ISBN 978-83-60810-22-4

The Fuzzy Genetic Strategy for Multiobjective Optimization

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

***Abstract*—This paper presents the idea of fuzzy controlling of evolution in the genetic algorithm (GA) for multiobjective optimization. The genetic algorithm uses the Fuzzy Logic Controller (FLC), which manages the process of selection of individuals to the parents' pool and mutation of their genes. The FLC modifies the probability of selection and mutation of individuals' genes, so algorithms possess improved convergence and maintenance of suitable genetic variety of individuals. We accepted the well-known LOTZ problem as a benchmark for experiments. In the experiments we investigated the operating time and the number of fitness function calls needed to finish optimization. We compared results of the elementary algorithms and the modified algorithm with the modification of probability of selection and mutation of individuals. Some good results have been obtained during the experiments.**

I. Introduction

In many practical problems, it's often expected that several indicators achieve optimal value at the same time, which is called multi-objective optimization problem [1][6][10]. 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 (MOP) 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

$$x = [x_1x_2x_3...x_n] \in \Re, n \in N, \tag{2}$$

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 $x, y \in S$, 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 a 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 the final solution from the pareto-optimal set.

Genetic algorithms stand for a class of stochastic optimization methods that simulate the process of natural evolution [3][5][7][9]. 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 the evolution they do not use the specific knowledge for given problem, except of fitness function assigned to all individuals. The specific knowledge for a given problem can set a trend for evolution and improve the efficiency of the algorithm.

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 genetic algorithm was published in [7]. We used this code as elementary genetic algorithm.

There are two parameters in elementary genetic algorithms which determine evolution: the probability of selection to the parents' pool and the probability of mutation. GA can be improved by the knowledge of experts. Experts can predict the course of the process of evolution. The experts' knowledge about evolution has a descriptive character and is often subjective, so we use the fuzzy logic controller to set a trend of evolution.

II. Adaptation of the Probability of Selection and Mutation

The probability of selection determines the ability of an individual to act as a parent and produce offspring. The chances of the individual for transferring its genetic material to the next generation increase with the probability of selection. Well-adapted individuals are the most wanted ones in the parents' pool. However, weak individuals should also hit for the parents' pool in order to prevent violent loss of their genetic material and premature convergence. We suggest introduction of an additional FLC for evaluating each individual in the population. The FLC modifies the probability of selection using the following rules:

- enlarge the probability of selection for the individuals with the value of the fitness function of above the average in generations in which the average value of the fitness function grows with relation to the preceding generation,
- don't change the probability of selection for individuals with the value of the fitness function equal to the average in generations in which the average value of the fitness function does not change the relation to the preceding generation,
- diminish the probability of selection for individuals with the value of the fitness function below the average in generations in which the average value of the fitness function decreases with relation to of the preceding generation.

The FLC calculates the adaptation ratio for all individuals based on two parameters:

- the increase of the average fitness function for the whole population (determines the change of the fitness function between the current and the previous populations)

$$\Delta F_{pop} = F_{pop}^{(T)} - F_{pop}^{(T-1)} \tag{4}$$

- an individual's fitness (determine the difference between the average value of the fitness function in the population and an individual's fitness)

$$W_i = F_i^{(T)} - F_{pop}^{(T)} \tag{5}$$

where:

$F_i^{(T)}$ - the fitness function of individual *i* in moment *T*,
$F_{pop}^{(T)}$ - the average fitness function of the whole population in moment *T*.

The FLC uses the center of gravity [9] defuzzyfication method. As the result from the controller we accepted:

wps_i - the adaptation ratio of the probability of selection of individual *i*.

The modified probability of selection of individual *i* obeys the formula:

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

where:

$ps'(i)$ - the modified probability of selection of individual *i*,
$ps(i)$ - the probability of selection of individual *i*,
wps_i - the adaptation ratio of the probability of selection calculated by the FLC for the individual *i*.

The knowledge base of FLC is shown in Table 1 (fuzzy values of the adaptation ratio of probability of selection for individual *i*)

Values in the table are:

TABLE I
FUZZY VALUES OF THE ADAPTATION RATIO OF PROBABILITY OF SELECTION FOR INDIVIDUAL i

		ΔF_{pop}			
W_i		NS	ZERO	PS	PL
	S	VS	S	A	A
	A	S	A	A	L
	L	S	A	L	L

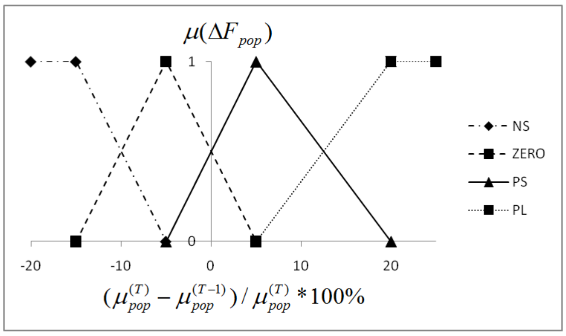

Fig. 1. The membership functions of the increase of the average fitness function for the whole population

- fuzzy sets of increase fitness function for whole population ΔF_{pop},
  - o NS - negative small,
  - o ZERO - closed to zero,
  - o PS - positive small,
  - o PL - positive large,
- fuzzy sets of an individual's fitness W_i and fuzzy sets of ratio wps_i,
  - o VS - very small,
  - o S - small,
  - o A - average,
  - o L - large.

Figures 1 - 3 shows the membership functions of the increase of the average fitness function for the whole population, an individual's fitness and the adaptation ratio of the probability of selection.

The probability of mutation determines the ability of the algorithm to explore and exploit the search space. In the initial period, mutations are frequent in order to find the solution in the whole search space (exploration mode). In the final period, mutations are rarer than at the start, so the algorithm can search the earlier established areas of possible optima (exploitation mode). The mutation of the gene can cause that the new, well adapted individual will translocate the population to the area of the total optimum. We suggest introduction of an additional FLC for evaluating each individual in the population. The FLC modifies the probability of mutation using the following rules:

- enlarge the probability of mutation of individuals with the value of the fitness function of less then the average in generations in which the average value of the fitness function decreases with relation to the preceding generation,

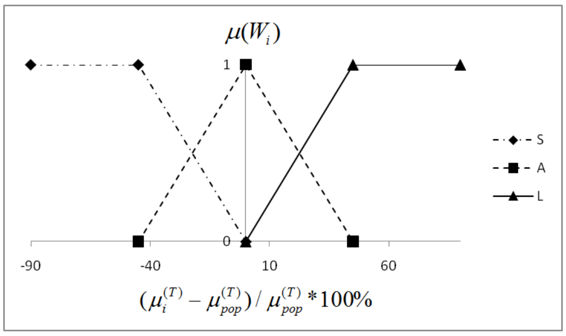

Fig. 2. The membership functions of an individual's fitness

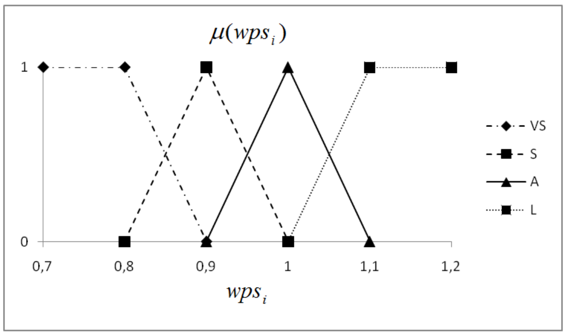

Fig. 3. The membership functions of the adaptation ratio of the probability of selection

- don't change the probability of mutation of individuals with the value of the fitness function equal to the average in generations in which the average value of the fitness function does not change in relation to the preceding generation,
- diminish the probability of mutation of individuals with the value of the fitness function above the average in generations in which the average value of the fitness function increases with relation to the preceding generation.

The FLC calculates the adaptation ratio for all individuals based on two parameters:

- the population's quality (defines the difference between the fitness function of the best individual discovered so far and the average fitness function of the current population)

$$\Delta F_{pop} = F_{max} - F_{pop}^{(T)} \tag{7}$$

- the individual's fitness (the same the parameter was used for the modification of the probability of selection)

$$W_i = F_i^{(T)} - F_{pop}^{(T)} \tag{8}$$

where:

$F_i^{(T)}$ - the fitness function of individual *i* in moment *T*,
$F_{pop}^{(T)}$ - the average fitness function for the whole population in moment *T*.

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

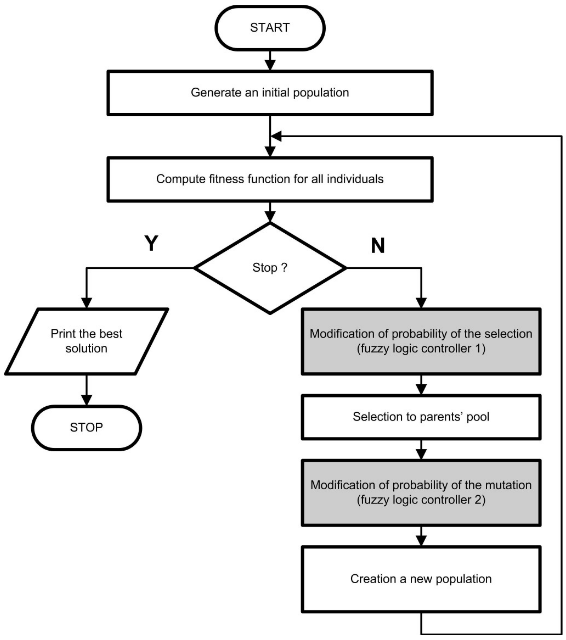

Fig. 4. The block scheme of the modified genetic algorithm

wpm_i - the adaptation ratio of the probability of mutation of individual *i*.

The modified probability of mutation of an individual *i* obeys the formula:

$$pm'(i) = pm(i) * wpm_i \text{ for } i = 1, ...N, \tag{9}$$

where:

- $pm'(i)$ - the modified probability of mutation of individual *i*,
- $pm(i)$ - the probability of mutation of individual *i*,
- wpm_i - the adaptation ratio of the probability of mutation calculated by the FLC for the individual *i*.

Figure 4 shows the block scheme of the modified genetic algorithm (the block of the fuzzy logic is noted with the grading). The construction of FLC for modification of the mutation is almost the same as FLC for modification of the selection. The construction of the fuzzy logic controller in details is considered in [8].

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 FLC estimates all individuals and modifies their probability of selection and mutation. The algorithm looks for any pareto-optimal solution. The LOTZ (Leading Ones, Trailing Zeroes) problem with the size from 50 to 100 was chosen as the test function. The LOTZ can be stated as the maximization problem of two objectives.

$$LOTZ1(x) = \sum_{i=1}^{n} \prod_{j=1}^{i} x_j, \tag{10}$$

TABLE II
THE AVERAGE VALUES OF THE RUNNING TIME AND THE NUMBER OF FITNESS FUNCTION CALLS

		Elementary	Modification of mutation	Modification of selection	Modification of selection and mutation	SEMO	NSGA2
LOTZ50	time [s]	0,063	0,097	0,133	0,199	17,15	14,26
	number of fitness function calls	25743	15196	22985	21350	7700	10700
LOTZ60	time [s]	0,172	0,207	0,361	0,575	38,32	30,41
	number of fitness function calls	60245	31401	57052	59160	36500	27500
LOTZ70	time [s]	0,564	0,528	0,92	1,309	49,76	31,13
	number of fitness function calls	164034	75335	134509	126560	44000	28500
LOTZ80	time [s]	1,098	1,036	1,956	3,377	49,78	57,11
	number of fitness function calls	275626	137082	267112	309490	45000	49000
LOTZ100	time [s]	5,233	3,932	7,879	12,99	51,09	77,57
	number of fitness function calls	1129902	460269	984622	1125690	48000	82000

$$LOTZ2(x) = \sum_{i=1}^{n}\prod_{j=i}^{n}(1 - x_j), \quad (11)$$

$$LOTZ(x) = (LOTZ1(x), LOTZ2(x)), \quad (12)$$

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

We compared algorithms with modification of selection and mutation with elementary genetic algorithm. The first population was generated randomly. All the algorithms started at the same point in the search space. The algorithms' 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,
- the number of individuals in the population = 10,
- the algorithms were stopped after finding any pareto-optimal solution.

To measure the achievements of modified algorithms, we choosed two other algorithms SEMO and NSGA2, usually used for solving LOTZ problem. SEMO is a population-based evolutionary algorithm for multiobjective optimization proposed in [4]. NSGA2 in an elitist multiobjective evolutionary algorithm proposed in [2]. We used a PISA-implementation of the algorithms written by Marco Laumanns. The SEMO and NSGA2 find a set of pareto optimal solutions, so they were stopped after finding any pareto optimal solution. For SEMO and NSGA2 we use:

- one-bit mutation,
- population size 100,
- probability of mutation 0,15
- probability of recombination 0,8.

Each algorithm was executed 10 times. In Table 2 there are average values of the running time and the number of fitness function calls obtained by the algorithms.

The graph in Figure. 5 illustrates the average running time of the algorithms.

The graph in Figure. 6 illustrates the average number of fitness function calls needed by the algorithms.

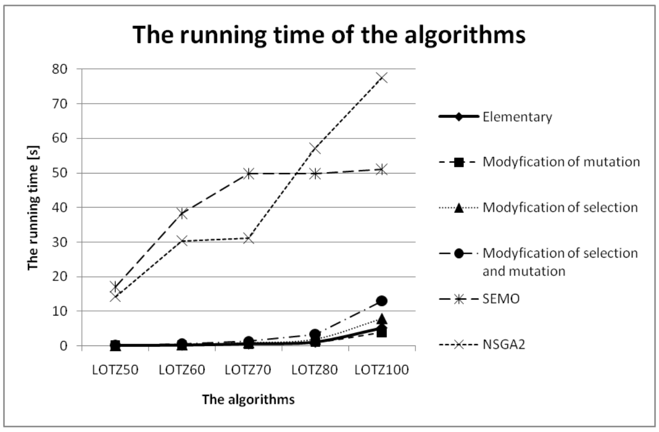

Fig. 5. The average running time of the algorithms

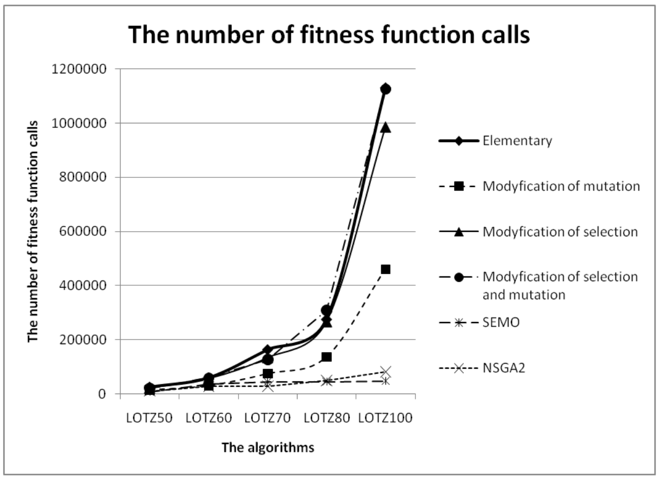

Fig. 6. The average number of the fitness function calls needed by the algorithms

IV. Conclusion

Modified algorithms need less fitness function calls in all experiments than elementary algorithms, but it need more fitness function calls than algorithm SEMO and NSGA2. The running time of modified algorithms is noticeably shorter than the algorithm SEMO and NSGA2.

The adaptation of parameters of the algorithms demands additional computational effort. The running time of the functions of the large computational complexity in the large search space can be diminished. The FLC effectively manages evolution in genetic algorithms by modification of the probability of selection and mutation.

The algorithms looked for any pareto-optimal solution. In the future, we are going to check whether the adaptation of parameters of the algorithm can direct evolution toward the point in pareto front preferred by the decision-maker.

References

[1] Coello, C.A.C. *The EMOO Repository: A Resource for Doing Research in Evolutionary Multiobjective Optimization* - IEEE Computational Intelligence Magazine, 45, February 2006.

[2] Deb K., Agrawal S., Pratap A., Meyarivan T., *A fast and elitist multiobjective genetic algorithm: Nsga-II*. IEEE Trans. Evolutionary Computation, 6(2):182197, 2002.

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

[4] Laumanns M., Thiele L., Zitzler E., *Running time analysis of multiobjective evolutionary algorithms on pseudo-boolean functions*, IEEE Trans. Evolutionary Computation, 8(2):170182, 2004.

[5] Lobo F.G., Lima C.F., Michalewicz Z, *Parameter setting in evolutionary algorithms*, Springer, 2007.

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

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

[8] 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.

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

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