I Krajowa Konferencja
Algorytmy Ewolucyjne
Murzasichle, 12-15.06.1996

Application of the Coevolutionary Strategy to the Rule Generation in Fuzzy Systems*.

Artur Wloszek
s141917@ia.pw.edu.pl

Pawel D. Domanski
domanski@ia.pw.edu.pl

Institute of Automatic and Computational Engineering
Warsaw University of Technology
ul. Nowowiejska 15/19
00-665 Warszawa, Poland

Abstract

This paper presents the decision rules searching algorithm for fuzzy systems based on the coevolutionary notion. At first the problem is stated and the coevolutionary idea is proposed. The classical rule exctraction problem is decomposed to the several cooperating systems. Each subsystem is based on the evolutionary strategy, and coevolution is obtained through the fitness function calculations and ranking rules selection. The method was investigated from different points of view. Mainly the number of subsystems and exchange rate of rule changes was considered.

1. INTRODUCTION.

The source of the fuzzy rules is the main problem in the fuzzy system design. In somee cases we can extract them from the verbal knowledge, but in many other situations it would be worth to incorporate knowledge from numerical experiments.

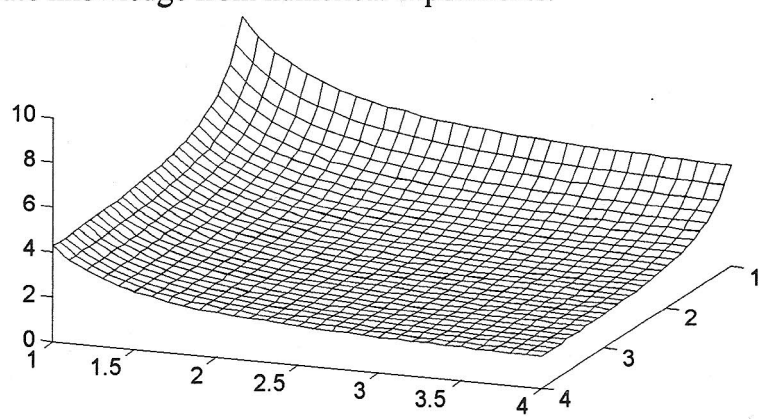


Fig. 1 The surface of nonlinear function.

There are many different approches applied to this specific problem: neural networks, fuzzy neural networks, decision trees and evolutionary techniques. Genetic approach to this problem was investigated in many papers, but the classical genetic algorithm is not too efficient. The introduction of the coevolution notion and problem decomposition can enlarge the overall algorithm efficiency.

2. PROBLEM FORMULATION.

The main goal is to build a fuzzy model. Lets say that the original system is a nonlinear function of two variables (1). The nonlinear function characteristics is sketched on Fig.1.

$$f(x_1, x_2) = (1 + x_1^{-2} + x_2^{-1.5})^2; \quad x_1 \in \langle 1,4 \rangle \text{ i } x_2 \in \langle 1,4 \rangle \tag{1}$$

* This work was supported from the Committee of the Scientific Research under Grant 8T 11A 003 10.

2.1. Fuzzy Model.

We are building a qualitative model basing on the fuzzy logic and fuzzy inference notion [1], [2]. The graphical representation of the fuzzy model is presented on Fig.2. The Fuzzyfication module translates the numerical data to the linguistic variables. The Rule Base keeps the collection of ***if-then*** rules while the Inference Machine provides the system with the inference implementation. The Defuzzyfication module decodes the linguistic variables back to the numerical data.

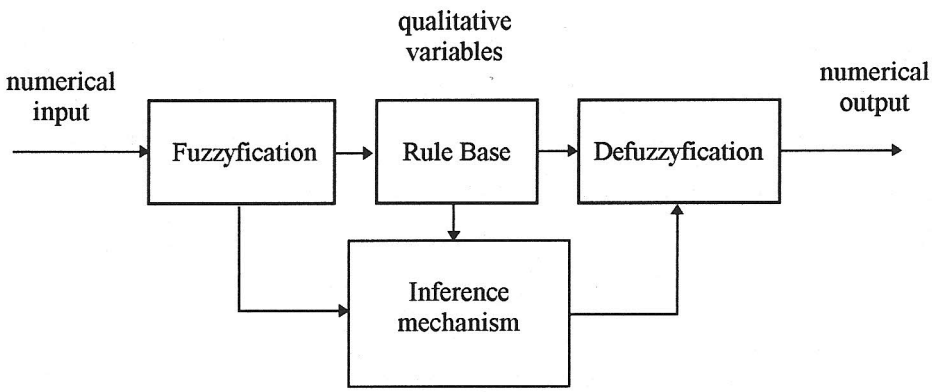


Fig. 2 The fuzzy inference scheme.

Variables x_1 and x_2 (inputs) and the function value y (output) are implemeted as the qualitative variables. Each of them has luinguistic values in labels T(x) = { $T_1^x .. T_j^x$ } and T(y) = { $T_1^y .. T_k^y$ }. All this labels are represented by fuzzy triangular sets. The membership function $\mu_{T_i^k}(u)$ represents the grade of membership of the numerical value u of signal k^{th} to the fuzzy label T_i^k. The decision rule can be written as follows:

IF x_1 is $T_{i_1}^{x_1}$ AND x_2 is $T_{i_2}^{x_2}$ THEN y is T_j^y

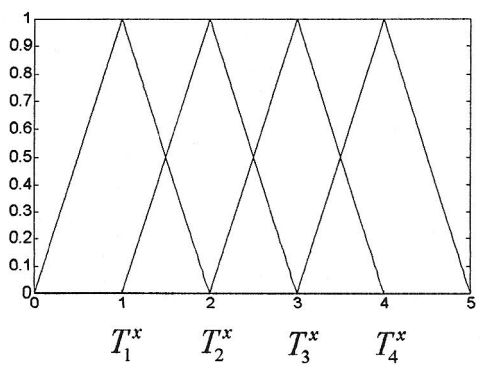


Fig. 3 Fuzzy partition for input signals.

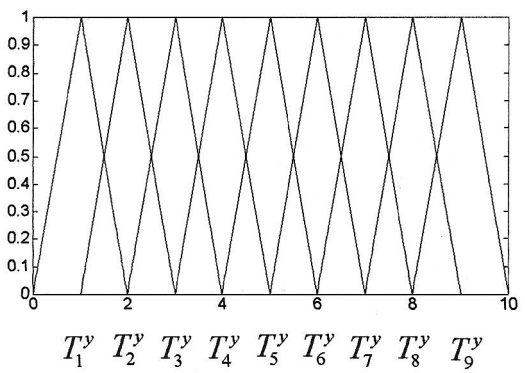
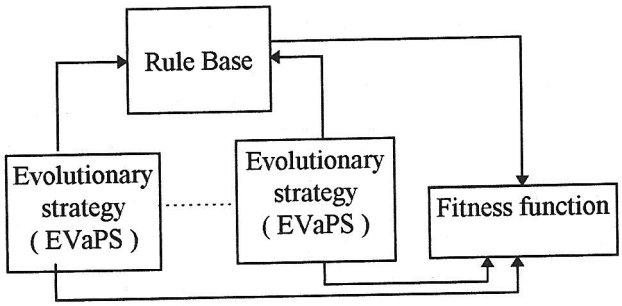


Fig. 4 Fuzzy partition for output signal.

On Fig. 3 and Fig. 4 the fuzzy partition of input signals and the output are presented. As one can see the partition for inputs is 4 and for output signal 9.

3. COEVOLUTIONARY SYSTEM.

The coevolutionary system is presented on Fig. 5. Each subsystems with implemeted evolutionary strategy is searching the decoposed system. The fitness function is playing the cooperation role between each subsystem. It keeps the information about the best results found by other subprograms. In the Rule Base the best rules from all subsystems are collected. The new rule can be added in the ranking way.



Rys. 5 Coevolutionary system

3.1. Evolutionary strategy.

The coevolutionary system is build from several subsystem. Each subprogram is based on evolutionary strategy. The number of these subsystems forms the algorithm parameter. As the local evoultionary strategy the EVaPS (Evolutionary algorithm with Varying Population Size [3]) algorithm is used.

It is the version differing from the standard evolutionary strategy with the parameter of the chromosome life time. This parameter is calculated for each new chromosome and informs for how many generations it will be kept in the population. The life time depends on the fitness value. The better is fitness the longer will be the life time. During each generation the secondary population is generated. Its size is proportional to the actual size of the primary population. The linear coefficient is constant (the algorithm parameter). Each chromosome from the primary population can be chosen to the secondary population with constant probability, independend from its fitness. The population is obtained through the application of the crossover and mutation operators on selected chromosomes.

As the genetic operators the uniform crossover is used, in which for two chosen chromosomes (with crossover probability p_c) it is decided for each allea whether it is exchanged or not. The exchange probability is set to *0.5*. While the mutation operator is used for each chromosome in the secondary population. During the mutation for each allea it is decided if its value will be changed. The mutation probability p_m is the algorithm parameter.

If the secondary population is finally formed the fitness value for each chromosome is calculated. The life time parameter is obtained on the basis of the fitness. There are many strategies for the life time calculation, for example:

a) proportional strategy;

b) linear strategy;

c) bilinear strategy.

In the considered scheme the bilinear strategy was applied:

$$life\_time = \begin{cases} MinLT + (AvgLT - MinLT)\dfrac{MaxFit - fitness}{MaxFit - AvgFit} \\ \qquad if \quad fitness > AvgFit \\ AvgLT + (MaxLT - AvgLT)\dfrac{AvgFit - fitness}{AvgFit - MinFit} \\ \qquad if \quad fitness \leq AvgFit \end{cases}$$

where: *AvgFit, MinFit, MaxFit* represents the average, minimum and maximum fitness value; *fitness* is the fitnees function value of the considered chromosome. The minimum and

maximum value of the life time parameter are *MinLT* and *MaxLT*, respectively, while $\eta = \frac{1}{2}(MaxLT - MinLT)$.

There is used a method based on the "*Michigan*" approach as the Rule Base coding strategy. It means that each chromosome represents only one fuzzy rule. The length of the chromosome is equal to the number of premises and consequences. Each allea is represented by the integer value playing the role of the fuzzy set label of the considered signal. For instance the chromosome:

2	1	7

represents the fuzzy rule:

IF x_1 **is** $T_2^{x_1}$ **AND** x_2 **is** $T_1^{x_2}$ **THEN** y **is** T_7^{y}

3.2. The form of cooperation (coevolution).

The evolutionary strategy works in the iterative mode. Its algorithm procedure is sketched below.

```
procedure C_AE
 begin
  t=0;
  init_Rule_Base();
  init_AE();
  while( not stop_criterium)
   begin
    t=t+1;
    for_each_EA_make_new_generation;
    form_new_Rule_Base();
   end;
 end;
```

Each AE algorithm can be calculated parallely. The exchange of information between each subsystem is performed during the fitness function calcultion basing on the Rule Base selected in the last generation and during the forming of the new Rule Base.

3.3. Rule Base.

The Rule Base is the set of the best rules found and selected during the algorithm action. Its size is the algorithm parameter and it represents the maximum number of fuzzy rules which can be selected. It can also happen that the final Rule Base size will be smaller. This fact comes from the way of the fuzzy rules selection based on the ranking mechanism.

3.4. Initilization of the Rule Base and EA.

At the biginning the Rule Base is initilized randomly and the fuzzy rules are not selected in any way. Afterwards each AE subsystem is initilized (initial popultions, crossover and mutation parameters, initial fitness and life_time calculation). During the fitness calcultion the last rule in the Rule Base is treated as the worst and it is exchaned with the actual rule. Afterwards the best N (Rule Base size) fuzzy rules are chosen from all subsystems, but without the repetition of similar rules. The method of Rule Base actualization is sketched below.

```
_end:= false;
repeat
 calculate_actual_Rule_Base_fitness();
 calculate_each_rule_fitness(calculate_actual_Rule_Base_fitness_without_this_rule
                              and compare_with_the_full_Rule_Base_performance);
  if (exists_rule_decreasing_performance) then
    delete_rule;
 else
   _end:= true;
until(_end);
```

Now each EA subsystem calculates new generation. During the fitness calculation the worst rule from the Rule Base is exchanged with the actual chromosome.

3.5. Rule Base actualization

Rule Base actualization is based on the exchange of the several worst rules (the algorithm parameter) with the best new chromosomes from the subsystems (the existing rule cannot be addeed). Afterwards the Rule Base actualization is performed. If the new rules set is better than the former one the new Rule Base is kept as the actual one.

As the stop criterium two parameters are considered: the maximum number of generations and the patiency of the algorithm, it means the number of generations without the enhancement of the Rule base performance index.

3.6. Fitness function.

During fitness calculation for the chromosome the Rule Base is applied to the fuzzy reasoning system with the worst rule exchanged with the actual chromosome. As the reasonong scheme the Mamdani operation rule is used with the centre of area defuzzyfication strategy. The performance measure comes from the average square error between the identification data output and the model output formed by the fuzzy rule based sysytem (2).

$$\frac{1}{N}\sum_{k=1}^{N}(y_k - \hat{y}_k)^2 \tag{2}$$

where:

N - number of identification data;

y_k - the real output;

$\hat{y}_k$ - the model output.

3.7. The final fuzzy rules tuning.

The final tuning is obtained by the simple nongradient optimization algorithm. In its simple version it can be found in [2]. In presented application it have been modified depending on the algorithm specification. All the fuzzy sets (input and output) are in the triangular shapes (Fig. 3 and 4). Each triangle can be characterised by the set of three points: p_1, p_2, p_3.

Starting from the first fuzzy set for the first input in the Data Base the first p_1 point is selected. This point is shifted right and left by δ and new points $p_1'=p_1-\delta$ and $p_1''=p_1+\delta$ are obtained ($p_1'' \leq p_2$). The fitness is calculated for both new fuzzy sets shapes and the best of the three is chosen as the actual p_1 point. And now the next p_i points are calculated, but one should remember that it must be kept the oder $p_1 \leq p_2$ and $p_2 \leq p_3$. This algorithm is repeated for all fuzzy sets in the Data Base, affecting different rules.

4. RESULTS.

Two data sets were generated before the experiment: one as the training (identification) set and second as the testing set. The parameters of each evolutionary subsystem were the same:

- maximum number of generations - *200*;
- patiency - *40*;
- minimum life_time - *1*;
- maximum life_time - *10*;
- crossover probability - *0.6*;
- mutation probability - *0.4*.

4.1. The influence of the coevolution parameters.

In this section the results considering the influance of the coevolution parameters on the overall system performance will be presented. It have been found that the most important parameters are: the Rule base size, the number of exchaged rules (exchange rate) and the number of AE subsystems.

4.1.1. Coevolution strategy with 10 rules.

With this Rule Base size the best results were obtained for the algorithm with two subsystems and the exchange rate of *20%*. The performance index for the training data was *0.1124* and for the testing data *0.1452*. But the efficiency (the number of fitness function calculations) was rather poor, about *1200*.

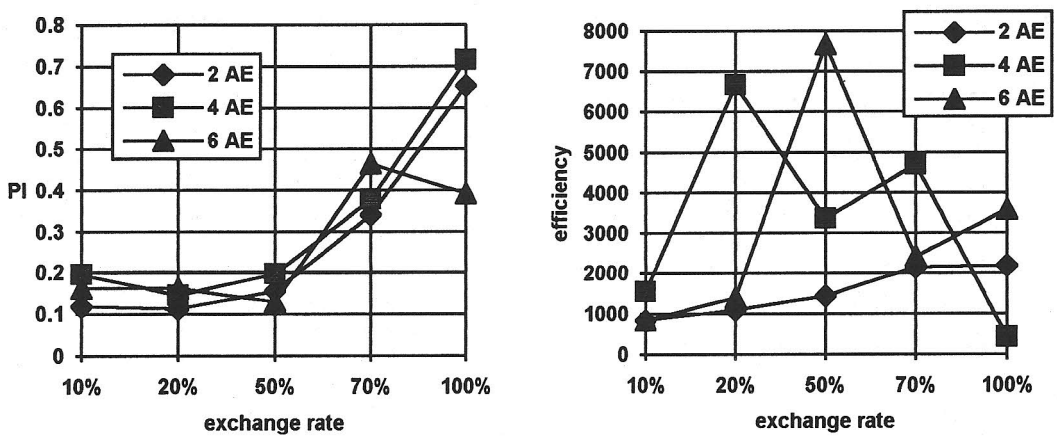


Fig. 5 Coevolutionary algorithm with 10 rules.

4.1.2. Coevolution strategy with 20 rules.

With this Rule Base size the best results were obtained for the algorithm with four subsystems and the exchange rate of *25%*. The performance index for the training data was *0.0366* and for the testing data *0.0696*. The efficiency was about *6000* but comparing to the much better performance index it is rather good result.

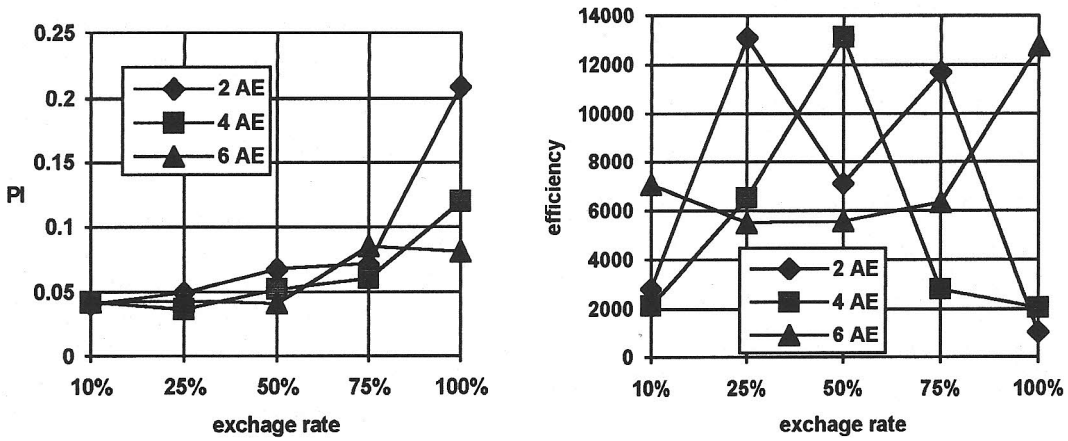


Fig. 6 Coevolutionary algorithm with 20 rules.

4.1.3. Coevolution strategy with 30 rules.

With this Rule Base size the best results were obtained for the algorithm with four subsystems and the exchange rate of *50%*. The system has 26 rules. The performance index for the training data was *0.0313* and for the testing data *0.0708*. The efficiency was about *6000* but comparing to the small or rather none performance index enhancement it is not enough good result.

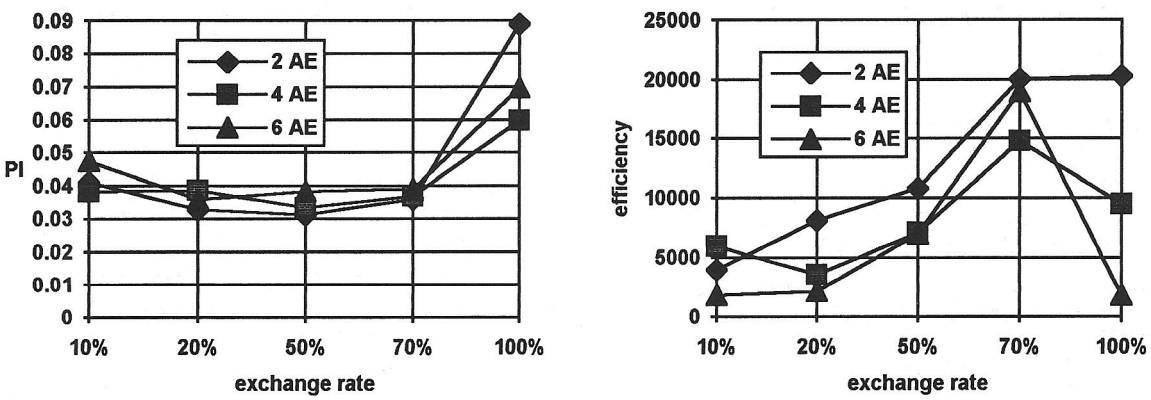


Fig. 7 Coevolutionary algorithm with 30 rules.

4.2. The final tuning.

After the application of the final fuzzy sets tuning algorithm to the best result obtained before (four AE substems and *25%* exchange rate) the performance index decreased to the value of *0.0073* on training data (former value *0.0366*) and on the testing date it reached *0.0286* (former value *0.0708*). During the tuning procedure the training set has been exchanged with the testing one. The nonlinear model function characteristics (before and after final tuning) are presented on Fig. 8.

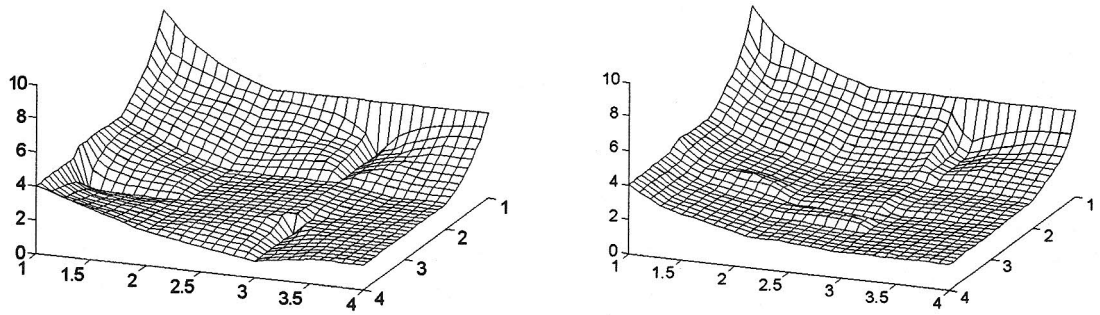


Fig. 8 The function characteristics for model with 4 subsystems and 25% exchange rate before (left) and after tuning (right).

5. CONCLUSIONS.

The coevolutionary system has many parameters, which should be set depending on the considered problem or the expert prefernces. On the basis of the abovee results there are some conclusions.

- The first problem lies in the correct selection of the number and the shape of fuzzy sets. This problem is solved mainly on the basis of the expert experience.
- The second parameter affecting coevolution and the model complexity is the Rule Base size. This value mainly depends on the expert preferences or is problem dependend.
- One can conclude that the exchange rate should be approximately between *20-50%.* to guarantee the best efficiency and performance.
- To obtain higher performance with bigger Rule Base (small PI and better efficiency) it should be chosen larger number of EA subsystems and bigger exchange rate.
- Too small number of AE subsystems gives worse efficiency and decrease in performance. In this case the algorithm has wirse searching possibilities.
- If the number of AE subsystems is too big there are many repeting ruleswhich gives worse convergence properties. Also the final performance is not enhanced.

6. REFERENCES.

[1] Zadeh L. Fuzzy Sets and Systems, Proc. Symp. Syst. Theory, Polytech. Inst. Brooklyn, pp.29-37, 1965.

[2] Sugeno M., Yasukawa T. „A Fuzzy Logic based Approach to the Qualitative Modelling" IEEE Trans. on Fuzzy Systems, Vol.1, No.1, pp.1-31, February 1993.

[3] Arabas J. „Evolutionary algorithms with varying population size and varying crossover range." PhD. thesis, Warszawa 1995 (in polish).

[4] De Jong K. An Introduction To Evolutionary Computation and Its Applications. in Fuzzy Logik, Theorie und Praxis, edit. B. Reusch, Springer-Verlag Berlin, Heidelberg, 1994.