



Influence of the Migration Process on the Learning Performances of Fuzzy Knowledge Bases

Akrout, Khaled; Baron, Luc; Balazinski, Marek; Achiche, Sofiane

Published in:

Annual Meeting of the North American Fuzzy Information Processing Society Conference

Publication date:
2007

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):

Akrout, K., Baron, L., Balazinski, M., & Achiche, S. (2007). Influence of the Migration Process on the Learning Performances of Fuzzy Knowledge Bases. In Annual Meeting of the North American Fuzzy Information Processing Society Conference (pp. 478-483). San Diego, USA: IEEE.

DTU Library Technical Information Center of Denmark

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Influence of the Migration Process on the Learning Performances of Fuzzy Knowledge Bases

Khaled Akrou, Luc Baron, Marek Balazinski and Sofiane Achiche

Department of Mechanical Engineering

École Polytechnique of Montréal

P.O. box 6079, succ. CV

Montréal, Québec, Canada, H3C 3A7

E-mail: Khaled.Akrou@..., Luc.Baron@polymtl.ca

E-mail: Marek.Balazinski@..., Sofiane.Achiche@polymtl.ca

Abstract— This paper presents the influence of the process of migration between populations in GENO-FLOU, which is an environment of learning of fuzzy knowledge bases by genetic algorithms. Initially the algorithm did not use the process of migration. For the learning, the algorithm uses a hybrid coding, binary for the base of rules and real for the data base. This hybrid coding used with a set of specialized operators of reproduction proven to be an effective environment of learning. Simulations were made in this environment by adding a process of migration. While varying the number of populations, the number of generations and the rate of migration or simply the migration of the best elements, on various types of problems. In general, simulations show a significant improvement of the results obtained with migration. The variation of these parameters makes it possible to conclude on the dominating importance of the number of migrant generations.

I. INTRODUCTION

Nowadays fuzzy logic is increasingly used in decision-aided systems because it offers several advantages over other traditional decision-making techniques. The fuzzy decision support systems (FDSS) can easily deal with incomplete or/and imprecise knowledge applied to either linear or nonlinear problems. In general, FDSS requires a fuzzy knowledge base (FKB) in order to support the decision-making process of end-users. The FKB can be either manually created by a human expert or automatically learned from a set of experimental data. Since the manual creation is very long and tedious, many research works have been conducted toward the automatic learning of fuzzy rules alone, e.g., [1], [2], or the fuzzy sets alone, e.g., [3], [4]. More recently, other research works proposed to perform a simultaneous learning of both fuzzy rules and sets, including their quantity and distribution over each premises and conclusion, e.g., [5], [6]. This learning process of FKBs is performed with a genetic algorithm (GA) along the Pittsburgh approach [7], [8].

In this paper, we use the Geno-Flou, an integrated learning environment, which allow the automatic learning of FKBs from different preprocessed sets of experimental data. GENO-FLOU is a strategic research project sponsored by the Natural Sciences and Engineering Research Council (NSERC) of Canada. As shown in Fig. 1, the architecture of GENO-FLOU is made of two main parts: the *data preparation and mining*

modules on the left and the *knowledge discovery modules* on the right. The latter includes a Mamdani's type *fuzzy system module* (described in section III) and a *genetic learning module* (described in section IV) that follows the Pittsburgh approach. The left part includes several data preprocessing such as preparation, formatting, analysis, clustering and feature selection modules. In this paper, we investigated the introduction of a migration process within the genetic learning module of Geno-Flou.

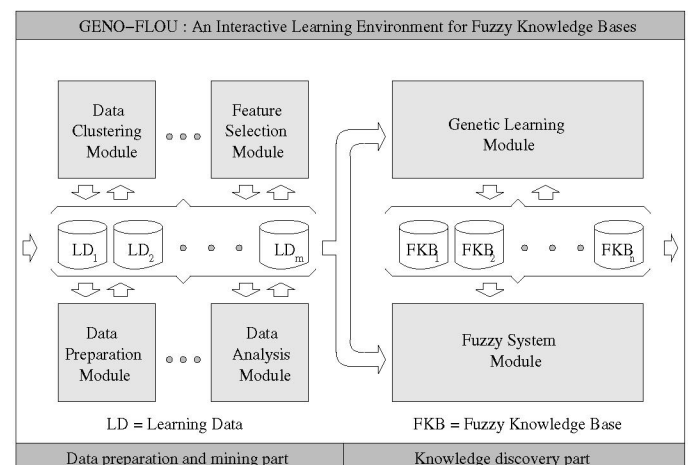


Fig. 1. Architecture of GENO-FLOU.

II. DATA PREPARATION AND MINING MODULES

As shown in Fig. 1, the data preparation module is aimed to account for different format of learning data sources, while the analysis module can be used to perform statistical and multi-regression analysis. The clustering module is aimed to get ride of superfluous repetition, and hence, result in a data reduction. Two main different clustering algorithms have been implemented [16]: AGNES (for Agglomerate NESTing) algorithm and the K-mean algorithm. Finally, the feature selection module is aimed to determine the most appropriate input variable that observe the desired output. This selection can be manually done or automatically done using another

GA that performs the supervision of the input learning data together with the FKB learning along the Pittsburgh approach.

III. FUZZY SYSTEM MODULE

A rule-based approach to the decision making using fuzzy logic techniques may consider imprecise vague language as a set of rules linking a finite number of conclusions. The knowledge base of such systems consists of two components: a linguistic terms base and a fuzzy rules base. The former is divided into two parts: the fuzzy premises (or inputs) and the fuzzy conclusions (or outputs). In general, the fuzzy systems are either of type: SISO (Single Input and Single Output); MISO (Multiple Inputs and Single Output); and finally MIMO (Multiple Inputs and Multiple Outputs). For the initial version of GENO-FLOU, we limit ourself to fuzzy systems of N multiple inputs and one single output (i.e., MISO). Moreover, for the sake of simplicity, we consider general non-symmetric triangular fuzzy sets on the premises and sharp-symmetric triangular fuzzy sets on the conclusion. The representation of such imprecise knowledge by means of fuzzy linguistic terms makes it possible to carry out quantitative processing in the course of inference that is used for handling uncertain (imprecise) knowledge. This is often called approximate reasoning [9]. Such knowledge can be collected and delivered by a human expert (e.g. decision-maker, designer, process planer, machine operator). This knowledge, expressed by $(k = 1, 2, \dots, K)$ finite heuristic fuzzy rules of the type MISO, may be written in the form:

$$R_{MISO}^k : \text{if } x_1 \text{ is } X_1^k \text{ and } \dots x_N \text{ is } X_N^k \text{ then } y \text{ is } Y^k, \quad (1)$$

where $\{X_i^k\}_{i=1}^N$ denote values of linguistic variables $\{x_i\}_{i=1}^N$ (conditions) defined in the following universe of discourse $\{X_i\}_{i=1}^N$; and Y^k stands for the value of the independent linguistic variable y (conclusion) in the universe of discourse Y . The global relation aggregating all rules from $k = 1$ to K is given as

$$R = \text{also}_{k=1}^K (R_{MISO}^k). \quad (2)$$

where the sentence connective *also* denotes any t- or s-norm (e.g., *min* (\wedge) or *max* (\vee) operators) or averages. For a given set of fuzzy inputs $\{X_i^t\}_{i=1}^N$ (or observations), the fuzzy output Y^t (or conclusion) may be expressed symbolically as:

$$Y^t = (X_1^t, X_2^t, \dots, X_N^t) \circ R, \quad (3)$$

where \circ denotes a compositional rule of inference (CRI), e.g., the *sup*- \wedge or *sup*-*prod* (*sup*- $*$). Alternatively, the CRI of eq.(3) is easily computed as

$$Y^t = X_N^t \circ \dots \circ (X_2^t \circ (X_1^t \circ R)). \quad (4)$$

In this fuzzy system module, there are four variants of CRI: the sentence connective *also* can be either \vee or *sum* (\sum); the compositional operator is the *supremum* (*sup*) of either \wedge or $*$, denoted *sup* \wedge and *sup* $*$; while the sentence connective *and* and the fuzzy relation are always identical to the second part of the latter. For the sake of brevity, all four variants of

CRI—i.e.: \vee -*sup* \wedge - \wedge - \wedge ; \vee -*sup* $*$ - $*$ - $*$; \sum -*sup* \wedge - \wedge - \wedge ; and \sum -*sup* $*$ - $*$ - $*$ —are expressed as

$$Y^t = \left\{ \sum_{k=1}^K \right\} \sup_{\{x_i \in X_i\}_{i=1}^N} \quad (5)$$

$$*_t(*_t(X_N^t, \dots, X_1^t), *_t(X_1^k, \dots, X_N^k, Y^k)),$$

where $*_t(\cdot)$ denotes the t-norm of (\cdot) defined as either \wedge or $*$. These variants of CRI mechanisms allow us to obtain different conclusions represented as the membership function Y^t . In this fuzzy system module, there are three defuzzification methods: the center of gravity (COG); the mean of maxima (MOM); and the height method (HM). All the results presented in this paper are obtained with the \sum -*sup* $*$ - $*$ - $*$ CRI and COG as defuzzification. This initial implementation of the fuzzy system module is based on the FDSS Fuzzy-Flou software previously developed at École Polytechnique (Canada) and the Technical University of Silesia in Gliwice (Poland). As shown in Fig. 2, the graphic interface of this software is made of two parts. The left window shows the rules base system, while the right window shows the fuzzy membership functions distributed on the two premises and the conclusion of this simple FKB.

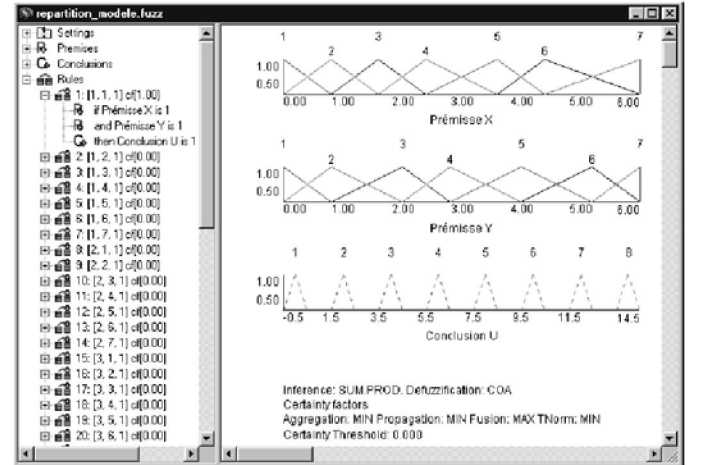


Fig. 2. Graphic interface of the FDSS Fuzzy-Flou software.

IV. GENETIC LEARNING MODULE

The GAs are powerful stochastic optimization methods presented first in the early 70's by John Holland [10]. These methods are based on the analogy of the mechanics of natural genetics, and imitates the Darwinian survival-of-the-fittest approach. A skilled FKB must satisfy two contradictory objectives: 1) the FKB must accurately reproduce the set of learning data, while interpolating or extrapolating fair conclusions for other data sets; 2) the FKB must be of low complexity such as measured by the number of fuzzy rules. This *minimalist objective* for the learning of FKBs allow a higher level of generalization, while obtaining FKBs of size still manageable by a human being.

As shown in Fig. 3, the genetic learning module follows the Pittsburgh approach. Each individual of the population is a potential FKB. The method uses four basic operations: *reproduction*; *mutation*; *evaluation* and *natural selection*, in order to iteratively improve the population of FKBs at each generation toward multiple optima simultaneously.

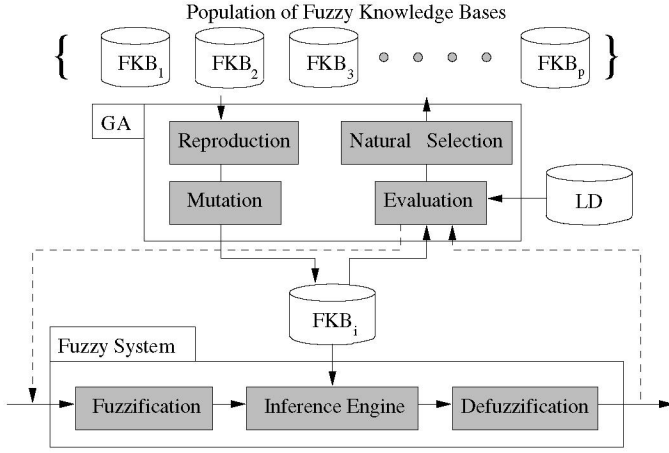


Fig. 3. Fuzzy system learning based on Pittsburgh approach.

A. FKB Coding

For the sake of implementation simplicity, most of the research works with GAs have been conducted with Binary Coded Genetic Algorithms (BCGA), thus resulting in poor accuracies of results. Many optimization problems require real number representation rather than binary, and hence, a Real Coded Genetic Algorithms (RCGA) is more appropriate. The representation of an FKB requires the coding of the following information:

- 1) the number of fuzzy sets and their locations as expressed by the location of the summit along all premises and along the conclusion;
- 2) the fuzzy rules connecting one fuzzy set of each premise to one fuzzy set on the conclusion.

Apparently, the first item has to deal with a list of real numbers (the location of the fuzzy set summit), while the second item is a list of integers (pointers to fuzzy sets). Consequently, an FKB must use a combination of both RCGA and BCGA [11], and hence, we call this combination a Hybrid Coded Genetic Algorithm (HCGA). We have performed a comparison study between HCGA and BCGA in [12]. In this context, the *genotype* of an FKB, namely G , is defined as

$$G \equiv \{ G_{sets}, G_{rules} \}, \quad (6)$$

where G_{sets} and G_{rules} are respectively the real coding of fuzzy sets and the binary coding of fuzzy rules. The latter G_{sets} has many sublists as the number of premises N together with the conclusion, i.e.,

$$G_{sets} \equiv \{ \underbrace{G_{X_1}}_{premise_1}, \dots, \underbrace{G_{X_i}}_{premise_i}, \dots, \underbrace{G_{X_N}}_{premise_N}, \underbrace{G_{X_C}}_{conclusion} \}, \quad (7)$$

where each G_{X_i} is given as

$$G_{X_i} \equiv \{ \underbrace{x_1}_{summit_1}, \underbrace{x_2}_{summit_2}, \dots, \underbrace{x_{K_i}}_{summit_{K_i}} \}, \quad (8)$$

with K_i being the number of fuzzy sets on the premise i (or the conclusion). The two fuzzy sets located at the extreme limits are not included in the coding since they are always there. It is noteworthy that K_i can be different for each premise i and can also be different, for the i , from an FKB to another FKB. As a result, the genotype G_{sets} has a variable length.

A fuzzy rule, namely r_i , is coded as an ordered lists of integer numbers pointing from one fuzzy set on each premise toward one fuzzy set of conclusion. Therefore, the genotype of the fuzzy rules is defined as

$$G_{rules} \equiv \{ \underbrace{r_1}_{rule_1}, \underbrace{r_2}_{rule_2}, \dots, \underbrace{r_K}_{rule_K} \}, \quad (9)$$

where K is the number of active fuzzy rules. It is noteworthy that two FKBs may have different values of K , and thus, different numbers of fuzzy rules. As a result, the genotype G_{rules} has also variable length.

B. FKB Reproduction

As shown in Fig. 3, the reproduction is obtained from a *multi-crossover mechanism* which apply three crossover operators on different parts of the genotype G .

First, the crossover on G_{sets} is implemented with the *blended crossover* α (BLX- α) [13], where α controls the exploitation/exploration level, knowing that:

- in the exploitation zone, the offspring inherits behaviors close to those of his parents;
- in the exploration zone, the offspring is a result of an exploration, therefore his attributes will be distant from his parent's average.

In order to avoid any bias in either direction (exploitation or exploration) the value of α is set to 0.5, which provides offsprings in the zone named the relaxed exploitation zone, as shown in Fig. 4. This multi-combinative strategy may overcome some aspects of the premature convergence encountered in the learning FKBs with the Pittsburgh approach [14]. We have also studied the different scheduling exploration/exploitation strategies for the learning of FKBs [15], which can be summarized as follows. The *exploration at the early stage, relaxing in the middle stage and exploitation at the end of the evolution* is the most promising strategy.

Second, the crossover on G_{rules} is implemented by a *simple crossover*. The operation is performed by inverting the end part of G_{sets} of the parents at a randomly selected crossover site. Third, the fuzzy set reducer is a mechanism aiming to increase the simplicity level of FKBs by randomly selecting a fuzzy set on a premise and erasing it together with the corresponding fuzzy rules. This mechanism allows one to try more simple FKB solutions (with less complexity).

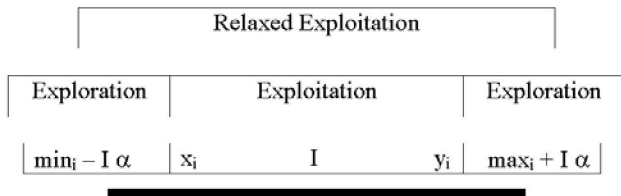


Fig. 4. Blended crossover α (BLX- α).

C. FKB Mutation

As shown in Fig. 3, the random uniform mutation is implemented by toggling one bit of an arbitrary FKB at a randomly selected mutation site.

D. FKB Natural Selection

The natural selection is performed on the population by keeping the most promising FKBs based on their fitness. The first generation starts with p randomly generated FKBs. At each generation, the most promising FKBs have more probabilities to reproduce themselves than others with lower fitness. This *elitist strategy* is crucial in GA, since it provides a direction of enhancement. The multi-crossover mechanism allows to double the size of the population at each generation, and hence, select only the first p FKBs that are kept for the future generations in order to keep the population size constant.

E. Migration between Populations

Three types of migration mechanisms between populations have been implemented, while keeping the population size constant.

1) Random Migration : As shown in Fig. 5, it consists to extract at each generation a percentage of randomly selected FKBs from each population and then redistributed them to other populations, while keeping the population size constant.

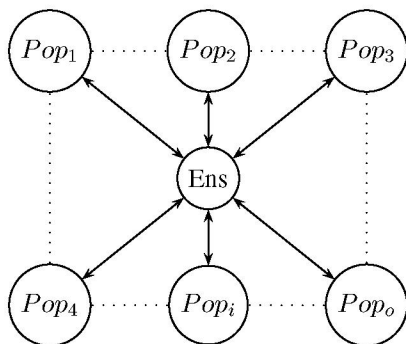


Fig. 5. Random migration between populations

2) Circular Migration : As shown in Fig. 6, the populations are placed into a circular list. At each generation a percentage of randomly selected FKBs are extracted from each population and transferred in the next population of the list.

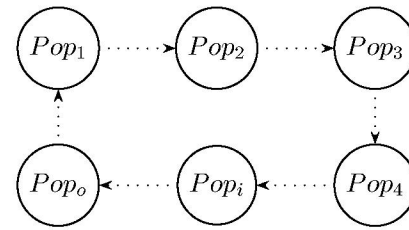


Fig. 6. Circular migration between populations

3) Single Migration : The single migration is similar to circular migration, but we extract at each generation only the best FKB from each population and then transfer it into the next population of the list.

V. LEARNING UNDER MIGRATION

In order to investigate the influences of the different migration strategies over the learning speed and fitness, we have conducted several learning from the same dataset. The population size is set to 100 FKBs. In all cases, the initial FKBs are randomly generated.

Figure 7 shows the fitness evolution of the best FKB among the 16 populations with 0%, 5% and 10% of random migration rate. Apparently, the random migration slightly accelerates the learning speed and fitness. However, after 100 generations the random migration mechanism does not help anymore the learning process.

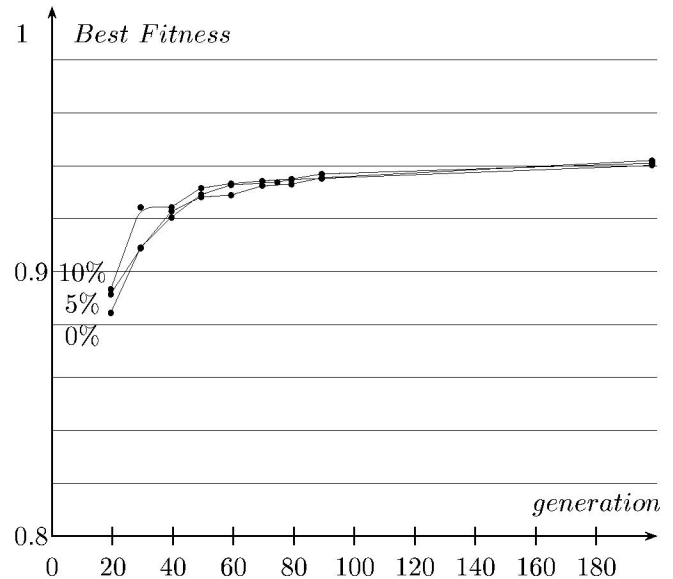


Fig. 7. Best fitness in terms of generation for 0%, 5% and 10% of random migration rate

As expected, the random migration produces the best learning performances, while the circular migration performs well and the single migration produces the same results as having no migration mechanism at all. This difference between the migration mechanisms appears only for a low number of generations.

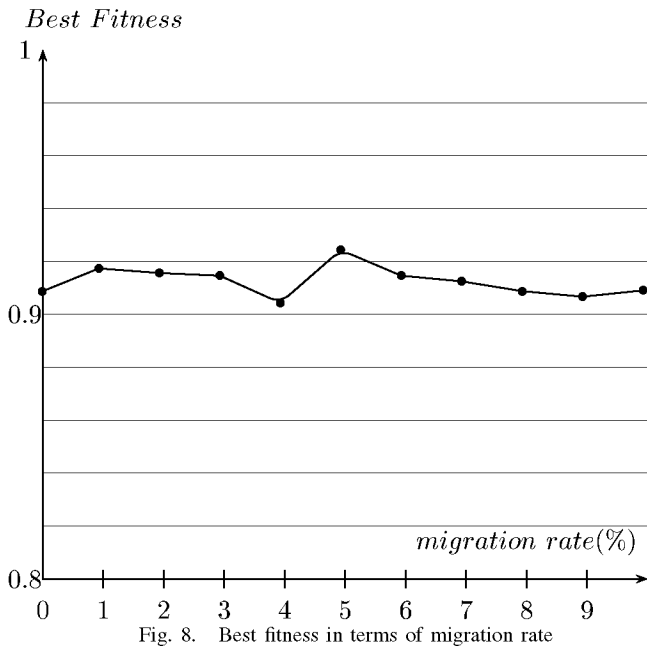


Fig. 8. Best fitness in terms of migration rate

Figure 8 shows the fitness evolution of the best FKB among the 16 populations at the 16th generation. Apparently, the best result is obtained at 5% of random migration rate.

Figure 9 shows the fitness evolution of the best FKB through the generation for different number of population from 2 to 16 (from the lower to the upper curves) with a random migration rate of 5%. At the 30th generation, the migration mechanism allows to increase the fitness by 2% to 4.5%. Apparently, the best result is obtained for 16 populations.

VI. CONCLUSIONS

In this paper, we have investigated the use of different migration mechanisms in a genetic-based learning system of FKBs from dataset. Three migration mechanisms have been implemented in Geno-Flou, an interactive teaching environment for FKBs. The knowledge discovery part of the system follows the Pittsburgh approach, while the data preparation and mining part is interactive. The learning uses a hybrid coded genetic Algorithm to learn the fuzzy sets and rules information. Its uses a multi-crossover mechanism together with the different migration mechanisms to accelerate the learning process. The migration mechanisms do not accelerate the learning process for high number of generation (say more than 100), while it enhanced the fitness by 2% to 4.5% at 30 generations. The random migration mechanism is most useful one, when used with low migration rate such as 5%. Consequently, we recommend the use of random migration mechanism, when the number of generation is low (say 30) such as in online applications.

ACKNOWLEDGMENT

The authors want to acknowledge the financial support from the Natural Sciences and Engineering Research Council

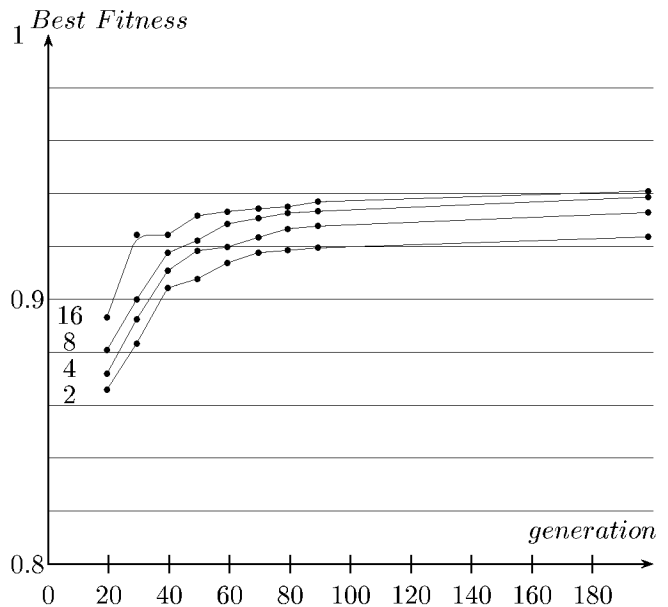


Fig. 9. Best fitness in terms of generation for 2, 4, 8 and 16 populations

of Canada under grants RGPIN-203618, RGPIN-105518 and STPGP-269579.

REFERENCES

- [1] G. Castellano, G. Attolico and A. Distanto, "Automatic generation of fuzzy rules for reactive robot controllers", *Robotics Autonomous Systems*, 22, pp. 133–149, 1997.
- [2] G. Casillas, O. Cordón and F. Herrera, "A methodology to improve adhoc data-driven linguistic rule learning methods by inducing cooperation among rules", *Technical report #DECSAI-000101*, University of Granada, Spain, 2000.
- [3] H. Nomura, I. Hayashi and N. Wakami, "A self-tuning method of fuzzy reasoning by genetic algorithm" in *Proceedings of the 7th International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, pp. 236–245, 1992.
- [4] J.R. Valesco, S. Lopez and L. Magdalena, "Genetic fuzzy clustering for the definition of fuzzy sets" in *Proceedings of the 6th IEEE International Conference on Fuzzy Systems*, FUZZ IEEE, III, pp. 1665–1670, 1997.
- [5] L. Baron, S. Achiche and M. Balazinski, "Fuzzy decision system knowledge base generation using a genetic algorithm", *International Journal of Approximate Reasoning*, vol. 28, no. 2–3, pp. 125–148, 2001.
- [6] S. Achiche, M. Balazinski and L. Baron, "Tool wear monitoring using genetically-generated fuzzy knowledge bases", *Engineering Applications of Artificial Intelligence*, vol. 15, no. 3–4, pp. 303–314, 2002.
- [7] S.F. Smith, "A learning system based on genetic adaptive algorithms", *Doctoral dissertation*, Department of computer science, University of Pittsburgh, USA, 1980.
- [8] S.F. Smith, "Flexible learning of problem solving heuristics through adaptive search" in *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, Morgan Kaufmann Publishers, pp. 422–425, 1983.
- [9] L.A. Zadeh, "Outline of new approach to the analysis of complex systems and decisions processes", *IEEE Transactions of Systems, Man and Cybernetics*, no. 3, pp. 28–44, 1973.
- [10] J. Holland, "Genetic algorithms and the optimal allocation of trials", *SIAM Journal on Computing*, vol. 2, no. 2, pp. 88–105, 1973.
- [11] O. Cordón, F. Herrera, F. Hoffmann, and L. Magdalena, *Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases*. Singapore: World Scientific, 2001.

- [12] S. Achiche, L. Baron and M. Balazinski, "Real/binary-like coded versus binary coded genetic algorithms to automatically generate fuzzy knowledge bases: a comparative study", *Engineering Applications of Artificial Intelligence*, vol. 17, no. 4, pp. 313–325, 2004.
- [13] F. Herrera and M. Lazano, "Gradual distributed real-coded genetic algorithms", *IEEE Transactions on Evolutionary Computation*, vol. 4, pp. 43–63, 2000.
- [14] S. Achiche, M. Balazinski and L. Baron, "Multi-Combinative Strategy to Avoid Premature Convergence in Genetically-Generated Fuzzy Knowledge Bases", in *Proceeding of the IEEE International Conference on Fuzzy Systems*, Banff, Canada, june 2004.
- [15] S. Achiche, L. Baron and M. Balazinski, "Scheduling Exploration/Exploitation Levels in Genetically-Generated Fuzzy Knowledge Bases", *Jouranal of Theoretical and Applied Mechanics*, vol. 42, no. 3, pp. 417–444, 2004.
- [16] A. Przybylo, S. Achiche, M. Balazinski and L. Baron, "Enghancing Fuzzy Learning with Data Mining Techniques", in *Proceeding of the AI-MECH*, Gliwice, Poland, nov. 2004.
- [17] S. Achiche, L. Baron, M. Balazinski and M. Benaoudia, "Chips Image Processing to Predict Pulp Brightness using Fuzzy Logic Techniques", to appear in *Proceeding of the 2005 Annual Meeting of the Pulp and Paper Technical Association of Canada*, Montreal, Canada, february 2005.