Synergy in Computational Intelligence

Christine L. Mumford

Abstract. This chapter introduces the book. It begins with a historical perspective on Computational Intelligence (CI), and discusses its relationship with the longer established term "Artificial Intelligence" (AI). The chapter then gives a brief overview of the main CI techniques, and concludes with short summaries of all the chapters in the book.

1 Introduction

In the early days of information technology computers were large, expensive and the property of the few government organizations, academic institutions and big businesses who could afford them. Centralized operating systems were developed and two classes of computer systems evolved: one for scientific computing and engineering, specializing in "number crunching" and the other for business computing focussing on data processing activities such as stock control and computerized customer accounts. Today computing devices are small and cheap, and pervade our every day lives. It is therefore not surprising that the style of software required for the twenty-first century is very different from that needed to run operations on the large mainframe computers of the past. It is in this climate that the field of "Artificial Intelligence" (AI) has given way to the newer study of "Computational Intelligence" (CI)[1]. AI grew out of attempts to emulate the human brain on mainframe computers, while CI is more pragmatic and relies on distributed computation, communication and emergence. CI is well suited to today's modern ubiquitous computing devices.

This book is about practical computational intelligence. It covers many techniques and applications, and focuses on novel ways of combining different CI

Christine L. Mumford
Cardiff University, School of Computer Science, 5 The Parade, Cardiff, CF24 3AA, United Kingdom
e-mail: C.L.Mumford@cs.cardiff.ac.uk

[1] Terms with very similar meanings have also emerged in the recent literature, such as "soft computing" and "natural computing".

C.L. Mumford and L.C. Jain (Eds.): Computational Intelligence, ISRL 1, pp. 3–21.
springerlink.com

techniques together, or hybridizing CI techniques with traditional computational techniques. Recognizing the need for pragmatism, authors in this collection propose some new and exciting problem-solving frameworks. The key themes emphasized in the book title are *collaboration*, *fusion* and *emergence*. *Fusion* refers to the amalgamation of CI techniques with each other or with more traditional computational methods. *Collaboration* involves effective communication and is essential, if the above mentioned "fused" techniques are to work harmoniously together. Finally, *emergence* can be viewed as a central goal of CI, asserting that complex behaviour can emerge from collaboration between simple processing elements. An essential ingredient of a CI system exhibiting emergent behaviour is *synergy* in which *the whole is greater than the sum of the parts*.

The remainder of this chapter is structured as follows. It will begin with some discussion on the origins of Computational Intelligence, and examine its relationships with Artificial Intelligence. This will be followed by a brief survey of some of the key CI paradigms. The chapter will conclude with a brief overview of the rest of the book.

2 The Birth of Computational Intelligence

The origin of the term "Computational Intelligence" (CI) has been widely attributed to Bezdek [1, 2]. Defining a new field devoted to computer-based intelligence can be viewed as a timely attempt to escape from some of the difficult issues and bad publicity associated with the longer established field of Artificial Intelligence (AI). Although AI and CI have much in common, the emphasis is subtly different. CI concentrates on practical application, self organization and the emergence of complex behaviour from simple components, while AI aims to build intelligent systems based on ideas of how the human brain works. John McCarthy originally coined the term "Artificial Intelligence" in 1955, in advance of a month long brainstorming conference held in Dartmouth College in the following year. The proposal for the Dartmouth conference [15] makes interesting reading. The introduction is reproduced below.

> We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.

The document goes on to discuss the "various aspects of the artificial intelligence problem" in more detail, including computers and computer programming, natural language processing, neural networks, the theory of computation, the need for automatic self-improvement, and aspects of abstraction and creativity. Most of these topics remain active research issues to this day. However, the assumption that human intelligence can be simulated by machine was perhaps a little overoptimistic. Indeed, it is one of the "big questions" remaining in computer science.

The two decades following the 1956 conference saw many high profile AI research projects, for example, the development of the LISP and PROLOG programming languages, the SHRDLU "microworlds" project, and the first expert systems (see standard texts on AI, such as [20, 21], for more information). Although few could argue that these projects had produced some highly successful results, and useful applications, there was, nevertheless, a general feeling of disappointment at the time, that the AI community had in some sense "failed to deliver". This perception was effectively articulated in a report to the British Science Research Council by the British academic James Lighthill in 1973 [14]:

> In no part of the field have discoveries made so far produced the major impact that was then promised.

In essence, the so-called "Lighthill Report" stated that AI researchers had failed to address the issue of the combinatorial explosion, i.e., AI techniques may work on small problem domains, but the techniques do not scale up well to solve more realistic problems. Following this very pessimistic view, the Science Research Council slashed funding for AI projects in the UK. Although a rather more optimistic view prevailed in much of the rest of the world, and major new investments continued throughout the 1980s (e.g., CYC in the USA [13], and the Fifth Generation Computer Systems project in Japan [6]). AI was becoming an increasingly fragmented study, consisting of many disciplines, such as reasoning, knowledge engineering, planning, learning, communication, perception, and so on. Despite the many successes that had been achieved using expert systems, logic programming, neural networks etc., it was blatantly obvious that the dream of properly emulating human intelligence had never come close to being realized. It was time to perhaps "move on" and capitalize on the substantial achievements provided by some of the "offshoots" of AI, and leave behind the very negative image that had become so closely associated with the term "AI" itself, not so much because AI had failed per se, but rather because of the over-inflated expectations that had become intrinsically tied up with the notion of it.

Bezdek's view of CI was as a system that exhibited some form of "intelligence", yet dealt with numerical (low level) data, as opposed to "knowledge", and in this sense differed from traditional Artificial Intelligence. Nevertheless, the view of Bezdek was very much focussed towards his personal research interests of pattern recognition and neural networks. In the following years the term "CI" became firmly established when it was adopted by the IEEE (the Institute of Electrical and Electronic Engineers), and in 2004 the Computational Intelligence Society (CIS) was established (as a name change from the Neural Network Society). The slogan of the IEEE CIS is "mimicking nature for problem solving", and its scope is stated as:

> The Field of Interest of the Society shall be the theory, design, application, and development of biologically and linguistically motivated computational paradigms emphasizing neural networks, connectionist systems, genetic algorithms, evolutionary

programming, fuzzy systems, and hybrid intelligent systems in which these paradigms are contained.

Artificial intelligence brings its connotations of "intelligence", which can be distracting. One can get sidetracked into pondering the meaning of intelligence, rather than asking more useful questions, about self-organization, and emergence of complex systems from simple components, for example. A useful definition taken from the Computer Science web site of Amsterdam University (http://www.cs.vu.nl/ci/) emphasizes the "bottom up" nature of CI:

> Enclosed in the name computational intelligence is a 'message', according to scientific folklore it is chosen to indicate the link to and the difference with artificial intelligence. While some techniques within computational intelligence are often counted as artificial intelligence techniques (e.g., genetic algorithms, or neural networks) there is a clear difference between these techniques and traditional, logic based artificial intelligence techniques. In general, typical artificial intelligence techniques are top-to-bottom, where the structure of models, solutions, etc. is imposed from above. Computational intelligence techniques are generally bottom-up, where order and structure emerges from an unstructured beginning.

Some interesting further discussions on the birth of AI and CI, and on some of the important philosophical issues on the essence of intelligence can be found in Chapter 2 of this book.

3 The Main CI Techniques

In this section we will look briefly at the following key CI paradigms: Evolutionary Algorithms, Neural Networks, Fuzzy Systems and Multi-Agent Systems. This will be followed by a short summary covering some other important techniques included by various authors in this collection.

3.1 Evolutionary Algorithms

Evolutionary algorithms (EAs) comprise a class of techniques inspired by evolution and natural selection. The best known EAs are undoubtedly the *genetic algorithms* (GAs) developed by John Holland [9] in the 1960's and 70's. Contemporaries of Holland independently developed some similar techniques however, for example of Rechenberg [19] introduced *evolution strategies* (ES) and Fogel, Owen and Walsh [7] developed *evolutionary programming* (EP). Since these early days, interest in evolutionary-inspired algorithms has grown extensively, and many new variations have appeared, often very different from the original models conceived by Holland, Rechenberg or Fogel. For example, in the early 1990s, John Koza proposed *genetic programming* [11]: an evolutionary style technique for evolving effective computer programs, mostly using the LISP programming language (see also Chapter 6). Other popular paradigms to have been derived from the more generic approach include artificial life [12], evolvable hardware [8], ant systems [4] and particle swarms [10] (Chapter 20), to name but a few. Artificial Immune Systems (Chapter 16) have also become a popular topic for research in recent years, drawing analogies with some of the ingenious problem-solving mechanisms observed in natural immune systems and applying them to a broad range of real-world problems. In addition, there are many examples of hybrid (or memetic) approaches where problem specific heuristics, or other techniques such as neural networks, fuzzy systems, or simulated annealing, have been incorporated into a GA framework. Thus, due to the growth in popularity of search and optimization techniques inspired by natural evolution during the last few decades, it is now common practice to refer to the field as *evolutionary computing* and to the various techniques as *evolutionary algorithms*. In addition, evolutionary techniques for simultaneously optimizing several objectives have recently become popular. These approaches, collectively known as multi-objective evolutionary algorithms [3] are very effective at balancing the frequently conflicting objectives to produce excellent trade-off solutions, from which a human decision maker can make an informed choice. Chapters 3 and 5 deal with multi-objective optimization problems.

Parallel evolutionary algorithms are discussed in Chapter 17. The analogy with natural population structures and their geographical distributions make parallel implementations highly desirable, to speed up processing and to facilitate complex emergent behaviour from simple components within the distributed populations.

Given the range of EAs mentioned above, it is not perhaps surprising that there is no rigorous definition of the term "evolutionary algorithm" that everyone working in the field would agree on. There are, however, certain elements that the more generic types of EA tend to have in common:

1. a population of chromosomes encoding candidate solutions to the problem in hand,
2. a mechanism for reproduction,
3. selection according to a fitness, and
4. genetic operators.

Figure 1 gives an outline of a generic EA. The process is initialized with a starting population of candidate solutions. The initial population is frequently generated by some random process, but may be produced by constructive heuristic algorithms, or by other methods. Once generated, the candidate solutions are evaluated to establish the quality of each solution, and based on this quantity, a fitness value will be computed, in such a way that better quality solutions will be assigned higher values for their fitness. Individuals will next be selected from the population to form the parents of the next generation, and these will be duplicated and placed in a mating pool. The selection process is frequently biased, so that fitter individuals are more likely to be chosen than their less fit counterparts. Genetic operators are then applied to the individuals in the mating pool. The idea is to introduce new variation, without which no improvement is possible. Recombination (also known as crossover) is achieved by combining elements of two parents to form new offspring. Mutation, on the other hand, involves very small random changes made to solutions. The final stage in the

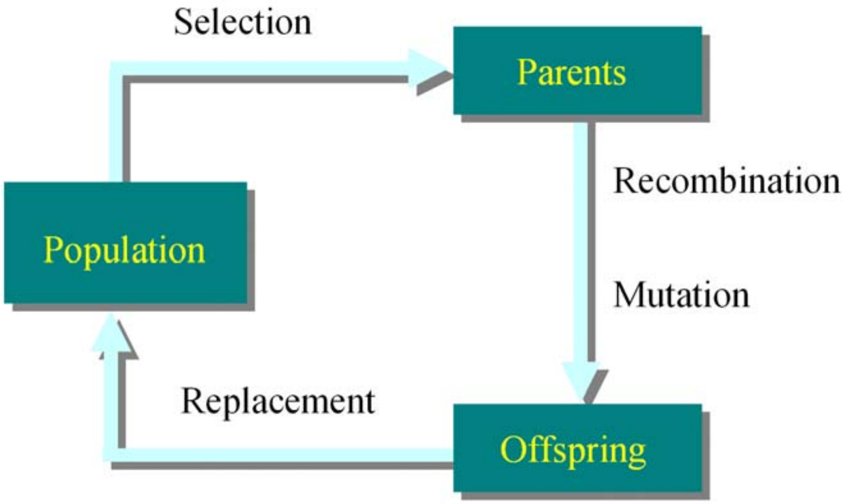

Fig. 1 The Evolutionary Cycle

cycle requires the population is updated with new individuals. Depending on the style of the EA, this may involve replacing the parent population in its entirety, or partial replacement is favoured by some researchers - perhaps replacing the poorest 10 % of the population by the best offspring, for example. A good general text on evolutionary algorithms is Eiben and Smith [5].

3.2 Neural Networks

Artificial Neural Networks (ANNs) are inspired by biological nervous systems, and emulate a simple "brain". They consist of large numbers of highly interconnected processing elements (neurons) working together and learning from experience. ANNs are specially configured for each application, and typical uses include pattern recognition and data classification. In a biological nervous systems, learning involves making adjustments to the synaptic connections between the neurons. In a similar way for ANNs, learning is accomplished through the adjustment of weights by application of some "learning rule" to the connections between the artificial neurons or nodes. Learning rules typically attempt to reinforce connections that contribute to a "correct output", and penalize connections that produce incorrect results. There are three main classes of ANN, distinguished by their different learning processes: 1) supervised learning, 2) unsupervised learning, and 3) reinforcement learning. With supervised learning a training stage uses a set of test data and a teacher to score the performance of the ANN, then adjusts the connection weights in an effort to improve the performance to better match the actual output to the predicted output. The most widely known supervised learning ANNs are the backpropagation nets. ANNs that use unsupervised learning do not have a training

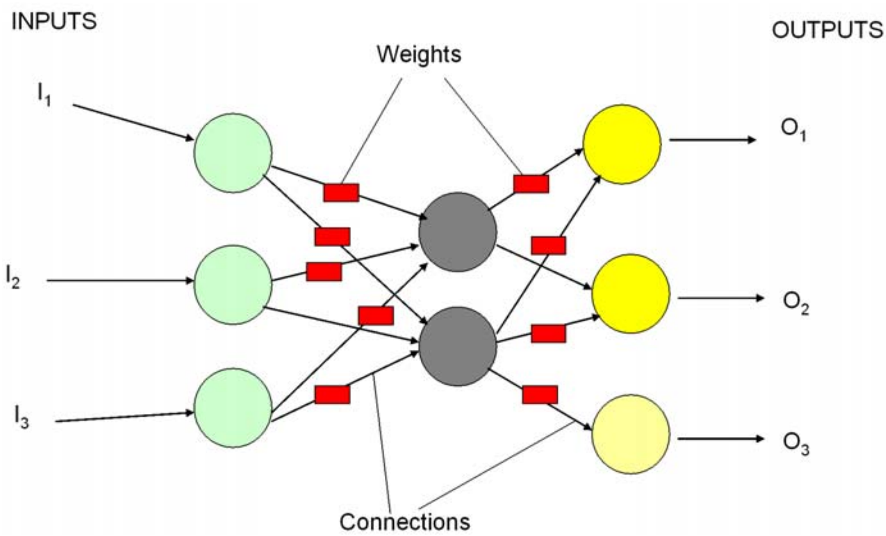

Fig. 2 A Neural Network with One Hidden Layer

stage, and these are frequently referred to as "self organizing networks". Kohonen nets are the best known example of this type. In reinforcement learning data is not usually available. Instead the aim is to discover a policy for selecting actions that minimize some measure of long-term cost. A schematic neural network is illustrated in Figure 2. For more details on ANN see Mehrotra, Mohan, and Ranka [16]. Chapters 12, 13 and 22 all utilize neural networks, in one form or another.

3.3 Fuzzy Systems

Fuzzy logic was first proposed by Lotfi A. Zadeh of the University of California at Berkeley in a 1965 paper [23]. It is a modification of boolean (or crisp) logic which allows approximate and common sense reasoning in the absence of "true" or "false" certainty. In crisp logic, set membership is "all or nothing". In contrast, fuzzy logic allows partial membership of sets, known as *fuzzy sets*, and forms the basis of *fuzzy systems*. Fuzzy Systems can deal with partial truth and incomplete data, and are capable of producing accurate models of how systems will behave in the real world, particulary when appropriate conventional system models are not available. Instead of supplying equations for a mathematical model, for example, a designer will need to produce appropriate fuzzy rules to describe the system he/she wishes to implement. The system operates when inputs are applied to the rules consisting of the current values of appropriate membership functions. Once activated, each rule will fire and produce an output, which will also be a partial truth value. In the final stage, the outputs from all the rules are combined, in some way, and converted into a single crisp output value. In summary, a fuzzy system consists of the following:

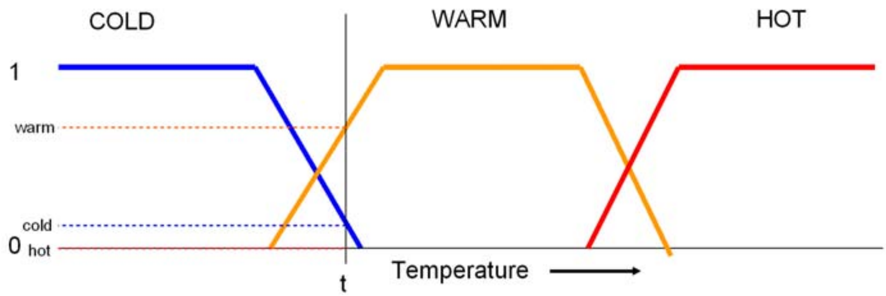

Fig. 3 A Fuzzy Temperature Control System

- a set of inputs
- a fuzzification system, for transforming the raw inputs into grades of memberships of fuzzy sets
- a set of fuzzy rules
- an inference system - to activate the rules and produce their outputs
- a defuzzification system - to produce one or more final crisp outputs

We will now look at a simplistic fuzzy system: a fuzzy controller for room temperature.

The fuzzy set membership diagram in Figure 3 characterizes three functions, identifiable as subranges of temperature: cold, warm and hot. Suppose we wish to keep a room at a comfortable temperature (warm) by building a control system to adjust a room heater. We can see in Figure 3 how each function maps the same temperature value to a truth value in the 0 to 1 range, so that any point on that scale has three "truth values", one for each of the three functions. It is these truth values that are used to determine how the room temperature should be controlled. The vertical line in the diagram represents a particular temperature, t. At this temperature it is easy to observe the degree of membership to "hot" (red) is zero, this temperature may be interpreted as "not hot". Membership of "warm" is about 0.7, and this may be described as "fairly warm". Similarly, examining membership of the "cold" function gives a value of about 0.15, which may describe it as "slightly cold". Adjectives such as "fairly" and "slightly", used to modify functions are referred to as "hedges", and can be a useful way to specify subregions of the functions to which they are applied.

To operate our fuzzy temperature control system, we require a number of fuzzy IF-THEN rules, in the form of "IF variable IS property THEN action". For example, an extremely simple temperature regulator that uses a heater might look like this:

1. IF temperature IS cold THEN turn heater to high
2. IF temperature IS warm THEN do nothing
3. IF temperature IS hot THEN turn off heater

Notice there is no "ELSE". All of the rules are evaluated, because the temperature will belong to all three sets (cold, warm and hot) at the same time, but to different degrees. At temperature t in Figure 3, for example, $M(cold) = 0.15, M(warm) = 0.7$ and $M(hot) = 0$.

Obviously, the greater the truth value of "cold", the higher the truth value of "turn the heater to high", although this does not necessarily mean that the output itself will be set to "high", since this is only one rule among many. In our example, the partial truth inputs for "cold", "warm" and "hot" will in turn produce partial truth values for the outputs "turn the heater to high", "do nothing" and "turn the heater off". The simplest way to produce a single crisp instruction, is to select the output with the maximum value (which will probably map to "do nothing" in the case of our temperature t). A more sophisticated method involves finding the centroid of all the outputs. This methods locates the "centre of mass" of the combined membership function curves.

More complex rules can be built for fuzzy systems, using AND, OR, and NOT operators. These are the counterparts of the familiar crisp logic operators, and they are usually defined (respectively) as the minimum, maximum, and complement. So, for the fuzzy variables x and y:

NOT x = (1 - truth(x))
x AND y = minimum(truth(x), truth(y))
x OR y = maximum(truth(x), truth(y))

Clearly, the simple temperature controller described above is for illustration only, and practical fuzzy systems will typically be made up from many more rules - perhaps hundreds or even thousands. In these more sophisticated systems, it is likely that the fuzzy rule set will be less "flat", and form more of a hierarchy, so that the outputs of some rules provide inputs to others. Systems with large rule sets will probably require more sophisticated inference systems to ensure the efficient processing of the rules, in a reasonable order.

To complete this section, it is worth mentioning a variation of fuzzy sets called *rough sets*. Rough Set Theory was introduced in the early 1980s by Zdzislaw Pawlak [18]. The basic idea is to take concepts and decision values, and create rules for upper and lower boundary approximations of the set. With these rules, a new object can easily be classified into one of the regions. Rough sets are especially helpful in dealing with vagueness and uncertainty in decision situations, and for estimating missing data. Uses include data mining, stock market prediction and financial data analysis, machine learning and pattern recognition.

For further reading on fuzzy systems [17] is a good introductory text. Also Chapter 4 in the present book, provides a good background to many of the important concepts, and chapters 3, 5, 18, and 22 also cover aspects of fuzzy systems.

3.4 Multi-Agent Systems

A multi-agent system (MAS) is a system composed of many interacting intelligent agents; each one is in itself simple and apparently acts only in its own interest, yet by collaborating and/or competing with each other agents, an MAS can be used to solve problems which would entirely defeat an individual agent or a monolithic system. MAS can exhibit self-organization and complex behaviour can emerge. Example applications include financial forecasting and online trading (see Chapter 8) and disaster response (see Chapter 10).

The agents in a multi-agent system have several important characteristics [22]:

- Autonomy: the agents are at least partially autonomous
- Local views: no agent has a full global view of the system
- Decentralization: there is no one controlling agent
- Typically multi-agent systems research refers to software agents. However, the agents in a multi-agent system could equally well involve robots, humans or human teams. A multi-agent system may contain combined human-software agent teams (see Chapter 8).

Generally, multi-agent systems are flexible and they are easily maintained or modified without the need for drastic rewriting or restructuring. MAS also tend to be robust and recover easily from a breakdown, due to built in duplication and redundancy of components. Chapters 8, 9, 10, 11 and 20 all deal explicitly with multi-agent systems.

3.5 Other Techniques Covered in the Book

Besides the main methods outlined above, a number of other CI techniques have been used by various authors in this text, including rule induction (Chapter 19), Bayesian Learning (Chapter 10), Likelihood Ratios (Chapters 18 and 19), Case-Based Reasoning (Chapter 21), Collaborative Clustering (Chapter 22), Blackboard Database Systems (Chapter 9), and Hyper-Heuristics (Chapter 6). Among the "traditional techniques" used in partnership with the CI methods, statistical methods are used in Chapters 13 and 21, and computer vision techniques in Chapters 12 and 13. Effective communications are essential for agent-based systems and all distributed CI techniques. These important issues are addressed in Chapters 14 and 15.

4 Chapters Included in This Book

This book is divided into nine parts:

Introduction
Fusing evolutionary algorithms and fuzzy logic
Adaptive solution schemes
Multi-agent systems
Computer vision
Communication for CI systems
Artificial immune systems
Parallel evolutionary algorithms
CI for clustering and classification

4.1 Part I: Introduction

This Part covers some of the history of computational intelligence, and sets the scene for the rest of the book.

Chapter 1: Synergy in Computational Intelligence

The present chapter, by Christine Mumford, introduces the book and begins Part I. It begins with a brief history of Artificial Intelligence and discusses the origins of the term "Computational Intelligence". Then follows an introduction to the main Computational Intelligence paradigms used by the various authors in the book; and finally, the chapter concludes with short summaries of all the individual chapters.

Chapter 2: Computational Intelligence: The Legacy of Alan Turing and John von Neumann

In this thought-provoking chapter, Heinz Mühlenbein recalls the fundamental research questions of computational intelligence, and explains how many of these issues remain unresolved to this day. In recent years, it has become fashionable to subdivide computational intelligence into many fields e.g. evolutionary computation, neural networks, fuzzy logic. This was not always the case. This chapter recalls the broader issues and reviews the seminal research of Alan Turing and John von Neumann in detail. The author discusses the many areas of computational intelligence that need to come together, if we are to create automata with human-like intelligence.

4.2 Part II: Fusing Evolutionary Algorithms and Fuzzy Logic

These three chapters cover some useful ways to combine evolutionary algorithms with fuzzy systems.

Chapter 3: Multiobjective Evolutionary Algorithms for the Electric Power Dispatch Problem

The main objective of the electric power dispatch problem is to schedule the available generating units to meet the load demand at minimum cost, while satisfying all constraints. However, thermal plants are a major source of atmospheric pollution. Recently the pollution minimization problem has attracted a lot of attention as the public demand clean air. Mohammad Abido explores the use of evolutionary multiobjective optimization to minimize cost and pollution, simultaneously. Furthermore, he uses fuzzy set theory to select the "best" compromise solution from the trade-off solution set.

Chapter 4:Fuzzy Evolutionary Algorithms and Genetic Fuzzy Systems: A Positive Collaboration Between Evolutionary Algorithms and Fuzzy Systems

Two alternative ways of integrating fuzzy logic and evolutionary algorithms are discussed in detail by F. Herrera, M. Lozano in this chapter. The first one, called a *genetic fuzzy system (GFS)* consists of a fuzzy rule based system (FRBS) augmented by a learning process based on evolutionary algorithms. In the second approach, fuzzy tools and fuzzy logic-based techniques are used for modeling different evolutionary algorithm components and also for adapting evolutionary algorithm control parameters, with the goal of improving performance. The evolutionary algorithms resulting from the second type of integration are called *fuzzy evolutionary algorithms*. This chapter includes some excellent introductory material on fuzzy logic, as well as a summary of state-of-the-art with respect to genetic fuzzy systems and fuzzy evolutionary algorithms. The potential benefits derived from the synergy between evolutionary algorithms and fuzzy logic are made clear.

Chapter 5: Multiobjective Genetic Fuzzy Systems

Hisao Ishibuchi and Yusuke Nojima describe the two conflicting goals in the design of fuzzy rule-based systems: one is accuracy maximization, and the other is complexity minimization. Generally, complex rules and large rule sets promote accuracy, and smaller rule sets with simple rules reduce complexity. The authors discuss the trade-off relation between these two goals, i.e., that improving the accuracy of a rule set will simultaneously increase its complexity. This chapter explains how various studies in multiobjective genetic fuzzy systems have experimented with the provision of non-dominated trade-off solutions, each solution being a complete candidate rule set for the decision maker's consideration. These rule sets will range from the simplest and least accurate to the most complex and most accurate.

4.3 Part III: Adaptive Solution Schemes

These two chapters describe two different approaches to adaptive problem solving, involving mechanisms to select from a portfolio of algorithmic alternatives, adapting to the best choices for particular problems and instances.

Chapter 6: Exploring Hyper-Heuristic Methodologies with Genetic Programming

Hyper-heuristics represent a novel search methodology that is motivated by the goal of automating the process of selecting or combining simpler heuristics in order to solve hard computational search problems. This approach operates on a search space of heuristics rather than directly on a search space of solutions to the underlying problem which is the case with most meta-heuristics implementations. In this chapter, Edmund Burke, Mathew Hyde, Graham Kendall, Gabriela Ochoa, Ender Ozcan and John Woodward look at the use of Genetic Programming to automatically generate heuristics for a given problem domain.

Chapter 7: Adaptive Constraint Satisfaction: The Quickest First Principle

James Borrett and Edward Tsang demonstrate the potential of adaptive constraint satisfaction in this chapter, using a technique known as algorithmic chaining. It is recognised that some constraint satisfaction instances are much easier to solve than others, and thus it makes sense to apply a simple and fast algorithm, whenever such an approach is adequate for solving the instance in question. However, when faced with exceptionally hard problem instances, a more complex (and slower) approach may be required. Algorithmic chaining presents a sequence of algorithms, which are applied to a problem instance in turn, if and when required. Thus, if the first algorithm is unsuccessful, the second in the sequence will be tried, and then the third, if required, and so on. The chapter describes the "Reduced Exceptional Behaviour Algorithm" (REBA), which is a technique based on algorithmic chaining. The REBA algorithm makes use of a mechanism for predicting when thrashing type behaviour is likely to occur, and results presented within the chapter clearly demonstrate the effectiveness of the approach in reducing susceptibility to exceptionally hard problem instances.

4.4 Part IV: Multi-Agent Systems

Multi-Agent Systems (MAS) provide increasingly popular paradigms for solving complex problems, using a distributed system of (simple) individual processing elements. These four chapters offer some novel solutions to difficult design and implementation issues associated with practical MAS.

Chapter 8: Collaborative Computational Intelligence in Economics

This chapter provides a general review of collaborative computational intelligence (CCI) in economics. Shu-Heng Chen demonstrates the potential of CCI by focussing on three research paradigms in economics: *heterogeneous agent-based economics*, *experimental economics*, and *financial data mining*. The essence of agent-based economics is a society of heterogeneous agents working together. Experimental economics is explored with respect to laboratories comprising both human agents and software agents. Finally, the chapter concludes with a survey of hybrid CI systems currently used in financial data mining.

Chapter 9: IMMUNE: A Collaborating Environment for Complex System Design

To address the dilemma of distributed versus central control in complex system design, decision support systems that enable robust collaboration amongst many design agents from different disciplines are required. The particular characteristics of such decision support systems must include immunity to catastrophic failures and sudden collapse that are usually observed in complex systems. This chapter, written by Mahmoud Efatmaneshnik and Carl Reidsema, lays the conceptual framework for IMMUNE as a robust collaborating design environment. Agents in IMMUNE are adaptive and can change their negotiation strategy and in this way can contribute to the overall capability of the design system to maintain its problem solving complexity.

Chapter 10: Bayesian learning for cooperation in multi-agent systems

Mair Allen-Williams and Nicholas R Jennings consider the problem of agent coordination in uncertain and partially observable systems. They present an approach to this problem using a Bayesian learning mechanism, and demonstrate its effectiveness on a cooperative scenario from the disaster response domain.

Chapter 11: Collaborative Agents for Complex Problems Solving

In a multi-agent system (MAS), agents that possess different expertise and resources collaborate together to handle problems which are too complex for individual agents. Generally, agent collaborations in a MAS can be classified into two groups, namely agent cooperation and agent competition. In this chapter Minjie Zhang, Quan Bai, Fenghui Ren and John Fulcher introduce two main approaches for complex problem solving via agent cooperation and/or competition, these being (i) a partner selection strategy among competitive agents, and (ii) dynamic team forming strategies among cooperative agents.

4.5 Part V: Computer Vision

Computer vision is a key application area for CI techniques. Chapters 12 and 13 discuss two extremely challenging applications: predicting human character traits from facial appearance and analyzing crowd dynamics, respectively.

Chapter 12: Predicting Trait Impressions of Faces Using Classifier Ensembles

Recent studies in social psychology indicate that people are predisposed to form impressions of a person's social status, abilities, dispositions, and character traits based on nothing more than that person's facial appearance. In this chapter Sheryl Brahnam and Loris Nanni present their work on building machine models of human perception, aimed at recognizing traits (such as dominance, intelligence, maturity, sociality, trustworthiness, and warmth) simply by observing human faces. They demonstrate that ensembles of classifiers work better than single classifiers, and also that ensembles composed of 100 Levenberg-Marquardt neural networks (LMNNs) seem to be as capable as most individual human beings are in their ability to predict the social impressions certain faces make on the average human observer.

Chapter 13: The Analysis of Crowd Dynamics: From Observations to Modelling

B. Zhan, P. Remagnino, D.N. Monekosso and S. Velastin describe how computer vision techniques, combined with statistical methods and a neural network, can be used to automatically observe, measure and learn crowd dynamics. New methods are proposed to measure crowd dynamics, and model the complex movements of a crowd.

4.6 Part VI: Communication for CI Systems

Distributed CI systems of all kinds need reliable, fast and efficient communications. These two chapters describe simple, low cost and effective ways to use the latest technology in a discriminatory way. Chapter 14 covers large scale collaborative sensor networks, and Chapter 15 focusses on opportunist networks.

Chapter 14 :Computational Intelligence for the Collaborative Identification of Distributed Systems

In this chapter Giorgio Biagetti, Paolo Crippa, Francesco Gianfelici and Claudio Turchetti suggest a new algorithm for the identification of distributed systems by large scale collaborative sensor networks. They describe how recent advances in hardware technologies have made it possible to realize low-power low-cost wireless devices and sensing units that are able to detect information from the distributed environment. Even though individual sensors can only perform simple local computation and communicate over a short range at low data rate, when deployed in large numbers they can form an intelligent collaborative network interacting with the surrounding environment in a large spatial domain. Sensor networks (SNs) characterized by low computational complexity, great learning capability, and efficient collaborative technology are highly desirable to discriminate, regulate and decide actions on real phenomena in many applications such as environmental monitoring, surveillance, factory instrumentation, defence and so on.

Chapter 15: Collaboration at the Basis of Sharing Focused Information: The Opportunistic Networks

This chapter is written by Bruno Apolloni, Guglielmo Apolloni, Simone Bassis, Gian Luca Galliani and Gianpaolo Rossi and discusses opportunistic networks. Opportunistic networks provide a communication protocol that is particularly suited to set up a robust collaboration within a very local community of agents. Like medieval monks who escaped world chaos and violence by taking refuge in small and protected communities, the authors point out that modern people may escape the information avalanche by forming virtual communities without relinquishing most of the benefits of the latest information and computer technology. A communication middleware to obtain this result is represented by opportunistic networks.

4.7 Part VII: Artificial Immune Systems

Chapter 16 provides a broad overview of artificial immune systems research, and focusses particularly on areas of natural immune systems that have been rather ignored by the AIS community in the past.

Chapter 16: Exploiting Collaborations in the Immune System: The Future of Artificial Immune Systems

This chapter, written by Emma Hart, Chris McEwan and Despina Davoudani, suggests some novel ways in which the natural immune system metaphor could be exploited to build new types of computational systems capable of meeting some of the challenges of the 21st Century, including self-configuration, self-maintenance, self-optimization and self-protection in an ever-changing environment. The authors focus particularly on aspects of the natural immune system which appear to have been largely overlooked by the artificial immune systems (AIS) research community in the past, and place significant emphasis on the design of *systems* rather than *algorithms*. The article puts forward some possible reasons why the potential promised by AIS has not yet been delivered, and suggests how this might be addressed in the future. The arguments are particularly relevant in light of recent advances in technology which present a new and challenging range of problems to be solved. A number of examples of systems in which steps are currently being taken to implement some of the mechanisms are then described. The chapter concludes with a discussion of an emerging field, that of *immuno-engineering* which promises a methodology which will facilitate maximum exploitation of immune mechanisms in the future.

4.8 Part VIII: Parallel Evolutionary Algorithms

Chapter 17 discusses the variety and importance of spatial interactions of populations in the natural world and demonstrates the relevance of these issues to parallel evolutionary algorithms.

Chapter 17: Evolutionary Computation: Centralized, Parallel or Collaborative

In this second chapter by Heinz Mühlenbein, the author focusses on the nature and importance of spatial interactions in evolutionary computation, and he also investigates cooperation and collaboration in this context. While "competition" is a fundamental component of Darwin's theory of natural selection, it can be argued that cooperation and collaboration also play a large role in evolution and population dynamics. In this chapter genetic algorithms with several different spacial interaction schemes are tested, and the results are discussed in relation to Darwin's ideas on the evolutionary gain achieved if subpopulations of individuals are periodically isolated from each other or from the main continental population of a species (i.e., the *continent-island cycle*).

4.9 Part IX: CI for Clustering and Classification

The four chapters in this section cover various aspects of pattern recognition, clustering and data mining.

Chapter 18: Fuzzy Clustering of Likelihood Curves for Finding Interesting Patterns in Expression Profiles

In this chapter Claudia Hundertmark, Lothar Jänsch and Frank Klawonn present a prototype-based fuzzy clustering approach that allows the automatic detection of regulatory regions within individual proteins. Cellular processes are mediated by proteins acting e.g. as enzymes (catalysts) in different metabolic pathways. Modifications are regularly made to specific regions of proteins within a living cell after that protein has been manufactured. The purpose of these post-translational modifications is to provide regulatory effects that will control the binding and activity properties of the modified proteins. In other words, the same protein will behave differently depending on the specific modifications made to it after its creation. Following the digestion of proteins into fragments (peptides), which is a necessary first stage of the work, the approach described in this chapter utilises likelihood curves to summarise the regulatory information of the peptides, based on a noise model obtained by an analytical process. Since the algorithm for the detection of peptide clusters is based on fuzzy clustering, their collaborative approach combines probabilistic concepts as well as principles from soft computing. However, fuzzy clustering is usually based on data points and its application to likelihood curves provided a considerable challenge for the authors. An interesting feature of this work is its potential transferability to noisy data from other applications, provided the noise can be specified by a noise model.

Chapter 19: A Hybrid Rule Induction/Likelihood Ratio-Based Approach for Predicting Protein-Protein Interactions

Mudassar Iqbal, Alex A. Freitas and Colin G. Johnson propose a new hybrid data mining method for predicting protein-protein interactions in this chapter. The purpose is to predict unknown protein interactions using relevant genomic information currently available. The new technique combines Likelihood-Ratios with rule induction algorithms and uses rule induction to discover the rules to partition the data. The discovered rules are subsequently interpreted as "bins" and used to compute likelihood ratios. In this way a rule induction algorithm learns classification rules, and these learned rules are used to improve the effectiveness of a likelihood ratio-based classifier, which is used to predict unknown protein interactions.

Chapter 20: Improvements in Flock-based Collaborative Clustering Algorithms

Esin Saka and Olfa Nasraoui begin their chapter with a brief survey of swarm intelligence clustering algorithms, and point out that since the early 90s, swarm intelligence (SI) has been a source of inspiration for clustering problems, and has been used in many applications ranging from image clustering to social clustering, and from document clustering to Web session clustering. The chapter then focuses mainly on a recent development: simultaneous data visualization and clustering using flocks of agents. The chapter presents some improvements to previous algorithms of this type and proposes a hybrid approach. Experiments on both artificial and real data confirm the validity of the approach and the advantages of the variants proposed in this chapter.

Chapter 21: Combining Statistics and Case-Based Reasoning for Medical Research

Case-based Reasoning (CBR) uses previous experience represented as cases to understand and solve new problems. A case-based reasoner remembers former cases similar to the current problem and attempts to modify solutions of former cases to fit the current problem. In this chapter Rainer Schmidt and Olga Vorobieva present a system, called ISOR, that helps to explain medical cases that do not fit a theoretical hypothesis. Indeed, it is often the case that no well-developed theory exists. Furthermore, at the start little knowledge or past experience may be available. This chapter focusses on the application of the ISOR system to the hypothesis that a specific exercise program improves the physical condition of dialysis patients. Additionally, for this application a method to restore missing data is presented.

Chapter 22: Collaborative and Experience-Consistent Schemes of System Modelling in Computational Intelligence

This study by Witold Pedrycz discusses a number of developments which form a conceptual and algorithmic framework for collaborative computational intelligence. First of all, the fundamentals of collaborative clustering are introduced in terms of information granules, i.e, fuzzy sets which emerge as a result of knowledge sharing. This is followed by the development of algorithmic definitions, which show the pertinent computing details. Hierarchies of clusters are also introduced, and experience-consistent fuzzy modeling is presented in the context of rule-based fuzzy models and neural networks.

References

1. Bezdek, J.C.: On the relationship between neural networks, pattern recognition and intelligence. International Journal of Approximate Reasoning 6, 85–107 (1992)
2. Bezdek, J.C.: What is computational intelligence? In: Zurada, J.M., Marks II, R.J., Robinson, C.J. (eds.) Computational Intelligence Imitating Life, pp. 1–12. IEEE Press, Los Alamitos (1994)
3. Deb, K.: Multi-objective optimization using evolutionary algorithms. John Wiley and Sons, Chichester (2001)
4. Dorigo, M., Maniezzo, V., Colorni, A.: The ant system: Optimization by a colony of cooperating agents. IEEE Trans. System Man Cybernetics Part B 26, 29–41 (1996)
5. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer, Heidelberg (2003)
6. Feigenbaum, E.A., McCorduck, P.: The Fifth Generation: Artificial Intelligence and Japan's Computer Challenge to the World. Addison-Wesley, Reading (1983)
7. Fogel, L., Owens, A., Walsh, M.: Artificial intelligence through simulated evolution. John Wiley, Chichester (1966)
8. Greenwood, G.W., Tyrrell, A.M.: Introduction to Evolvable Hardware: A Practical Guide for Designing Self-Adaptive Systems. Wiley-IEEE Press, Chichester (2006)
9. Holland, J.H.: Adaptation in natural and artificial systems. The University of Michigan Press, Ann Arbor (1975)
10. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, pp. 1942–1948 (1995)
11. Koza, J.: Genetic programming. MIT Press, Cambridge (1992)
12. Langton, C. (ed.): Artificial life: An overview. MIT Press, Cambridge (1995)
13. Lenat, D.B.: Cyc: A Large-Scale Investment in Knowledge Infrastructure. Communications of the ACM 38(11) (November 1995)
14. Lighthill, J.: Artificial Intelligence: A General Survey. In: Artificial Intelligence: a paper symposium, Science Research Council, UK (1973)
15. McCarthy, J., Minsky, M.L., Rochester, N., Shannon, C.E.: A proposal for the Dartmouth summer research project on artificial intelligence, Stanford University, August 31 (1955), `http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html`
16. Mehrotra, K., Mohan, C.K., Ranka, S.: Elements of Artificial Neural Networks. MIT Press, Cambridge (1996)
17. Nguyen, T.H., Walker, E.A.: A First Course in Fuzzy Logic, 3rd edn. Chapman and Hall, Boca Raton (2006)
18. Pawlak, Z.: Rough sets. Int. J. Computer and Information Sci. 11, 341–356 (1982)
19. Rechenberg, I.: Cybernetic solution path of an experimental problem. Technical Report Translation number 1122, Ministry of Aviation, Royal aircraft Establishment, Farnborough, Hants, UK (1965)
20. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn. Prentice Hall, NJ (2003)
21. Winston, P.H.: Artificial Intelligence, 3rd edn. Addison Wesley, MS (1992)
22. Wooldridge, M.: An Introduction to MultiAgent Systems. John Wiley & Sons Ltd, NY (2002)
23. Zadeh, L.A.: Fuzzy Sets. Information and Control 8, 338–353 (1965)