BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **12**, No 2

Sofia • 2012

MultiObjective Genetic Modified Algorithm (MOGMA)

Elica Vandeva

Institute of Information and Communication Technologies, 1113 Sofia
Email: el_@abv.bg

Abstract: *Multiobjective optimization based on genetic algorithms and Pareto based approaches in solving multiobjective optimization problems is discussed in the paper. A Pareto based fitness assignment is used − non-dominated ranking and movement of a population towards the Pareto front in a multiobjective optimization problem. A MultiObjective Genetic Modified Algorithm (MOGMA) is proposed, which is an improvement of the existing algorithm.*

Keywords: *Multiobjective optimization, genetic algorithms, population, Pareto based approaches, ideal point, ideal function, MultiObjective Genetic Modified Algorithm.*

1. Introduction

Genetic algorithms are nowadays a very popular tool to solve optimization problems. They are based on the application of evolutionary principles in the search for an optimal solution. These algorithms have been recognized as most appropriate for the purposes of multiobjective optimization [3]. The genetic algorithm is regarded as an optimizing element in the multicriteria optimization process, including also a Decision Maker (DM). It represents interaction between the two, leading to a satisfactory solution.

The basis of this article is multiobjective optimization using genetic algorithms. In multicriteria optimization problems several criteria (objective functions) in the feasible set of solutions (alternatives) are simultaneously optimized.

The multiobjective optimization problem can be described in the following way:

to find a vector $x = (x_1, x_2, ..., x_n)$ that is a solution of

$$\mathrm{Min}_{x\in\Omega} \quad F(x) = \left(f_1(x), f_2(x), ..., f_m(x)\right)^{\mathrm{T}}$$

$$\text{subject to } \Omega \begin{cases} q_i(x) \le 0, & i = 1, 2, ..., k, \\ h_j(x) = 0, & j = 1, 2, ..., l, \end{cases}$$

where Ω is the set of the solutions vectors, m is the number of the criteria, $q_i(x), \quad i = 1, 2, ..., k,$ are k inequality constraints, and $h_j(x), \quad j = 1, 2, ..., l,$ are l equality constraints.

In the general case there does not exist one solution optimizing all the criteria. However, there is a set of solutions in the variables space and a corresponding set of criteria values in the criteria space, which are characterized by the following feature: any improvement in the value of one criterion leads to worsening of the value of at least one other criterion. These sets are called Pareto optimal sets [11]. A Pareto optimal set of variables is also called an efficient set, and the Pareto optimal set of criteria – a non-dominated set. Each element of these sets could be a solution of the multiobjective problem. In order to choose one element, additional information is required, that is set by the so called Decision Maker. The information that the DM specifies reflects his/her global preferences with respect to the quality of the solution sought.

A modification of one Pareto based algorithm – the MultiObjective Genetic Algorithm (MOGA) [4, 5, 15] is discussed in the paper. The idea of the modified algorithm is to avoid the hardly computed parameters. This is achieved by determining the distance of the solutions up to the ideal point of the Pareto front (a Pareto front means the non-dominated vectors distributed in the criteria space). The algorithm differs from the standard genetic algorithm only in the fitness assignment procedure, which is Pareto based. This means that it accomplishes non-dominated ranking and movement of a population towards the Pareto front in the multiobjective optimization problem.

2. Genetic algorithms

The Genetic Algorithm (GA) is a stochastic method for solving optimization problems with or without constraints based on natural selection, i.e., the process underlying the biological evolution [13]. They are an attempt to model the Darwinian evolution in terms of a computer program [10]. Thus, for a given optimization problem several arbitrary sub-optimal solutions are initially built [8]. The most successful of them remain, and on their base new ones are constructed with the intention that they will prove even better. In this way the sub-optimal and non-promising solutions are isolated from further consideration. The analogy with the process of the evolution of species is obvious – in nature this model is known as "survival of the fittest".

The Genetic Algorithms are becoming still more popular and used in variable fields [1, 9, 12, 14, 16]. The Genetic Algorithms may be applied to solve numerous optimization problems, the solution of which cannot be realized by the existing standard optimization methods, including those, in which the objective function is not continuous, non-differentiable, stochastic or definitely nonlinear.

3. Pareto based approaches

The idea to use a Pareto based fitness assignment was first proposed by Goldberg [6]. He suggested the use of non-dominated ranking and movement of a population towards the Pareto front in a multiobjective optimization problem.

The basic idea is to find the set of strings in the population that are Pareto nondominated in relation to the rest of the population. These strings are then assigned the highest rank r_i and eliminated from further consideration. Another set of Pareto nondominated strings is defined from the remaining population and it is assigned the next highest rank. This process continues until the population is suitably ranked. Goldberg also suggested the use of some kind of a niching technique that keeps the GA from converging to a single point on the front. A niching mechanism, such as sharing [7] would enable the GA to preserve all the individuals around the non-dominated frontier.

The main problem of Pareto ranking is that there is no efficient algorithm checking for a non-dominated solution in the set of feasible solutions [2]. The conventional algorithms seriously worsen their operation in cases when the population increases in size or in number of objectives. The use of sharing is also required to compute the value of σ_{share} [4] which is not easy, but the implementation of the method relies significantly on this value.

4. MultiObjective Genetic Modified Algorithm (MOGMA)

This algorithm differs from the standard Genetic Algorithm in the way of evaluating the population. The remaining part of the algorithm is the same as in the classical Genetic Algorithm.

Since MOGA gives certain inaccuracies in the fitness, it is modified with the purpose to improve the algorithm and to avoid the calibrated parameters, such as σ_{share}, $\text{Sh}(d_{ij})$, ηc_i. The basis of the fitness assignment procedure of this algorithm is the ideal point. MOGMA fitness assignment procedure is described below in six steps.

Step 1. Determine the ideal point f^0 of the Pareto front with the objective function value, instead with the parameter values. It is denoted as $\mu(j)=0$ for all possible ranks $j=1,\ldots,N$. Set the solution counter $i=1$.

Step 2. Compute the number of solutions (η_i) that dominate the solution i. For solution *i*, the rank is equal to one plus the number of solutions η_i that dominate the solution, $i : r_i = 1 + \eta_i$. In this way the non-dominated solutions are assigned a rank equal to 1, since no solution will dominate a non-dominated solution in the population. This means that in any population there must be at least one solution with a rank of one, and the maximal rank of every population member cannot exceed *N* (the population size).

Step 3. If $i < N$, increment *i* by one and go to **Step 1**. Otherwise, go to **Step 4**.

Step 4. The maximal rank r^* is identified by checking the largest r_i which has $\mu\left(r_i\right) > 0$. The sorting according to the rank and the averaging fitness yields the following problem of average fitness for each solution $i = 1,...,N$:

$$F_i = N - \sum_{k=1}^{r_i - 1} \mu(k) - 0.5\left(\mu(r_i) - 1\right).$$

The average fitness of the ideal point is conditionally denoted as $F_0 = N$.

For each solution i with a rank $r_i = 1$, the above equation assigns a fitness equal to $F_i = N - 0.5\left(\mu(1) - 1\right)$, which is the average value of $\mu(1)$ successively integer from N up to $N - \mu(1) + 1$. Set a rank counter $r = 1$.

Step 5. For each solution i with a rank *r*, compute the ideal fitness using $F_i' = \dfrac{F_i}{d_{0i}}$, where $d_{0i} = \sqrt{\sum_{k=1}^{M} \left(f_k^{(0)} - f_k^{(i)} \right)^2}$. The fitness is called "ideal" because it uses the distance up to the ideal point. In order to preserve the average fitness, the ideal fitness is scaled as follows:

$$S_i = \frac{F_0 . F_i}{r_i . \sum_{k=1}^{\mu(r)} F_k'} F_i'.$$

Step 6. If $r < r^*$, increment r by one and go to **Step 5**. Otherwise, the process is complete.

The avoiding of the parameters σ_{share}, $\text{Sh}(d_{ij})$, ηc_i which are used in MOGA makes the algorithm simpler and easier to apply. Since the selection of σ_{share} requires additional time and is relatively difficult, its exclusion considerably facilitates the algorithm and reduces the time for its execution.

The including of the ideal point on the Pareto-front guarantees that the solutions with smaller distance up to it will have a better ideal fitness. Because this distance is included in the denominator, the less it is, the greater the value of the fitness assessment will be.

5. Procedure for determining the next generation

5.1. Transfer of the best solution

Given that the best solution is well distinguished among the others, it would be a good step to directly transfer it to the new generation, thus ensuring that even in the worst case it will remain the best solution in the next generation as well. This means that the individuals of the new generation will not be worse than those in the current one.

5.2. Mutation of the worst solution

The mutation must be strategic – determined by the individual Decision Maker, the purpose of the mutation being the approaching of the solution towards the ideal point on the Pareto-optimal front.

5.3. Participation of the DM in the selection of every generation

The participation of the DM in the selection of each generation would provide him/her with better understanding of the problem and thus would facilitate his/her choice. It is not difficult, since there are not very complex calculations.

5.4. Interactive procedure for determination of the next generation

Let us compute the ideal fitness assignment in MOGMA, using a method of determining the next generation that could be realized in the following algorithmic scheme:

Step 1. Let the ideal point be $f^0 = (f_1^0, f_2^0, \ldots, f_p^0)$, where

$$\left| \begin{array}{l} f_j^0 = \min f_j(x), \\ x \in \Omega. \end{array} \right.$$

Step 2. Determination by the Decision Maker of the solutions, which he/she wishes to be changed.

Step 3. Depending on the determination process, the decisions are divided into three classes:

- Class I_1 – solutions which the decision maker is unwilling to change – they are directly transferred to the new generation;
- Class I_2 – solutions, which the decision maker agrees to be changed – they are subject to crossover;
- Class I_3 – solutions, which the decision maker wishes to be improved – they are transferred to the new generation through a controlled mutation, with the purpose to bring these solutions close to the ideal point of the Pareto front.

5.5. The stop criteria

There must be defined criteria for algorithm end in order to prevent its cycling. They may be the following:

- when a solution is reached that satisfies the preferences of the DM; in the case with the method determining the next generation, the class must contain all the solutions;
- the presence of high percentage of the states of the current population, identical or similar to each other;
- if the fitness assignment values of the best solutions of the previous few populations are close or coincide;
- if for several successive performances of MOGMA procedure no better solution can be found.

6. Test problems

Test problems [3] used to compare the performance of MOGA with MOGMA are described. The performance metrics are illustrated and the results are discussed. The population size was set to 20 in all the problems. The encoding in value is used and the proposed interactive method is applied to determine the next generation.

Figs 1-10 show the 200-th final generation and the Pareto-optimal front. These figures demonstrate the ability of MOGMA and MOGA to approach the true Pareto-front and to find diverse solutions in the front.

$$\text{ZDT1: } \begin{cases} \min f_1(x_1, x_2) = x_1, \\ \min f_2(x_1, x_2) = \dfrac{1+x_2}{x_1}, \\ \quad \text{where}: 0.1 \le x_1 \le 1, \\ \qquad\qquad 0 \le x_2 \le 5. \end{cases}$$

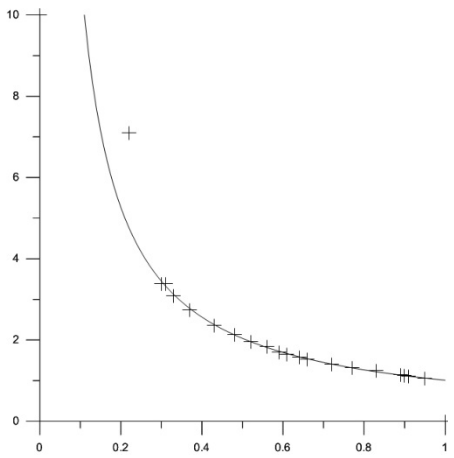

Fig. 1. MOGA

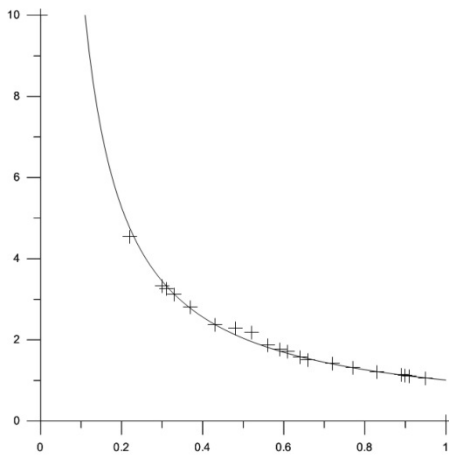

Fig. 2. MOGMA

Fig. 1 shows distinctly one point, which is more distant from the Pareto front, while in Fig. 2 this point belongs to the Pareto front. It can be seen that MOGMA manages to preserve the diversity of the solutions, like MOGA, which makes it more appropriate for the solution of this problem.

In Fig. 4 three of the points do not belong to the Pareto-front, as in Fig. 3. Since the idea of MOGMA is to provide the DM with the points located most closely to the ideal point for his/her selection, this is not of great importance. The remaining points are equally well ranked and the diversity of the solutions is again preserved.

$$\text{ZDT2:} \begin{cases} \min\ f_1(x_1, x_2) = x_1^2 \\ \min\ f_2(x_1, x_2) = \dfrac{1 + x_2^2}{x_1^2} \\ \text{where}: \sqrt{0.1} \le x_1 \le 1, \\ \qquad 0 \le x_2 \le \sqrt{5}. \end{cases}$$

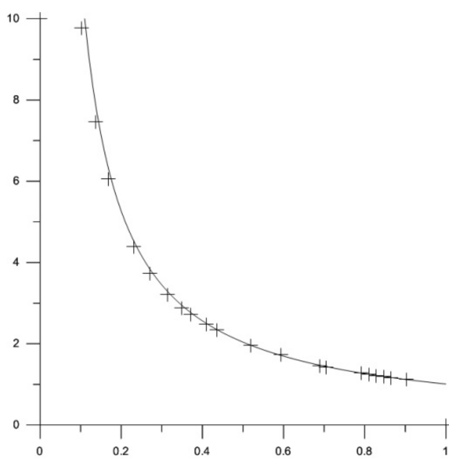

Fig. 3. MOGA

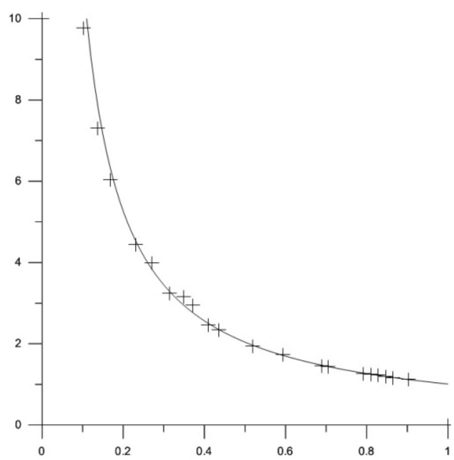

Fig. 4. MOGMA

$$\text{ZDT3:} \begin{cases} \min\ f_1(x_1, x_2) = x_1^2 + x_2^2, \\ \min\ f_2(x_1, x_2) = (x_1 + 2)^2 + x_2^2, \\ \qquad \text{where}: -5 \le x_1 \le 5, \\ \qquad\qquad -5 \le x_2 \le 5. \end{cases}$$

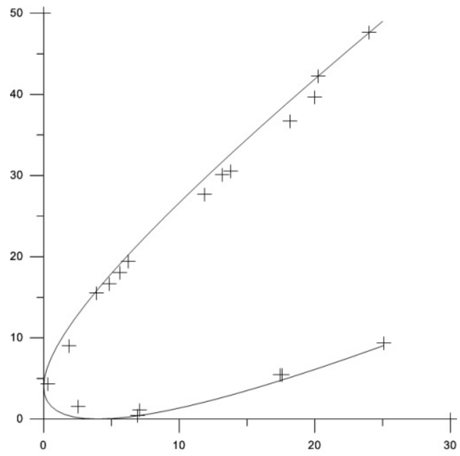

Fig. 5. MOGA

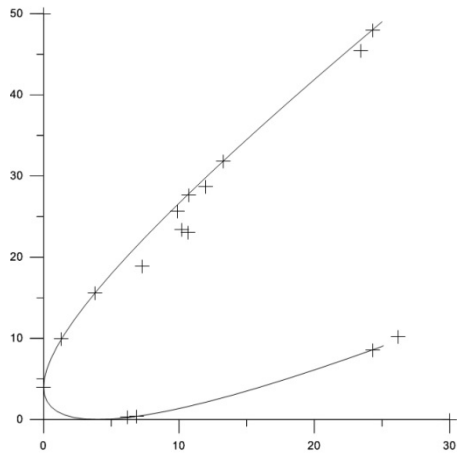

Fig. 6. MOGMA

Fig. 6 shows that more points belong to the Pareto front in comparison with Fig. 5. This proves that MOGMA operates better than MOGA.

$$\text{ZDT4: } \begin{cases} \min\ f_1(x_1, x_2) = x_1, \\ \min\ f_2(x_1, x_2) = 1 + x_2 - x_1^2, \\ \quad \text{where}: 0 \le x_1 \le 1, \\ \quad\quad 0 \le x_2 \le 3. \end{cases}$$

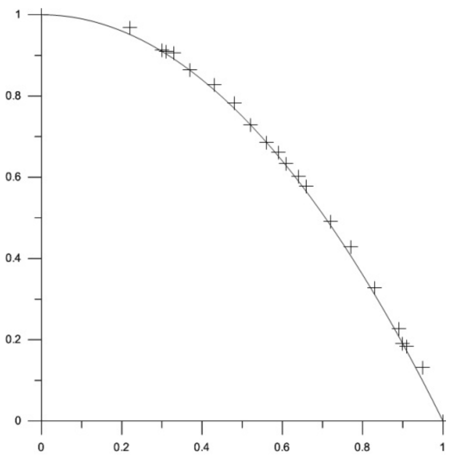

Fig. 7. MOGA

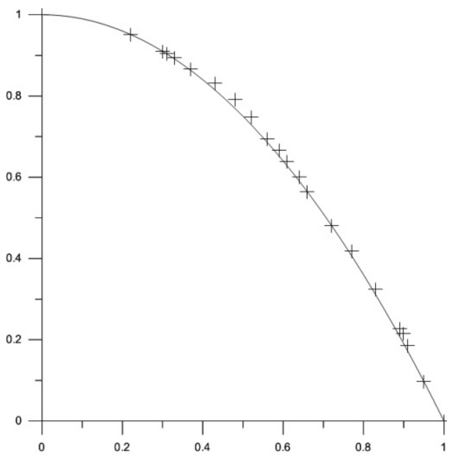

Fig. 8. MOGMA

The number of the points belonging to the Pareto front in Figs 7 and 8 is equal. Since MOGMA is more simplified and comprehensible to the DM, it is more appropriate for this problem.

$$\text{ZDT5:}\begin{cases}\min\ f_1(x_1,x_2)=x_1^2+x_2^2,\\ \min\ f_2(x_1,x_2)=x_2^2-4x_1,\\ \quad \text{where}:-2\le x_1,x_2\le 2.\end{cases}$$

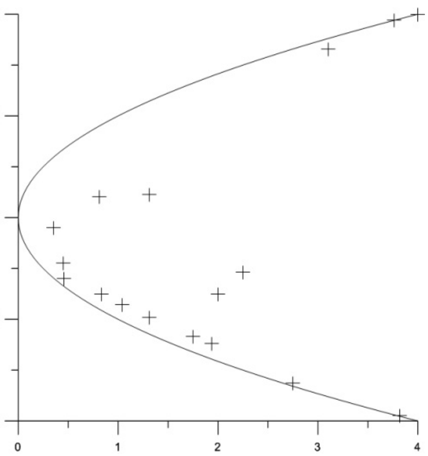

Fig. 9. MOGA

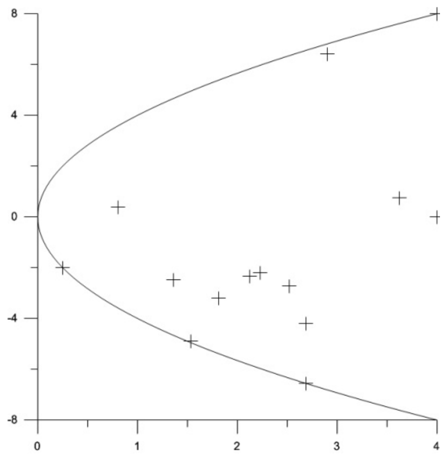

Fig. 10. MOGMA

In this problem stop criteria are activated, executing the maximal number of iterations – 200. Though in Fig. 9 there are more points, relatively close to the Pareto front, only three points belong to it. In Fig. 10 four points belonging to the Pareto front are observed, three of them being very close to the ideal point, so that the DM can easily choose one of them as the best solution.

7. Conclusion

The MOGMA algorithm is a more simpified algorithm in comparison with MOGA. Since some hardly identifiable parameters are avoided, its operation time is shorter. Moreover, it ensures easier introduction of the Decision Maker into the problem. The parameters which are used, are easily determined and do not require very complicated knowledge. That is why it would be easy to train the Decision Maker during the execution of the algorithms. The basic idea is to focus the attention of the DM on the Pareto optimal front by providing external information for the selection algorithm – the ideal point of the Pareto front. By determining the distance of each solution to it, the advantages of the separate solutions are revealed. The resulting scores are easily identifiable, allowing the DM to be easily convinced in the quality of the best solution. It is easy to see its benefits and to determine whether they satisfy the preferences of the DM.

The proposed procedure for determining the next generation ensures that it will not be worse than the current one. The idea is to determine the best solutions that will go directly into the new generation. Modification of the worst solution is also suggested, which leads to its approach to the ideal point on the Pareto front.

Thus it can be seen that the new generation will be better than the current one, or at least it will not be any worse.

Using the program results, one can definitely notice that the solutions obtained with the help of the original and the modified algorithms are similar. This means that the modified algorithm also maintains a good variety of solutions.

By introducing clear rules of operation, MOGMA can be easily implemented in a decision support system, solving multiobjective optimization problems.

Acknowledgement: The paper is partially supported by the Bulgarian National Science Fund, Grant No DTK02/71 "Web-Based Interactive System, Supporting the Building Models and Solving Optimization and Decision Making Problems".

References

1. A w a d, A. R., I. V. P o s e r. Calibrating Conceptual Rainfall-Runoff Models Using a Real Genetic Algorithm Combined with a Local Search Method, Latest Trends on Computers. – In: 14th WSEAS International Conference on COMPUTERS, Corfu Island, Greece, WSEAS, WSEAS Press, Vol. **I**, 2010, 174-181.
2. C o e l l o, C. A. C. An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineeting Design. Ph. D. Thesis. Department of Computer Science, Tulane University, New Orleans, LA, 1996. **http://delta.cs.cinvestav.mx/~ccoello/Thesis.html**
3. D e b, K. Multi-Objective Optimization Using Evolutionary Algorithms. Department of Mechanical Engineering. Indian Institute of Technology, Kanpur, India, 2001, 200-216.
4. D e b, K. Solving Goal Programming Problems Using Multi-Objective Genetic Algorithms. Indian Institute of Technology, Kanpur, 2002, 77-84.
5. F o n s e c a, C. M., P. J. F l e m i n g. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. – In: S. Forrest, Ed. Proceedings of the 5th International Conference on Genetic Algorithms (San Mateo, California, 1993), University of Illinois at Urbana-Champaign: Morgan Kauffman Publishers, 1993, 416-423.
6. G o l d b e r g, D. E. Genetic Algorithms in Search Optimization and Machine Learning. – In: Reading, Massachusetts, Addison-Wesley Publishing Company, 1989.
7. G o l d b e r g, D. E., J. R i c h a r d s o n. Genetic Algorithm with Sharing for Multi-Modal Function Optimization. – In: J. J. Grefenstette, Ed. Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms, Lawrence Erlbaum, 1987, 41-49.
8. I v a n o v a, D., M. P o p o v a, N. V a l o v. Application of Genetic Algorithm for Tuning PID Regulators. – Scientific Works of the Ruse University, Vol. **48**, 2009, Series 3.1, 123-127.
9. K e l l e r, A. A. Genetic Search Algorithm to Fuzzy Multiobjective Games: A Mathematica Implementation for Multi-Objectives Problems under Its Objective Boundary. – In: Selected Topics in Applied Computer Science, 10th WSEAS International Conference on Applied Science, Iwate Prefectural University, Japan, WSEAS, WSEAS Press, 2010, 351-359.
10. K u b a t, M., A. R o z s y p a l. Using the Genetic Algorithm to Reduce the Size of a Nearest-Neighbor Classifier and to Select Relevant Attributes. – In: Proceedings of ICML'2001, 2001, 449-456.
11. P o d i n o v s k i, V. V., V. D. N o g g i n. Pareto-Optimal Solutions to Multicriteria Problems. Moskow, FIZMATLIT, 2007. 256 p.
12. S o r n m u a n g, S., S. S u j i t j o r n. GA-Based PIDA Control Design Optimization with an Application to AC Motor Speed Control. – International Journal of Mathematics and Computers in Simulation, Vol. **4**, 2010, Issue 3, NUAN, 67-80.
13. S t a n i m i r o v i ́c, Z. A Genetic Algorithm Approach for the Capacitated Single Allocation P-Hub Median Problem. – Computing and Informatics, Vol. **29**, 2010, 117-132.

14. T a g a w a, K., H. K i m. A Local Search Technique for Large-Scale Optimum Design Problems of Double Mode SAWfilters. – WSEAS Trans. on Circuits and Systems, Vol. **6**, 2007, Issue 1, 1-8.
15. V a n d e v a, E., M. V a s s i l e v a. GA–Based Multiobjective Optimization Techniques. – In: Proceedings of International Conference Automatics and Informatics 2009, John Atanasov Union of Automation and Informatics, Sofia, Bulgaria, 2009, 105-108.
16. Y o s h i k a w a, M., H. N i s h i m u r a, H. T e r a i. A New Genetic Coding for Job Shop Scheduling Problem Considering Geno Type and Pheno Type, Recent Advances in Computer Engineering and Applications. – In: Proceedings of the 4th WSEAS International Conference on Computer Engineering and Applications, Harvard University, Cambridge, 27-29 January 2010, WSEAS, WSEAS Press, 59-62.