Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 137–140

ISBN 978-83-60810-51-4

The Fuzzy Genetic System for Multiobjective Optimization

Krzysztof Pytel
Faculty of Physics and Applied Informatics
University of Lodz,
Lodz, Poland
Email: kpytel@uni.lodz.pl

***Abstract*—The article presents the idea of a hybrid system for multiobjective optimization. The system consists of the genetic algorithm and the fuzzy logic driver. The genetic algorithm realizes the process of multiobjective optimization. The fuzzy logic driver uses data aggregated by the genetic algorithm and controls the process of evolution by modifying the probability of selection of individuals to the parental pool. The controlling of the evolution process makes it possible to choose the preferred area with pareto-optimal solution. In experiments we investigated the ability of the proposed system to search solutions in a given area of the search space. We compared the results of the standard genetic algorithm and the proposed system. The experiments showed that the proposed system is able to control the process of evolution toward pareto-optimal solutions in the given area of searching.**

I. Introduction

IN MANY practical problems, it's often expected that several indicators will achieve an optimal value at the same time, which is called the multi-objective optimization problem (MOP) [2][6]. These multiple objectives, often conflicting with each other, can accept the maximum or minimum in other points of the search space. The multi-objective optimization problem can be stated as follows:

$$\begin{cases} maximize F(x) = [f_1(x) f_2(x) f_3(x) ... f_m(x)] \\ \text{subject to: } g_j(x) \leq 0 \text{ for } j = 1, 2, ..., k \\ x \in S \end{cases} \tag{1}$$

where

$$x = [x_1 x_2 x_3 ... x_n] \in \Re, n \in N, \tag{2}$$

is an n-dimensional vector of decision variables,

$$F(x) = [f_1(x) f_2(x) f_3(x) ... f_m(x)], \tag{3}$$

are objective functions, and
$g_i(x)$ are constrains.

Let us choose an optimization problem, with *m* objectives, which are, without loss of generality, all to be maximized. The set of potential solutions can be parted to two subsets: dominated and not dominated.

Let us choose two solutions $x, y \in S$, x is said to be dominated by *y*, if $f_i(y) \geq f_i(x)$ for all $i = 1, 2, ..., m$ and $f_j(y) > f_j(x)$ for at least one index *j*. A solution $x^* \in S$ is said to be pareto-optimal if there does not exist another solution *x*, such that x^* is dominated by *x*. $F(x^*)$ is then called a pareto-optimal objective vector. The set of all the pareto-optimal objective vectors is called the pareto-optimal front. A set of pareto-optimal solutions is usually found as a result of multiobjective optimization. The decision-maker can use his preferred method to choose a final solution from the pareto-optimal set. For the decision-maker it would be profitable if the process of searching pay special attention to the preferred area.

Genetic algorithms (GA) stand for a class of stochastic optimization methods that simulate the process of natural evolution [1][3][5]. In a genetic algorithm, a population of strings (called chromosomes), which encode candidate solutions (called individuals) to an optimization problem, evolves toward better solutions. They usually search for approximate solutions of composite optimization problems. A characteristic feature of genetic algorithms is that in the process of evolution they do not use specific knowledge for a given problem. They use only the fitness function assigned to all individuals.

The genetic algorithm consists of the following steps:

1) the choice of the first generation,
2) the estimation of an individuals' fitness,
3) the check of the stop condition,
4) the selection of individuals to the parents' pool,
5) the creation of a new generation with the use of operators of crossing and mutation,
6) the printing of the best solution.

The source code of standard genetic algorithm was published in [3]. We used this code as a base genetic algorithm in our experiments.

The probability of selection of an individual to the parents pool is a parameter determining the process of evolution in genetic algorithms. This parameter determines the abilities of an individual to act as a parent. By changing the probability of selection we can select individuals that can create offspring and transfer their own genetic material to the next population. Modifications must be made carefully, not to disturb the process of the natural selection. The knowledge of experts about the evolution can be used for these modifications. This knowledge has a descriptive character and is often subjective, so we use the fuzzy logic controller. In the proposed system, the knowledge is represented by the set of rules in the fuzzy logic driver.

II. Modification of the Probability of Selection to the Parents Pool

The probability of selection to the parents pool is a basic parameter determining the efficiency of the genetic algorithm. The likelihood that an individual will transfer its own genetic material to the next generation increases with the probability of selection. Well-adapted individuals are the most wanted ones in the parents' pool. However, weak individuals should be selected to the parents' pool too, in order to prevent violent loss of their genetic material and premature algorithm convergence. The enlargement of the probability of selection of individuals with preferred characteristics, can result in these individuals producing more descendants. The genetic variety of the population will be diminished, but the algorithm convergence will be enlarged.

For the realization of this strategy we suggest the introduction of the additional fuzzy logic driver. The FLC estimates each individual in the population. The goal of the FLC is evaluation of every individual and computation of the new probability of selection for each individual in the population. The FLC uses the knowledge of experts and the knowledge aggregated by the genetic algorithm. The FLC computes a new probability of selection, based on an individual's fitness function and fitness functions of all individuals in the current population. The probability of selection is enlarged for well-adopted individuals and is diminished for poor individuals. The FLC modifies the process of selection using the following rules:

- enlarging the probability of selection if the value of the chosen objective function for an individual is greater than the average value of this function in the current generation,
- not changing the probability of selection if the value of the chosen objective function for an individual is equal to the average value of this function in the current generation,
- dimindiminishing the probability of selection if the value of the chosen objective function for an individual is smaller than the average value of this function in the generation,
- enlarging the probability of selection if the value of the chosen objective function for an individual is near the optimal value,
- diminishing the probability of selection if the value of the chosen objective function for an individual is far-away from the optimal value.

We accepted the FLC with two inputs and one output. The FLC estimates the values of the inputs of all individuals and determines the value of the output. As inputs we have chosen:

- the difference between the chosen objective function of an individual and maximum values of this function in the current population (it shows whether the given solution is near the optimum - from the point of view of the chosen objective function)

$$\Delta f_i = f_i(mem) - max(f_i) \tag{4}$$

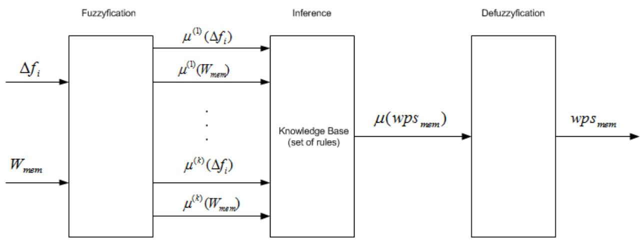

Fig. 1. The block scheme of the fuzzy logic controller

where:

Δf_i - the difference between the chosen objective function of individual mem and maximum values of this function in the current population,

$f_i(mem)$ - the chosen objective function of individual *mem*,

$maxf_i$ - maximum values of the chosen objective function in the current population.

- an individual's fitness (determines the difference between the chosen objective function of an individual and the average value of this function in the population)

$$W_{mem} = f_i(pop) - f_i(pop) \tag{5}$$

where:

W_{mem} - the fitness function of an individual *mem*,

$f_i(mem)$ - the value of the chosen objective function of an individual *mem*,

$f_i(pop)$ - the average value of this function in the current population.

The FLC uses the center of gravity [6] defuzzyfication method. As the result from the controller we accepted:

- the adaptation ratio of the probability of selection for individual wps_{mem}

Figure 1 shows the fuzzy logic controllers scheme. The modified probability of selection of individual k obeys the formula:

$$ps'(k) = ps(k) * wps_k \text{ for } k = 1, ...N, \tag{6}$$

where:

$ps'(k)$ - the modified probability of selection of individual k,

$ps(k)$ - the probability of selection of individual k,

wps_k - the adaptation ratio of the probability of selection calculated by the FLC for the individual k.

Figure 2 shows a block scheme of the modified genetic algorithm (the block of the fuzzy logic is noted with the grading). The construction of the fuzzy logic controller in details is considered in [4][5].

III. Computational Experiments

The goal of the experiment is the verification of the idea of fuzzy controlling of evolution in the modified genetic algorithm for multiobjective optimization. The LOTZ (Leading Ones, Trailing Zeroes) problem with size from 50 to 120 was

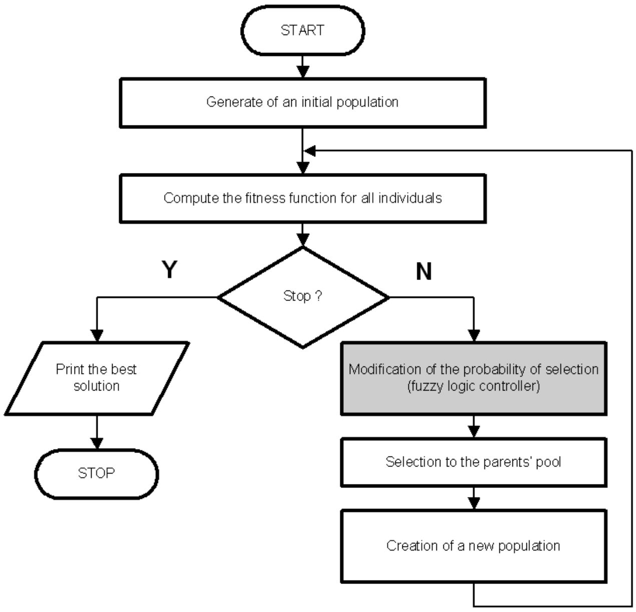

Fig. 2. The block scheme of the modified genetic algorithm

chosen as the test function. The algorithm searches for any of pareto-optimal solutions. The FLC estimates all individuals in all generations and modifies their probability of selection. The LOTZ can be stated as the maximization problem of two objectives.

$$LOTZ1(x) = \sum_{i=1}^{n} \prod_{j=1}^{i} x_j, \tag{7}$$

$$LOTZ2(x) = \sum_{i=1}^{n} \prod_{j=i}^{n} (1 - x_j), \tag{8}$$

$$LOTZ(x) = (LOTZ1(x), LOTZ2(x)), \tag{9}$$

where: $x = (x_1, x_2, ..., x_n) \in \{0, 1\}$.

There are two objective functions defined in the LOTZ problem. Function LOTZ1 maximalizes the number of ones, while function LOTZ2 maximalizes the number of zeros in a binary string of an individual. The function optimized is the sum of the functions LOTZ1 and LOTZ2. Both the functions, LOTZ1 and LOTZ2, have equal weight. The standard genetic algorithm usually finds solutions in which the number of ones is approximately equal to the number of zeros. In the proposed system we use an additional FLC, which modifies the probability of selection of individuals to the parent's pool. The FLC can direct the process of evolution toward the solutions with the greater number of zeros or ones.

The first population was generated randomly. All algorithms started at the same point in the search space. The algorithm's parameters used in the experiment:

- the genes of individuals are represented by binary numbers,
- the probability of crossover = 0,8,
- the probability of mutation = 0,15,

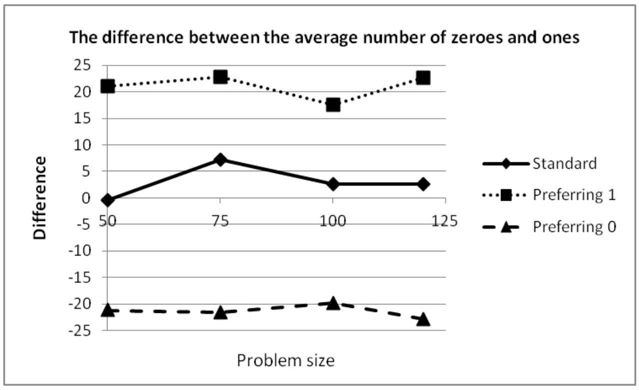

Fig. 3. The difference between the average number of zeroes and the average number of ones

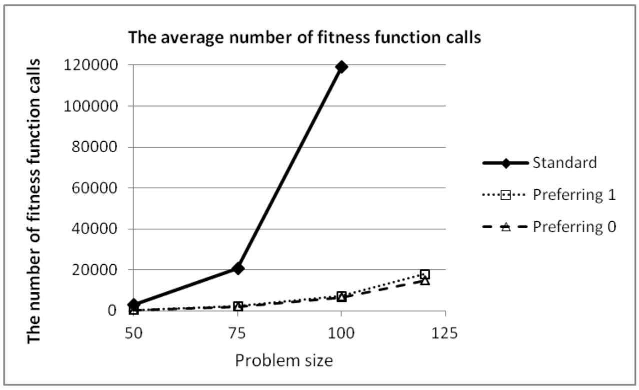

Fig. 4. The average number of fitness function calls

- the number of individuals in the population = 25,
- the algorithms were stopped after finding any pareto-optimal solution.

Each algorithm was executed 10 times. In table 1 there are average numbers of ones and zeroes and the number of fitness function calls obtained by the algorithms.

The graph in Figure. 3 illustrates the difference between the average number of zeroes and the average number of ones in the algorithms. We can see that in the standard algorithm the number of zeros and ones is almost equal. In the algorithm preferring ones, their number is greater than the number of zeros. In the algorithm preferring zeros, their number is greater than the number of ones.

The graph in Figure. 4 illustrates the average number of fitness function calls needed by the algorithms. We can see that in the standard algorithm the number of fitness function calls grows quickly, together with the problem size. In modified algorithms the number of fitness function calls is noticeably smaller than in the standard algorithm.

IV. Conclusion

In all the experiments modified algorithms needed less fitness function calls. In the standard algorithm many individuals with a different genotype can have the same value

TABLE I
THE AVERAGE NUMBER OF ZEROES AND ONES AND THE NUMBER OF FITNESS FUNCTION CALLS

		Standard	Preferring of ones	Preferring of zeroes
LOTZ50	the number of zeros	24,8	35,5	14,4
	the number of oness	25,2	14,5	35,6
	the number of fitness functions calls	3057	458	454
LOTZ75	the number of zeros	41,1	48,9	26,7
	the number of ones	33,9	26,1	48,3
	the number of fitness functions calls	20745	2172	2122
LOTZ100	the number of zeros	51,3	58,8	40,1
	the number of ones	48,7	41,2	59,9
	the number of fitness functions calls	119036	7317	6614
LOTZ120	the number of zeros	61,3	71,3	48,5
	the number of ones	58,7	48,7	71,4
	the number of fitness functions calls	471107	17805	14773

of the fitness function. In the special case the maximum value of the fitness function can be associated to many individuals, in spite of the fact that every one represents an individual located in another area of the search space. For example, individuals represented by "only zeroes in a string" and "only ones in a string" have associated the same, maximum value of the fitness function, in spite of the fact that their genetic similarity is equal to zero. The mechanism of selection based only on the value of the fitness function is not able to classify individuals on the base of their genetic relation. The search in this case gathers the character of fate searching. The FLC modifies the probability of selection of individuals in such a way that individuals different in genetic sense, have different values of the probability of selection. In the proposed algorithm the system of classifying of individuals is more effective, so the algorithm convergence is enlarged.

In the proposed algorithms, the difference between the number of ones and zeros in the solution is greater than in the standard algorithm. The solution "only zeroes in a string" or "only ones in a string" was never found. The FLC is able to direct the process of evolution toward a preferred area, but cannot guarantee finding of a specific solution.

The proposed system can be used for more complex multi-objective optimization problems.

REFERENCES

[1] Kwasnicka H., *Evolutionary Computation in Artificial Intelligence*, Publishing House of the Wroclaw University of Technology, Wroclaw, Poland, (1999) (in Polish).

[2] Masatoshi Sakawa, *Genetic Algorithms and Fuzzy Multiobjective Optimization*, Kluwer Academic Publications, Boston 2002.

[3] Michalewicz Z., *Genetic Algorithms + Data Structures = Evolution Programs*, - Springer Verlag, Berlin (1992).

[4] Pytel K., Nawarycz T., *Analysis of the Distribution of Individuals in Modified Genetic Algorithms* [in] Rutkowski L., Scherer R., Tadeusiewicz R., Zadeh L., urada J., Artificial Intelligence and Soft Computing, Springer-Verlag Berlin Heidelberg 2010.

[5] Pytel K., *The Fuzzy Genetic Strategy for Multiobjective Optimization* [in] Federated Conference on Computer Science and Information Systems, IEEE Explore Digital Library, ISBN 978-83-60810-34-4, IEEE Computer Society Press 2011.

[6] Rutkowska D., Pilinski M., Rutkowski L., *Neural Networks, Genetic Algorithms and Fuzzy Systems*, PWN Scientific Publisher, Warsaw, (1997) (in Polish).

[7] Zitzler E., *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*, PhD thesis, ETH Zrich 1999.