16

Multi-objective Evolutionary Algorithm for Temporal Linguistic Rule Extraction

Gary G. Yen

School of Electrical and Computer Engineering
Oklahoma State University
Stillwater, OK 74078-5032, USA
gyen@okstate.edu

Summary. Autonomous temporal linguistic rule extraction is an application of growing interest due to its relevance to both decision support systems and fuzzy controllers. In the presented work, rules are evaluated using three qualitative metrics based on their representation on the truth space diagram. Performance metrics are then treated as competing objectives and Multiple Objective Evolutionary Algorithm is used to search for an optimal set of non-dominated rules. Novel techniques for data pre-processing and rule set post-processing are developed that deal directly with the delays involved in dynamic systems. Data collected from a simulated hot and cold water mixer and a two-phase vertical column is used to validate the proposed procedure.

16.1 Introduction

When performing process management decisions, a control algorithm makes use of the knowledge of the plant in the form of a model. In most cases, this knowledge is commonly derived from the first principles and/or laboratory and pilot plant experiments; and often such "ideal" knowledge is of little practical use under real world complications due to unaccounted factors and modeling uncertainties.

Human operators, on the other hand, make use of another type of model when in charge of process management decisions. After a long time in contact with the plant, process operators are capable of attaining some understanding of what factors govern the process and derive relationships between process variables based on intuition and past experience. This process was best described in [23] as "a cognitive skill of experienced process operators that fits the current facts about the process and enables the operators to assess process behavior and predict the effects of possible control actions." However, the knowledge attained in this fashion also presents critical deficiencies since wrong impressions on what is going on with the process will lead to operator misjudgment as described in [17]. Furthermore, incoherencies inside

G.G. Yen: *Multi-objective Evolutionary Algorithm for Temporal Linguistic Rule Extraction*, Studies in Computational Intelligence (SCI) **16**, 365–383 (2006)
www.springerlink.com

such knowledge propagate itself as "mis-knowledge" or "technical folklore" are passed down from one generation of process operators to the next.

By making use of linguistic information in the form of IF/THEN logical statements or rules, Expert Systems and Fuzzy Logic Controllers (FLCs) are technologies capable of enabling better process monitoring and control. FLCs have found applications in a variety of fields such as robotics [7], automated vehicles [5], and process control [19], to name a few. Expert Systems have been used in the Chemical Process Industry (CPI) to control, monitor [6], and understand process behaviors. Other applications of such knowledge based systems have been in operator training and for planning and scheduling of operations in control and maintenance [11], especially for getting a plant back online after a failure or abnormal operating condition.

Expert Systems can be built from knowledge inserted by human experts or acquired from historic data from the system. Knowledge bases made by polling information from experienced personnel not only incorporate the before mentioned "technical folklore," but also is intrinsically incomplete. Such rules pertain only to information that is critical or obvious to the operators; it is related to information just necessary for them to maintain desired plant conditions. Such information does not incorporate the knowledge of events that are lesser in significance or rarer in occurrence, but which affect the operation of the plant nonetheless. A complete rule base should possess information on almost all plant events that have an effect on the desired output or may change the variable under control. Finally, knowledge collected from experts is usually in the form of static rules loosely related to the real numerical world. Due to its lack of a mechanism to deal with the temporal behavior of the process, the rigid, non-adaptive knowledge devised in this fashion becomes inadequate for complete supervisory control of dynamic systems. Therefore, the solution lies on the development of an algorithm capable of autonomously extracting and improving a dynamic rule set for an expert system directly from process data.

As pointed out in an extensive survey of Knowledge Discovery [8], many technical fields are exploring rule extraction from data. Data Mining views the rule extraction problem as one of finding patterns that describe the information in a database [14]. Machine Learning on the other hand approaches the same problem by generating deterministic finite-state automata with generalizing abilities capable of describing the relationship between antecedents and consequents [25]. Yet, the most appreciable contributions made in process applications are all based on computational intelligence paradigms, such as Neural Networks, Fuzzy Logic, or Evolutionary Computation [9, 16, 12, 27]. Still, temporal reasoning is missing in most studies.

It is fundamental for the modeling of a dynamic system that the model used incorporates the concept of time. In [18], a temporal restrictor is first proposed as an entity that restricts a fuzzy proposition, expressed as 'A be B T,' which translates into 'A is satisfying the fuzzy predicate B, taking into account the temporal restrictor T.' The temporal restrictor is expressed as a membership function. The conjunction *be* is a generic verb of being which can take different temporal connotations according to the rule. In this way, it is possible to add to the extracted rules information in the form of 'temperature was hot one_hour_ago.' However, this approach is rudimentary and incomplete; it does not account for persistence of excitation, and treating time as a restricting factor greatly limits the generalization of the rules.

Bakshi and Stephanopoulos exploited temporal knowledge and proposed reducing the dimensionality in [2] by describing process behavior with *triangles*. The design is specifically crafted for fault detection applications. However, our goal is to explore relations between general variables, not just discrete faults. Data mining in a time series database was approached in [15] by using three different temporal attributes: *length*, *slope*, and *fluctuation* (signal-to-noise ratio over time). Nevertheless, the application is not targeted on rule extraction, but on time-series prediction.

Based on the widely applied Autoregressive Moving Average (ARMA) [1] models, Tsai and Wu [24] proposed to incorporate temporal relationships into fuzzy rules by matching antecedents with consequents a fixed number of time steps in the future. In [21] the architecture was extended to allow different discrete time delays to be used for each antecedent in single consequent rules. Due to the usage of discrete time delays however, the representation capability of the rule set was largely affected by the particular choice of time delays and it displayed great sensitivity to noise, especially related to datasets composed of data sampled from continuous systems. Displaying applications related to the stock market and the weather, Last et al. [15] applied stochastic preprocessing techniques to improve the meaningfulness of the data provided to the rule extraction mechanism. Based on the concept of internal clocks that biological organisms use for the learning of period and interval timing, Carse and Fogarty [3] proposed the usage of a temporal membership function for the averaging of sampled data in order to generate crisp values related to fuzzy time periods. Applications of such an approach have been documented in distributed adaptive routing control in packet switched communication networks [3]. Therefore, in this study, a particular fuzzy delay is assigned to the temporally averaged consequent of each rule, generating a structure of the form:

IF *condition 1* ***AND*** *condition 2* ***AND*** *condition 3...* ,
THEN *after a certain fuzzy delay, a control variable will be such.*

The statement between the IF and the THEN conjunction is the *antecedent* while the statement after the THEN conjunction is the *consequent.*

A crucial step in the autonomous extraction of rules is the method used to validate and compare those that are created. An optimal rule should be accurate, properly describes the dynamic relationship between its antecedent and consequent, and possesses enough data to support it. In a methodology introduced in [20], three metrics based on a Truth Space Diagram (TSD) capable of encapsulating and measuring each of these three goals were introduced and tested. However, it was also shown that in the general scenario such metrics cannot be independently optimized due to inherent confliction among them.

Multiple Objective Evolutionary Algorithm (MOEA) is a tool capable of performing efficient searches on high dimensional spaces to locate the Pareto front, a set of solutions that contain the best rule for each possible trade-off between competing goals. A growing research field, MOEA has already demonstrated successful applications in solving challenging benchmark problems [26] and real world applications [4].

In this chapter, three metrics developed in [20] are used under a novel dynamic treatment of the data to evaluate linguistic rules against process data. MOEA is then introduced to locate inside the high dimensional rule space the Pareto front of the antecedents that best describe (in the sense of different combination of metrics) a given consequent.

The remainder of the chapter is organized as follows. Section 16.2 describes the three metrics generation process, detailing the novel temporal data pre-processing, the development of the truth space diagrams, and the choice of relevant metrics. Section 16.3 introduces the use of MOEA for the automated extraction of rules and the necessary rule set post-processing. The simulation results of applying the proposed algorithm to the Hot and Cold water mixer and a two-phase vertical column are explored in Section 16.4, following by some final remarks and conclusions in Section 16.5.

16.2 Rule Evaluation

In order to guide and automate process management decisions, a rule must display three basic characteristics: high accuracy, precise antecedent/consequent relationship, and sufficient data support. However, it is seldom possible to maximize these three characteristics at the same time. For example, if an event is observed only a single time, it is trivial to develop a rule with 100% accuracy, however it will lack support from historical data and the probability that it will describe a whole family of similar events with comparative accuracy is small. For this reason there is a need to develop three qualitative metrics, each focusing on one of such competing characteristics.

In [20] a series of metrics were suggested, each capable of specifically representing a different quality of a rule. All metrics were designed with a qualitative measure manifested through the Truth Space Diagram (TSD). In the first part of this section, the efficiency of the TSD is further enhanced with the introduction of a novel pre-processing strategy for the representation of the temporal behavior of the plant into the linguistic rules. The concept of the TSD is then introduced taking in consideration the enhanced temporal representation and finally the three metrics of interest are introduced.

16.2.1 Data Pre-processing

In previous applications of the TSD methodology, data pre-processing took place in the following manner: 1) The consequent data was shifted backwards in time by fixed intervals so that each set of antecedents matched three consequent values corresponding to short, medium, and long delays; 2) The crisp input-output data was fuzzified. Fuzzification proceeded through the application of Equation (16.1) by using triangular membership functions to classify each variable into three fuzzy categories – low, medium, and high:

$$\mu_x^{i,j} = \begin{cases} \frac{a_j - x_i}{a_j - b_j}, & x_i \in [a_j, b_j] \\ 0, & \text{otherwise}, \end{cases} , \tag{16.1}$$

where j=1 to 3, i=1 to n, x_i is the crisp numerical value of the i^{th} input or output variable, $\mu_x^{i,j}$represents the fuzzy membership value of x_i in the j^{th} fuzzy category, a_j and b_j are the fuzzy set break points for category j, and n is the maximum number of datasets in the input-output data. Figure 16.1 illustrates the fuzzy classification of one variable into three fuzzy categories.

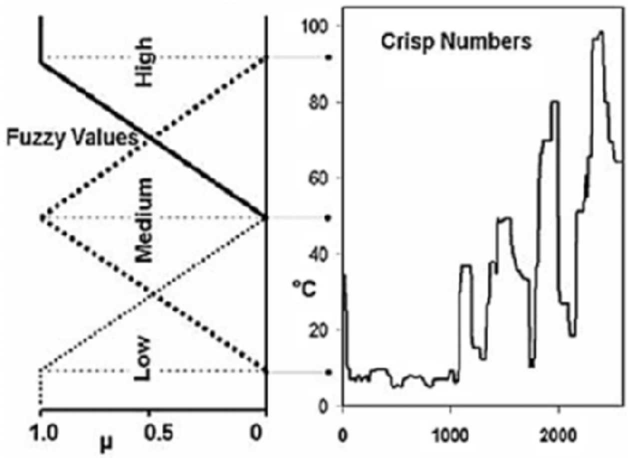


Fig. 16.1. Fuzzy classification procedure for the antecedents. In this case, temperature with centers at 10^o, 50^o, and 90^oC.

The fuzzification of the physical crisp data in such a manner leads to great generalization capabilities, inherent noise rejection, and direct rule interpretation by human operators. Although ideal for the treatment of the antecedents, the manner through which the dynamic temporal element is incorporated into the consequent lacks such benefits. In essence, consequences for plant characteristics at any given time are only observed at discrete instants. Since a consequent's time delay contains variability as much as physical characteristics of a system, although rule extraction is possible, it requires extensive data in order to determine the true correlations through time. Moreover, the previous approach relies on *accurate* knowledge on the inherent major delays of the plant in order to set up the number of iterations that correspond exactly to the operator's understanding of small, medium, and long delays.

In the present work, such deficiencies are addressed by dealing with time uncertainties in a novel manner that is different in essence from the fuzzification of physical variables. The application of such approach leads to a meaningful linguistic description that maintains all the previously stated benefits while better capturing the temporal characteristics of dynamic plants. Instead of simply shifting the data to obtain a single measurement to represent a delay, averaged values of a consequent are obtained for each fuzzy delay region through Equation (16.2) and it is those averaged values that are then classified into the membership function of the consequent by Equation (16.3).

$$y^i_\delta(t) = \frac{\sum_{k=t^1_\delta}^{t^3_\delta} M_\delta(k) \cdot y^y(t+k)}{\sum_{k=t^1_\delta}^{t^3_\delta} M_\delta(k)}, \tag{16.2}$$

$$\mu^{i,j}_{y,\delta}(t) = \frac{a_j - y^i_\delta(t)}{a_j - b_j}, \tag{16.3}$$

where $y^i(t)$ is the crisp measurement of the i^{th} consequent at time t, $y^i_\delta(t)$ corresponds to its arithmetical average for a given fuzzy delay δ, $\delta \in$[short , medium , long], t^i_δ are the fuzzy set break points ($i \in$[1, 2, 3]) for category δ, $\mathrm{M}_\delta(t)$ denotes the membership function of a fuzzy delay δ, and $\mu^{i,j}_{y,\delta}(t)$ is the fuzzy membership value of the i^{th} consequent for a fuzzy delay δ. An example of the application of the procedure involving Equations (16.2) and (16.3) can be seen in Figure 16.2, where the fuzzy membership value of the consequent $y^1(t)$ is calculated at time 20s for *medium* delay and *low* temperature, i.e. $\mu^{1,l}_{y,m}(20)$.

By processing the process data from the antecedents using (1) and that from the consequents with (2-3), crisp data is translated into linguistic variables. It is important to note that although each antecedent relates to a single linguistic variable, due to the introduction of the fuzzy delay, each consequent is represented by three fuzzy variables, each related to a different delay membership function.

16.2.2 Truth Space Diagram

The TSD is a two-dimensional space in which a series of metrics capable of quantifying the quality of a particular cause-and-effect rule can be obtained. Each TSD relates to a single rule. For every data point extracted either from mathematical simulations, pilot plant experiments, or real-word sensory data, a point is plotted in the TSD according to its truth of the antecedent *Ta* and the truth of the consequent *Tc*. Both parameters are calculated as geometrical

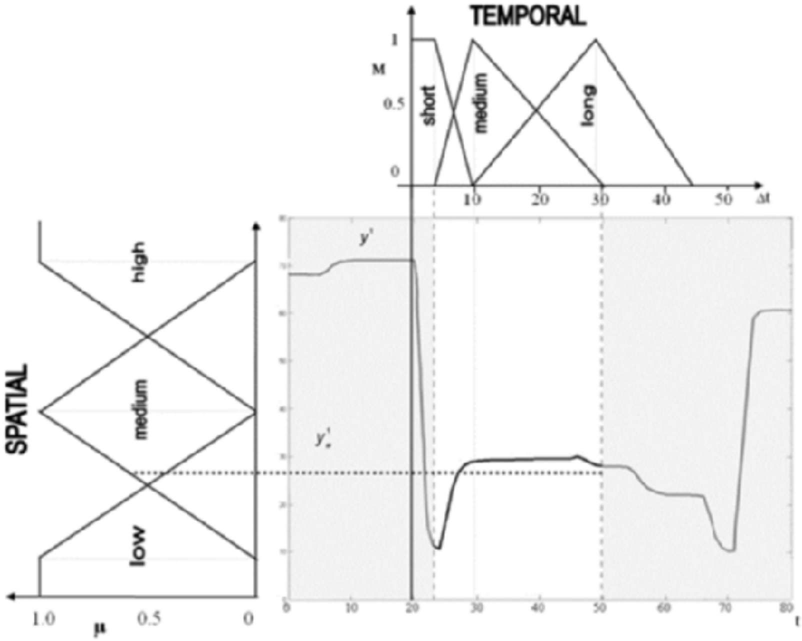


Fig. 16.2. Proposed physical and temporal two-step fuzzification procedure.

means of the fuzzy membership function of each variable of the antecedents and consequents. Hence, the truth space delimited by *Ta* and *Tc* is bounded between 0 and 1 in which a value equal to 0 means absolute false while a value of 1 means absolute truth.

Designed in this fashion, the TSD represents a one-to-one mapping from the dataset from the real (numerical) space to a new (truth) space defined by the linguistic statements of a specific rule. The TSD can be divided into four quadrants and each quadrant provides different information about the linguistic rule. For example, consider point A in Figure 16.3. The values for *Ta* and *Tc* are high for this data point, i.e. the predicted consequent follows the appointed antecedent or the cause and effect match according to the relevant rule statement. This reveals that the information expressed in the linguistic rule is contained within the numerical data. Hence, many points in Quadrant II (denoted as Quad **II**) of the TSD reflect the validity of the rule in question. Consequentially, points in Quadrant IV (denoted as Quad **IV**) show that the rule statement is false, i.e. what the antecedent of the rule express does not lead, in most cases, to the predicted consequent. An example of this can be seen from data point B in Figure 16.4. Similarly, points in Quadrant I demonstrate the incompleteness of the rule, since the predicted consequent was due to an event(s) other than the one expressed in the antecedents of the rule. Finally, the presence of a cluster of points in Quadrant III show the possibility

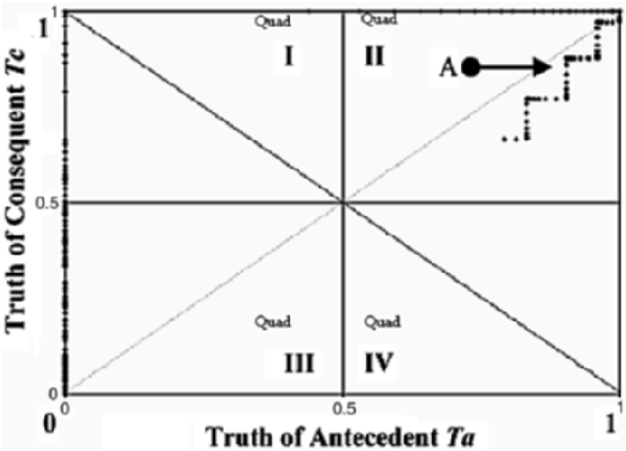


Fig. 16.3. TSD for a meaningful rule extracted from process data with sufficient supporting evidence.

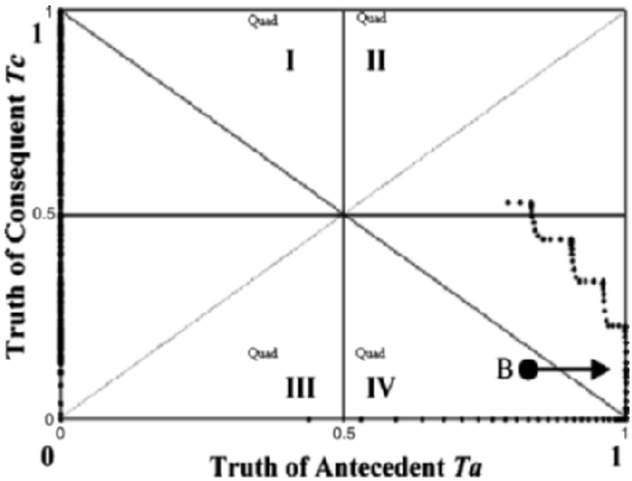


Fig. 16.4. TSD for a rule that was proven inaccurate in a significant number of points in the process data.

of that a rule is valid, however the amount of data currently available does not allow yet for a conclusion to be drawn with enough confidence. The points that lie on the vertical and horizontal axis show that either the antecedent or the consequent of a particular rule were not expressed in the data.

16.2.3 Numerical Metrics

As mentioned previously, the goal of the presented work is to extract rules that present high accuracy, precise antecedent/consequent relationship, and that are supported by sufficient data. By using the TSD, it is possible to obtain metrics for each of these conflicting goals. In order to transform the problem into one of minimization however, the actual metrics of interest are converted into: rule inaccuracy, antecedent/consequent mismatch, and lack of supporting evidence in the dataset. To improve the performance of the rule extraction algorithm, all metrics presented here are normalized to the interval [0,1].

Metric 1: *rule inaccuracy* – A rule is deemed inaccurate when its antecedent is observed but the consequence that follows after the prescribed delay does not match the predicted behavior. As mentioned previously, in the TSD this concept relates to the points in the Quadrant IV (i.e. set Q4), which relate to high truth of the antecedent but low truth of the consequent. The number of data points in Q4 (i.e. n_4) can therefore be used as a relative measure of inaccuracy, however it is necessary to normalize this number by dividing n_4 by the total number of data points in which the rule antecedents were observed with sufficient confidence, i.e. the sum of the points in Quadrant II (n_2) and in Quadrant IV (n_4). Equation (16.4) summarizes m_1, the rule inaccuracy metric.

$$m_1 = \begin{cases} \frac{n_4}{n_2+n_4}, & n_2 + n_4 > 0 \\ 1, & \text{otherwise} \end{cases} . \tag{16.4}$$

Metric 2: *antecedent/consequent mismatch* – from a good rule it is expected that the value of Tc should match the value of the Ta. In other words, the intensity in which the antecedents are observed should be equal to the intensity of the resulted consequent. In real world scenarios however, Tc is affected by both the quality of the rule and the quality of the available data (e.g. noise corruption). By analyzing the data points in Quadrant II (i.e. set Q2), it is possible to measure antecedent/consequent mismatch directly by summing all distances from each data point in it to the diagonal of the TSD. Defining Ta_i and Tc_i respectively as the truth of the antecedent and the truth consequent for rule i, $\|.\|$ as the Euclidian norm, and since 0.3536 is the maximum distance to the diagonal, this second metric is stated as shown in Equation (16.5):

$$m_2 = \begin{cases} \frac{\sum_{i \in Q2} ||Ta_i - Tc_i||}{0.3536 \cdot n_2}, & n_2 > 0 \\ 1, & \text{otherwise} \end{cases} . \tag{16.5}$$

Metric 3: *lack of supporting evidence in the dataset* – Since there is a need for sufficient information inside the available data for any conclusion to be drawn, this metric is crucial for the success of any data driven rule extraction method. Using the TSD representation, Equation (16.6) is built for this purpose.

$$m_3 = 1 - \frac{n_{TSD}}{n_{data}}, \tag{16.6}$$

where n_{TSD} is the total number of data points mapped in the TSD excluding points on the abscissa ($Ta = 0$); and n_{data} is the total number of data points available in the dataset.

16.3 Rule Extraction

Evolutionary Algorithms (EA) is commonly regarded as a family of stochastic search procedures that is inspired by computational models of natural evolutionary processes to develop computer based optimization problem solving systems [22]. Being a population based algorithm, in EA each candidate solution is represented as an individual. When evolving towards better solutions, the individuals that better meet the optimization goal (individuals with greater *fitness*) have a greater probability of being selected to take part in the creation of the individuals of the new generation.

For problems that have multiple conflicting goals that cannot be directly combined into a single scalar measure of fitness, Multiple Objective Evolutionary Algorithm (MOEA) provides a method through which a population of solutions can evolve towards a set of solutions within which no solution is better than another in all optimization goals. By defining that an individual *dominates* another when all of its optimization goals are closer to the ideal values than those of the other individual, such a set can be referred to as the non-dominated set. The non-dominated set among all possible solutions is called the Pareto front and its determination is then the ultimate goal of MOEA. Therefore, as shown in (7), in MOEA the fitness F is a vector of the optimization goals, in this case represented by the three goodness metrics.

$$F = [m_1, m_2, m_3]. \tag{16.7}$$

In the present work, MOEA in the form presented in [26] is used once for every consequent to evolve an initial random population of related rules towards the Pareto front of the tri-dimensional space defined by F. Therefore, for each consequent, a set of equally good (in the sense of the minimization of the three previously defined metrics) antecedents is extracted based on their relative success.

As laid down in the pseudocode in Table 16.1, the first step of implementing an MOEA algorithm is the generation of the initial population of candidate solutions. In order to guarantee an unbiased and diverse population while maintaining a low computational demand, 20 initial individuals are generated with random antecedent values, clearly 20 is an ad hoc choice that needs to be quantified in future research. In the following step, the three metrics are calculated for the rules formed by the antecedents of each individual and the consequent related to the current MOEA run. It is also in this step that each individual is assigned a rank value according to their relative success in minimizing the elements of the fitness vector F (i.e. the concept of Pareto optimality). In particular, the ranking scheme discussed in [10] is implemented, in which an individual is assigned a rank value equal to one plus the number of individuals it is dominated by.

Table 16.1. MOEA pseudocode

1. Generate initial population;
2. Evaluate the metrics of all individuals and rank them;
3. for (i=1: *maximum_generation*)
4. Choose parents with probability inversely proportional to their ranks;
5. Perform crossover operation on parents to generate new individuals;
6. With probability equal to the mutation rate, perform mutation procedure;
7. Evaluate all three metrics on the new individuals;
8. Update population ranking.

The third step in the presented pseudocode is the first in its main loop and it relates to the selection of two individuals that will be involved in the generation of new individuals to the population. The selection is performed stochastically by assigning a greater selection probability to individuals with smaller rank values, and therefore individuals with greater fitness. The individuals in this way chosen are denominated parents and, in step 4, part of their individual solutions is exchanged in the operation termed crossover. Through the crossover operation, two new individuals (solutions) are formed, combining elements of both parents. In the following step, mutation, another biologically inspired process, may affect with a specific probability (defined as the mutation rate) the newly generated individuals. In MOEA, mutation takes place by randomly modifying an arbitrary portion of the solution related to a given individual. Independent of the occurrence of mutation in step 5, on step 6 the fitness vector F is evaluated for the two new individuals, followed by the updating of the ranks of all individuals in the population. A generation is then concluded and the algorithm returns to step 3 until a maximum number of generations is reached.

After MOEA generates a set of non-dominated rules for each consequent, thresholds are used over each metric to eliminate outliers and establish minimum acceptable performances (e.g. minimum degree of accuracy required of a rule). Another post-MOEA data processing involves removal of time-redundant information from the rule set. If, for instance, a consequent should develop quickly and remain unchanged for a long time throughout the dataset, the antecedents would be credited with both short and long-term effects even though the long-term effect is only a matter of persistence. Time redundancy then refers to rules with equal antecedents and physical consequents, but with different consequent delays. In such cases, the rule related to the longer consequent delay is removed.

16.4 Simulation Results and Discussion

The results of the application of the proposed rule extraction algorithm to two examples are presented in this section. The first is a proof-of-the-concept computer simulation. The simulation pertains to a hot and cold water mixer that is sufficiently challenging to demonstrate the impact of each of the algorithm sub-systems while at the same time it remains simple enough to be intuitively understandable. The second pertains to the results of rule extraction over process data collected from the operation of a laboratory scale two phase column. Different from the computer simulation, the two phase column is an actual plant, subject to real world noise levels, sensor calibrations and other implementation imperfections.

16.4.1 Hot and Cold Water Mixer

To demonstrate the feasibility and clarify implementation details of the proposed process for extracting temporal cause and effect relationships, data was acquired from a Hot and Cold water mixer simulator shown in Figure 16.5. The simulator incorporates real world dynamics such as transport and measurement delays and is capable of adding deviations such as measurement bias and process drifts that have an ARMA stochastic behavior, noise and valve "sticktion." This is a simple example, but incorporates behaviors which are representative of a majority of unit operations within the CPI. The simulation was non-linear, had multiple inputs and its dynamics (such as hydrodynamic delay) depended upon operation conditions. A detailed description containing the mathematical model of the Hot and Cold water mixer simulator is available in [27].

For the purpose of generating data, the four input variables were manipulated, the flow of each input tube (F_1 and F_2) changing randomly at every 20 seconds and the input temperatures (T_1 and T_2) changing randomly at every 40 seconds. The periods of manipulation of the variables were shifted so as not to lead into two changes occurring at the same time. Their effect on the temperature at the output of the mixer stream (T_3) was measured over time. All flow variables were restricted to the interval [0,30] kg/min and the temperature variables to [0,100] ^{o}C.

According to the proposed data pre-processing procedure, physical variables were fuzzified with centers at 10, 50 and 90 ^{o}C for the temperatures and at 2.5, 15 and 25 kg/min for low, medium and high flow rates respectively. For the fuzzy delay, centers were placed at 3, 7 and 20 seconds for short, medium and long delays respectively. Note that long delay rules will be harder to calculate since the input variables will change randomly at faster rates. The dataset is intentionally devised in this form to challenge the rule extraction procedure with data of different degrees of quality.

The proposed MOEA based on the three selected metrics was implemented over the pre-processed data generating a non-dominated set of rule candidates

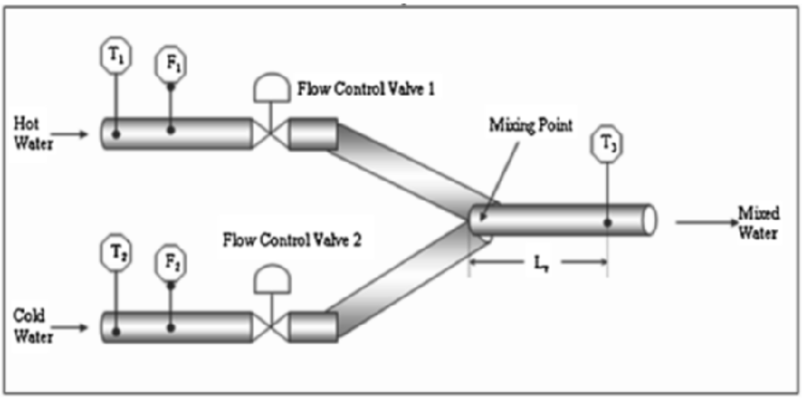


Fig. 16.5. The hot and cold water simulator used for validation of the rule extraction algorithm.

for each consequent. An initial population of 20 randomly generated individuals was allowed to evolve through the course of 200 maximum generations. The individuals received a rank equal to one plus the number of individuals it was dominated by. At each generation, parents were chosen according to a probability inversely proportional to their rank. For the generation of new individuals, crossover was implemented with a single crossover point and a mutation rate of 0.01 was used. An elitism scheme was implemented to guarantee that all non-dominated solutions were preserved during the evolution process. Figure 16.6 displays the obtained non-dominated set containing 13 antecedent combinations relating to high temperature at T_3 after a long delay.

For post-processing, the minimum acceptable accuracy of the extracted rules was set at 90%, a minimum of 1% of the information inside the observed dataset was necessary to validate a rule, and a maximum spread of 0.6 around

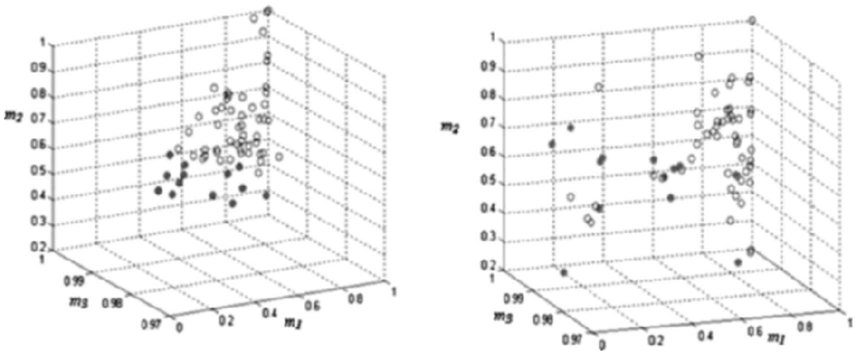


Fig. 16.6. Distributions of individuals related to two different consequents in the metric space at generation 200. Filled circles form the non-dominated set.

the diagonal of the TSD was allowed. In terms of the minimization metrics m_1, m_2, and m_3, the corresponding thresholds were 0.1, 0.99 and 0.6 respectively. Finally, rules that were time-redundant were removed to generate the final rule set.

Through the outlined process, the presented algorithm was capable of extracting 49 rules out of a possible set of 729 rule combinations (containing both "good" and "bad" rules). Some examples of the obtained rules are shown below:

- **IF** T_1 is high **AND** F_1 is medium **AND** T_2 is medium **AND** F_2 is low **THEN** after a medium delay T_3 will be high.
- **IF** T_1 is low **AND** F_1 is low **AND** T_2 is low **AND** F_2 is low **THEN** after a long delay T_3 will be low.
- **IF** T_1 is medium **AND** F_1 is low **AND** T_2 is high **AND** F_2 is high **THEN** after a short delay T_3 will be high.

Good rules are those that express the phenomenological-based, cause-and-effect mechanism as a logical relation between their antecedent and consequent parts. Consequentially, bad rules are defined as those that are inconsistent with the process phenomena. Therefore, in order to evaluate the quality of the 49 extracted rules, each one of them had its antecedents implemented in the simulator and those that demonstrated matching consequents were deemed good rules. As a result, 5 of those rules were rejected demonstrating a success ratio of 89.8% of the proposed rule extraction algorithm. Moreover, most rejected rules pointed to borderline consequents (e.g. the measured T_3 would be 78^oC, when the maximum value acceptable for a medium fuzzy range was 70^oC). Such scenarios reflect the choice of fuzzy membership function centers, left at the discretion of the operator.

As mentioned previously, any rule extraction procedure can only produce results as good as the data provided. This simulation was intentionally designed to provide much sparser and more noise corrupted data for the extraction of rules related to long delays. Among the 44 good rules in the final set, only 6 of those portrayed long delays, while the expected from a fully representative dataset would be one third of the total. Since the algorithm minimizes inaccuracy (m_1) while at the same time evaluating the amount of supporting evidence (m_3), the lower number of long delay rules extracted demonstrates the success of the procedure in avoiding unsupported rules to be presented to the operator in the final set.

16.4.2 Two-Phase Flow Column

Gas-liquid two-phase flows are defined as the flow of a mixture of the two phases, flowing together, through a system. The description of a two-phase flow in pipes is highly intricate due to the various existence of the interface between the two phases [27]. For gas-liquid two-phase flows, the variety of interface forms depends on the flow rates, phase properties of the fluid and on the inclination and the geometry of the tube. Generally, for vertical gas-liquid two-phase flows, the flow regimes are mainly determined by the phase flow rates. In this case, Bubbly, Slug, Churn and Annular are four significant regimes that can be recognized as standard patterns in the chemical industry. The characteristics of these four patterns are shown in Figure 16.7. Each of these four patterns has a distinguished air/water density and flow speed ratio. These characteristics have a strong influence on pressure drop and heat and mass transfer mechanisms in a system, and are very important in the chemical process industries.

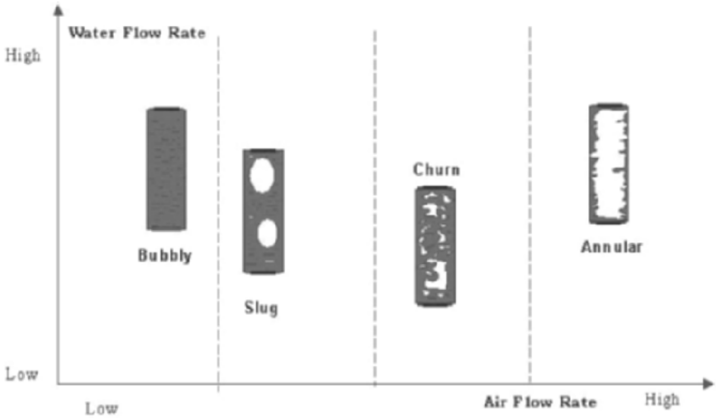
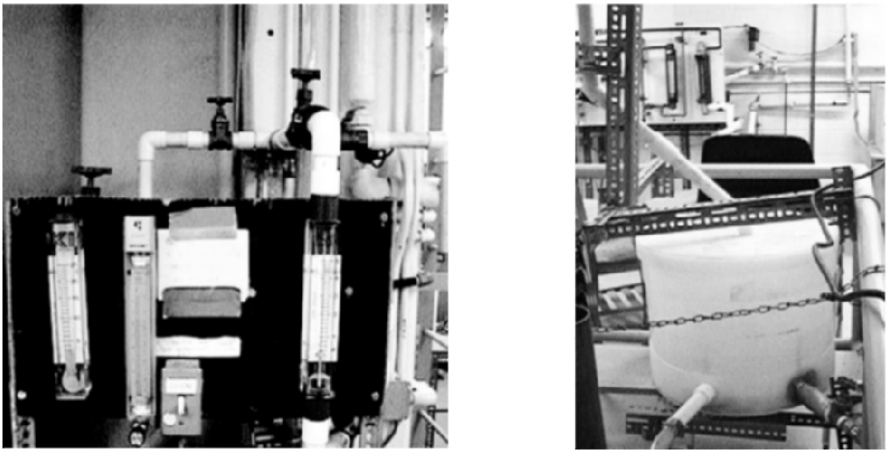


Fig. 16.7. Water/air flow ratio of four major two-phase vertical flow patterns.

Using the laboratory-scale vertical two-phase column shown in Figure 16.8, real process data (subjected to ambient noise) was acquired from the pressure drop in the column (ΔP) while independently varying the flow rates of air (F_a) and water (F_w). As illustrated in Figure 16.9, the two inputs were alternately modified at every 30 seconds (150 samples).

Due to the second order nature of the response of the two-phase column, higher fuzzyfication definition was required. Therefore, five membership functions were used to generate the levels *very low*, *low*, *medium*, *high* and *very high*; leading to 375 possible rules. Using the same procedure discussed for the hot and cold water mixer simulator, 22 satisfactory rules were extracted from the two-phase column data. Some examples of the extracted rule set are shown below.

- **IF** F_a is high **AND** F_w is very low **THEN** after a short delay ΔP will be very low.
- **IF** F_a is medium **AND** F_w is low **THEN** after a medium delay ΔP will be low.



(a) (b)

Fig. 16.8. Details of the base (a) and the top (b) of the two-phase flow column.

- **IF** F_a is very low **AND** F_w is medium **THEN** after a long delay ΔP will be medium.

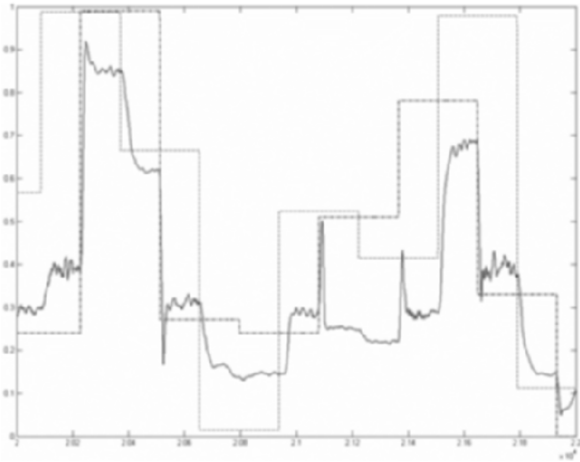


Fig. 16.9. Two-phase column pressure drop (solid) in response to variations in water flow (dashed) and air flow (dot-dashed).

16.5 Conclusions

As demonstrated though the application of the procedure on the data collected from the simulated hot and cold water mixer, the proposed rule extraction procedure succeeded in autonomously generating a viable rule set from a less than completely representative data set. The use of MOEA as an optimization algorithm allowed for three conflicting metrics to be evaluated simultaneously leading to the final extraction of optimal non-dominated rule sets. Both pre-processing, involving the representation of each rule inside a TSD, and post-processing, which allowed for the removal of time-redundant rules, were applied successfully and with beneficial outcomes.

The presented work assumes that all antecedents remain unaltered until the consequent is observed, however its ultimate goal is to achieve rules that specifically account for antecedent persistence such as: "**IF** (the reactor temperature has been high for a short period of time) **AND** (the speed is in manual for an extended period of time) **THEN** (in a short while the product will be slightly yellow)." To achieve such a goal, persistence must also be added to the dynamic persistence of the rule, which will lead to different data pre-processing requirements. To further improve the dynamic representation of complex plants, there may also be need to explore novel operators as opposed to the logical ones currently in use, such as "shortly succeeded by." Future work also focuses on the autonomous definition of fuzzy membership centers and on procedures capable of improving the quality of the extracted rule set as new batches of data becomes available over time.

References

[1] K. Astrom, T. Soderstrom. Uniqueness of the maximum likelihood estimates of the parameters of an ARMA model. *IEEE Transactions on Automated Control*, 19:769 –773, 1974
[2] B. Bakshi, G. Stephanopoulos. Representation of process trends - IV: Introduction of real-time patterns from operating data for diagnosis and supervisory control. *Computers & Chemical Engineering*, 18:303–332, 1994
[3] B. Carse, T. Fogarty. Evolutionary learning of temporal behavior using discrete and fuzzy classifier systems. In: *Proceedings of the International Symposium on Intelligent Control*, pp.183–188, 1995
[4] F. Chen, Z. Chen, Z. Jiao. A novel processing for multiple gases detection. In: *Proceedings of the World Congress on Intelligent Control and Automation*, pp. 2186–2189, 2002
[5] K. Cheng, J. Chen. A fuzzy-nets training scheme for controlling nonlinear systems. *Computers & Industrial Engineering*, 31:425–428, 1996
[6] C. Cimander, T. Bachinger, C. Mandenius. Integration of distributed multi-analyzer monitoring and control in bioprocessing based on a real-time expert system. *Journal of Biotechnology*, 103:327–348, 2003

[7] C. Collewet, G. Rault, S. Quellec, P. Marchal. Fuzzy adaptive controller design for the joint space control of an agricultural robot. *Fuzzy Sets and Systems*, 99:1–25, 1998

[8] M. Craven, J. Shavlik. Rule extraction: where do we go from here? *Technical Report 99-1*, University of Wisconsin Machine Learning Research Group, Madison, WI, 1999

[9] M. Delgado, A. Gomez-Skarmeta, F. Martin. A fuzzy clustering-based rapid prototyping for fuzzy rule-based modeling. *IEEE Transactions on Fuzzy Systems*, 5:223–233, 1997

[10] C. Fonseca, P. Fleming. Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In: *Proceedings of the International Conference on Genetic Algorithms*, pp.416–423, 1993

[11] D. Fonseca, G. Knapp. An expert system for reliability centered maintenance in the chemical industry. *Expert Systems with Applications*, 19:45–57, 2000

[12] X. Fu, L. Wang. Rule extraction by genetic algorithms based on a simplified RBF neural network. In: *Proceedings of the Congress on Evolutionary Computation*, pp.753–758, 2001

[13] F. Hewitt. Measurement of Two Phase Flow Parameters. Academic: London, 1978

[14] I. Jagielska. Linguistic rule extraction from neural networks for descriptive data mining. In: *Proceedings of the Knowledge-Based Intelligent Electronic Systems Conference*, pp.89–92, 1998

[15] M. Last, Y. Klein, A. Kandel. Knowledge discovery in time series databases. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, 31:160–169, 2001

[16] S. Mitra, Y. Hayashi. Neuro-fuzzy rule generation: survey in soft computing framework. *IEEE Transactions on Neural Networks*, 11:748–768, 2000

[17] E. Oshima. Computer aided plant operation. *Computers & Chemical Engineering*, 7:311–329, 1983

[18] A. Pesonen, A. Wolski. Quantified and temporal fuzzy reasoning for active monitoring in rapidbase. In: *Proceedings of Symposium on Tool Environments and Development Methods for Intelligent Systems*, pp.227–242, 2000

[19] R. Rhinehart, P. Murugan. Improve process control using fuzzy logic. *Chemical Engineering Process*, 91:60–65, 1996

[20] N. Sharma. Metrics for evaluation of the goodness of linguistic rules. MS Thesis, Oklahoma State University, School of Chemical Engineering, 2003

[21] N. Sisman, F. Alpaslan. Temporal neurofuzzy MAR algorithm for time series data in rule-based systems. In: *Proceedings of the International Conference on Knowledge-Based Intelligent Electronic Systems*, pp.316–320, 1998

[22] M. South, C. Bancroft, M. Willis, M. Tham. System identification via genetic programming. In: *Proceedings of the UKACC International Conference on Control*, 912–917, 1996

[23] G. Stephanopoulos, C. Han. Intelligent systems in process engineering: A review. *Computers & Chemical Engineering*, 20:743–791, 1996

[24] C. Tsai, S. Wu. A study for second-order modeling of fuzzy time series. In: *Proceedings of the IEEE International Fuzzy Systems Conference*, pp.719–725, 1999

[25] A. Vahed C. Omlin. Rule extraction from recurrent neural networks using a symbolic machine learning algorithm. In: *Proceedings of the International Conference on Neural Information Processing*, pp.712–717, 1999

[26] G. Yen, H. Lu. Dynamic multiobjective evolutionary algorithm: adaptive cell-based rank and density estimation. *IEEE Transactions on Evolutionary Computation*, 7:253–274, 2003

[27] G. Yen, H. Lu. Acoustic emission data assisted process monitoring. *ISA Transactions*, 41:273–282, 2002