances in Soft Computing **2**, 725–731 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

Leski [8], Cao and Kandel [7]. It was shown that dynamics of fuzzy control system are very similar for different reasoning. This paper presents some comparison of the best reasoning methods possible to use in fuzzy control, choused by the author and other researchers [7] [17] from large set of methods.

2 Fuzzy control

The methods used in fuzzy control generally are based on triangular norms, but there are also another ideas using mean operations, as arithmetic, geometric, harmonic mean, generalized mean operations, and some others as simple sum. The author of the paper, using three examples of fuzzy control systems, performed a comparison of the results of fuzzy PID and PD control. The examples are simple, but can be considered as representative. Input linguistic variables: error e, derivative d, and integral of error i have seven fuzzy sets: NL, NM, NS, ZE, PS, PM, PL. The fuzzy PD (FPD) part of the controller has typical rules. The fuzzy PI (FPI) part has separate rules. The output signals of the PD and PI part are added. The membership functions are triangular. Previous investigations [2] [4] [][6] and other researchers (see ex. [17] [18] [19] allows to choose a collection of possible reasoning methods for fuzzy control. In the paper as reasoning is understand sequence of operations:
Reasoning = [operation1-operation2-operation3-defuz]
where operation1 is t-norm or mean operation, operation2 is s-norm or dual mean operation, operation3 is implication or other operation, defuz is defuzzification method. Only three defuzzification methods are compared hear: area (AM), gravity(GM), and height (HM).
The reasoning methods were applied to the systems described below. In the Tables some abbreviation are used:
p - product: ab
s -sum: $a + b$
f - force implication: $ab = a(1 - |a - b|)$
bp - bounded(Lukasiewicz) product: $max(0, a + b - 1)$
bs - bounded (Lukasiewicz) sum: $min(1, a + b)$
dp - drastic product: a if b=1, b if a=1, 0 if $a, b < 1$
ds - drastic sum: a if b=0, b if a=0, 1 if $a, b > 0$
dpm - modified dp: a if $b > \alpha$, b if $a > \alpha$, 0 if $a, b \leq \alpha$; (α=0.01)
dsm - modified ds: a if $b < 1 - \alpha$, b if $a < 1 - \alpha$, 1 if $a, b \geq \alpha$
ep - Einstein product: $ab/(1 - a - b + ab)$
es - Einstein sum: $(a + b)/(1 + ab)$
hp- Hamacher product: $ab/(a + b - ab)$
hs - Hamacher sum: $(a + b - 2ab)/(1 - ab)$
pp - probabilistic (algebraic) product: ab
ps - probabilistic (algebraic) sum: $a + b - ab$
yp - Yager product: $1 - min(1, [(1 - a)^p + (1 - b)^p]^{1/p})$ with p=2
ys - Yager sum: $min(1, [a^p + b^p]^{1/p})$

min - logic product: $min(a, b)$
max - logic sum: $max(a, b)$
arit - arithmetic mean: $(a + b)/2$ (dual arithmetic is identical)
ge - geometric mean: $\sqrt{ab}$
dge - dual geometric mean: $1 - \sqrt{(1-a)(1-b)}$
ha - harmonic mean: $2ab/(a + b)$
dha - dual harmonic mean: $1 - 2(1-a)(1-b)/(2-a-b)$
a - area defuzzification method (AM)
g - gravity defuzzification method (GM)
h - height defuzzification method (HM)
Apart from these operations it is possible to use some others as ($0 \leq \gamma \leq 1$):
"fuzzy and" : $\gamma min(a, b) + (1-\gamma)(a + b)/2$
"fuzzy or": $\gamma max(a, b) + (1-\gamma)(a + b)/2$
min-max: $\gamma min(a, b) + (1-\gamma)max(a, b)$
comp-and: $(ab)^{1-\gamma}[1 - (1-a)(1-b)]^{\gamma}$
However, the results of investigations were rather negatives and are not shown here. Comparison of reasoning was performed using system presented in the

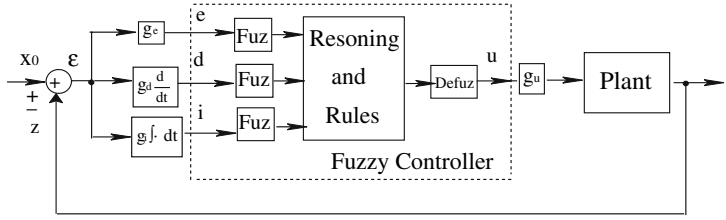


Fig. 1. General structure of the system

Fig. 1. The system step responses were investigated. The output gain g_u was adjusted to obtain the same overshoot of 15%. Rise time (for level 0.9), value of first minimum y_{1min}, square and time errors were chosen as comparative criteria. The time-error is defined as integral of product $t \cdot error(t)$. Every time 30..40 reasoning methods were tested. The range of discourse was found for every step response parameter and divided into 30-33 levels. Any parameter value was scored comparing to the levels. The best value obtained score equal to 1. Total score is sum of scores. Three systems were investigated.

System 1. FPID with linear plant of first order with delay [17]. The plant transfer function $H(s) = e^{-2s}/(1 + 20s)$.
A part of best results obtained is shown in the Table 1.

System 2. FPID controller and nonlinear plant of second order described by equation $\ddot{y} + 2\zeta\omega_n\dot{y} + y = \alpha_1 u(t) + \alpha_2|u(t)|$. The parameters of the system: g_e=2, g_d=2, g_i=0.002, ω_n=100, ζ=0.7, α_1=1, α_2= -0.1.

Table 1. The results of step response for the System 1

Reasoning op1-op2-op3-def	Rise time	Value y_{1min}	Square error	T-err $\times 10^3$	Total score	Rank
dp-ds-p-g	8.10	no*	217	74.8	4	1
dp-ds-f-g	8.70	no*	223	79.0	10	2
dpm-dsm-p-g	10.80	.988	267	99.9	40	3
p-s-f-g	11.92	.989	276	101.2	45	4
bp-bs-p-g	12.01	.984	281	108.1	53	5
bp-bs-p-a	12.20	.985	284	108.9	55	6
p-s-p-g	12.76	.982	291	113.9	64	7
bp-bs-bp-g	12.89	.981	291	111.4	65	8
p-s-p-a	12.87	.981	291	113.4	66	9
bp-bs-f-g	12.94	.978	291	110.9	67	10

Table 2. The results of step response for the System 2

Reasoning op1-op2-op3-def	Rise time	Value y_{1min}	Square error	T-err $\times 10^3$	Total score	Rank
dp-ds-f-g	14.44	.874	370	128	37	1
dp-ds-p-g	13.56	.849	348	132	44	2
bp-bs-bp-g	16.41	.896	404	139	46	3-4
bp-bs-bp-a	-"-	-"-	-"-	-"-	-"-	3-4
p-s-f-g	15.06	.864	377	135	51	5-7
p-s-f-a	-"-	-"-	-"-	-"-	-"-	5-7
mdp-mds-p-g	15.38	.874	383	143	51	5-7
.....						
min-max-p-g	19.52	.904	453	169	78	20-21
min-max-min-h	-"-	-"-	-"-	-"-	-"-	20-21
min-max-min-g	21.64	.924	498	179	92	22
arit-arit-p-g	18.81	.825	449	191	97	23

System 3. FPID with non-stable fuzzy dynamical plant. The system contains FPID controller with parameters: g_e=30, g_d=100, g_i=0.3 and fuzzy non-stable plant proposed by Chen [6]. The plant consists with four objects described by differential equations:
Ob1: if y is S and $\dot{y}$ is S then $\ddot{y} - 2.391\dot{y} + 2.484y = 1.277u$
Ob2: if y is S and $\dot{y}$ is L then $\ddot{y} + 0.91\dot{y} = 0.23u$
Ob3: if y is L and $\dot{y}$ is S then $\ddot{y} + 0.722\dot{y} - 1.425y = 2.825u$
Ob4: if y is L and $\dot{y}$ is L then $\ddot{y} - 0.6\dot{y} = 0.16u$
Depending on the linguistic value (Small or Large) of y and derivative of y objects are operative. Fuzzy sets have the membership functions with semi-gaussian shapes [6].

Table 3. The results of step response for the System 3

Reasoning op1-op2-op3-def	Rise time	Value y_{1min}	Square error	T-err $\times 10^3$	Total score	Rank
dp-ds-f-g	1.0709	.9625	27.75	431	23	1
dp-ds-p-a	1.0751	.9575	28.08	448	44	2
p-s-f-g	1.1230	.9658	28.81	439	47	3
bp-bs-f-g	1.1219	.9637	28.80	441	52	4
bp-bs-bp-g	1.1175	.9608	28.81	440	56	5
dp-ds-p-g	1.0692	.9546	28.01	443	62	6
bp-bs-bp-a	1.1243	.9619	28.95	444	64	7
dpm-dsm-p-g	1.0750	.9533	28.20	462	67	8
p-s-p-g	1.1214	.9552	29.10	437	70	9/10
bp-bs-p-g	1.1152	.9550	28.99	440	70	9/10
bp-bs-p-a	1.1232	.9569	29.14	447	79	11/12
p-s-p-a	1.1281	.9574	29.19	442	79	11/12
hm-dhm-p-g	1.1265	.9550	29.17	446	81	13
gm-dgm-p-g	1.1270	.9538	29.27	442	83	14
ep-es-p-g	1.1233	.9523	29.21	442	84	15
p-bs-p-g	1.1248	.9524	29.20	442	84	16
yp-ys-p-g	1.1256	.9517	29.24	443	86	17
pp-ps-p-g	1.1264	.9514	29.26	445	93	18
arit-arit-bp-g	1.1230	.9548	29.19	459	95	19
hp-hs-p-g	1.1285	.9503	29.31	449	97	20
min-max-p-g	1.1309	.9498	29.37	453	104	21
min-max-min-g	1.1428	.9524	29.69	453	107	22
arit-arit-p-g	1.1470	.9515	29.91	455	117	23

3 Comparison of the results

General comparison of the results shows that many methods give very similar results. It can be seen first of all from graphical comparison of responses. However, there are some methods, which in any example obtain high rank (small score). The results show Table4. Now, the total score is the sum of ranks.

Of course, not all combinations were tested. Example, defuzzification GM gives generally slightly better results than AM and better than HM. Thus, operations were tested firstly with GM. The best method is drastic product-drastic sum-force-gravity (dp-ds-f-g). Second place takes drastic product-drastic sum-product-gravity method. Both methods have a strange nonlinear effect. If the output's gain g_u rises then in some regions overshoot diminishes. Thus, the methods cannot be recommended without reserve. Good properties have product-sum and bounded product-bounded sum operations with force or bounded product operations.

Conventional Mamdani reasoning (min-max-min-g) and Larsen reasoning (min-max-p-a or min-max-p-g) take very far places. Thus, there are no any

Table 4. The comparison of results

Reasoning	Syst. 1	Syst. 2	Syst. 3	Score	Rank
dp-ds-f-g	2	1	1	4	1
dp-ds-p-g	1	2	6	9	2
p-s-f-g	4	6	3	13	3
bp-bs-bp-g	8	3.5	5	16.5	4
dpm-dsm-p-g	3	6	8	17	5
bp-bs-f-g	10	11	4	25	6
p-s-p-g	7	9	9.5	25.5	7
bp-bs-p-g	5	15	9.5	29.5	8
p-s-p-a	9	9	12.5	30.5	9
bp-bs-p-a	6	15	11	32	10

reasons to use them. Other investigations do not presented here show that product-sum and bounded product-bounded sum can be recommended as first pair of operations. For operation 3 force implication, product or bounded product is recommended. Rarely HM is better than GM and AM, however it happens.

Some general important remarks must be added. The author shown [2] [3] that any stable system with FPD controller tends in steady state to the same value independently on the pair of triangular norms used for reasoning. Similarly, error of a system with FPID controller tends to zero independently on the pair of triangular norms used for reasoning.

In [5] the comparison of reasoning was performed using another criteria. The last observed value y_{Tmax} was used instead of time-error. Moreover, the scoring was performed in different way. All values of the parameters were similarly ordered monotonically, but the score value is equal to consecutive number. The best method was prod-sum-force-gravity. Second place obtained dr.prod-dr.sum-prod-gravity method. The scoring method proposed here, based on partition on levels, seems more reasonable. Also, the last observed value of y_{Tmax} used before in [5] is rather random, so the result obtained here are more reliable.

References

1. Butnariu D., Klement E.P., Zafrany S. (1995). On triangular norm-based propositional fuzzy logics. Fuzzy Sets and Systems v. 69, pp. 241-255.
2. Butkiewicz B.S. (1998). Steady-State Error of a System with Fuzzy Controller. IEEE Transactions on System, Man, and Cybernetics -Part B: Cybernetics, 6, 855-860.
3. Butkiewicz B.S. (2001). Fuzzy Control System with B-operations. In: R. John, R. Birkenhead (eds.), Advances in Soft Computing, Developments in Soft Computing, Physica-Verlag, Heidelberg, New York, pp. 48-55.

4. Butkiewicz B.S. (2001). About Robustness of Fuzzy Logic PD and PID Controller under Changes of Reasoning Methods. In: Advances in Computational Intelligence and Learning Methods and Applications, H-J. Zimmermann, G. Tselentis, M. van Someren, G. Dounias (eds.), Kluwer Academic Publishers, pp. 307-318.
5. Butkiewicz B. S. (2004). Comparison of Reasoning Methods for Fuzzy Control. In: L. Rutkowski, J. Siekmann, R. Tadeusiewicz, L. Zadeh (eds.), Springer, pp. 278-283.
6. Chen C.-L., Chen P.-C., Chen C.-K. (1993). Analysis and Design of Fuzzy Control System. Fuzzy Sets and Systems, v. 57, pp. 125-140.
7. Cao Z., Kandel A. (1989). Applicability of Some Fuzzy Implications Operators. Fuzzy Sets and Systems, v. 31, pp. 151-186.
8. Czogaa E., ski J. (2201). On equivalence of approximate reasoning results using different interpretations of fuzzy if-then rules. Fuzzy Sets and Systems, v. 117, pp. 279-296.
9. Dyckhoff H., Pedrycz W. (1984). Generalized means as model of compensative connectives. Fuzzy Sets and Systems, v. 14, 143-154.
10. Gupta M.M., Qi J. (1991). Theory of t-norms and fuzzy inference methods. Fuzzy Sets and Systems, v. 40, No. 3, pp. 431-450.
11. Kerre E. (1992). A comparative study of the behavior of some popular fuzzy implication operators on the generalized modus ponens. In Zadeh L., Kacprzyk J. (eds.) Fuzzy logic for management of uncertainty. Wiley, New York, pp. 281-295.
12. Klement E.P., Mesiar R., Pap E. (2000). Triangular norms. Kluwer Academic Publishers, Dordrecht.
13. Larsen P.M. (1980). Industrial Applications of Fuzzy Logic Control. Int. J. Man Machine Stud., v. 12, No. 1, 3-10.
14. Lee E.S., Zhu Q. (1995). Fuzzy and Evidence Reasoning. Physica-Verlag, Heidelberg.
15. Li Z. (1999). Suitability of fuzzy reasoning methods. Fuzzy Sets and Systems, v. 108, pp.299-311.
16. Mizumoto M. (1985). Extended fuzzy reasoning. In: Approximate Reasoning in Decision Analysis, M.M. Gupta, A. Kandel, W. Bandler, J.B. Kiszka (eds.), North-Holland, New York, pp. 71-85.
17. Mizumoto M. (1995). Improvement of Fuzzy Control Method. In: Fuzzy Logic and Intelligent Systems, Li H. Gupta M. (ed.), Kluwer Acad. Publishers, pp. 1-16.
18. Pedrycz W. (1996). Fuzzy Control and Fuzzy Systems. John Wiley & Sons, New York.
19. Pedrycz W. (1980). On the use of fuzzy ukasiewicz logic for fuzzy control. Arch. Autom., v. 3, pp. 301-313.
20. Schwartz D.G. (1989). Fuzzy inference in a formal theory of semantic equivalence. Fuzzy Sets and Systems, v. 31, pp. 205–216.
21. Takagi T., Sugeno M. (1985). Fuzzy Identification of Systems and its Application to Modeling and Control. IEEE Trans. on Systems, Man, and Cybernetics, v. 15, No.1, 116-132.

Evolving Scientific Knowledge

Jürgen Paetz[1,2]

[1] J.W. Goethe-Universität Frankfurt am Main, FB 14, Institut für Organische Chemie und Chemische Biologie, Molecular Design Laboratory, Marie-Curie-Straße 11, D-60439 Frankfurt am Main, Germany
paetz@bioinformatik.uni-frankfurt.de
[2] J.W. Goethe-Universität Frankfurt am Main, FB 15, Institut für Informatik, Robert-Mayer-Straße 11-15, D-60054 Frankfurt am Main, Germany

1 Introduction

The generation of new, useful knowledge is the mission of scientists. In the data mining area researchers try to find interesting new knowledge about data [1, 2]. Especially in text mining interesting knowledge is extracted from texts [3]. Automated theorem proving aims at finding automatically theorems and proofs [4]. Genetic programming is used for automatically generating new programs [5]. To find new and interesting knowledge *automatically*, a system should be "computational intelligent" as much as possible.

If we consider a limited field of scientific activity, a kind of "knowledge domain", e.g. soft computing, then most of the actual knowledge of such a knowledge domain has been written down and is available as papers, published in journals, books, technical reports, theses, or it is electronically published.

Our aim is to evolve a given knowledge domain to obtain new knowledge. In Fig. 1 we show the parts that would be needed for a complete automatic system. At first relevant "content parameters" of the elements of a knowledge domain need to be extracted. This could be a task for text miners. Then, a component is introduced that we call "knowledge evolution". A final part is the evaluation of the evolved knowledge by the domain experts.

In our paper we will focus on the second step, the knowledge evolution. In section 2 we present the motivation behind the knowledge evolution procedure. Then, in section 3, we will give examples for knowledge evolution in the knowledge domain of soft computing, considering neural network models and fuzzy logic. In section 4 we discuss our ideas and give a conclusion.

2 A Motivation for Knowledge Evolution

We explain knowledge generation ("science") partly as an evolutionary process, and demonstrate that the principles of an evolutionary strategy can be

J. Paetz: *Evolving Scientific Knowledge*, Advances in Soft Computing **2**, 733–738 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

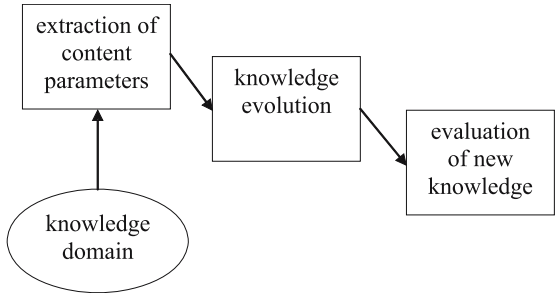


Fig. 1. Knowledge extraction system using knowledge evolution.

understand as an analogy for scientific research. For an overview of evolutionary strategies see [6] for example. Our insights will lead to a proposal of *automatic knowledge evolution* based on scientific knowledge in the next section.

The ambition of scientists is to create something "new" and something "better". Usually, a new work is based on older work where parts of the older work are changed or combined. We state that the changes can be interpreted as *mutating* knowledge to another knowledge.

Let us consider a method in the area of soft computing, e.g. a neural network based method. Much work is done to construct better topologies and to create better learning algorithms. When reading papers, often one gets the impression that this goal is not reached with a significant global proof, i.e. the new network is not the best network so far, better than all the others, but it is better than one or several existing methods ("local improvement"). A lot of mutations of algorithms exist, especially when applications are considered. Then, usually some parameters are changed or added or deleted compared to other existing methods.

In the last years the interest in hybrid methods has increased. For example neural networks have been combined with evolutionary strategies or fuzzy systems. This has been done in fact not really in a goal-oriented way, meaning to find the optimal hybrid method, but one hybrid method that works well or in some aspects better than basic single methods. This is as well discussed in [7]. We notice that knowledge of different papers have been *recombined*.

Having mutated or recombined knowledge to new knowledge, its quality, the *fitness*, needs to be evaluated. This is formally done by several fitness criteria in a review process, as for this conference. Once a paper, containing the new knowledge, is published, it may be indirectly evaluated by citation factors of journals or directly by a high number of following applications or following theoretical papers. But this shows as well that such a paper is a suboptimal one.

Having reached the point that papers are mutated, recombined and evaluated for their fitness, it is straightforward to treat the formalized knowledge

of a paper as an evolving individual in the knowledge domain. Now, we have all the ingredients collected for interpreting knowledge generation by an evolutionary strategy as knowledge evolution:

Algorithm 1: (Evolving Scientific Knowledge)
1. *Initialization*: The set of individuals contains the content parameters of selected papers in the knowledge domain.

2. **while** knowledge evolution has not ended (= everything can be explained in the knowledge domain) **do**

2a. *Mutation*: Modify the content parameters.

2b. *Recombination*: Combine content parameters.

2c. *Evaluation*: Evaluate the new content parameters, i.e. determine a fitness for one individual.

2d. *Selection*: Select the individuals with the highest fitness.

end % while

The optimization by evolution in Algorithm 1 may be replaced by swarm algorithms. Since working groups are not working without interaction or communication with other working groups, content parameters of actual knowledge is interchanged. This can be seen as the global optimization part in swarm algorithms where local and global elements of optimization are combined. Swarm algorithms are described in [8] for example. If we discuss this aspect further, one may interpret the set of individuals as a swarm, optimizing its research results.

At this point we have defined the paradigm of "evolving scientific knowledge". Before we present examples in the next section, we give additional thoughts about knowledge evolution.

In our context the results in [9], where it is stated that a single run of an evolutionary strategy with a higher population achieves better results than multiple runs with smaller populations, can be seen as a demand for more interaction between working groups: Research based on more working groups is more successful than the sum of the research on an area done by several distinct working groups.

Automatically mutated and recombined knowledge may be useful as a basis for a tool for research assistance. Such a tool may found knowledge that would not be found otherwise. Although such results may not be the best ones, they may be helpful for the researcher for further research. Of course, scientific knowledge that outlines new research directions, not based on mutations or recombinations as before, is needed to extend the knowledge

base. It is assumed that it will be far in the future that an automatic system will become as inventive as a researcher. Surely, a knowledge generation system is a complex system. Thus, we restrict our considerations in the next section to knowledge that can be systematically formalized by content parameters.

3 Examples

We will give a concrete example for knowledge evolution, followed by an extension of the first example, and one example that is not as concrete as the following one, showing the limitations of the approach.

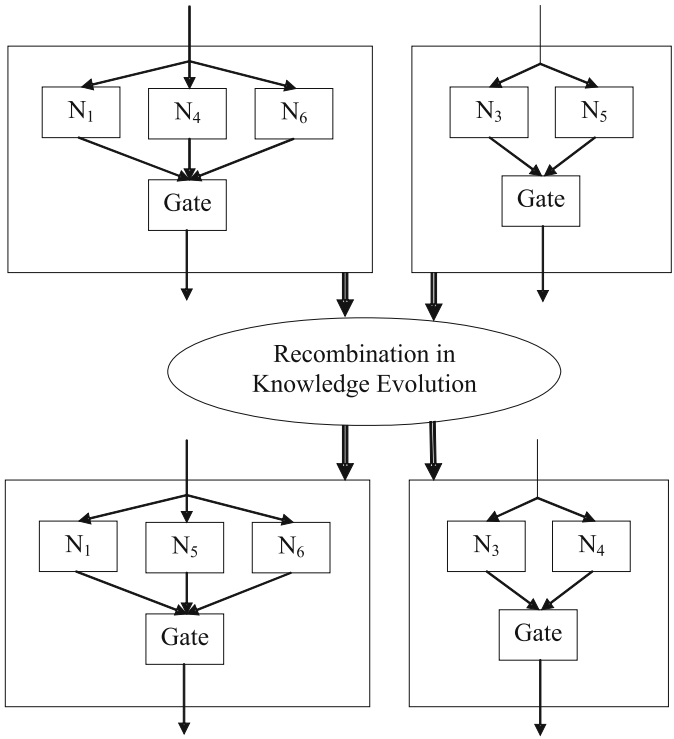


Fig. 2. Example of knowledge evolution: model exchange in neural network ensembles. The sets $N_i, i = 1, \ldots, 6$ are the content parameters of neural network models, published in different papers.

Example 1) Consider m not necessarily equal neural network models with parameter sets $N_1, \ldots, N_n$ as content parameters, e.g. stemming from several papers. Then, we introduce a basic mutation of one parameter set as a

modification of the parameter settings of the parameters in the set. As a recombination we can define a combination to an ensemble machine [10], Ch. 7 with a voting gate. Later on, the exchange of two models can be used as a recombination, cp. Fig. 2. This recombination represents a new network model N_{n+1} with a new, possibly larger, parameter set. Different voting schemes could also be integrated in the knowledge evolution. If we consider a classification problem with the trained network models, then as a fitness operator the classification performance on a test set can be used. After a suitable number of generations the best evolved model with its content parameters is the result of knowledge evolution.

Example 2) If we add content parameters of similar applications or different formulizations of a problem, given by $A_1, \ldots, A_m$, then one can combine the neural network model parameter sets $N_1, \ldots, N_n$ with the application parameters, e.g. different representations of molecules. The optimized result is the best solution for an application problem, i.e. a neural network ensemble for an optimized formulation of a problem.

Example 3) The work of Zadeh [11] is an example for research, that cannot easily performed automatically, but the definition of fuzzy sets has been the initial point for a new successful research direction. If one would interpret fuzzy sets as a mutation of naive sets, for example by recombining probabilities with sets, then a fitness operator would have to evaluate this recombination with a very high fitness, although it is not clear how the fitness should be determined automatically in this case.

4 Conclusion

The great challenge concerning the generation of scientific knowledge is its *automization* by knowledge evolution. Our insights are a first step towards a meta-research tool, (semi-)automatically finding new and optimal knowledge that is based on existing knowledge. Until now we have only presented ideas and the example of ensemble machine evolution. A main problem that needs to be solved is the determination of content parameters. Of course, there will be scientific achievements that could not or not yet be automized, e.g. an automatic proof of Fermat's Last Theorem, proved in [12], [13], although there is an analogy in the more than 350 year old history of this problem to an evolutionary process since many approaches were modified and combined. For modeling an automization for such extremely difficult problems with a huge search space a much deeper understanding of brain functions is supposed to be needed. A hypothesis could be that the brain is able to evolve knowledge internally and unconciuosly. The next moderate step of this ongoing work is the development of more general and useful models for knowledge evolution.

Acknowledgement

The author thanks one of the anonymous reviewers for helpful comments.

References

1. Han J, Kamber M (2000) Data mining: concepts and techniques. Morgan Kaufmann Publishers, San Francisco
2. Data Mining, Knowledge Discovery, Genomic Mining, Web Mining http://www.kdnuggets.com
3. Berry MW (ed) (2004) Survey of text mining : clustering, classification, and retrieval. Springer, New York
4. Newborn M (2001) Automated theorem proving : theory and practice. Springer, New York
5. Koza JR (1992) Genetic programming on the programming of computers by means of natural selection. MIT Press, Cambridge, MA
6. Beyer H-G, Schwefel H-P (2002) Evolution strategies. Natural Computing 1:3–52
7. Cordon O, Gomide F, Herrera F, Hoffmann F, Magdalene L (2004) Ten years of genetic fuzzy systems: current framework and new trends. Fuzzy Sets and Systems 141(1):5–31
8. Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm intelligence: from natural to artificial systems. Oxford University Press, New York
9. Cantú-Paz E, Goldberg DE (2003) Are multiple runs of genetic algorithms better than one? In: Cantú-Paz E, Foster JA, Deb K, et al. (eds) Proc. of the Genetic and Evolutionary Computation Conference (GECCO), Chicago, IL, USA, Springer, Berlin, 801–812
10. Haykin S (1999) Neural networks: a comprehensive foundation. 2nd ed., Prentice Hall, Upper Saddle River, NJ
11. Zadeh L (1965) Fuzzy sets. Information and control 8:338–353
12. Wiles A (1995) Modular elliptic curves and Fermat's Last Theorem. Annals of Mathematics 141:443-551
13. Taylor R, Wiles A (1995) Ring theoretic properties of certain Hecke algebras. Annals of Mathematics 141:553-572

Coding of Chaotic Orbits with Recurrent Fuzzy Systems

Alexander Sokolov[1] and Michael Wagenknecht[2]

[1] National Aerospace University Tchkalov str. 17, Kharkiv 61070, Ukraine
asokolov@xai.edu.ua
[2] University of Applied Sciences Zittau/Goerlitz, IPM, Theodor-Koerner-Allee 16, Zittau 02763, Germany
m.wagenknecht@hs-zigr.de

Abstract. In this paper we investigate the dynamic systems, which are represented by recurrent Takagi-Sugeno rule bases that are widely used in many applications. The main question to be answered is under what conditions the recurrent rule base can reconstruct the chaotic bit series. We use for this purpose so-called 'backward interval mapping'.

1 Modeling of Tent Mapping with Fuzzy Rule Base

In the simplest case the recurrent Takagi-Sugeno (TS) fuzzy rule base of 0^{th} order can be presented in the following form [1]

$$\begin{aligned} &R_1: \ if\, x_k\, is\, L_1 then\ x_{k+1} = A_1, \\ &R_2: \ if\, x_k\, is\, L_2 then\ x_{k+1} = A_2, \\ &\vdots \\ &R_N: \ if\, x_k\, is\, L_N then\ x_{k+1} = A_N, \end{aligned} \tag{1}$$

where x is state variable, $L = \{L_1, \ldots, L_N\}$ are linguistic variables and $A = \{A_1, \ldots, A_N\}$ are numerical constants. As was shown in [2] three rules like (1) are necessary and sufficient for producing chaotic mapping (when normality conditions are held for membership functions of L_i and $x \in I = [0,1]$,$A_i \in I = [0,1]$). In this case we have the transition function $f : I \to I$, namely

$$f : x_k \to x_{k+1}. \tag{2}$$

Then mapping (2) is chaotic in sense of Li-Yorke [3]. When we have triangular membership functions the mapping (2) is isomorphic to well known tent mapping. In such case we can rewrite the mapping (2) as the slopping tent mapping

$$x_{k+1} = \begin{cases} f_1(x_k) = \frac{1}{\lambda} x_k, if 0 \le x_k \le \lambda, \\ f_2(x_k) = \frac{1}{\lambda - 1} x_k + \frac{1}{1-\lambda}, if \lambda < x_k \le 1, \end{cases} \tag{3}$$

where $x_k \in [0,1]\,, \lambda \in (0,1)$ (Fig. 1).

A. Sokolov and M. Wagenknecht: *Coding of Chaotic Orbits with Recurrent Fuzzy Systems*, Advances in Soft Computing **2**, 739–746 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

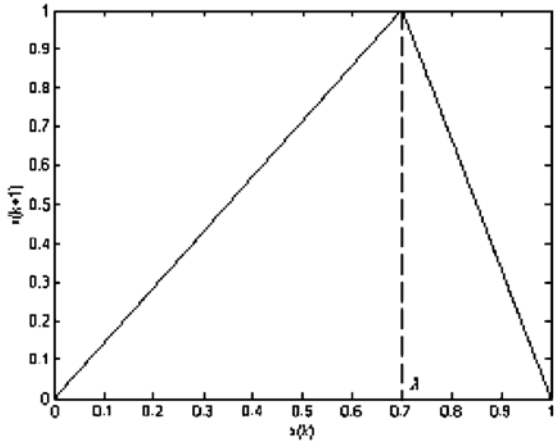


Fig. 1. Slopping mapping

2 Reconstruction of Chaotic Bit Series with TS Models

Let us consider the following bit sequence

$$C = \{c_i\}_{i=1}^{N} \tag{4}$$

with length $N, c_i \in \{0, 1\}$. It is necessary to restore the source sequence (4) as a bit sequence

$$\tilde{C} = \{\tilde{c}_i\}_{i=M}^{N} \tag{5}$$

according to the rule

$$\tilde{c}_i = \begin{cases} 1, if x_i \geq \lambda, \\ 0, if x_i < \lambda, \end{cases} \tag{6}$$

for$i = \overline{M, N}, M \leq N$. Namely, it is necessary to find such value $x_M$that gives the same values for restoring sequence like in the source one. Besides, it is important to find such value of λ that gives maximum members of restored sequence. The best case is when $M = 1$.

The novelty of approach proposed consists in substitution of chaotic bit series by initial value x_M that can reconstruct original orbit with mapping (3). Underlying methodology is based on so called backward interval mapping.

We propose the following solution of this task. First of all we propose to use so-called *backward interval mapping*[2].

Let us consider

$$g(x) = f^{-1}(x) \tag{7}$$

The mapping (7) is contracting mapping if $f(x)$ is chaotic one. For $(k-1)^{th}$ step we can write

$$x_{k-1} = g(x_k) \tag{8}$$

Definition[4]. Mapping $f(x) : I \to I$ is chaotic if

1. it is topologically transitive, i.e., if there exists a $k > 0$ such that $f^k(U) \cap V \neq \emptyset$ where $f^k(U) = \left\{ f^k(x) \middle| x \in U \right\}$for any pair of open sets, $U, V \subseteq I$.
2. it is sensitive to the initial conditions, i.e. if there exists a $\delta > 0$ such that $x \in I$ and any neighborhood N of x there exists a $y \in N$ and $n > 0$ such that $|f^n(x) - f^n(y)| > \delta$.
3. the periodic points of f are dense in I .

According to Definition we need to find the topological transitivity for inverse mapping (8) as

$$g^{-n}(U) \bigcap V \neq \emptyset. \tag{9}$$

We use the interval mapping instead of point mapping, namely we consider the mapping

$$g(I) = \begin{cases} g_1(I) = I_1 \subset I \\ g_2(I) = I_2 \subset I \end{cases} \tag{10}$$

Because $g(x)$ is contracting mapping, then

$$I_1 \bigcup I_2 = I \text{ and } I_1 \bigcap I_2 = \emptyset. \tag{11}$$

The second step of backward mapping is

$$g^2(I) = \begin{cases} g(g_1(I)) = \begin{cases} g_1(g_1(I)) = I_{11} \subset I_1, \\ g_2(g_1(I)) = I_{21} \subset I_2, \end{cases} \\ g(g_2(I)) = \begin{cases} g_1(g_2(I)) = I_{12} \subseteq I_1, \\ g_2(g_2(I)) = I_{22} \subset I_2. \end{cases} \end{cases} \tag{12}$$

We propose the following lemma.

Lemma 1. *Let there be given $g(x) = (g_1(x), g_2(x))$ where $g_i, i = 1, 2$ are monotonous and continuous mapping on $g(x) : I \to I$ and $g(x)$ is constructed in the form (10) and it is contracting mapping ((11) is being satisfied) and we have*

$$g^{K+1}(I) = \begin{cases} g_1\left(g^K(I)\right) = I_{1\{K\}} \\ g_2\left(g^K(I)\right) = I_{2\{K\}} \end{cases} \tag{13}$$

where $\{K\} = \{11...1, 11...2, ...22...2\}$ is set of indexes length of K that were used for marking subset of I on K^{th} step and the next conditions are fulfilled

$$\begin{array}{l} g_1^{K+1}(I) \subset g_1^K(I), K = 0, 1, ..., \\ g_2^{K+1}(I) \subset g_2^K(I), K = 0, 1, ..., \\ g_2^K(I) \neq \emptyset, K = 0, 1, ... \end{array}$$

then

$$\begin{array}{l} I_{1\{K\}} \bigcup I_{2\{K\}} = I, \\ \left(\bigcap\limits_{\{K\}} I_{1\{K\}} \right) \bigcap \left(\bigcap\limits_{\{K\}} I_{2\{K\}} \right) = \emptyset \end{array}$$

and $g(x) = (g_1(x), g_2(x))$*is contracting mapping for the set* I *and all of it subsets.*

If $g(x) = (g_1(x), g_2(x))$ is fulfilled to Lemma conditions then $f(x) = g^{-1}(x)$ is chaotic in sense of Definition. For slopping tent mapping (3) we have folowing backward mapping [2] (Fig. 2)

$$x_k = \begin{cases} g_1(x_{k+1}) = \lambda x_{k+1}, \\ g_2(x_{k+1}) = (\lambda - 1)\, x_{k+1} + 1. \end{cases} \tag{14}$$

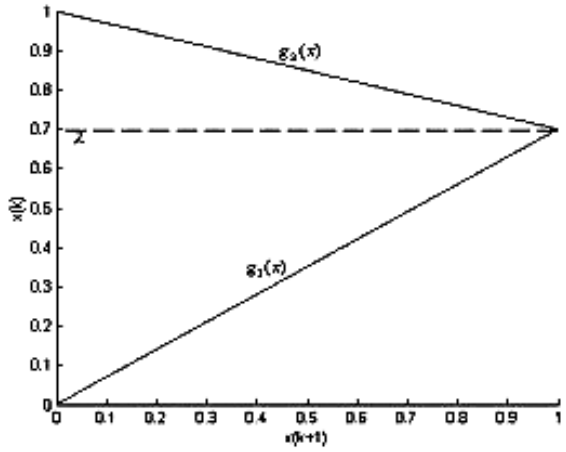


Fig. 2. Backward slopping mapping

Let us consider the action of this mapping when argument of function g is interval (Fig. 3). For initial interval I we have

$$g(I) = \begin{cases} g_1(I) = I_1 = [0, \lambda] \subset I, \\ g_2(I) = I_2 = [\lambda, 1] \subset I. \end{cases} \tag{15}$$

The following iterative procedure for backward interval mapping takes place. Let choose initial interval according to rule

$$I_N = \begin{cases} [\lambda, 1], if c_N = 1, \\ [0, \lambda], if c_N = 0. \end{cases} \tag{16}$$

Then define the possible transitions with the backward interval mapping

$$\tilde{I}_{N-1} = g(I_N) = \begin{cases} g_1(I_N) \subset [0, \lambda], \\ g_2(I_N) \subset [\lambda, 1]. \end{cases} \tag{17}$$

The next interval is made more precise in according to the value c_{N-1}:

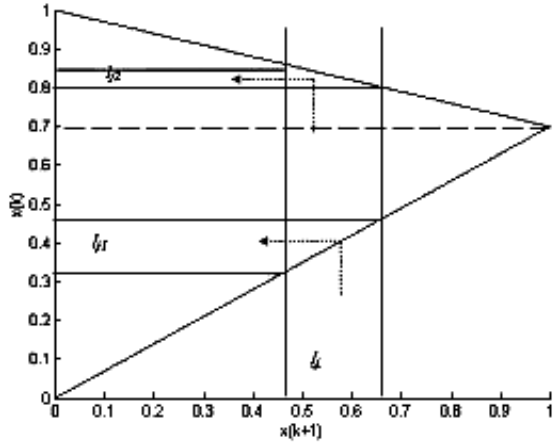


Fig. 3. Backward interval slopping mapping

$$I_{N-1} = \begin{cases} g_2\left(I_N\right), if c_{N-1} = 1, \\ g_1\left(I_N\right), if c_{N-1} = 0. \end{cases} \tag{18}$$

The procedure (17), (18) is repeated until obtaining the limit of accuracy

$$diam\left(I_{M-1}\right) = \varepsilon \tag{19}$$

Then any value $x_M \in I_M$ restores the source sequence (4) with forward mapping (3) starting from M number of sequence. Parameter ϵ is machine accuracy. So we can not restore arbitrary sequence with backward methods (only for moment when interval became ϵ - length).

To increase the quantity of restoring numbers we need to find optimal value of λ. According to (14) total coefficient of contraction of iterative procedure for backward interval mapping is determined as

$$K = \lambda^n \left(\lambda - 1\right)^m \tag{20}$$

where n is percent of 'zeros' in the source sequence and m is percent of 'ones' (Fig. 4).

It is easy to see that the following **statement** holds true:

$$\arg \min_{\lambda \in (0,1)} \lambda^n \left(\lambda - 1\right)^m = n \tag{21}$$

Thus, it is possible to decrease speed of contraction of intervals for iterative backward mapping procedure.

3 Example

Let's consider application of the described approach to the coding and restoration of bitmap image (Fig. 5). Let dark pixels of bitmap be encoded by 'ones'

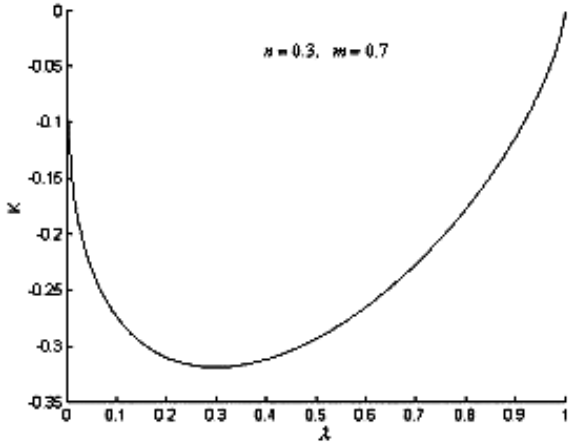


Fig. 4. Coefficient of contraction in iterative mapping procedure

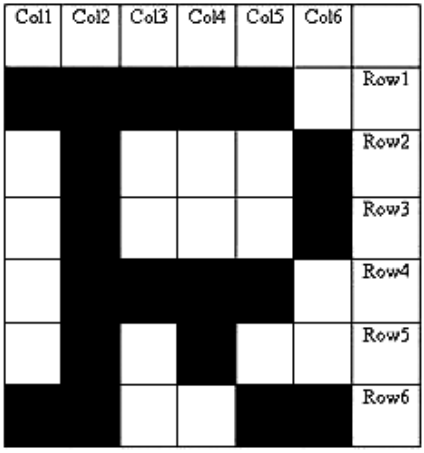


Fig. 5. Graphic presentation of the letter 'R'

and light pixels - by a 'zeros'. This bitmap was transformed to one-dimensional sequence with the length $N = Col * Row$ by reading all lines from top to bottom. Thus we have the next example of bit sequence (4):

$$\mathrm{C} = \{111110010001010001011110010100110011\}\,. \tag{22}$$

Let's define the quantity of 'zeros' and 'ones' in (22) for optimum slopping tent mapping construction:

$$n = 17/36 = 0.472222222222222, m = 19/36 = 0.527777777777778.$$

Thus in equation (3) $\lambda = 0.472222222222222$. According to rule (16) initial interval I_{36} is defined as:

$$I_{36} = [\lambda, 1] = [0.472222222222222, 1].$$

In accordance with iterative rules (17) and (18) the next interval is:

$$I_{35} = g_2\left(I_{36}\right) = [0.472222222222222, 0.750771604938272]$$

The next step of iterative procedure gives:

$$I_{34} = g_1\left(I_{35}\right) = [0.222993827160494, 0.354531035665295].$$

The final result of implementation of our iterative procedure is:

$$I_1 = [0.672604526057511, 0.672604526072895].$$

Now if we choose any $x_1 \in I_1$ we can restore a bit sequence (15) using chaotic mapping (3) and rule (6). For example if $x_1 = 0.67260452606$ then slopping tent mapping (3) gives us chaotic sequence shown on Fig. 6.

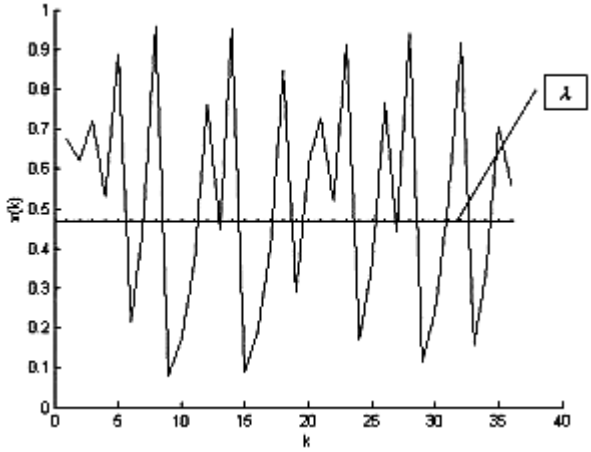


Fig. 6. Chaotic mapping for letter 'R'

We know value of λ and according to rule (6) one can restore all members of target sequence (15).

4 Summary

Proposed approach can be used for coding and data compression as well. We can increase the amount of encoded elements for bit sequence with few 'ones' or 'zeros' in it, because reduction of total coefficient of contraction takes place in such case. Obviously that critical factor of the algorithm is accuracy of software. It depends on sensitivity to initial conditions of chaotic mapping.

References

1. Kempf R, Adamy J (2003) Fuzzy Sets and Systems 140: 259-284
2. Sokolov A, Wagenknecht M (2003) Investigation of chaotic behavior of fuzzy Takagi-Sugeno models with regard to simulation and control of technological processes. Scientific Report, Univ. of Zittau/Goerlitz, IPM
3. Li TY, Yorke JA, (1975) Amer. Math. Monthly 82: 985-992
4. Kloeden PE (1991) Fuzzy Sets and Systems 42: 37-42

Genetic-Based Tuning of Fuzzy Dempster-Shafer Model

Zenon A. Sosnowski and Jaroslaw S. Walijewski

Department of Computer Science, Technical University of Bialystok, Wiejska 45A, 15-351 Bialystok, Poland

{Zenon,jarekw}@ii.pb.bialystok.pl

Abstract. In this study, we discuss the use of Dempster-Shafer theory as a well-rounded algorithmic vehicle in the construction of fuzzy decision rules. The concept of fuzzy granulation realized via fuzzy clustering is aimed at the discretization of continuous attributes. Next we use Genetic Algorithms (GA) to find the best points of division for discretization of continuous attributes. The rules, generated using Fuzzy Dempster-Shafer model (FDS), were verified by GA methods. The natural crossover improved by random changes (mutation and selection) can help us to find the best set of rules.

1 Introduction

Fuzzy modeling is regarded to be one of the possible classification architecture of machine learning and data mining. There have been a significant number of studies devoted to generating fuzzy decision rules from sample cases or examples. These include attempts to extend many classical machine learning methods to learn fuzzy rules. The objective of this paper is to employ the Dempster-Shafer theory (DST) as a vehicle supporting the generation of fuzzy decision rules. More specifically, we concentrate on the role of fuzzy operators, and on the problem of discretization of continuous attributes and show how these can be effectively used in the quantization of attributes for the generation of fuzzy rules.

The material is arranged in the following way. First, we summarize the underlying concepts of the Fuzzy Dempster-Shafer model and briefly discuss the nature of the underlying construction. Next we explain essential features of our model. Finally, we report exhaustive experimental studies concerning well-known medical data sets available on the Web.

Z.A. Sosnowski and J.S. Walijewski: *Genetic-Based Tuning of Fuzzy Dempster-Shafer Model* Advances in Soft Computing **2**, 747–755 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

This paper is a continuation of our earlier work [6,7]. Here we apply theoretical vehicle, introduced in the previous research, to new input data in order to find possible area of applications. Our important objective here is to reveal a way in which this approach becomes essential to a more comprehensive treatment of continuous attributes.

2 Fuzzy Dempster-Shafer Model

In Fuzzy Dempster-Shafer (FDS) model [2] we consider rules R_r as:

$$If \left(X_1 is\ A_{r,1,j_1}\right) ... \ And \ ... \ \left(X_n is\ A_{r,n,j_n}\right) \quad Then \quad \left(D\ is\ m_r\right)$$

where X_i and D stand for input and output respectively, and m_r is a fuzzy belief structure, that is a standard belief structure with focal elements $S_{r,p}$ as fuzzy subset of frame of discernment Θ with basic probability assignment m_r ($S_{r,p}$), and m_r ($S_{r,p}$) is the believe that the conclusion should be represented as class $S_{r,p}$.

2.1 Learning –Rules Construction

In antecedent construction of the Fuzzy Dempster-Shafer model (FDS) [2] let us assume that we have n features (attributes) in antecedents of testing example. We consider a collection of m generic linguistic terms characterized by membership functions defined in a universe of discourse being a domain of each attribute.

For each element of data t we build a collection:

$$\begin{matrix} A_{1,1,t} & A_{2,1,t} & \cdots & A_{n,1,t} \\ A_{1,2,t} & A_{2,2,t} & \cdots & A_{n,2,t} \\ \vdots & \vdots & \ddots & \vdots \\ A_{1,m,t} & A_{2,m,t} & \cdots & A_{n,m,t} \end{matrix} \qquad (1)$$

where:

$A_{i,j,t}$ – the values of j-th membership function for i-th feature and for t-th element of data.

On the base of (1), for a given data point t we can calculate vectors:

$$A_{\mu,t}: \quad A_{1,\max_1,t} \quad A_{2,\max_2,t} \quad \cdots \quad A_{n,\max_n,t}$$

and

$$I_{c,t}: \quad I_{1,\max_1,t} \quad I_{2,\max_2,t} \quad \dots \quad I_{n,\max_n,t}$$

called index of membership functions . Here $A_{i,\max_i,t}$ is a maximum value of all membership functions designed for the feature i and $I_{i,\max_i,t}$ is the number of the best membership function for feature i.

Then we have the following candidate for a rule

$$R_t: \quad I_{1,\max_1,t} \quad I_{2,\max_2,t} \quad \dots \quad I_{n,\max_2,t}$$

The firing level of the rule is calculated according to the following formula

$$\tau_t = \overset{n}{\underset{i=1}{\phi}} \left(A_{i,\max_i,t} \right)$$

where ϕ means the operator of fuzzy matching.

The rule candidate is added to rules set if $\phi[\tau_r, m_r] \geq Th$ (where Th threshold value, and ϕ matching operator). This can help to eliminate the worst rule from the final rule set.

More ten one rule can have the same antecedent part and it is also possible that conclusion of these rules are different. Then we have to use appropriate counters $c_{t,1},.., c_{t,|S|}$, where $|S|$ denotes the power of decision class set. These counters can show us how many data, according to rule pattern, vote for each decision class.

The product is a new belief structure on X

$$\hat{m}_r = \tau_r \wedge m_r$$

Focal elements are fuzzy subset given as

$$F_{r,p}(x) = \tau_r \wedge S_{r,p}(x)$$

and appropriate distributions of new focal elements are defined as:

$$\hat{m}_r(F_{r,p}) = m_r(S_{r,p})$$

So we can build an aggregate:

$$m = \prod_{r=1}^{R} \hat{m}_r .$$

Than for each collection

$$\Im = \left\{ F_{r_1,p_1}, F_{r_2,p_2}, \ldots, F_{r_R,p_R} \right\}$$

where F_{r_t,p_t} are focal elements of $\hat{m}_r$ we have focal element E of m described as

$$E = \bigcap_{t=1}^{R} F_{r_t,p_t}$$

with appropriate probability distribution

$$m(E) = \prod_{t=1}^{R} m\left(F_{r_t,p_t}\right).$$

At this point, the rule generalization process is complete.

2.2 Test

In testing we ignore the value from the last column, that is decision class number, because our goal is to calculate it. To compute the firing level of a rule k for a given data

$$X_k : \quad X_{1,k} \quad X_{2,k} \quad \ldots \quad X_{n,k} \quad D_k$$

where $X_{i,k}$ – feature's value, D_k – conclusion decision class that we have to compare with the result of inference;

we build a rule matrix $\mu_{k,t} = \Phi_{i=1}^{n} \left(A_{i,l,k} \left(X_{i,t} \right) \right), \quad l = I_{i,\max,k}$

We are interested only in active rules i. e. rows with matching value $\mu_{k,t} > 0$.

For each collection of F_{r_t,p_t} focal elements $\hat{m}_r$ we define an aggregate

$$E = \bigcap_{t=1}^{R} F_{r_t,p_t}$$

with basic probability assignment

$$m(E) = \prod_{t=1}^{R} m\left(F_{r_t,p_t}\right).$$

The results of classification are *D is m*, with focal elements E_k $(k=1,...,R^{|S|})$ and distribution $m(E_k)$. These results are calculated using focal elements and appropriate counters $C_{t,1},..,\ C_{t,|S|}$.

Then we perform defuzzification according to COA method [1]

$$\bar{y} = \sum_{k=1}^{R^{|S|}} \bar{y}_k m(E_k)$$

where $\bar{y}_k$ are defuzzified values for focal element E_k defined as

$$\bar{y}_k = \frac{\sum_{1\le t\le n} x_t \mu_{k,t}(x_t)}{\sum_{1\le t\le n} \mu_{k,t}(x_t)}.$$

In the next step, the rules structure is simplified to

If Antecedent $_r$ Then (D is H_r),

where $H_r = \left\{\frac{1}{\gamma_r}\right\}$ is a singleton fuzzy set for factor

$$\gamma_r = \sum_{p=1}^{|S|} \bar{y}_p m_r(S_{r,p}).$$

3 Genetic Tuning of the given Rules

In Table 1 we summarize the performance of several membership functions and matching operators. We start from standard solution used in introduction to fuzzy modeling then we consider more complicated models. We compute results for the following membership function: *Linear, Quadratic* and *Gaussian.* We concentrate on *Minimum, Multiply* and *Implication* as matching operators. In general, the most interesting is *Gaussian* function and *Implication* operator. The most valuable is comparing the results of all calculations. To apply GA methods we use standard procedures from package SUGAL [3].

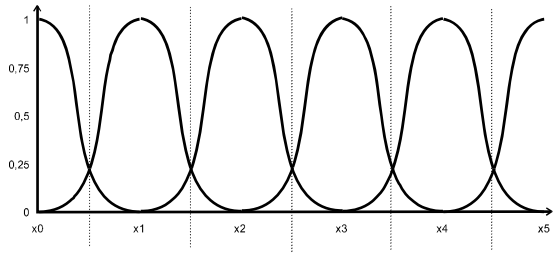


Fig. 1. Gausian Membership Function (FDS)

In the beginning we divide the space of all possible values of each feature. We assume that all of the intervals are equal – see Fig.1. But, in general, it is not a good decision. To find the points of the division we should consider the decision class for the point of data. But how we can find a decision class before running the application ? Of course it is not possible to say anything about worth of the rule set before testing. But if we have the results of the rules then we can use those results to tuning the rule set. The points of division are of particular interest. In general it is difficult to find direction of point's changes. Almost all of them seem to be the same. We can move them randomly, but every time we have to exam the given rules. If we find the rule set that can produce more accurate conclusion, then we can use these rules in the future. In opposite case we reject the rules.

This algorithm is simplifying of genetic algorithms(GA). The natural way is to use GA theory to find the better division points. First we start from base Fuzzy Dempster-Shafer model (FDS). All the intervals, that we divide feature of data, are equal. It is not optimal, but in same situations it is quite good. Next we generate new random division points and then we can test them. We generate rules by FDS algorithm using new division points. The accuracy of the rules are survive function. We use accuracy on the training data set. Using GA methods we generate next generation of population. Of course instead of only one division, we can generate more random division and then crossover them

In next point we use previous population to produce new one. After a few generations we can get division points as shown on Fig.2.

We can observe that after few generation the quality of calculated rules does not changes.

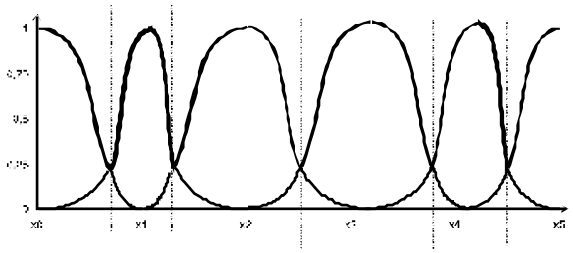


Fig. 2. Gausian Membership Function (GA)

When we analyze the results of the classification using the rule set, it is easy to see that the cost of rule generation is only few times more then base FDS. Let us show it on the sample of Dermatology data set. Data contains 366 records, 244 of then we use for learn, 122 for test. There is no floating data, but 33 discrete and only 1 binary. The rules generated by FDS have the best accuracy 51,64 for function *Gaussian* and operator Multiply. After 10 generation of GA the conclusion was good almost is 100% of cases of testing data. For function *Gaussian* and operator Implication the accuracy indicator set at 100%.

4 Conclusions

The study has focused on the use of Fuzzy Dempster-Shafer model for generating of fuzzy decision rules. Fuzzy sets are useful in discretization of continuous attributes. The approach is discussed in the concrete applications of two real medical data sets (especially to problems of identification of diseases) and few well-known data sets available on the Web [4]. The results are used to classify objects. The vehicle of Genetic Algorithms, as an additional approach for generation of the rules give us better indicator of accuracy. It can be used in the case of features with many possible values.

We compare our results with classification using Decision Trees [5]. In all points we receive better accuracy. When we use standard FSD model then for Dermatology data set we got poorer results then in Decision Trees. But after genetic tuning the rules produce much better results then previous. Sometimes we can not get an increase accuracy of the rules. It means that in that cases the change points of division has no influence on the final result.

Table1. Experimental results for testing data sets

		Linear		Quadratic		Gaussian		Decision
		FDS	GA	FDS	GA	FDS	GA	Trees
Minimum								
	Iris	93.33	93.33	96.67	96.67	93.33	93.33	91.30
	Ulcers	9.52	**16.67**	28.57	**59.52**	52.38	**59.52**	-
	Diabe-	0.00	**2.70**	37.84	**62.16**	56.76	**82.22**	-
	Derm.	0.00	**5.55**	66.39	66.39	50.82	**94.26**	87.50
	EKG	5.56	5.56	38.89	**66.67**	66.67	**99.44**	59.00
Multiply								
α = -1	Iris	0.00	0.00	96.67	96,67	93,33	93,33	
	Ulcers	0.00	**4,76**	2.38	**61,90**	2.38	**59,52**	
	Diabe-	0.00	0,00	29.73	**64,86**	54.05	**67,67**	
	Derm.	0.00	**5,55**	66,67	**77,87**	51.64	**88,52**	
	EKG	5.56	5,56	5,56	**72,22**	11.11	**100.0**	
Implication								
α = 20	Iris	83.33	83.33	96.67	96.67	96.67	96.67	
	Ulcers	9.52	9.52	28.57	**57.14**	52.38	**64.28**	
	Diabe-	0.00	**2.70**	37.84	**67.57**	56.76	**96.30**	
	Derm.	0.00	**5.55**	86.89	86.89	50.82	**100.0**	
	EKG	0.00	**5.56**	38.89	**77.78**	66.67	**88.89**	

Acknowledgement

This work was supported from the Technical University of Bialystok grant No W/WI/8/02.

References

1. J. C. Bezdek, M.J. Sabin, W.T. Tucker, "Convergence Theory for Fuzzy c-Means: Counterexamples and Repairs." *IEEE Trans. On System, Man, and Cybernetics*, 1987, vol.smc-17, no 5: p. 873-877.

2. E. Bianaghi and P. Madella "Inductive and Deductive Reasoning Techniques for Fuzzy Damper-Shafer Classifiers", *Proceeding of 1997 Seventh IFSA World Congress*, Prague, 1997, vol. 1, p.197-302.

3. Andrew Hunter , " SUGAL Genetic Algorithm Simulator" University of Sunderland 1995

4. C. J. Merz, P.M. Murphy, UCI Repository for Machine Learning Data-Bases [http://www.ics.uci.edu/~mlearn/MLRepository.html], *Irvine, CA: University of California, Department of Information and Computer Science,* 1996.

5. J.R. Quinlann, "Induction of decision trees", *Machine Learning* 1, 1986, p. 81-106.

6. Z.A. Sosnowski and J.S. Walijewski, "Generating Fuzzy Decision Rules with the Use of Dempster-Shafer Theory", *ESM'99,* Warsaw 1999; p. 419-426

7. Z.A. Sosnowski and J.S. Walijewski, "Granular Modeling and Fuzzy Dempster-Shafer Decision Rules ", 7th Workshop of the Polish Society for Computer Simulation, Zakopane ,Poland, 2000, p.273-276.

A Novel Design for Classifying Multi-Field Internet Packets Using Neural Networks

Modjtaba Rouhani[1], M.J. Solaymanpour[2], and Naser Nematbakhsh[3]

[1]Azad University of Gonabad, Gonabad, Iran
[2]Khorasan Telecom.Co., Mashad, Iran
[3]Isfahan University, Isfahan, Iran

Abstract. Classification of internet packets is a common task for internet routers. More rapid information transformation in communication links requires better processing and classifying algorithms. This paper presents a novel classification algorithm for internet packets profits parallel processing of neural networks. It is based on an extended able design of simple neural network blocks which are fast to learn and fast to respond. The classifier could be updated by re-learning simple neural networks, whenever a rule changes.

1. Introduction

Packet classification is the process of categorizing packets in an internet router into flows [1]. All packets obey a predefined rule and must treated in the same manner in router belong to the same flow. Classifiers are the collection of rules. Based on some criteria applied to a packet header, each rule specifies the flow that the packet belongs to. High speed packet classification is also essential for providing different services (*e.g.* packet filtering, policy routing, accounting, traffic rate limiting, *etc*) to different users. A good classifier is the one has high speed, needs low storage requirements, could be updated fast and is able to handle large real-life classifications.

Although most traditional classification algorithms use hierarchical trees [2-6], in this paper we introduce a novel classification design benefits from parallel processing structure of neural networks. By introducing

M. Rouhani et al.: *A Novel Design for Classifying Multi-field Internet Packets Using Neural Networks*, Advances in Soft Computing **2**, 757–761 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

neural networks, not only a higher processing speed could be achieved, but also the classifier could be updated whenever a rule has changed.

2. Internet Packet Classifier

Internet consists of routers connected together via communication links. Internet nodes communicate with each other based on *Internet Protocol* (IP). IP packets, in their path to final destination, are transferred from a router to the next router via communication links. Each router makes a decision to select next router.

In addition to capability of packet transfer, an IP router may take special decision on an incoming packet (e.g. a packet may be blocked due to security reasons; a packet may require high level of priority, or a packet may be had to sent not later than a specified delay time). To perform such decisions, routers classify incoming packets to some *flows*. All packets belongs to a flow, obey a predefined *rule* and managed similarly by router (e.g. all packet with the same source address filed may belong to the same flow). A classifier consists of a set of rules. Every rule assigns a flow to every incoming packet, based on content of its header. Header of a packet has some fields (e.g. destination address, source address, destination port, source port, protocol and service type.

Consider a classifier C with N rules R_j, $1 \le j \le N$. Each R_j has three elements:

1. To be classified by rule R_j, every of d header fields of a packet must have a mathematical representation $R_j[i], 1 \le i \le d$.
2. Priority of rule R_j is represented by $pri[R_j]$.
3. Each rule R_j corresponds to an action.

Packet P with d header field is shown as $(p_1, p_2, \ldots, p_d)$. The d-dimensional classification is defined as process of assigning the rule Rj with highest priority to packet P, such that p_i corresponds to $R_j[i], 1 \le i \le d$.

3. Neural network IP Classifier

Unlike most applications of neural networks, which need only limited precision, an internet packet classifier has to be perfect. In fact, neural net must learn all possible inputs and classify them based on their

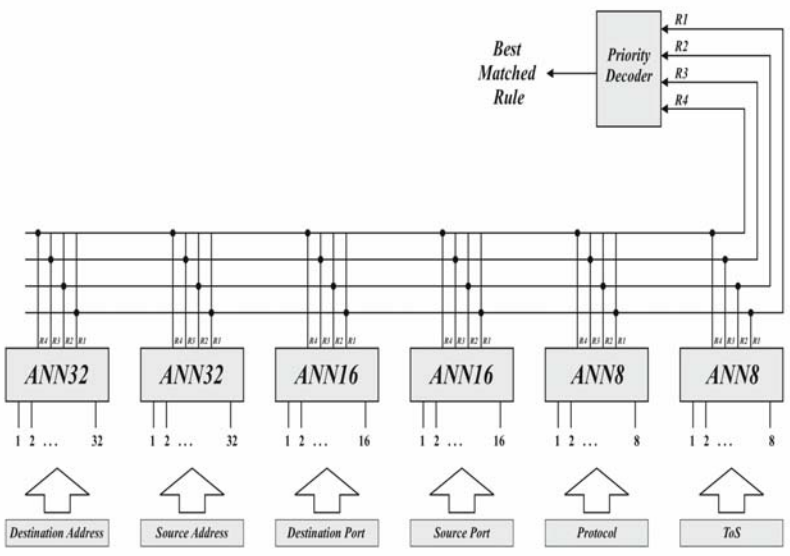


Fig. 1. Modular architecture for internet packet classifier

headers. A packet header consists of few fields is normally over a hundred bits. So it seems impossible to present all training patterns to the net.

We overcame this problem by introducing a modular architecture, as shown in figure 1. A 112-bits header consist of destination IP address (32 bits), source IP address (32 bits), destination port (16 bits), source port (16 bits) and type of service (8 bits) has been considered[3,4]. The classifier assumed to have four rules. Each field is presented to its corresponding network, which categorizes every input pattern to at least one output rule. If an input pattern belongs to more than one category, all corresponding outputs will be selected. The next stage make an *AND* operation to determine if input pattern belong to a category. The final stage determines the rule with higher priority among active ones.

Furthermore, each of these modules consists of simple two layer neural networks named *ANN8*. An *ANN8* has eight inputs and four outputs, each corresponding to one of the rules (see figure 2). There are only 128 input patterns and as there are normally much more zeros in output space, the learning is straightforward and requires few hidden neurons (figure 3).

Our experience showed that a two layer net with five hidden neurons are quit enough to achieve zero percent error. Every *ANN8* could be learned independently, thus we can learn them in parallel to reduce learning time.

The proposed classifier is compared with those previously reported [7-8] and shown to have excellent results considering speed, flexibility, and simplicity as shown in table 1 below. Learning time is measured based on our experience with a Pentium IV 1.7GHz processor and MatLab neural network toolbox. Implementation of proposed NN classifier using FPGA technology (200MHz clock), leads to a classification speed of about 40Gbps.

Table 1.

	Direct implementation of MLP [7-8]	New NN IP Classifier with modular structure
No. of classifying field	Destination address field	All fields of header
Learning time	Few hours	12-18 sec
Percentage of error	1-4 %	0%
Classification speed	60Gbps	40Gbps
Required memory	19Kb(one field)	30Kb(three fields)

4. Conclusion

In this paper, a novel neural network design for internet packet classification is presented. Due to its modular design, it is very flexible and extendable. The neural classifier consists of simple blocks named *ANN8*. Each *ANN8* has eight inputs and learns its 128 binary input space very fast. Increasing the number of rules, need more outputs for *ANN8*. Here we consider a 4 rule classifier, but it could be extended without major change in results.

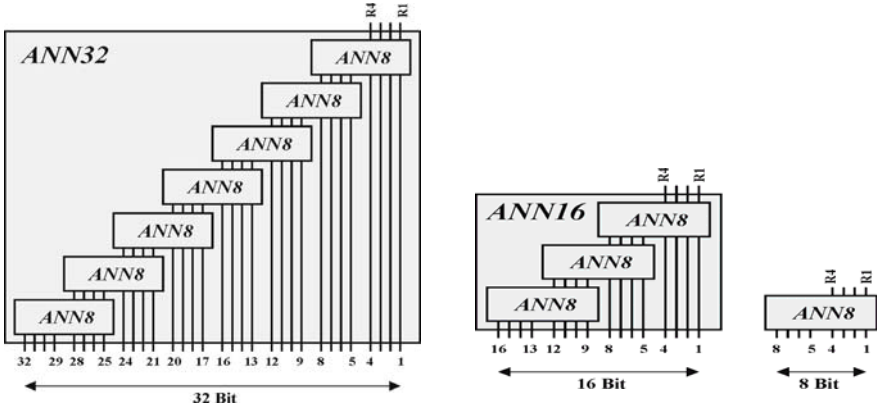


Fig. 2. Modules with different input space

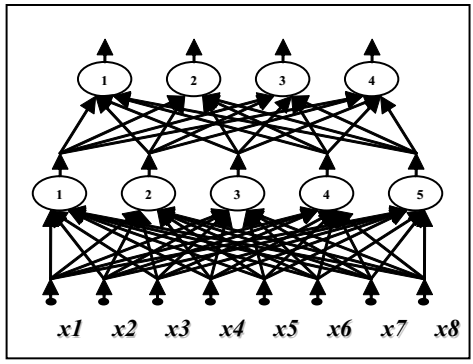


Fig. 3. *ANN8*: Two layer feedforward net with five hidden neurons

5. References

1. Pankaj Gupta, "Algorithms for routing lookups and packet classification," PhD Thesis, Ch 1 and 4, Dec 2000, Stanford University, at http://yuba.stanford.edu/pankaj/phd.html

2. F. Baker, editor. "Requirements for IP version 4 routers," RFC 1812, Jun 1995, at http://www.ietf.org/rfc/rfc1812.txt.

3. Pankaj Gupta and Nick McKeown, "Packet Classification on Multiple Fields," Proc. Sigcomm, Computer Communication Review, V29, N4, pp 147-160, Sep 1999, Harvard University.

4. Pankaj Gupta and Nick McKeown, "Algorithms for Packet Classification," IEEE Network Special Issue, V15, N2, pp 24-32, Mar/Apr 2001.

5. M. de Berg, M. van Kreveld, "Computational Geometry: Algorithms and Applications," Springer-Verlag, 2nd rev. ed. 2000.

6. M.M. Buddhikot, S. Suri, and M. Waldvogel. "Space decomposition techniques for fast layer-4 switching," Proceedings of Conference on Protocols for High Speed Networks, pp 25-41, Aug 1999.

7. M. Mahramian, N. Yazdani, k.Faez and H. Taheri, "A Multilayer Neural Network for IP Lookup and Packet Classification," Proceeding of the ninth Asia-Pacific Conference on Computer, Malaysia, Sep 2003.

8. M. Mahramian, N. Yazdani, k.Faez and H. Taheri, "Neural Network based Algorithms for IP Lookup and Packet Classification," Proceeding of the first Eurasia-ICT Conference, Shiraz, Iran, Oct 2002.

Modeling Uncertainty in Decision Support Systems for Customer Call Center

Alexander Holland

University of Dortmund, Department Computer Science I, D-44221 Dortmund, Germany
Alexander.Holland@udo.edu

Abstract. Customer call centers are the preferred and prevalent way for many companies to communicate with their customers. The customer call center industry is thus vast and rapidly expanding in terms of both workforce and economic scope. Most major companies have reengineered their communication with customers via one or more call centers, either internally managed or outsourced. Call centers constitutes a set of resources which enable the delivery of services via telephone, email or web portal access. Customer inquiries contains different types of uncertainties regarding the problem description, the recommended system solution and precise cause study. We develop a decision support system for customer call centers using soft computing techniques for automating, maintaining and maximizing the value of the decision process. Fuzzy logic as soft computing technique is a methodology for the representation and manipulation of imprecise and vague information. Bayesian networks are formal graphical languages for the representation and communication of decision scenarios requiring reasoning under uncertainty. We discuss decision support system scenarios under uncertainty using Bayesian networks and fuzzy logic. Real customer requests as support cases contain cause action coherence under uncertainty. We will model these types of uncertainty scenarios in a decision support system selecting the appropriate technique of supporting the decision process.

Key words: Call centers, decision making, decision support systems, fuzzy sets, bayesian networks, probabilities, uncertainties.

1 Customer Call Centers

Customer call centers constitutes a set of resources (personnel, computers, telecommunication equipment) which enable the delivery of services via the telephone, email or web portal access. Most call centers support Interactive Voice Response (IVR) units including the possibilities of interactions. A current trend is the extension of a call center into a contact center. The latter is a call center in which the traditional phone service is enhanced by some additional multimedia customer contact channels like fax, chat and web portal

A. Holland: *Modeling Uncertainty in Decision Support Systems for Customer Call Center*, Advances in Soft Computing **2**, 763–770 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

access. There already exist several academic surveys on call centers. Pinedo et al. [1] describes basics of call center management, including some analytical models. Anupindi & Smythe [2] introduce the technology that enables current and plausibly future call centers. Grossman et al. [3] and Mehrotra [4] both short overviews of some OR challenges in call center research and practice and Anton [5] provides a managerial survey of the past, present and future of customer contact centers. Our survey deals with call centers having a help desk functionality providing different agents characterize low-skilled, highly-trained, single and multi-skilled agents via web portal access. A developed decision support system including different soft computing techniques like fuzzy logic or Bayesian networks as graphical representation deals with different types of uncertainty.

2 Bayesian networks

In this chapter we will first give a short introduction to the basic probability calculus, which is needed for understanding the technology of Bayesian networks. We define a function P: $\epsilon \rightarrow [0,1]$ to be a probability function if it satisfies the Kolmogorov axioms. Let A and B be any two events such that $P(B) \neq \emptyset$. The conditional probability of A given B is defined by $P(A \mid B) = P(A \cap B)/P(B)$. Given the conditional probability we can introduce the Bayes rule. The Bayes rule tells how the posterior replaces the prior probability after receiving evidence, i.e. the observed data. Let $A_1, ..., A_n$ be a partition of Ω and let B be an event, such that $P(B) > 0$ and $P(A_i) > 0$ for all i, $i = 1, ..., n$. Then we can express the Bayes rule as $P(A_i \mid B)$ as

$$P(A_i \mid B) = \frac{P(B \mid A_i)P(A_i)}{\sum_{i=1}^{n} P(B \mid A_i)P(A_i)}. \tag{1}$$

The use of probabilistic models based on directed acyclic graphs apply within the field of artificial intelligence. Such models are known as Bayesian networks [6]. Their development was motivated by the need to model the top-down semantic and bottom-up perceptual combination of evidence in reading. The capability for bidirectional inferences, combined with a rigorous probabilistic foundation, were the reason for the appearance of Bayesian networks as a method of choice for reasoning under uncertainty in artificial intelligence and expert systems. A Bayesian network can be described as a graphical model for probabilistic relationships among a set of variables. It is therefore a graph in which the following holds:

- The nodes of the network represent a set of variables. The variables can be described as propositional variables of interest.
- Pairs of the nodes are connected with a set of directed links. A link represents informational or causal dependencies among the variables.

- Each node has a conditional probability table $P(A \mid B_1, ..., B_n)$ attached that quantifies the effects that the parents $B_1, ..., B_n$ have on the node.

In fact graphical models describe the distribution of a large number of random variables simultaneously. We can now represent a Bayesian network as n-dimensional discrete random variable $X_1, ..., X_n$. Each random variable X_k has the range $x_{k1}, ..., x_{km_k} \in \Re$. We can define the conditional probability of X_k given C in the following:

$$p(x_{1j_1}, ..., x_{nj_n} \mid C) = P(X_1 = x_{1j_1}, ..., X_n = x_{nj_n} \mid C) \quad \text{with} \quad j_k \in \{1, ..., m_k\}, k \in \{1, ..., n\} \tag{2}$$

or in compact form as

$$p(x_1, ..., x_n \mid C) = P(X_1 = x_1, ..., X_n = x_n \mid C). \tag{3}$$

3 Decision support systems

A decision maker come to a decision by combining his own knowledge, experience and intuition with that available from other sources. Modeling this behaviour lead to mathematical models which represent knowledge relevant to decision making processes. A model describes the modelled object by means of variables, which represent the elements of the object the model users want to study. The basic function of a decision support system (DSS) is to support the decision maker in finding values of decision variables, which will result in a behaviour of the system that meets best with the decision maker's preferences. A DSS based on a model typically uses the following concepts visualized in figure 1.

- Decision input x, controlled by the decision maker
- External decision input z, not controlled by the decision maker
- Outcomes y, used for measuring the consequences of inputs
- Model F, which represents the relations between decisions x and z, and outcomes y

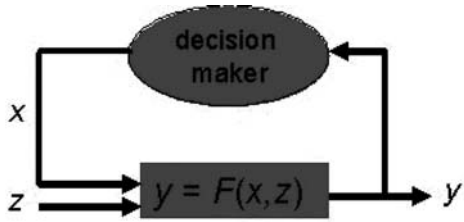


Fig. 1. Mathematical model to represent relations between decisions and outcomes

DSS are used to realize computer based systems intended to help decision makers to use data and models to identify and solve problems and to make decisions. An important task in planning and design of a DSS is selecting the appropriate technique of supporting the decision process.

4 Model uncertainty in decision support systems

As described in the section two we can develop Bayesian networks as graphical models handling with uncertainty [6]. The Bayesian network can be used in DSS to reason about the propositions that are represented by the nodes. Another approach is the integration of Fuzzy logic aspects. Fuzzy logic is a methodology for the representation of imprecise and vague information [7]. One approach in this sense is creating a fuzzy rule based system. A fuzzy rule based system has the advantages that it can represent domain knowledge in the form of rules, similar to an expert system, but it can also reason with uncertain information and perform numerical calculations.
We first develop a Bayesian network handling with uncertainty [8]. Specific user cases include different types of uncertain information regarding the user preference, the user problem description and the difficulty to detect precise problem causes.
A Bayesian approach can model this uncertainty under probability perspectives. For instance consider the following customer printing problem using a HP Laserjet 4100 series printer. The customer access a support web page having the problem scenario where printing is not possible from a specific software application. A DSS collect different problem symptoms and recommend in cooperation with additional user requested information troubleshooting suggestions. For instance to verify that the occuring problem is not a network printing problem, connect first the computer directly to the printer with a parallel cable and try to print. Based on the customers information confirm the local print successful so that the DSS next suggest to print the configuration page and verify the protocol information that server and node name match the names in the printer driver. This process scenario can built up in an underlying Bayesian network to reason about the propositions represented by the nodes. We have also developed a customer front end with integrated tree control, search functionality and full text illustrated in the following figure 2.

4.1 DSS web front end

Utilisation of DSS web front end requires different system demands. To reply customer requests the DSS offer different search scenarios while solving the problem case. A navigation tree can support the customer while navigating through the solution space and restrict the number of calculated solutions

Fig. 2. DSS customer front end

belong to a specific question. A new aspect considered in this DSS is the automatic generation of following questions based on the entry customer problem description. For instance the customer cannot print the system offer following questions like cable problems to reach the appropriate solution stored in the system database. Our aim is to recommend only one solution based on a single customer problem description. The main system requirements from the customer point of view are finding a solution as fast as possible and finding good solutions by containing the relevant documents through offering questions answered by the customer. The leading part of the architecture is to offer good and only a few solutions. The system is in an ongoing process of learning from current queries to minimize the search time of same or similar requests. If there is no solution offered by the system, there is the uncomplicated and quick way to contact a call center agent embedded via a specific system interface. In this case, the agent can track previous questions answered by the customer and carry on the next solution steps.

From the structural point of view we differentiate between questions, answers and appropriate solutions. Determining the solution documents stored in the database regarding primary and secondary documents. Primary documents are extensions for which we store the solution navigation way, special keywords regarding the solution, last call information or count of all calls concerning this solution. Secondary documents contains no additional information like keywords. The reason for this differentiation is easy to understand. In the economy are monetary aspects fundamental. Rare specific solutions never or

only few requested would increase the number of cases taking into consideration. For instance different operating systems like HP UX 10.20, HP UX 11.00 or HP UX 11.11 implemented on customers workstations require different patches and different applications. It makes no sense to buil up all possible combinations of cases in a DSS. Another reason is the possibility that specific problems occur only on a specific system for one customer. We can define a specific threshold ϵ as measurement for requested documents. If the number of requested secondary solutions exceed this threshold, we can take over this documents in the primary document list. The search template is the central working space. The customer can place a new query enter in a search field or traverse a navigation tree and scroll through different problem areas like for instance general printers, next HP LaserJet printer family and next to specific HP LaserJet printers like 5P, 5,L or 6L. Here the customer can choose between different issues like "What to do when print media does not feed correctly into the HP LaserJet 6L printer family?" or "Why am I unable to print to my LaserJet printer when shelling from DOS (accessing the DOS prompt within Windows)?". The queries are associated with one or more other questions or solutions as child nodes. A history function provide an opportunity to return to a previous point of search within the navigation tree. The search field allow the following entries: case identification, keywords, phrases and complete questions(see figure 3). Based on the entry request the

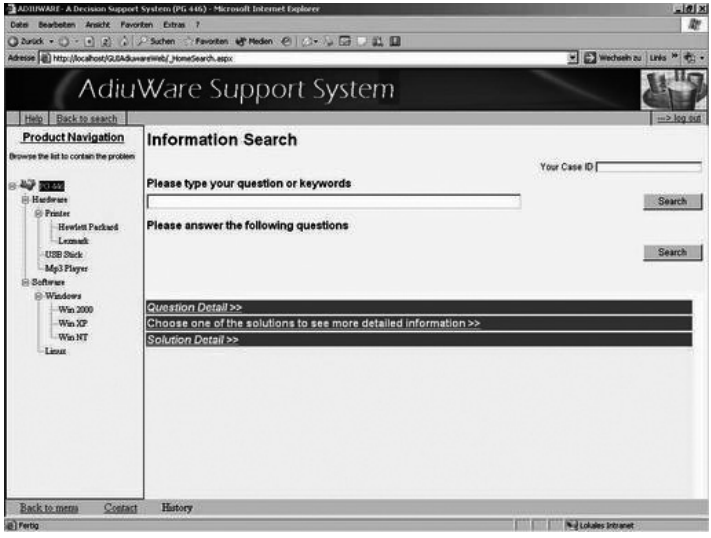


Fig. 3. Information search by question or keywords

system offer the following questions answered by the customer. If the system could not offer a solution the customer can start a secondary search. Now the solution space contains only documents with less relevance which can cover

also other problem areas (for instance operating system solutions in the case of network printing problems).

4.2 Search and Indexing

To allow an efficient way to detect solution documents within the database match with customer requests we must first identify document keywords. We administrate the position of each keyword in the document as an integer array. The indexed documents affect both solutions and cases. The keyword list can increased through manual word elements specified by the system administrator. Our DSS apply lookup lists as inverted index lists to identify keywords. Lookup lists contain keywords in alphabetical order with a reference to the document or case ID. We support lookup lists regarding the primary solutions and secondary solutions as ccCases. The search algorithm can calculate first primary and next secondary solutions.

4.3 Inference

We use Bayesian networks in our DSS to calculate new pobabilities when achieve particular information. For instance let A have n states with $P(A) = (x_1, ..., x_n)$ and assume that we get the information e that A can only be in state i or j. This statement expresses that all states except i and j are impossible, so we can illustrate the probability distribution as $P(A, e) = (0, ..., 0, x_i, 0, ..., 0, x_j, 0, ..., 0)$. Assume we have a joint probability table $P(U)$ and $\underline{e}$ is the preceding finding (n-dimensional table of zeros and ones). Using the chain rule for Bayesian networks [6], we can express the following

$$P(U, e) = \Pi_{A \in U} P(A \mid parents(A)) \cdot \Pi_i \underline{e}_i \tag{4}$$

and for $A \in U$ we have

$$P(A, e) = \frac{\Sigma_{U \setminus \{A\}} P(U, e)}{P(e)}. \tag{5}$$

The inference implemented in our DSS is a recursive algorithm based on the exact inference well known from Bayesian networks. The DSS Bayesian network contains solutions as leaf nodes and questions for all other nodes. The conditional probability tables of the solutions contains an additional column expressing the utility values to determine the ongoing question order. The evidence list hold question nodes and appropriate states, specific all questions respond before. The algorithmn calculate the probability of each node in summarizing the probabilities of the states of all parent nodes. We can understand our DSS as extension of standard Bayesian aprroaches regarding the utility values. In the first step we calculate based on the occurrence of question or solution nodes utility values (for questions) or probability values for the solutions as in the standard Bayesian case. This scenario occurs only

in the first call. In the recursive case the algorithm detect only question nodes based on the recursive behavoir in calculating probabilities for parent nodes. Declare states for the parent nodes and calculate the probability for certain answers. We apply these probabilities to compute the utility value for the original called node.

5 Conclusions

The Decision support system outlined above has been implemented and used as experimental system for customer call centers handling hardware and software requests for first and second level support cases. Most of the today's computer-based decision support has focused on the support for the behavioral aspects of decision making and extensions of the analytical capabilities of decision makers. We built up and model the uncertainty aspect occurring in customer call center scenarios. Through the differentiation of primary and secondary solutions we increase the achievement while reducing the system complexity having a look at more relevant solutions in the first step. The Bayesian network was implemented through question and solution nodes as leaf nodes. Further work will be done with other problem domains and the contemplation of dependences between different problem domains, for instance hardware components and their influence with specific customer applications. Another consideration is to model possibilistic networks and learn global structures based on the available customer cases.

References

1. Pinedo, M.; Seshadri, S.; Shanthikumar, J.G. (1999) Call centers in financial services: Strategies, technologies and operations. Kluwer.
2. Anupindi, R.; Smythe, B.T.(1997) Call centers and rapid technology change. Teach Note.
3. Grossman, T.A.; Samuelson, D.A.; Oh, S.L.; Rohleder, T.R.(1999) Call centers. Encyclopedia of Operations Research and Management Science, 2nd edition.
4. Mehrotra, V.(1997) Ringing up big business. OR Today.
5. Anton, J.(2000) The past, present and future of customer access centers. International Journal of Service Industry Management, 11:120-130.
6. Russel, S.; Norvig, P.(2003) Artificial Intelligence - A modern approach. Prentice Hall Inc.
7. Ruspini, E.; Bonissone, P.; Pedrycz, W.(1998) Handbook of Fuzzy Computation. Institute of Physics Publishing.
8. Das, B.(1999) Representing uncertainties using Bayesian networks. Electronics and Surveillance Research Laboratory, Australia.

A New GA-Based Real Time Controller for the Classical Cart-Pole Balancing Problem

N. Seifipour

Electrical engineering Department., Azad University, Tehran, Iran

M. B. Menhaj

Department of Electrical Enginnering, Amirkabir University, Hafez Ave. #529, Tehran-Iran,
menhaj@aut.ac.ir

Abstract

This paper introduces a new application of the genetic algorithm for on-line control application. It acts as a model free optimization technique that belongs to the class of reinforcement learning. Its concepts and structure is first investigated and then the ability of this algorithm is highlighted by an application in a real-time control (pole balancing) problem. The simulation results approves the better the merit of the proposed technique.

1 Introduction

Most traditional optimization methods employed in science and engineering can be divided into two broad categories, direct and gradient search techniques [1]. The methods in the first category need only the objective function values while the techniques in the second category require only gradient information either exactly or numerically with the common characteristic that they all work on a point-by-point basis, i.e. they begin with an initial solution (usually supplied by the user) and a new solution is calculated according to the steps of the algorithm. In general, direct search methods however require no gradient calculations, are computationally extensive and in most cases they work on some simple unimodal objective functions. On the other hand, gradient-based techniques require the know-

N. Seifipour and M.B. Menhaj: *A New GA-Based Real Time Controller for the Classical Cart-Pole Balancing Problem*, Advances in Soft Computing **2**, 771–786 (2005)
www.springerlink.com © Springer-Verlag Berlin Heidelberg 2005

ledge of gradients of functions and constraints and in most cases they suffer from the problems of gradient calculations and local optimality.

The Genetic Algorithm is in-depth different from the two major classes of classical calculus-based and enumerative techniques [3-8]. Because of its ease of implementation procedure, the GA has been successfully used as an optimization tool for a variety of optimization problems. Some of these applications are machine learning [9], delay estimation of sampled signals [10-11], robotic trajectory planning [12], fuzzy controllers [13-16], neural networks [17-20], linear transportable process [21], identification and control [22-23], NP-hard problems [24], job shop scheduling [25] and so on. The GA is good for ill-defined or multi-modal cost functions in which local optima are frequently observed. Since it tends to seek the global optimum solution without getting trapped at local minima.

The applications of GA for on-line problems are few in number, however GA is useful and the volume of literature published on this subject, i.e. applications GA in complex, multivariable, and real-world problems, is remarkable. However the conventional form of GA implemented so far, as the authors know, could not be guaranteed to yield an on-line optimum solution, and in fact oftentimes performed rather poorly. The modifications to the conventional GA proposed in this paper make the GA become (more) suitable for on-line problems.

Since population plays a crucial role in natural evolution, the genetic algorithms inspired by natural biologics relies on population. This factor, however produces some problems, whose optimal cases have not been specified yet. Some of these problems are as follows:

- 1. Optimal population size,
- 2. Efficient selection method of individuals in population,
- 3. Efficient crossover operator,
- 4. Efficient fitness function,
- 5. Suitable function converting minimization problems to the maximization ones,
- 6. Quality of using individual's age in the genetic process,
- 7. How to use GA in problems with two sides of fitness functions that have both, positive and negative values.

Many researchers have employed different ways to solve the above problems by proposing some corrections on the procedure of conventional GA, i.e., optimization of control parameters of GA [26], population size [27-29], fitness function [30], aging of individual [31], mutation operator [32-33], crossover operator [34-36], migration operator [37-38], and multi-population [39-40].

Nevertheless, these corrections have been dependent on the problem, which GA is implemented. In other words, these corrections may have

good performances in some cases while have bad performances in other ones. Another restriction of the conventional GA, is that it acts in off-line situations and then could not been used for on-line applications.

This paper introduces an On-line Genetic-Based Algorithm (OGA) as an emerging optimization technique for on-line signal processing. The remainder of the paper is organized as follows. Section 2 fully describes the OGA. In Section 3; the application of OGA for control is then presented. To show the effectiveness of the proposed genetic algorithm, in section 4, through an example (cart-pole balancing problem) the performances of OGA is investigated. Finally, section 5 concludes the paper.

2 THE OGA

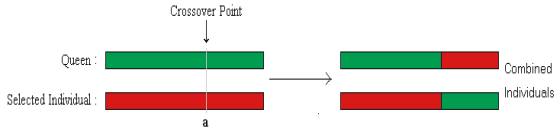
The genetic algorithm was proposed as a novel algorithm mainly based on random search technique for optimization. The reason behind choosing GA name is the fact that the evolution in each biological process is based on heredity. Many researchers [26-40] showed deep interest in GA and improved its performance in various aspects such as convergence speed and properties. It is worth mentioning that these modifications make the GA improves its performance on some problems. Although some of these modifications caused the GA converge to the optimal points in less iteration, the amount of computations at any iteration gets increased and further they are not well suited for each problem.

To distinguish the proposed GA based method which will be fully discussed in this section with all other GA based techniques developed so far, we call all the existing GA methods by Conventional Genetic Algorithms (CGA). The CGA may be completely described by the following steps, [41]:

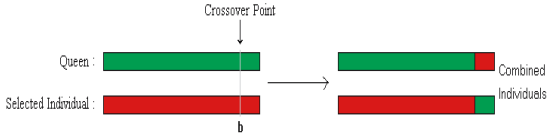
- 1- Select a coding scheme to represent adjustable parameters,
- 2- Initialize and evaluate a population of chromosomes,
- 3- Repeat the following steps till stopping criteria are met,
- 3-1- Reproduction,
- 3-2- Crossover,
- 3-3- Mutation,
- 3-4- Population evaluation.

The proposed genetic algorithm is inspired by unusual human recombination. To make it more clear suppose as the CGA, produce a population of individuals by stochastic selection at response space. After that all of individuals in population are evaluated by using a suitable fitness function, identify the best individual from present population and call it *queen* of the

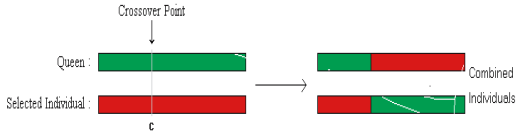
population. Similar to the CGA, individuals are selected by a method as roulette wheel lemma to doing recombination [41]. Against in CGA that selected individuals are recombined two by two with each other, in proposes GA, any selected individual is only recombined with the queen. It is visible in Figure 1. It shows one individual in each recombination process is queen while another individual is randomly selected from the population. After completion of queen with selected individuals, mutation operator will be accomplish and then, as CGA, all of individuals of population are evaluated by fitness function and maybe new queen are obtained. Then again above process will be repeated.



(a) Crossover point selected at point **a**



(b) Crossover point selected at point **b**



(c) Crossover point selected at point **c**

Fig. 2.1. The proposed unusual human recombination

Just as it is seeing, the striking difference between the propose GA with the CGA is that instead of combination of selected individuals in form of two by two in CGA, the selected individuals are only combined by present queen in propose GA. Of course by this change, our expectancy are that the propose GA is trapped fast in local response. For preventing of this occurrence, probability of doing the mutation operator, P_m, must bigger in compare to the CGA. The combination process that is shown in Fig. 2.1 is investigated precisely. Just as it is seeing in the figure, all of produced individuals are really queen that a part of them are changed. In other word, new individuals or chromosomes present some individual is queens that

inherit some genes from main queen. In another word, all of combined individuals can consider as Fig. 2.2.

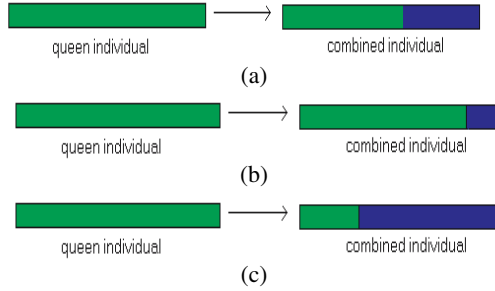


Fig. 2.2. The conversion procedure of queen individual into a new individual

After this, as done in CGA, to enter new independent genes to the population, the mutation operator is performed, see Fig. 2.3.

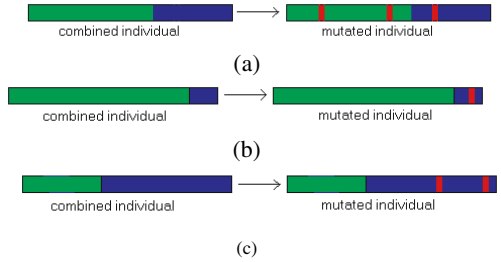


Fig. 2.3. After Mutation

Eventaullay all the produced individuals at the end of one generation cycle will quite resemble the queens, see Fig. 2.4.

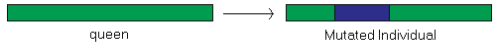


Fig. 2.4. Production of new individuals from the queen

By a close look at the above figures, it is clear that the present queen will change, and the final individual at the end of a generation cycle becomes a mutated form of the present queen, see Fig. 2.5.

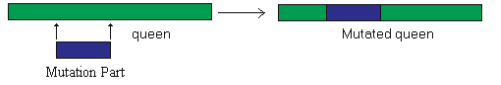


Fig. 2.5. Process of Production new individuals from Queen

This is the main idea of the OGA. At the end of each generation cycle, all mutated queen will be evaluated and the best of them will be compared with the present queen. Now if the best mutated queen outperformes the present queen, it will be choosen as the new queen for the next generation cycle. If the size of population is reduced to one, the on-line genetic algorithm is made The OGA is summarized below.

- 1- Select a coding scheme to represent adjustable parameters
- 2- Initiate a chromosome and evaluate its performance and choose it as the queen
- 3- Repeat the following steps until stopping criteria are met
- 3-1- Create a candidate queen by *mutation* of the present queen
- 3-2- Evaluate the candidate queen
- 3-3- Choose it as the queen of the next generation cycle if it outperforms the old one.

3 The Proposed OGA- Based Control

In general, as shown if Fig. 3.1 the structure of the adaptive closed-loop control system has two parts. One part contains model reference by input-output information (r,y^d), where y^d is the desired output with respect to reference nput signal r, and the other part contains plant and controller. The second part must behave like the reference model.

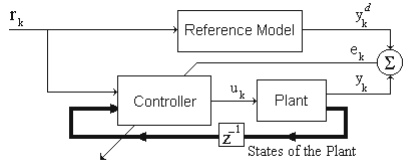


Fig. 3.1. The Adaptive Control Scheme

By modifying the general structure of adaptive control system, the proposed OGA based adaptive controller is given in Fig. 3.2. The OGA tunes the controller. The top part represents the reference model with the input-output pairs of (r,y^d). The other bottom part also is made up of two parts. One part performs the control task and the other performs the prediction task. The controller is tuned so that the output signal y_k tracks the desired

output, y_k^d and the predictor will tune the controller. Below these two tasks are fully explained.

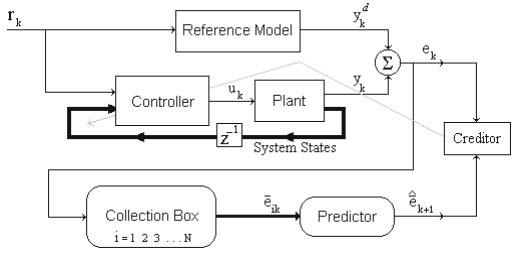


Fig. 3.2. The proposed control process related to OGA

3.1 The Control Task

The adaptive controller, which is aimed to be designed so that the overall closed-loop system acts like the reference model, can be structured by classical techniques such as PID, pole-placement, LQG or by intelligent techniques such as neural networks, Fuzzy networks, Neuro-Fuzzy networks. After choosing a suitable structure for the controller, its adjustable parameters will be tuned by the OGA. In the first step, the OGA will randomly select parameters of the controller and consider it as a queen. Then, the motion of the plant will begin at some arbitrary initial state X_0, and continue to a specified period of time, i.e. T seconds. During this period of time, the tracking errors, differences between the outputs of the plant and those of the reference model, will be collected in the Collection-Box. This box produces the input-output training data of the predictor. After training of the predictor, its output represents the average of tracking errors of the next τ seconds, $\hat{\bar{e}}_{\tau}$, (note that τ is a random number in (0,T]). At the end of τ seconds, a new queen candidate will be produced and the plant moves with this new controller, for the next τ seconds. After some iterations, if the total average errors between outputs of the plant and the reference model $\bar{e}_{\tau}$ becomes less than that of of the predictor, $\hat{\bar{e}}_{\tau}$, the new queen candidate will be selected for the next T- seconds. Otherwise, the old queen will not be replaced for the next cycle. Note that only in the present T seconds, information is collected in the Collection-Box and delivered to the predictor, while in the next τ seconds, performance of the new queen candidate

will be evaluated. It is evident that the controller's parameters are tuned with a policy of reward and punishment.

3.2 The Prediction Task

The goal of designing a predictor is to estimate an average tracking error of the current τ seconds by using the data collected during the last T seconds, ($\tau \leq T$). Note that the tuning of the controller is an individually oriented based procedure; it means that the predictor sees whether the queen has to be kept in servicefor the next cycle or it must be replaced with the queen candidate of the present τ-seconds. The predictor estimates the expected value of the tracking errors for the present τ seconds by using information obtained from the queen controller in the previous τ seconds, while the candidate queen controller remains unchanged during this period of time. Now, if the mean of tracking errors produced by the candidate queen controller gets less than what the predictor has been estimated, then the queen candidate controller replaces the last queen controller, otherwise the candidate queen controller fails. This means that the last queen controller represents the winner and remains active.

Note that the process of training and that of the of predictor can be done in parallel. In other word, two independent processors implement simultaneously the control and predictor tasks. Neural networks, fuzzy logic or neuro-fuzzy structures may represent the structure of the predictor. This paper uses MLP based neural networks to represent the predictor. Initially, the parameters of predictor are chosen randomly. Thus, it is necessary to train the NN during the control process. For purpose of training we are required to have a desired set of input/output data pairs. This can be done as follows.

Input-Output Information for Predictor training: Suppose that the plant affected by the queen controller from time interval t_o to t_o +T; where T is a given duration time of movement of the plant. Now, with a collection of tracking errors at the interval [t_o , t_o +T] at the Collection-Box, the input-output training data, will be collected. If we can arrange th collected data so that the predictor learns the average of the next section through information gathered from the present section, the input-output training data will be made. For example, assume that the predictor uses four previous successive errors to estimate the next error. Then, the neural networks will have four inputs and one output. Divide the time T into four subsections. Each of these sections is in turn divided into four other sub-

sections as shown in Fig. 3.3. Note that if the number of predictor inputs were n, we would have n subsections in each partitioning.

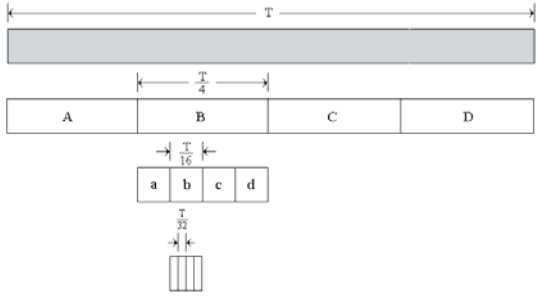


Fig. 3.3. The dividing procedure for a time interval of length T

If the expected value of errors for the time interval 'B' represents the desired output of the predictor, then we have to consider time interval 'A' whose length is exactly the same as that of the time interval 'B' as a region, in which the training data of the predictor are chosen. To do so, the subsection 'A' is divided into four parts and the expected values of these four parts make the inputs of the predictor. Also, if we want the expected value of errors at time interval 'd' be the desired output of the predictor, then the time interval 'c' must be divided into four subsections and the expected values of these subsections are to be considered as the inputs of the predictor. Consequently, by repeating this procedure we can end up with a set of input-output data pairs to train the predictor.

During the present T seconds that the predictor learns to predict the expected value of tracking errors, the plant can independently operates with the new candidate queen controller. In other words, we have results of both the predictor and the new candidate queen controller in order to evaluate the performance of the last candidate queen controller. For instance, if we want the predictor to estimate the expected value of errors at the present T-second time inerval, the inputs of the predictor must be the expected values of errors at four subsections just before this time interval, as shown in Fig. 3.4.

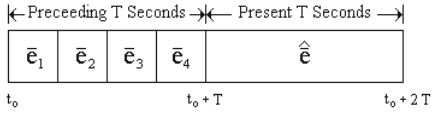


Fig. 3.4.. Inputs and Output of the predictor for $\tau = T$

4 Control of Cart-Pole balancing system

In this section the well-known classical pole-balancing problem originally introduced by [42-43] is simulated and served to evaluate the OGA performance. This problem was also simulated by Berenji [42], Lin and Lee [43]. Since this problem shows the system operation in concrete form and can be used as benchmark to compare the performance of OGA with that of the Berenji's method, it will be described in some detail in this section.

As shown in Fig. 4.1, a motor driven cart with a vertical pole attached is placed on a short rail track. The cart is free to move to its right or left but not forward or backward and the pole has only one degree of freedom of rotating about the hinge pivot point. The primary controller's task is to keep the pole balanced vertically as long as possible by moving the cart to its both right and left with impulsive pushes without moving the cart beyond the rail track boundaries. The equations of motion are determined by the parameters: pole length and mass, cart mass, friction in the pole-cart hinge, friction between cart and track, force of gravity, impulsive driven force applied to the cart, and simulation time step, and are modeled by:

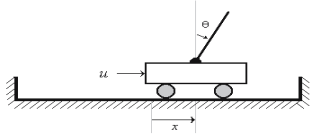


Fig. 4.1. The cart-pole balancing system

$$\ddot{\theta}(t) = \frac{m \cdot g \cdot Sin\theta(t)}{(4/3) \cdot m \cdot l - m_p \cdot l \cdot Cos^2\theta(t)} - \frac{Cos\theta(t) \cdot \left(u(t) + m_p \cdot l \cdot \dot{\theta}^2(t) \cdot Sin\theta(t) - \mu_c \cdot Sign(\dot{x}(t))\right) - \frac{\mu_p \cdot m \cdot \dot{\theta}(t)}{m_p \cdot l}}{(4/3) \cdot m \cdot l - m_p \cdot l \cdot Cos^2\theta(t)}$$

$$\ddot{x}(t) = \frac{u(t) + m_p \cdot l \cdot \left(\dot{\theta}^2(t) \cdot Sin\theta(t) - \ddot{\theta}(t) \cdot Cos\theta(t)\right) - \mu_c \cdot Sign(\dot{x}(t))}{m}$$

where

1) g = -9.8m/s^2, acceleration due to gravity.
2) m = 1.1 kg, combined mass of the pole and the cart.
3) m_p = 0.1 kg, mass of the pole.
4) l = 0.5 m, half-pole length.
5) μ_c = 0.0005, coefficient of friction of the cart on the track.
6) μ_p = 0.000002, coefficient of friction of the pole on the cart.

For control of a system, at first we need to choose kind of inputs that must feed to the controller. In the cart-pole problem, there are four states θ, $\dot{\theta}$, x and $\dot{x}$. On the other hand, the cart-pole has only one input, and then with this input we can control only θ or x. In this experiment, we want to control the angle of pole, θ , then we cannot control the position of the cart. Therefore, the position of the cart isn't important and it won't have any effect on the angle of cart. But, by noting to the dynamics of the cart-pole system, we can see that the velocity of the cart, $\dot{x}$, affects the angle of pole. Therefore, we can choose the inputs of the controller as θ, $\dot{\theta}$ and $\dot{x}$.

There are some constraints on the variables of the cart-pole system as, "$-12° \le \theta \le 12°$, $-2.4m \le x \le 2.4m$" and "$-20N \le u \le 20N$" and assume the dynamic equations of the cart-pole balancing system are not known. Furthermore, if the cart is in $|x| \ge 2.4\ m$, it is set to x=0.

It is remarkable to note that there is no information about parameters of the system. Practically, the system runs, till a failure occurs. The system fails and receives a penalty signal of -1 when the pole falls down to the vertical bound (±12° is used here). On the other hand, to design the controller with the OGA method, we assume that the dynamic equations of system are unknown.

Author use the multilayer feedforward neural networks structure for the controller. The inputs of this controller are θ, $\dot{\theta}$ and $\dot{x}$ and the neural network has 8 neurons in the hidden layer with hyperbolic tangent activation functions and one linear neuron in the output layer. The goal here is to show how the controller can produce a sequence of forces with proper magnitudes in such a way that it can balance the pole as long as possible without failure. For the control of cart-pole system, the OGA based control strategy that mentioned in section 3 is applied with the adaptive structure in Fig. 4.2. After some failures of cart-pole system, it became balance and there was not any new queen candidate to win.

The vertical angle of the pole and force to the cart (output of controller) are shown in Fig. 4.3 and Fig. 4.4, respectively. Easily observed, the proposed controller could balance the pole. Therefore, the system learned to solve the complex cart-pole balancing problem with no knowledge of the dynamics of the system (the system outputs must be available at each time instant). From the simulation results, one can easily judge the merit of the proposed OGA.

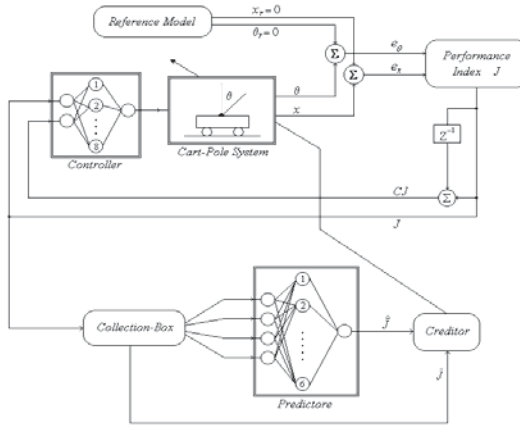


Fig. 4.2. Neural networks control process for Cart-Pole system

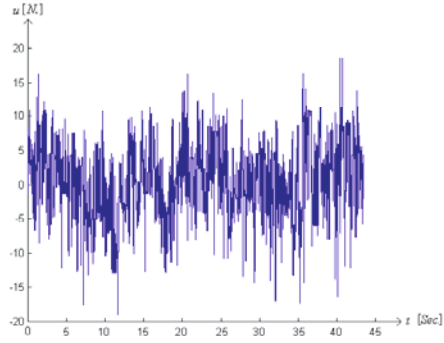


Fig. 4.3. Control effort r

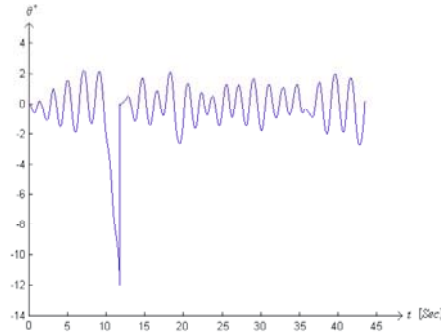


Fig. 4.4. Trajectory of the Angle of Pole

6 Conclusion

This papers introduces a new genetic algorithm. This algorithm is called On-line Genetic Algorithms (OGA) that can be used to solve problems such as optimization and real time control. By this technique, without any knowledge about dynamic equations of the plant, the controller can be trained so that the closed-loop system tracks a desired trajectory.

7 References

1. D. G. Luenberger, Linear and Nonlinear Programming, 1989, New York
2. J. H. Holland, Adaptation In Natural And Artificial Systems, Ann Arbor, MI: The University of Michigan Press, 1975
3. E. Scales, Introduction to Nonlinear Optimization, Springer-verlag, N. Y. 1985
4. R. E. Bellman, Dynamic Programming, Princetion univ. Press, princetion, N. J., USA, 1957.
5. K. A. DeJong, "Genetic Algorithms: A 10 Years Perspective," Proceedings of an International Conference on Genetic Algorithms and Their Applications, pp. 169-177, 1985
6. J. J. Grefenstetle, "Optimization of Control Parameters for Genetic Algorithms," IEEE Trans. on Systems, Man and Cybern., SMC-16, No. l, pp. 122-128, Jun-Feb., 1988
7. D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Publishing Company 1988
8. L. Davis and M. Steenstrup, "Genetic Algorithms & Simulated Annealing," In GAs & SA, L. Davis, Ed., Pitman, London, UK, pp. 1-11,1987.
9. D. B. Fogel, System Identification Through Simulated Evolution: A Machine Learning Approach to Modeling, Needham Heights, MA: Ginn Press, 1991.
10. Y. T. Chan, J.M. Riley and J.B. Plant, "Modeling of Time Delay and it's Application to Estimation of Nonstationary Delays," IEEE Trans. Acoust. Speech, Signal Processing, Vol.: ASSP-29, pp. 577-581, June 1981.
11. D. M. Elter and M.M. Masukawa, "A Comparison of Algorithms for Adaptive Estimation of The Time-Delay Between Sampled Signals," proce. of IEEE int. conf. on ASSP., pp. 1253-1256, 1981.
12. Y. Davidor, Genetic Algorithms & Robotics, A Heuristic Strategy for Optimization, World Scientific, Singapore, 1991.
13. C.L. Karr, "Genetic Algorithms for Fuzzy Controllers," A.I. Expert, Vol. 6, No. 2, pp. 26-33, 1991.

14.D. Park, A. Kandel, and G. Langholz, "Genetic-Based New Fuzzy Reasoning Models with Application to Fuzzy Control," IEEE Trans. on Sys., Man, and Cyb., Vol. 24, No. 1, January 1993.
15.H. Nomura, I. Hayashi and N. Wakami, "A Self Method of Fuzzy Reasoning by Genetic Algorithms," in Fuzzy Control System, Kandel and Langholz (ed), CRC Press, Inc., London, pp337-354, 1994.
16.D. A. Linkens, and H. O. Nyongesa, "Genetic Algorithms for Fuzzy Control Part 2: Online System Development and Application," IEE Proc.-Control Theory Appl., Vol. 142, No. 3, May 1995.
17.D. J. Montana and L. Davis, "Training Feed-forward Neural Networks Using Genetic Algorithms," Proc. of Int. Joint Conf. on Artificial Intelligence (Detroid),
pp. 762-767, 1989.
18.F. Z. Brill, D. E. Brown, and W. N. Martin, "Fast Genetic Selection of Features for Neural Networks Classifiers," IEEE Trans. on Neural Networks, Vol. 3, No. 2, March 1992.
19.V. Maniezzo, "Genetic Evolution of The Toplogy and Weight Distribution of Neural Networks," IEEE Trans. Neural Networks, Vol. 5, No. 1, pp. 39-53, Jun 1994.
20.K.S. Tang, K.F.Man and C. Y. Chan, "Genetic Structure for NN Topology and Weight Optimization," 1st IEE/IEEE int. conf. on GAs in Engineering System, innovations & Applications, pp. 280-255, 12-14, Sept. 1995.
21.GA Vignaun and Z. Michalewicz, "A Genetic Algorithms For The Linear Transportation Problem," IEEE Trans. on Systems Man and Cybernetics, Vol. 21, pp. 445-452, March-April, 1991.
22.K. Kristinsson, and G. A. Dumont, "System Identification and Control Using Genetic Algorithms," IEEE Trans. on Sys., Man, and Cyb., Vol. 22, No. 5, September/October 1992.
23.A. Varsek, T Urbancic, and B. Filipic, "Genetic Algorithms in Controller Design and Tuning," IEEE Trans. on Sys., Man, and Cyb., Vol. 23, No. 5, September/October 1993.
24.F. T. Lin, C. Y. Kao, and C. C. Hsu, "Applying the Genetic Approach to Simulated Annealing in Solving Some NP-Hard Problems," IEEE Trans. on Sys., Man, and Cyb., Vol. 23, No. 6, November/December 1993.
25.K. Morikawa, T. Furuhashi, and Y. Uchikawa, "Single Populated Genetic Algorithms and its Application to Jobshop Scheduling," Proc. of IEEE, International Conf. on Power Electronics and Motion Control, Vol. 2, pp. 1014-1019, 1992.

26.J. J. Grefenstette, "Optimization of Control Parameters for Genetic Algorithms," IEEE Trans. on Sys., Man, and Cyb., Vol. Smc-16, No. 1, January/February 1986.
27.M. O. Odetayo, "Optimal Population Size for Genetic Algorithms: An Investigation," IEEE, Colloquium on Genetic Algorithms for Control Systems Engineering, pp. 2/1-2/4, 1993.
28.J. T. Alander, " On Optimal Population Size of Genetic Algorithms," CompEuro '92. Computer Systems and Software Engineering', Proc. pp. 65-70, 1992.
29.J. Arabas, Z. Michalewicz, and J. Mulawka, "GAVaPS - a Genetic Algorithms with Varying Population Size," Evolutionary Computation, IEEE World Congress on Computational Intelligence, Proceedings of the IEEE Conf., Vol. 1, pp. 73-78, 1994.
30.J. A. Lima, N. Gracias, H. Pereira, and A. Rosa, "Fitness Function Design for Genetic Algorithms in Cost Evaluation Based Problems," Proc. of IEEE, International Conf. on Evolutionary Computation, pp. 207-212, 1996.
31.A. Ghosh, S. Tsutsui and H. Tanaka, "Individual Aging in Genetic Algorithms," Conf. on Intelligence Information System, Australian and New Zealand, pp. 276-279, 1996.
32.J. Hesser and R. Manner, "Towards an Optimal Mutation Probability for Genetic Algorithms," in Proc. First Int. Workshop on Parallel Problem Solving from Nature, Dortmuntd, 1990, Paper A-XII.
33.X. Oi and F. Palmieri, "Theoretical Analysis of Evolutionary Algorithms With an Infinite Population Size in Continuous Space Part I: Basic Properties of Selection and Mutation," IEEE Trans. on Neural Networks, Vol. 5, No. 1, January 1994.
34.X. Oi and F. Palmieri, "Theoretical Analysis of Evolutionary Algorithms With an Infinite Population Size in Continuous Space Part II: Analysis of the Diversification Role of Crossover," IEEE Trans. on Neural Networks, Vol. 5, No. 1, January 1994.
35.Y. Shang and G. J. Li, "New Crossover in Genetic Algorithms," Proc. of IEEE, Third International Conf. on Tools for Artificial Intelligence, TAI '91, pp. 150-153, 1991.
36. M. Coli, G. Gennuso, P. Palazzari, "A New Crossover Operator for Genetic Algorithms," Proc. of IEEE, International Conf. on Evolutionary Computation, pp. 201-206, 1996.
37.J. C. Potts, T. D. Giddens, and S. B. Yadav, "The Development and Evaluation of an Improved Genetic Algorithms Based on Migration and Artificial Selection," IEEE Trans. on Sys., Man, And Cyb., Vol. 24, No. 1, January 1994.

38.M. C. Moed, C. V. Stewart and Robert B. Kelly, “Reducing The Search Time of A Steady State Genetic Algorithms Using the Immigration Operator,” Proc. of IEEE, Third International Conf. on Tools for Artificial Intelligence, TAI ’91, pp. 500-501, 1991.
39.S. Tsutsui and Y. Fujimoto, “Phenotypic Forking Genetic Algorithms (p-fGA),” Proc. of IEEE, International Conf. on Evolutionary Computation, Vol. 2, pp. 566-572, 1995.
40.S. Tsutsui and Y. Fujimoto and I. Hayashi, “Extended Forking Genetic Algorithms for Order Representation (o-fGA),” Proc. of the First IEEE, Conf. on IEEE World Congress on Computational Intelligence, Vol. 2, pp. 566-572, 1994.
41.L. Davis, Handbook of Genetic Algorithms, Van Nostrand Reinhold, 1991.
42.H.R. Berenji and P. Khedkar, “Learning and Tuning Fuzzy Logic Controllers Through Reinforcements,” IEEE Transactions on Neural Networks, Vol. 3, No. 5, 724-740, 1992.
43.C. T. Lin and G. Lee, “Reinforcement Structure/Parameter Learning for Neural-Networks-Based Fuzzy Logic Control System,” IEEE Trans. on Fuzzy Systems, Vol. 2, No. 1, Februry 1994.
44.Fuzzy Logic Control System,” IEEE Trans. on Fuzzy Systems, Vol. 2, No. 1, Februry 1994

Depth Control of Desflurane Anesthesia with an Adaptive Neuro-Fuzzy System

A. Yardimci[1], N. Hadimioglu[2], Z. Bigat[2], S. Ozen[1]

[1] Department of Instrumentation & Process Control Technology, TBMYO, Akdeniz University, Antalya, Turkey
yardimci@akdeniz.edu.tr
sukruozen@akdeniz.edu.tr

[2] Departments of Anesthesia and Reanimation, Faculty of Medicine, Akdeniz University, Antalya, Turkey
hadimioglu@hotmail.com
zbigat@akdeniz.edu.tr

Abstract. This paper is the first step of a multi-sensor fusion system for control of dept of desflurane anesthesia. In this study, depth of desflurane anesthesia was examined through cardiovascular-based an adaptive neuro-fuzzy system according to changing in the blood pressure and heart rate taken from the patient. The second step, in the next paper will be based on auditory evoked responses. The system designed for anesthetic agent, desflurane, because it is very popular and among the first choices of anesthesiologist for inhalation anesthesia. Intraoperative awareness resulting from inadequate anesthetic is a rare but serious complication during general anesthesia. In order to prevent possible intraoperative awareness, anesthesiologists usually apply anesthetics at level much above the minimal necessary. Anesthetic overdosing prolongs the recovery period, which may cause severe hemodynamic depression and a life-threatening scenario in critically ill patients. To increase patient safety and comfort is one of the most important potential benefits of the system. The second important aim of the study is to relase the anesthesiologist so that he or she can devote attention to other tasks that can't yet be adequately automated. Also, to make the optimum in the area of anesthetic agent and to economize by lessening the costs of an operation are included the benefits which are coming with this system.

1 Introduction

Fuzzy-based control systems have recently found various applications in different fields ranging from consumer electronics and automotive systems to process engineering. Since the 1980s new techniques have appeared from which fuzzy-based has been applied extensively in medical systems. Although medicine is a science which isn't related to control engineering, it is being affected

A. Yardimci et al.: *Depth Control of Desflurane Anesthesia with Cardiovascular-based an Adaptive Neuro-Fuzzy System*, Advances in Soft Computing **2**, 787–796 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

to such an extent that it is now possible to use available control techniques for on-line devices, especially during surgical operations and in intensive care units [1]. Nevertheless, no standard methods exist for transforming human experience into the rule base of the fuzzy inference control system. Beside this the shape of the membership functions associated with the linguistic expressions are determined by trial and error methods so that the performance index is optimised.

The depth control of anesthesia is very important problem in anesthesia. Since depth of anesthesia (DoA) is a very challenging area with no direct measurements being available. Many models have been developed to formulate control strategies for depth control of anesthesia under general anesthesia. Some of these models are used to describe the relationship between depth of anesthesia and clinical signs such as blood pressure and heart rate. Some are used for identifying the EEG process in obtaining derived parameters and evaluating change with anesthetic administration. Howewer, despite these advancements, anesthesiologists still rely heavily on training and experience when determining anetshetic dosage during surgical operations. Only a robust control system built on a more accurate model of consciousness under anesthesia may offer an alternative to the existing paradigm of anaesthesiology practices.

2 Adaptive Neuro-Fuzzy System for Cardiovascular Based DoA Control

ANFIS, as a neuro-fuzzy method, combines fuzzy logic and neural nets into five-layer adaptive network architecture. Using a hybrid learning procedure, ANFIS can learn an input-output mapping based on human knowledge in the form of if-then rules. ANFIS performs the identification of an input-output mapping, available in the form of a set of N input-output examples, using a fuzzy architecture, inspired by the Takagi-Sugeno approach.

To build a derived fuzzy-knowledge model based on ANFIS for estimating DoA, model structure and parameter are required. Structure tuning concerns the structure of the rules: input and output variables selection variable universe of discourse partition, linguistic labels determination and type of logical operation to compose each rule. Parameter tuning mainly concerns the fine adjustment of the position of all membership functions, together with their shape, controlled by premise parameters and the Takagi-Sugeno type.

3 Pharmacological Profile of Desflurane Anesthetic

This volatile anesthetic is a nonflammable fluorinated methyl ethyl ether. It has a vapor pressure of 673 mm Mercury at 20 degree Celsius and boils at

23.5 degree Celsius. The blood/gas coefficient is 0.42 and the MAC in 100 percent oxygen is 6.0 and in 60 percent nitrous oxide 2.8.

Unlike other inhalation anesthetics, desflurane cannot be delivered by standard vaporizers. It requires the use of electrically heated vaporizers. Desflurane is very resistant to degradation by soda lime and can therefore be used during low flow or closed system anesthesia. Desflurane produces a dose-dependent reduction in arterial blood pressure due to peripheral vasodilatation. It might as well cause an increase in heart rate. It should therefore not be used in patients with aortic valve stenosis. It does not sensitize the heart to arrhythmias or cause coronary artery steal syndrome. Like other inhalation anesthetics, it can trigger malignant hyperthermia.

Induction of anesthesia can be achieved by using 6 to 10 percent desflurane in air or in oxygen, or by using 5 to 8 percent desflurane in 65 percent nitrous oxide. Desflurane may cause coughing and excitation during induction and should therefore rather not be used without intravenous anesthetics. Maintenance of anesthesia can be achieved with 5 to 7 percent desflurane. The low tissue solubility of desflurane results in rapid elimination and awakening.

4 Patients and Application

The study was approved by the Hospital Ethics Committee. In Akdeniz University, in the operating room of the Emergency and Trauma Hospital, for eight months, data were collected from the ASA I-II patients, who underwent an arthroscopy operation, and who were administered desflurane. Data gathered from 25 patients. 25 operations were recorded and 20 of them (operation time under 2 hours) were evaluated for the system. The data base was constructed of the records from a total of 20 patients who were asked permission in advance of the operation and were informed about the study. During the operation every five minutes the blood pressure, the heart rate, and the rate of anesthetic agent were recorded. 20 patients were studied, 8 females and 12 males, mean age 48.1 (range 17–69) yr, mean weight 80.7 (54–110) kg, mean height 161.3 (154–194) cm.

A system based on fuzzy logic provides good control algorithms with self-learning capabilities, with measurement inference based on cardiovascular indicators, changes in blood pressure (systolic arterial pressure-BP) and heart rate (HR). Neuro-fuzzy methods facilitate knowledge acquisition, refinement, and interpretation for building the fuzzy inference system, such as the adaptive-network-based fuzzy inference system (ANFIS). An input-output map, which consists 350 lines, is produced from this gathered data. Using a hybrid learning procedure, ANFIS performed an input-output mapping based on human knowledge in the form of if-then rules.

The neuro-fuzzy training process starts with an initial fuzzy logic system. The neuro-fuzzy learning can either start from the default rules. The advantage of coming up with such initial rules is that it enhances the performance

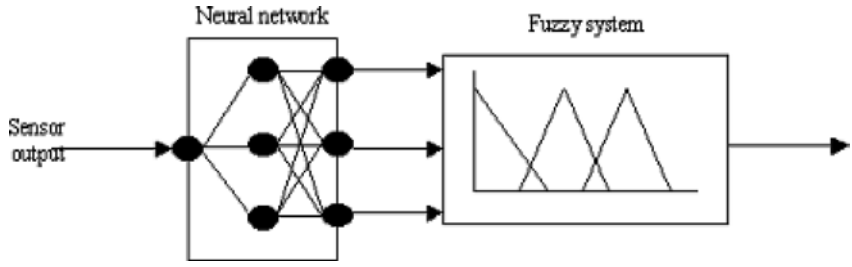


Fig. 1. Neuro-Fuzzy System Training for Membership Functions

of the learning process. Every piece of information that you can put in the initial system out of experience the neuro-fuzzy algorithm cannot extract from training data. This is a very important fact since poor data quality is the reason why neural nets most often fail to deliver good solutions. Figure 1 and 2 shows structure of neuro-fuzzy systems.

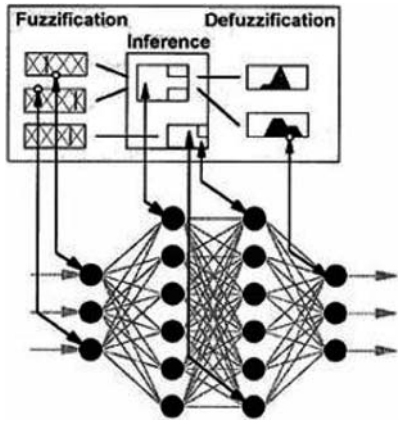


Fig. 2. Neuro-Fuzzy System Structure

In the next step, we selected the parts of the system that the system may modify. This is a major advantage over learning in a neural net, since it lets us control the learning process. Especially when we use neuro-fuzzy to optimize an existing fuzzy logic system, we can exclude parts of the system from learning. For example, if we are confident that parts of the system already work fine with the initial fuzzy logic system, exclude them from learning and training becomes much quicker. Also, if we have data sets representing different aspects of the system performance, we may direct the system to learn different parts of the system from different data sets. If we have defined rules from experience as part of your initial fuzzy logic system, we can also open them for learning. This is useful if we are not completely confident that the

rules or membership functions we defined are correct. The learning method defines how the errors between the results computed by the current fuzzy logic systems and the desired output values shall modify the rules and the membership functions of the fuzzy logic system. To optimize performance, we can parameterize the learning method. Two parameters are important. One is the learn rate for rules, the other one the learn rate for membership functions. These parameters define how much the neuro-fuzzy algorithm modifies a rule or a membership function in each learning step.

In addition to automatic generation of fuzzy logic systems (membership functions and rule base using sample data, neuro-fuzzy can do more. To tune systems while they are running, either to increase their performance or to cope with changing parameters in the environment, we may design adaptive solutions with Neuro-fuzzy. If we use fuzzy logic, we define the system behavior explicitly. With neuro-fuzzy, it is possible to define it implicitly through examples, but with neuro-fuzzy adaptive systems, we only tell it what the goal of its behaviour is. The neuro-fuzzy adaptive strategy will now optimize the running controller to achieve the goals set. The membership functions and the rule base of the neuro-fuzzy system were checked under the inspection of specialists by abiding by the data base information. The system was designed for anesthetic agent, desflurane, which is among the first choices of anesthesiologist for inhalation anesthesia. Because a biological process like anesthesia has a nonlinear, time-varying structure and time-varying parameters, modeling it suggests the use of rule-based controllers like neuro-fuzzy controllers.

Fuzzy rule-based systems include many aspects of fuzzified values, such as the rules antecedents and consequence. The rules structure are usually of the form of *if.. then*. In its basic form this type of the control is equivalent linguistically to a PI controller, and depending on the output, whether it is incremental or absolute, the controller is known as PI or PD respectively. An example of such a rule is *if the blood pressure is above the target and decreasing slowly, then reduce the drug infusion*. A more sophisticated structure is a PID, where the input, its derivative, and integral are considered as three inputs. The rules are checked and composed either from the expert (anesthesiologist) or crafted by hand depending on the experience of the researcher. This includes tuning the membership functions in terms of the shape, width and position. This type of controller is widely used and is the most applicable control type in anesthesia [13, 14, 15, 16, 17].

Blood pressure values always contrast to depth of anesthesia. If the depth of anesthesia in deep blood pressure goes down. In other case, it goes up[10, 11]. Depth of anaestesia is controlled by using a mixture of drugs that are injected intravenously and inhaled gases. Desfluorane is widely used, most often in a mixture of 0 to 10 percent by volume of desflurane in oxygen and/or nitrous oxide. Figure 3 shows the system block diagram. System has two inputs and one output.

We used fuzzy TECH 5.52 MP Explorer and MPLAB 6.01 programs for this application. MPLAB is a windows based application which contains a

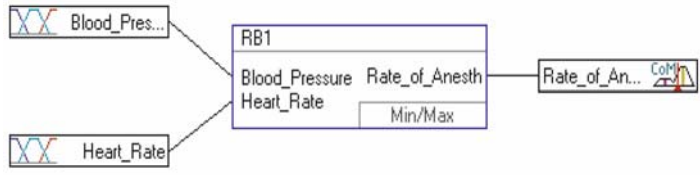


Fig. 3. System block diagram

full featured editor, editor, emulator and simulator operating modes and a project manager. MPLAB allows us to edit our source files and one touch assemble and download to PIC16/17. But designer have to write main program byself. Collected data map firstly were used to training process to produce membership functions for blood pressure, hearth rate and anesthetics gas out. Figure 4, 5, and 6 illustrates these trained membership functions. Linguistic variable ranges on the functions were determined by systems neuro-learning part, from the operations data which were collected by anesthesiologist.

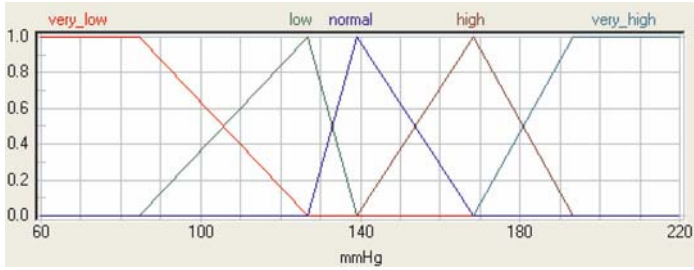


Fig. 4. The MBF of "Blood_Pressure"

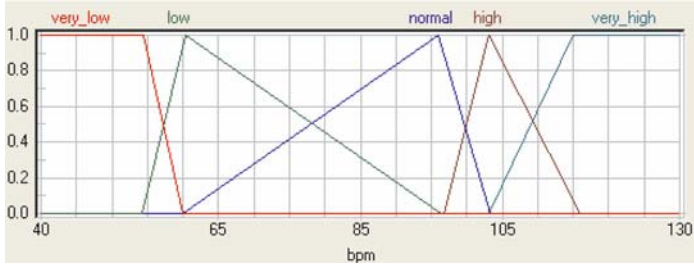


Fig. 5. The MBF of "Heart_Rate"

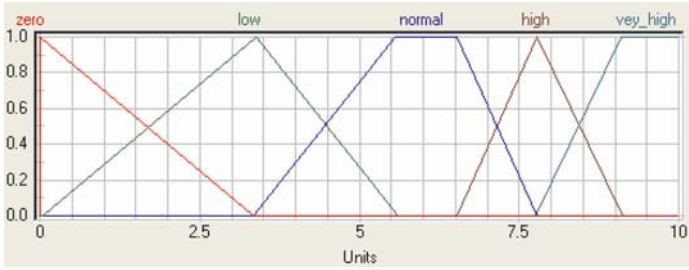


Fig. 6. The MBF of "Rate_of_Anesth"

Rule base includes 17 rules. Figure 7 shows the system rule base. Some of the rules are eliminated by neuro- fuzzy system. Eliminated rules are related to conditions which never happened. Input variable ranges imply 16 bits for 65536 steps.

#	IF		THEN	
	Blood_Pressure	Heart_Rate	DoS	Rate_of_Anesth
1	very_low	very_low	1.00	zero
2	very_low	low	1.00	zero
3	very_low	normal	1.00	low
4	low	very_low	1.00	low
5	low	low	1.00	low
6	low	normal	1.00	normal
7	normal	very_low	1.00	low
8	normal	low	1.00	normal
9	normal	normal	1.00	normal
10	normal	high	1.00	high
11	normal	very_high	1.00	high
12	high	normal	1.00	high
13	high	high	1.00	vey_high
14	high	very_high	1.00	vey_high
15	very_high	normal	1.00	vey_high
16	very_high	high	1.00	vey_high
17	very_high	very_high	1.00	vey_high

Fig. 7. The system rule base

5 Simulation and Simulation Results

In order to test the system, simulation studies have to be carried out to validate the system and also to test its reliability. Therefore a simulator environment was developed which consists one laptop computer and one patient emulator device. In order to provide a mobile simulation system the complete system has also been implemented on a laptop computer which facilitates a multi-window environment providing graphical presentations of the cardiovascular system, the depth of anesthesia and for the patient data.

Results are reported for different types of simulations runs. These simulation results were produced by real recorded operations data. Figure 8 iIllustrates BP, HR DoA values variation that were occurred in every five minutes during operation and Figure 9, shows a sensitive the oldest patient (66 yrs) operations results. Figure 10 shows nominal patient who was in middle age (33 yrs), operations results. Figure 11 shows a resistive the youngest patient operation results. All data which were used during simulation process specially, were not included the neuro-fuzzy system training map.

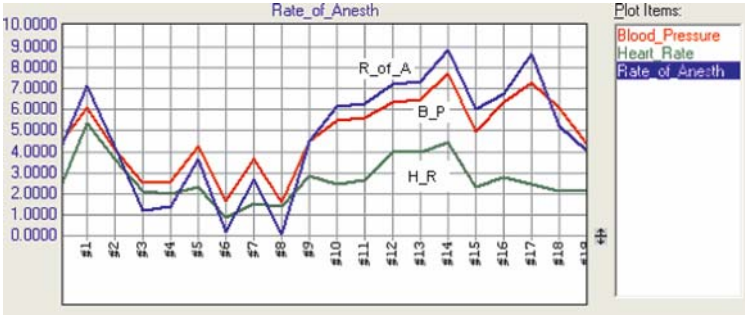


Fig. 8. Real records during the oldest patient operation

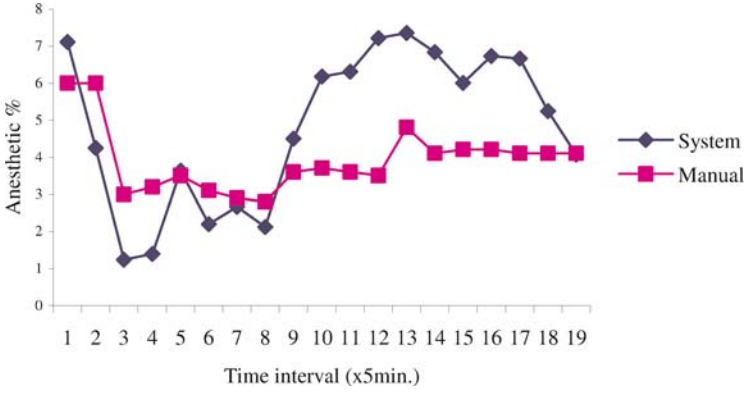


Fig. 9. Anesthetic out for the oldest patient

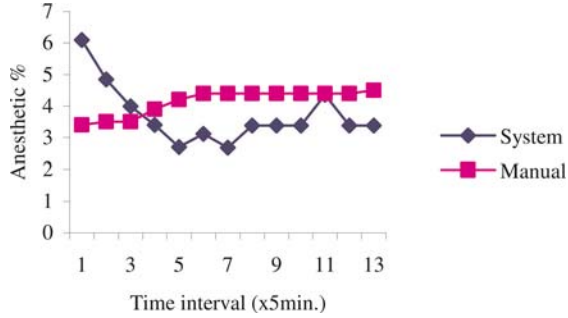


Fig. 10. Anesthetic out for the nominal patient

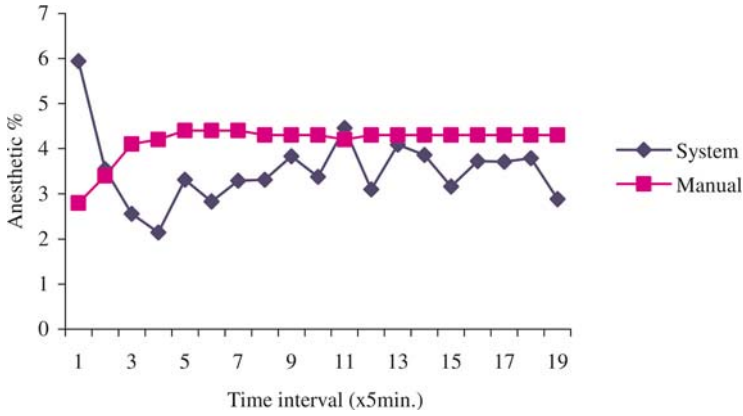


Fig. 11. Anesthetic out for the youngest patient

6 Conclusion

A closed-loop adaptive neuro-fuzzy control system for control of depth of desflurane anesthesia has been developed based on the cardiovascular system. The system has been validated different patients sensitivities. The simulation results show that the system effectively maintains the patient at the clinically acceptable anesthetic depth.. Also, the results were found quite good and acceptable by specialist from the Akdeniz University Emergency and Trauma Hospital Department of Anesthesia and Reanimation. If the desflurane anesthesia adaptive neuro-fuzzy control system does not seem to ensure the patient's safety as an equipment which works independently without the anesthesiologist, it can easily be used as a monitor which helps keep track of depth of anesthesia. The system is to relase the anesthesiologist so that he or she can devote attention to other tasks that can't yet be adequately automated. Our future work is in DoA control will involve more advanced signal processing and improved mathematical modeling methods. We will try to estimate

DOA from electroencephalogram, midlatency auditory evoked potentials and electrocardiogram.

Acknowledge

The authors acknowledge financial support from the Akdeniz University Unit of Scientific Research Projects Managements.

References

1. Bengston JP, Sonader H, Stenqvist O (1998) Comparision of costs of different anaesthetic techniques" Acta Anaesthesiologica Scandinavica, 32: 33–35
2. Bickford RG (1950) Automatic control of general anesthesia, Electroencephalog. Clin. Neurophysiol., Vol.2, pp.93–96
3. Bickford RG (1951) The use of feedback systems for the control of anesthesia, Elect. Eng., vol.70, pp. 852–855
4. Collins VJ (1993) General Anesthesia Fundamental Considerations, 3th Edition, Philadelphia, Lea&Febiger, pp:314–359
5. Greenhow SG, Linkens DA, Asbury A (1993) Pilot Study of an Expert System Adviser for Controlling General Anaesthesia, British Journal of Anaesthesia, 71: 359–365, 1993
6. Gilbert HC, Vender JS (1992) Monitoring the Anesthetized Patient, Clinical Anesthesia, pp: 742–743, 1992
7. Isaka S (1993) Control Strategies for Arterial Blood Pressure Regulation, IEEE Trans. Biomed. Eng., vol. 40, pp. 353–363
8. Kraft HH, Lees DE (1984) Closing the loop: How near is automated anesthesia?, Southern Med. J., vol. 77, pp. 7–12
9. Linkens DA, Abbod MF, Mahfouf M (1999) An initial survey of fuzzy logic monitoring and control utilisation in medicine, Journal of Biomedical Engineering , Applications, Basis Communications, Special Issues on Control methods in medicine, Taiwan
10. Bickford RG (1950) VICKERS, M.D., MORGAN, M., SPENCER, P.S.S., "General Anaesthetics", 7th edition, Butterwurth-Heineman Ltd., Oxford, pp:118–159, 1991
11. Virant-Klun I, Virant J (1999) Fuzzy Logic Alternative for Analysis in the Biomedical Sciences, Computers and Biomedical Research 32, 305–321
12. Vishnoi R, Roy RJ (1991) Adaptive control of closed-circuit anesthesia, IEEE Trans. Biomed. Eng., vol. 38, pp. 39–47
13. Yardimci A, Onural A (1999) Fuzzy Logic Control of Child Blood Pressure During Anaesthesia, Computational Intelligence Theory and Applications, Lecture Notes in Computer Science Vol:1625 p:701–704, Springer, Berlin Germany
14. Yardimci A, Ferikoglu A (2000) Analysis of Biological Process with Microcontroller based Fuzzy Logic Systems, Workshop on Biomedical Information Engineering Proceedings, 25–27 June, p:261–269 Isik University Istanbul, Turkey

Ultrasound Intensity and Treatment Time Fuzzy Logic Control System for Low Cost Effective Ultrasound Therapy Devices

A.Yardimci[1], O.Celik[2]

[1] Department of Instrumentation&Process Control Technology, TBMYO, Akdeniz University, Antalya, Turkey
yardimci@akdeniz.edu.tr

[2] Departments of Communication, TBMYO, Akdeniz University, Antalya, Turkey
ocelik@akdeniz.edu.tr

Abstract. Therapeutic ultrasound is an emerging field with many medical applications. High intensity focused ultrasound provides the ability to localize the deposition of acoustic energy within the body, which can cause tissue necrosis and hemostasis. The ultrasound applied in therapy is usually ranged from 1MHz to 1000MHz. Even the least vibration of 1MHz would be as keen as a sharp knife to cut off steels, if we reinforce its amplitude. However, the output of the ultrasound used in treating people must be decreased substantially. A specific increase in temperature is necessary to achieve a temperature-mediated therapeutic impact by ultrasound in rehabilitation. On a large scale ultrasound intensity determines the temperature level on the tissue. High intensity causes a marked mechanical peak loading of the tissue. This may even lead to tissue damage. The extreme pressure differences developing as a consequence of exposure to ultrasound may cause cavitations in the tissues. Opinions in the literature on the duration of treatment also vary. The duration of treatment depends on the size of the body area to be treated. Lehmann fixes the maximum duration of treatment at 15 minutes. This refers to a treated area of 75-100 cm^2 which he considers the maximum area that can reasonably be treated. New medical applications have required advances in biomedical equipment design and advances in numerical and experimental studies of the interaction of sound with biological tissues and fluids. In this study a fuzzy logic control system will be explained which was developed in order to obtain optimum ultrasound intensity and determine optimum treatment time during ultrasound therapy (UT). This system also increases patient safety and comfort during UT.

1 Introduction

Therapeutic ultrasound is a common mode of treatment in rehabilitation and physiotherapy. Ultrasound refers to mechanical vibrations like sound waves but with a frequency beyond the range of the human hearing. Typical frequencies used in rehabilitation range from 0.8 to 3 Mhz. The absorption of such waves by the human body results in molecular oscillatory movements. This energy transfer is converted into heat proportional to the intensity of the ultrasound. If this heat is not dissipated by physiological means, a localized increase in temperature will occur and thermal therapeutic effects may arise. If the dissipation of heat equals the generation of it, any effects is said to be non-thermal. It is believed that such effect could be achieved by low intensities or a pulsed output of ultrasonic energy[4,7].

A. Yardimci and O. Celik: *Ultrasound Intensity and Treatment Time Fuzzy Logic Control System for Low Cost Effective Ultrasound Therapy Devices*, Advances in Soft Computing **2**, 797–807 (2005)
www.springerlink.com
© Springer-Verlag Berlin Heidelberg 2005

The thermal effects are best known by research and thought to be more controllable than the non-thermal effects, as they can in fact be measured in a rather easy way. Lehmann and colleagues investigated this matter extensively [8]. Although these experiments did not reveal practical guidelines for present rehabilitation protocols and ultrasound equipment, it was not until the early 1990s that new and more appropriate clinical research was carried out.

From former investigations it is known that specific rises in temperature are needed to obtain a beneficial influence on human tissue. According to Castel and Lehmann each increase of tissue temperature by 1 ^{o}C will result in a 13% increase of the metabolic rate. A moderate heating of 2-3 ^{o}C should reduce muscle spasms, pain, chronic inflammation and promote blood flow, although randomized trials do not support the clinical relevance of such estimated heating. A strong heating decreases the viscoelastic properties of callagenous tissue. According to Forrest therapeutic thermal effects of ultrasound can only be expected when tissue temperature exceeds 40^{o}C.

To obtain such temperatures in deeper tissue layers a rehabilitation specialist or physiotherapist has a variety of technical ultrasound parameters at his or her disposal like intensity, ultrasound frequency, mode of energy transfer, a static or dynamic treatment protocol and the option of the treatment duration.

2 Physical Foundation of Ultrasound

The unit of the rate of ultrasound is Hertz and that of the intensity (the density of the output power) of ultrasound is Watt, while the action strength is W/cm2, i.e. one unit of output intensity working on one square centimeter. We have to set for different intensities for different purposes, or else powerful ultrasound could hurt very much like a keen knife. However, the kind of ultrasound is not so easy for common people to touch with.

There are two kinds of waveforms that common ultrasound devices provide. One is continuous waveform and the other, pulses. The difference between the two is that the pulse waveform, unlike the continuous one having heat efficacy, produces more powerful output suddenly in every circle. It is better for us to treat certain abnormalities. Hence, we use it on the field of therapy rather than the others. On the other hand, heat helps beautification so much that we prefer the continuous waveform. Figure 1 shows the nature of ultrasound.

2.1 The rate of ultrasound

The rate of ultrasound means the numbers of vibration occurred in a second. Hertz is the unit of measurement. 1000 wave circles occurring in one second means 1000Hz, 10000 wave circles, 10000Hz. As to ultrasound over 20000Hz, it

is not to be heard by human beings. At present, the ultrasound devices for both medicine and beauty are ranged from 1MHz to 5MHz or so.

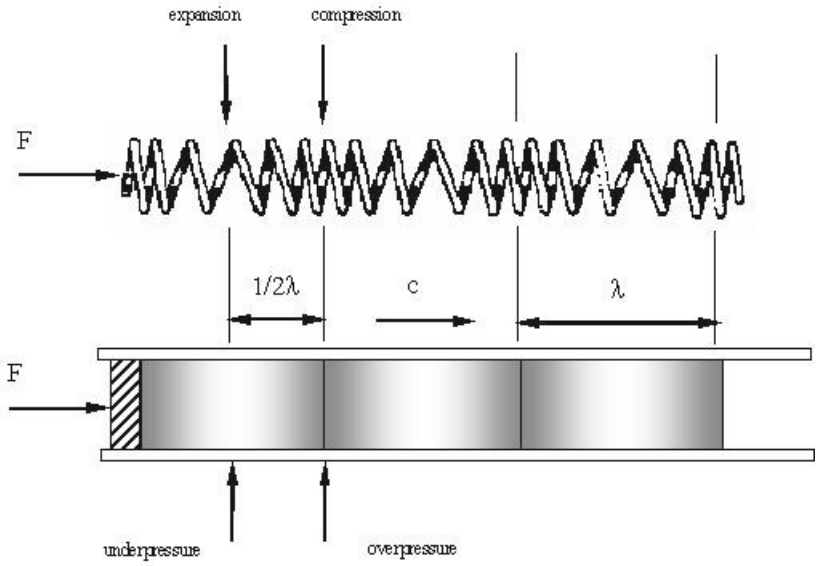


Figure.1. Elastic waves in spring and a liquid (Enraf Nonius-Ultrasound Therapy, 1996)

2.2 The intensity of ultrasound

The intensity of ultrasound is measured as Watt, while the action strength is W/cm^2, i.e. one unit of output intensity working on one square centimeter. Passed through the safe test, the safe standard of continuous waveform is $1W/cm^2$ and pulse waveform, $240mW/cm^2$. Cleaning equipment uses 10 to $100W/cm^2$. Continuous ultrasound of 0.5~3 W/cm^2 affects the transmission of the peripheral nerves obviously. 5~10 minutes 2~3 W/cm^2 causes the axis cylinder of the sciatic nerve to bulge up and then block the nervous pulses. The values mentioned have been found empirically over the years, an experience gained mainly with ultrasound frequencies between 800 kHz and 1 MHz. On theoretic grounds it is assumed that the therapeutic intensity is lower at 3 MHz than at 1 MHz. This conclusion is based mainly on the greater absorption and greater mechanical effect of ultrasound at 3 MHz [6].

In any case during treatment the patient may not feel unpleasant sensations amounting to pain. A mild sensation of excitation is permissible. If as a result of treatment, headache, vertigo, fatigue or other reactions develop, subsequent treatment should be given at a lower intensity. With continuous and pulsed ultrasound at high intensity a sensation of heat may be felt. Only a mild sensation

of warmth is acceptable. As to beautification, the action strength must be much lesser, for not over-stimulating the tissues. On the face, the output is 0.5W/cm^2, to muscles, 1~2W/cm^2. That is why almost all beauty devices limit the highest output. The higher output is used on the body or to relieve the sore and pain of the muscles, the lower output, to the facial beautification. The output level of action strength is displayed as following:

1. Low output <0.3W/cm^2
2. Middle output 0.3~1.2W/cm^2
3. High output 1.2~3W/cm^2

For pulsed ultrasound the mean value must be considered. For instance an intensity of 1 W/cm^2 in position 1:5 pulsed ultrasound is equivalent to 0.2 W/cm^2 continuous ultrasound. However the peak intensity of the pulse must also be considered because of the mechanical effect [3].

2.3 Duration of Treatment

Opinions in the literature on the duration of treatment also vary. The duration of treatment depends on the size of the body area to be treated. One who experienced ultrasound may feel nothing but a little warm. Nevertheless, we can not neglect the long-time stimulation, for it causes the dermis tissue fatigued. Generally speaking, the maximum time is suggested not over 15 minutes. Lehmann fixes the maximum duration of treatment at 15 minutes. This refers to a treated area of 75-100 cm^2, which he considers the maximum area that can reasonably be treated [8]. Naturally the effective radiating area of the treatment head is of importance in this respect. Areas no larger than the treatment head are in general treated for a few minutes. On the body, 10-15 minutes is suitable, on face, about 10. As to the guys over weight, prolonging few minutes would be fine. The time between two treatments is better over than 5 hours [1,2].

2.4 The transmission of ultrasound

Ultrasound can not transmit in the air. If the treatment head does not completely contact with the skin, the result would not meet your expectation. Researches display, the transmission proportion of ultrasound in the air is 0, about 50% in general water, 60% in distilled water, near 70% in glue matter, and above 70% in special gel [10,11]. Besides, another efficacy of ultrasound is to promote the skin absorbing outside matters, no matter nutrition or dirt. Hence, the transmission is worthy to be noticed. Except cleaning your skin before using ultrasound device, we also suggested you use special gels or creams (they must be certified by hygiene organization) in case side effects.

2.5 Absorption and penetration of ultrasound

As the ultrasound energy penetrates into the body tissues, biologic effects can be expected to occur only if the energy is absorbed by the tissues. Due to the absorption the intensity of the sound waves will decrease as they penetrate further in to the tissues. The absorption of ultrasound energy by biologic tissues varies. The absorption coefficient (*a*) is used as a measure of the absorption in various tissues. The absorption is frequency-dependent. For low frequencies absorption in the tissues is lower than for high frequencies[5,9]. This relationship is linear for all tissues except bone between 1 and 10 MHz. Therefore there is always a link between the frequency, absorption and action at depth of ultrasound. In effect the absorption coefficient together with reflection determines the spread of ultrasound in the body. For ultrasound, among other things, the following formula applies. This formula is true for ultrasound consisting of longitudinal waves with perpendicular incidence on homogeneous tissues;

$$I(x)=I_0.e^{-ax} \quad (1)$$

Where;

$I(x)$ = the intensity in W/cm^2 at a depth x in cm

I_0 = the intensity in W/cm^2 at the surface of the body, but in the body tissue

e = 2,7 (base of natural logorithms)

a = absorption coefficient (cm^{-1})

From this formula it emerges that the intensity of ultrasound at a certain depth depends on the absorption coefficient (*a*).

3 Fuzzy Logic Ultrasound Intensity and Treatment Time Control System for UT

This system has been developed to obtain optimum therapy conditions for physiotherapy patients. A fuzzy logic control system determines the ultrasound intensity (UI) and treatment time related with the surface of treatment area, position of the affected tissue, nature and thickness of surrounding tissues and absorption of surrounding tissues. Fig. 1 shows the system structure.

A low-cost effectual ultrasound therapy device (LCUTD) was designed two years ago. This control system has been applied the LCUTD. Test results are evaluated by Akdeniz University Hospital Physiotherapy Department's physiotherapy specialists. It shows that the control unit brings very good solutions under very specific patient data inputs.

Membership functions and rule bases were determined by physiotherapy specialist. They used their experience and knowledge about UT. Fig. 2, 3, 4, 5, shows input membership functions. Fig. 6 is UI output membership function for rule base 1, but also it is input membership function for rule base 2. Finally Fig.7 shows the treatment time membership function.

The rule base contains the control strategy of a fuzzy logic system. The rules' 'if' part describes the situation, for which the rules are designed. The 'then' part describes the response of the fuzzy system in this situation. The degree of support (DoS) is used to weigh each rule according to its importance. There are four inputs to rule base 1 and each has 3 fuzzy zones. Hence the number of rules in the rule base 1 is equal to 81 (3 x 3 x 3 x 3). There are five inputs to rule base 2 and each has 3 fuzzy zones. Hence the number of rules in the rule base 2 equal to 243 (3 x 3 x 3 x 3 x 3). But, rule base 1 includes 54 rules, and rule base 2 includes 216 rules. 27 rules are eliminated from each rule bases by specialist. Eliminated rules are related to conditions which never happened.

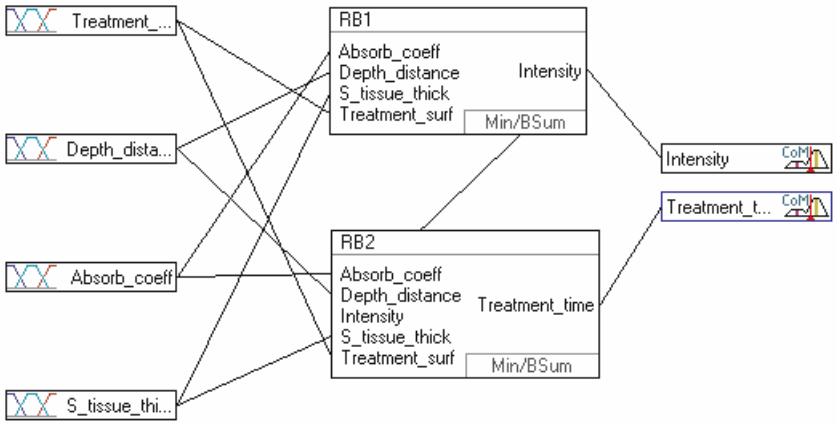


Figure 2. Structure of the Fuzzy Logic UI and TT Control System

Table 1.Input and output variables

#	Variable Name	Type	Unit	Min	Max	Default	Term Names
1	Absorb_coeff		cm^{-1}	0	9	4.5	low,,medium, high
2	Depth_distance		mm	0	120	60	close, medium, far
3	S_tissue_thick		mm	0	120	60	low, medium, high
4	Treatment_surf		cm^2	0	75	37.5	small, medium, large
5	Intensity		W/cm^2	0	3	0	low, medium, high
6	Treatment_time		min	0	15	7.5	very_short, short,normal long, very_long

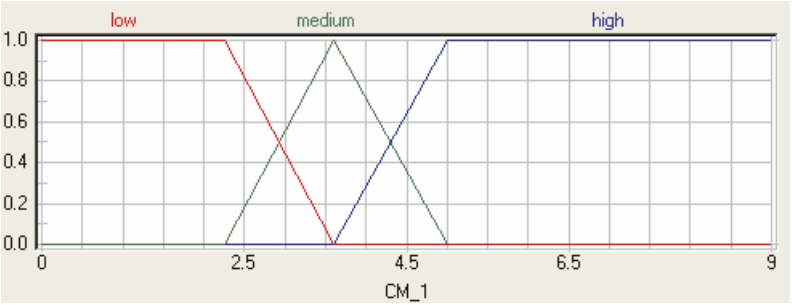


Figure 2. MBF of "Absorb_coeff"

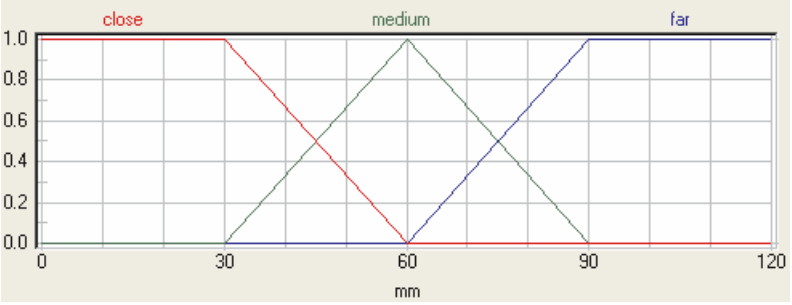


Figure 3. MBF of "Depth_distance"

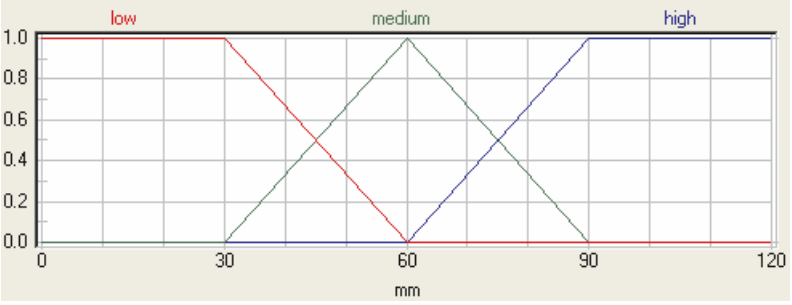


Figure 4. MBF of "S_tissue_thick"

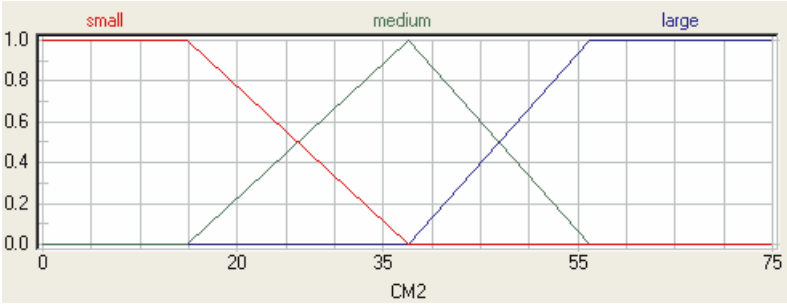


Figure 5. MBF of "Treatment_surf"

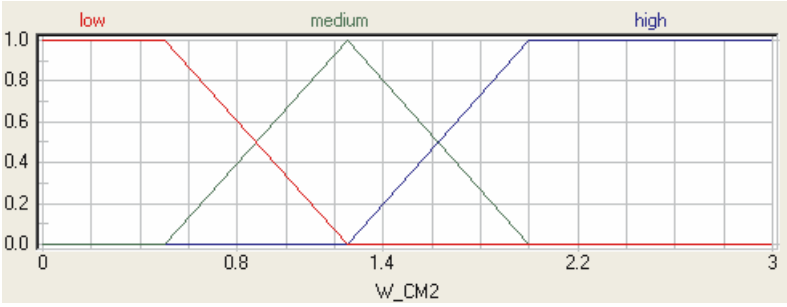


Figure 6. MBF of "Intensity"

Figure 7. MBF of "Treatment_time"

Most fuzzy logic based application solutions use production rules to represent the relationship between the linguistic variables and to derive actions from sensor inputs. Production rules consist of a precondition (IF-part) and a consequence (THEN-part). The IF-part can consist of more than one condition linked by linguistic conjunctions like AND and OR. The computation of fuzzy rules is called fuzzy rule inference. The software which we used for this application FuzzyTECH, calculates the inference in two steps: input aggregation and composition with degree of support (DOS). Aggregation uses fuzzy logic operators to calculate the

result of the IF part of a production rule when the rule consists of more than one input conditions. One of the linguistic conjunctions, AND or OR, links multiple input conditions. Composition uses the fuzzy logic operator, PROD, to link the input condition to the output condition.

Composition links the validity of the entire condition with the DOS. Thus, composition, the second calculation step of each production rule, uses the validity of the condition to determine the validity of the consequence. In standard MAX-MIN or MAX-PROD inference methods, the consequence of a rule is considered equally as true as the condition.

The Center-of-Maximum (CoM) defuzzification method is an approximation of the more computationally-intensive Center-of- Area method. Let R be the linguistic variable to be defuzzified, let μ_{Ri} be the membership functions of all linguistic terms i defined for the base variable internal X ($x \in X$), and let μ_{Ii} be the inference result for every term i. The crisp output value $r \in R$ is computed by the following equation:

$$r = \frac{\sum_i [\mu_{Ii} \bullet \max_x(\mu_{Ri}) \bullet \arg(\max_x(\mu_{Ri}))]}{\sum_i \mu_{Ii}} \tag{2}$$

5 Test Results

In order to test the system, simulation studies have to be carried out to validate the system and also to test its reliability. This simulation results was produced random patients data. Fig 8. illustrates UI results according to depth of distance and treatment surface. The other input variables , position of the affected tissue, nature and thickness of surrounding tissues and absorption of surrounding tissues are chose constant during this test. All data which were used during simulation process were included system response map.

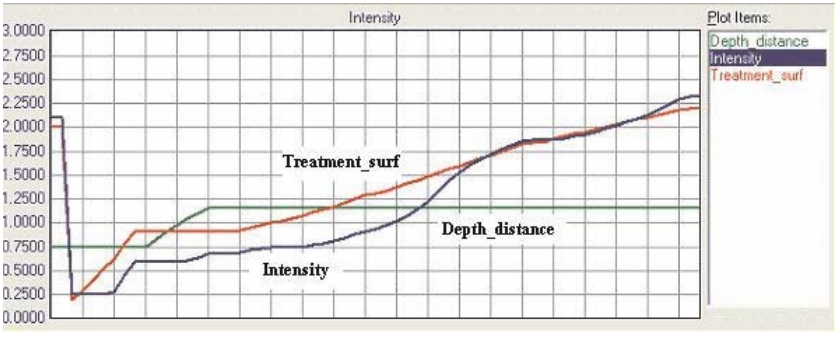


Figure.8. System UI response according to patient conditions

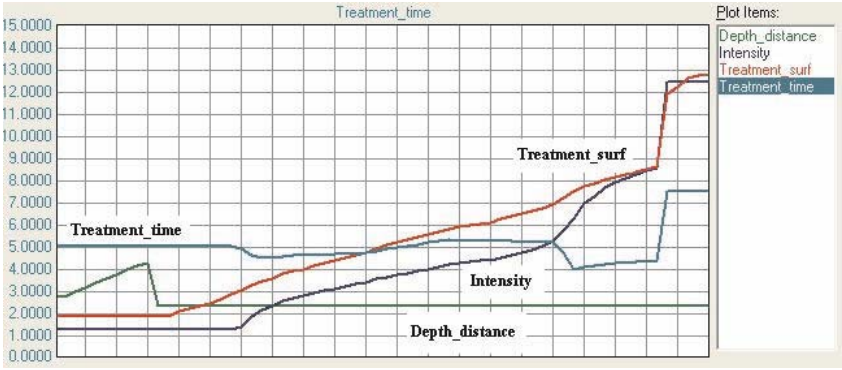


Figure. 9. System TT response according to patient conditions

6 Conclusion

New medical applications have required advances in biomedical equipment design and advances in numerical and experimental studies of the interaction of sound with biological tissues and fluids. The developed system obtains optimum ultrasound intensity and treatment time during UT. Test results show the increasing at patient safety and comfort during UT.

Acknowledgement

The authors acknowledge financial support from the Akdeniz University Unit of Scientific Research Projects Managements.

References

1. Conradi E, (1982) 'Zum gegenwärtigen Stand der Therapie mit Impulsultraschall'. Zeitschrift für Physiotherapie, pp 28 6, 371-376.
2. Conradi E, Fritze U, Hoffmann B, (1983) 'Untersuchungen zur Verteilung der Wärmeenergie in verschiedenen Gewebsschichten beim Schwein nach Ultraschalltherapie im Gleich- und Impulsbetrieb'. Zeitschrift für Physiotherapie, 35, pp 271-280.
3. Daniel S, Prince DJ, (1992) Sonoluminesce in water agar gels during irradiation with 0,75 MHz continuous-wave ultrasound, IEEE International Conference, pp 297-308.
4. Daniel S, Kodoma T, Prince DJ, (1995) Ultrasound in medicine and Biology, IEEE Biomedical Engineering, V.21, pp 105-111.
5. De Lisa JA, Gans M, (1993) Rehabilitation Medicine Principles and Practice, Second Edition, pp 408-410.
6. Edel H en A.Lange (1979) 'Schmerzmodulation durch elektrische Reize und Ultraschall'. Zeitschrift für Physiotherapie, 31, 4.

7. Enraf Nonius, (1996) Ultrasound Therapy, june, The Netherlands.
8. Lehmann JF, (1982) Therapeutic Heat and cold, 3e druk. Williams and Wilkins Baltimore, London
9. Payton OD, Lamp RL, Kasey ME, (1975) 'Effects on therapeutic ultrasound on bone marrow in dogs'. Physical therapy, USA, pp 20-27.
10. Pohlman R, (1951) Die Ultraschalltherapie, Verlag Hans Huber, Bern.
11. Shriber WJ, (1975) A manual of electrotherapy. Philadelphia, pp 339-342.

Author Index