

# Negative Selection with Ranking Procedure in Tabu-Based Multi-criterion Evolutionary Algorithm for Task Assignment

Jerzy Balicki

Naval University of Gdynia, ul. Smidowicza 69,  
81-103 Gdynia, Poland  
J.Balicki@amw.gdynia.pl

**Abstract.** In this paper, an improved negative selection procedure to handle constraints in a multi-criterion evolutionary algorithm has been proposed. The problem that is of interest to us is the complex task assignment for a distributed computer system. Both a workload of a bottleneck computer and the cost of system are minimized; in contrast, a reliability of the system is maximized. Moreover, constraints related to memory limits and computer locations are imposed. Finally, an evolutionary algorithm with tabu search procedure and the improved negative selection is proposed to provide effective solutions.

## 1 Introduction

Evolutionary algorithms (EAs) have to exploit a supplementary procedure to incorporate constraints into fitness function in order to conduct the search precisely. An approach based on the penalty function is the most commonly used to satisfy constraints. Likewise, this technique is frequently used to handle constraints in multi-criteria evolutionary algorithms. However, it has some limitations, from which the most remarkable is the complicatedness to identify penalty coefficients.

The homomorphous mapping has been proposed as the constraint-handling procedure of EA to deal with parameter optimisation problems in order to avoid some impenetrability of the penalty function [10]. After that, a constrained-handling negative selection has been tested to optimisation problems with one criterion [3].

As a result, we propose an improved negative selection to handle constraints in multi-criterion optimisation problems. Both a workload of a bottleneck computer and the cost of machines are minimized; in contrast, a reliability of the system is maximized. Furthermore, constraints related to computer memory limits are imposed on the feasible task assignment. After all, an evolutionary algorithm with a tabu search procedure [12] and the negative selection is proposed to provide Pareto-optimal task assignments for the distributed systems. To avoid a limitation of the negative selection, we suggest introducing some distance measures from the state of an antibody to the state of the selected antigen, according to the constraints. In consequence, an improved negative selection with ranking procedure can be designed.

## 2 Negative Selection Algorithm

The immune system can be seen as a distributed adaptive system that is capable for learning, using memory, and associative retrieval of information in recognition [9]. Many local interactions provide, in consequence, fault tolerance, dynamism and adaptability [2]. The model of immune network and the negative selection algorithm are major outcomes on which most of the current works are based [8].

The negative selection algorithm (NSA) for detection of changes has been developed by Forrest et al. [6]. This algorithm is based on the discrimination principle that is used to know what a part of the immune system is and what it is not [5]. Detectors are randomly generated to reduce those detectors that are not capable of recognising themselves. Subsequently, detectors proficient to distinguish trespassers are kept. An adjusted detection is performed probabilistically by the NSA.

The negative selection can be used to manage constraints in an evolutionary algorithm by isolating the contemporary population in two groups [13]. Feasible solutions called “antigens” create the first cluster, and the second cluster of individuals consists of “antibodies” – infeasible solutions. For that reason, the NSA is applied to generate a set of detectors that verify the state of constraints.

We assume the fitness for antibodies is equal to zero. Then, a randomly chosen antigen  $G^-$  is compared to the selected antibodies. After that, the distance  $S$  between the antigen  $G^-$  and the antibody  $B^-$  is calculated due to the amount of similarity at the genotype level [1]:

$$S(G^-, B^-) = \sum_{m=1}^M s_m(G^-, B^-), \tag{1}$$

where

$M$  – the length of the solution,

$$s_m = \begin{cases} 1 & \text{if } G_m^- \text{ is matching to } B_m^- \text{ at position } m, \\ 0 & \text{in the other case.} \end{cases} \quad m=1, \overline{M};$$

The measure of genotype similarity between the antigen and the antibody depends on their representation. This assessment of similarity for the binary representation (1) can be re-defined for the integer version, as follows:

$$S'(G^-, B^-) = \sum_{m=1}^M |G_m^- - S_m^-|. \tag{2}$$

The negative selection is a modified genetic algorithm in which infeasible solutions that are similar to feasible ones are preferred in the current population. Although, almost all the random choices are based on the uniform distribution, the pressure is directed to improve the fitness of appropriate infeasible solutions.

## 3 Ranking Procedure for Negative Selection

The situation that the fitness of the winner is increased by adding the magnitude of the similarity measure may pass over a non-feasible solution with the relatively small

value of this assessment. Nevertheless, some constraints may be satisfied by this alternative. What is more, if a constraint is exceeded and the others are performed, the value of a similarity measure may be low for some cases. That is, the first of two similar solutions, in genotype sense, may not satisfy this constraint and the second one may satisfy it.

To avoid this limitation of the NSA, it was suggested introducing some distance measures from the state of an antibody to the state of the selected antigen, according to the constraints. Equalities and inequalities that are of interest to us are, as follows:

$$g_k(x) \leq 0, \quad k = \overline{1, K}, \tag{3}$$

$$h_l(x) = 0, \quad l = \overline{1, L}. \tag{4}$$

The distance measures from the state of an antibody  $B^-$  to the state of the selected antigen  $G^-$  are defined, as below:

$$f_n(B^-, G^-) = \begin{cases} g_k(B^-) - g_k(G^-), & k = \overline{1, K}, n = k, \\ h_l(B^-), & l = \overline{1, L}, n = K + l, \end{cases} \quad n = \overline{1, N}, N = K + L. \tag{5}$$

The distance  $f_n(B^-, G^-)$  is supposed to be minimized for all the constraint numbers  $n$ . If the antibody  $B^-$  is marked by the smaller assessment  $f_n(B^-, G^-)$  to the selected antigen than the antibody  $C^-$ , then  $B^-$  ought to be preferred to  $C^-$  due to the improvement of the  $n$ th constraint. Moreover, if the antibody  $B^-$  is characterized by all the shorter distances to the selected antigen than the antibody  $C^-$ , then  $B^-$  should be preferred for all constraints. However, situations may occur when  $B^-$  is characterized by the shorter distances for some constraints and the antibody  $C^-$  is marked by the shorter distances for the others. In this case, it is difficult to select an antibody.

In this paper, we suggest introducing a ranking procedure to calculate fitness of antibodies and then to select the winner. In the previous works, a ranking idea for non-dominated individuals has been applied to avoid the prejudice of the interior Pareto alternatives [1, 3].

Now, we adjust this procedure to the negative selection algorithm (NSA\*). Firstly, distances between the chosen antigen and some antibodies are calculated. Afterwards, the nondominated antibodies are determined according to their distances (5) to the antigen, and then, they get the rank equal to 1. Next, they are temporary eliminated from the population. Next, the new nondominated antibodies are found from the reduced population and they get the rank equal to 2. In this procedure, that level is increased and it is repeated until the subset of antibodies is exhausted. All the non-dominated antibodies have the same fitness because of the equivalent rank.

If  $B^-$  is the antibody with the rank  $r(B^-)$  and  $1 \leq r(B^-) \leq r_{\max}$ , then the increment of the fitness function value is estimated, as below:

$$\Delta f(B^-) = r_{\max} - r(B^-) + 1. \tag{6}$$

Afterwards, the fitness of all the chosen antibodies are increased by adding their increments. The antibodies are returned to the current population and this process is repeated typically three times the number of antibodies. Each time, a randomly chosen antigen is compared to the same subset of antibodies. Next, the same procedure as for the NSA is carried out. Afterwards, a new population is constructed by reproduction, crossover and mutation without calculations of fitness. That process is repeated until a convergence of population emerges or until a maximal number of iterations is exceeded. At the end, the final population of the negative selection algorithm is returned to the external evolutionary algorithm.

### 4 Task Assignment in Distributed Computer System

To test the ability of NSA\* for handling constraints, we consider a multi-criterion optimisation problem for task assignment in a distributed computer system [1].

Finding allocations of program modules may decrease the total time of a program execution by taking a benefit of the particular properties of some workstations or an advantage of the computer load. An adaptive evolutionary algorithm has been considered for solving multi-objective optimisation problems related to task assignment that minimize  $Z_{max}$  – a workload of a bottleneck computer and  $F_2$  – the cost of machines [1]. The total numerical performance of workstations is another criterion for assessment the quality of a task assignment and it has been involved to multi-criterion problem formulated in [1]. Moreover, a reliability  $R$  of the system is an additional criterion that is important to assess the quality of a task assignment.

In the considered problem, both a workload of a bottleneck computer and the cost of machines are minimized; in contrast, a reliability of the system is maximized. In addition, constraints related to memory limits and computer locations are imposed on the feasible task assignment. A set of program modules  $\{M_1, \dots, M_m, \dots, M_M\}$  communicated to each others is considered among the coherent computer network with computers located at the processing nodes from the set  $W = \{w_1, \dots, w_i, \dots, w_I\}$ . A set of program modules is mapped into the set of parallel performing tasks  $\{T_1, \dots, T_v, \dots, T_V\}$ . Some modern scheduling algorithms for tasks are proposed in [12].

Let the task  $T_v$  be executed on computers taken from the set of available computer sorts  $\Pi = \{\pi_1, \dots, \pi_j, \dots, \pi_J\}$ . The overhead performing time of the task  $T_v$  by the computer  $\pi_j$  is represented by an item  $t_{vj}$ . Let  $\pi_j$  be failed independently due to an exponential distribution with rate  $\lambda_j$ . We do not take into account of repair and recovery times for failed computer in assessing the logical correctness of an allocation. Instead, we shall allocate tasks to computers on which failures are least likely to occur during the execution of tasks. Computers and tasks can be allocated to nodes in purpose to maximize the reliability function  $R$  defined, as below [1]:

$$R(x) = \prod_{v=1}^V \prod_{i=1}^I \prod_{j=1}^J \exp(-\lambda_j t_{vj} x_{vi}^m x_{ij}^\pi), \tag{7}$$

where

$$x_{ij}^\pi = \begin{cases} 1 & \text{if } \pi_j \text{ is assigned to the } w_i, \\ 0 & \text{in the other case.} \end{cases}$$

$$x_{vi}^m = \begin{cases} 1 & \text{if task } T_v \text{ is assigned to } w_i, \\ 0 & \text{in the other case,} \end{cases}$$

$$x = [x_{11}^m, \dots, x_{1I}^m, \dots, x_{v1}^m, \dots, x_{vI}^m, x_{11}^\pi, \dots, x_{1J}^\pi, \dots, x_{ij}^\pi, \dots, x_{I1}^\pi, \dots, x_{Ij}^\pi, \dots, x_{IJ}^\pi]^T.$$

A computer with the heaviest task load is the bottleneck machine and its workload is a critical value that is supposed to be minimized [1]. The workload  $Z_{\max}(x)$  of the bottleneck computer for the allocation  $x$  is provided by the subsequent formula:

$$Z_{\max}(x) = \max_{i \in 1, I} \left\{ \sum_{j=1}^J \sum_{v=1}^V t_{vj} x_{vi}^m x_{ij}^\pi + \sum_{v=1}^V \sum_{u=1}^V \sum_{i=1}^I \sum_{k=1}^I \tau_{vui k} x_{vi}^m x_{uk}^m \right\}, \tag{8}$$

where  $\tau_{vui k}$  – the total communication time between the task  $T_v$  assigned to the  $i$ th node and the  $T_u$  assigned to the  $k$ th node.

Figure 1 shows the workload of the bottleneck computer in the distributed computer system for generated task assignments in a systematic way. The function  $Z_{\max}$  takes value from the period [40; 110] for 256 solutions. What is more, even a small change in task assignment related to the movement of a task to another computer or a substitution of computer sort can cause a relatively big alteration of its workload.

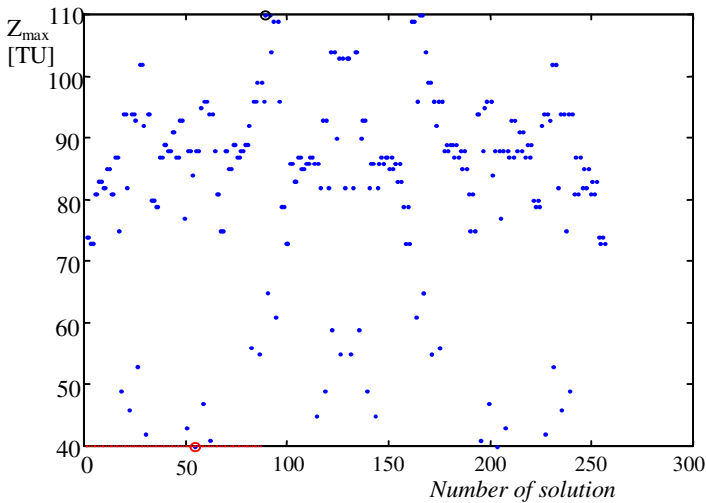


Fig. 1. Workload of the bottleneck computer for generated solutions

A computer should be equipped with necessary capacities of resources for a program execution. Let the following memories  $z_1, \dots, z_r, \dots, z_R$  be available in an entire system and let  $d_{jr}$  be the capacity of memory  $z_r$  in the workstation  $\pi_j$ . We assume the task  $T_v$  reserves  $c_{vr}$  units of memory  $z_r$ , and holds it during a program execution. Both values  $c_{vr}$  and  $d_{jr}$  are nonnegative and limited. The memory limit in a machine cannot be exceeded in the  $i$ th node, what is written, as follows:

$$\sum_{v=1}^V c_{vr} x_{vi}^m \leq \sum_{j=1}^J d_{jr} x_{ij}^\pi, \quad i = \overline{1, I}, \quad r = \overline{1, R}. \tag{9}$$

The other measure of the task assignment is a cost of computers [1]:

$$F_2(x) = \sum_{i=1}^I \sum_{j=1}^J \kappa_j x_{ij}^\pi, \tag{10}$$

where  $\kappa_j$  corresponds to the cost of the computer  $\pi_j$ .

### 5 Adaptive Evolutionary Algorithm with NSA\* and Tabu Mutation

The total computer cost is in conflict with the numerical performance of a distributed system, because the cost of a computer usually depends on the quality of its components. The faster computer or the higher reliability of it is the more expensive it is. Additionally, the workload of the bottleneck computer is in conflict with the cost of the system. Let  $(X, F, P)$  be the multi-criterion optimisation question for finding the representation of Pareto-optimal solutions [7]. It is established, as follows:

1)  $X$  - an admissible solution set

$$X = \{x \in B^{I(V+J)} \mid \sum_{v=1}^V c_{vr} x_{vi}^m \leq \sum_{j=1}^J d_{jr} x_{ij}^\pi, \quad i = \overline{1, I}, \quad r = \overline{1, R};$$

$$\sum_{i=1}^I x_{vi}^m = 1, \quad v = \overline{1, V}; \quad \sum_{j=1}^J x_{ij}^\pi = 1, \quad i = \overline{1, I} \}$$

where  $B = \{0, 1\}$

2)  $F$  - a quality vector criterion

$$F : X \rightarrow \mathbb{R}^3 \tag{11}$$

where

$\mathbb{R}$  – the set of real numbers,

$$F(x) = [-R(x), Z_{\max}(x), F_2(x)]^T \text{ for } x \in X,$$

$R(x), Z_{\max}(x), F_2(x)$  are calculated by (7), (8) and (10), respectively

3)  $P$  - the Pareto relation [14].

An overview of evolutionary algorithms for multi-objective optimisation problems is submitted in [3], [4]. An analysis of the task assignments has been carried out for two evolutionary algorithms. The first one was an adaptive evolutionary algorithm with tabu mutation AMEA+ [1]. Tabu search procedure was applied as an additional mutation operator to decrease the workload of the bottleneck computer. However, some numerical examples indicated that obtained task assignments have not satisfied constraints in many cases. Therefore, we suggest introducing a negative selection algorithm with ranking procedure to improve the quality of obtained solutions.

Better outcomes from the NSA\* are transformed into improving of solution quality obtained by the adaptive multi-criterion evolutionary algorithm with tabu mutation AMEA\*. This adaptive evolutionary algorithm with the NSA\* gave better results than the AMEA+. After 200 generations, an average level of Pareto set obtaining was 1.5% for the AMEA\*, 1.9% for the AMEA+ (Fig. 2).

Fifty test preliminary populations were prepared, and each algorithm started from these solutions. For integer constrained coding of chromosomes there were 12 decision variables in the test optimisation problem. The binary search space consisted of  $1.0737 \times 10^9$  chromosomes and included 25 600 admissible solutions.

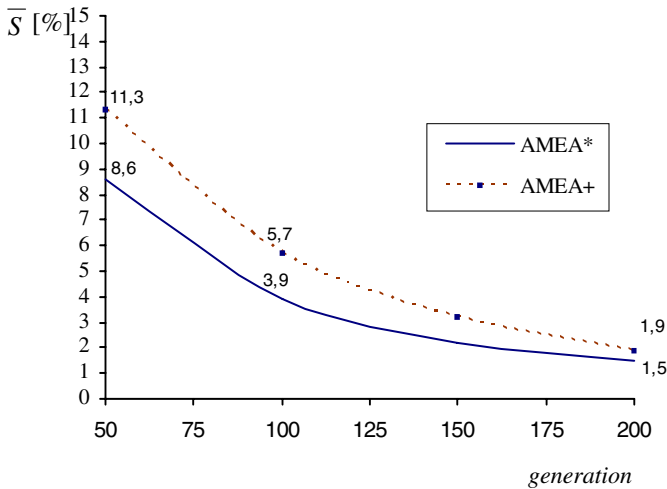


Fig. 2. Convergence for outcomes of the AMEA\* and the AMEA+

## 6 Concluding Remarks

The adaptive evolutionary algorithm with the improved negative selection and tabu mutation can be applied for finding Pareto-optimal task allocations in a three-objective optimisation problem. In this problem, the workload of the bottleneck computer and the cost of computers are minimized. Moreover, the system reliability is maximised.

The negative selection algorithm can be used to handle constraints and improve a quality of the outcomes obtained by an evolutionary algorithm. Our future works

will concern on a development the AMEA\* for finding Pareto-optimal solutions of the multi-objective optimisation problems.

## References

1. Balicki, J., Immune Systems in Multi-criterion Evolutionary Algorithm for Task Assignments in Distributed Computer System. *Lectures Notes in Computer Science*, Vol. 3528 (2005) 51-56
2. Bernaschi, M., Castiglione, F., Succi, S.: A High Performance Simulator of the Immune System. *Future Generation Computer System*, Vol. 15 (1999) 333-342
3. Coello Coello, C. A., Van Veldhuizen, D. A., Lamont, G.B.: *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York (2002)
4. Deb, K.: *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Chichester (2001)
5. Farmer, J.D.: A Rosetta Stone for Connectionism. *Physica D*, Vol. 22 (1990) 153-187
6. Forrest, S., Perelson, A.S.: Genetic Algorithms and the Immune System. *Lecture Notes in Computer Science* (1991) 320-325
7. Greco, S., Slowinski, R., Stefanowski J.: Mining Association Rules in Preference-ordered Data. *Lectures Notes in Computer Science*, Vol. 2366 (2005) 442-451
8. Jerne, N.K.: Idiotypic Networks and Other Preconceived Ideas. *Immunological Review*, Vol. 79 (1984) 5-25
9. Kim, J. and Bentley, P. J.: Immune Memory in the Dynamic Clonal Selection Algorithm. *Proc. of the First Int. Conf. on Artificial Immune Systems*, Canterbury, (2002) 57-65
10. Koziel, S., Michalewicz, Z.: Evolutionary Algorithms, Homomorphous mapping, and Constrained Parameter Optimisation. *Evolutionary Computation*, Vol. 7 (1999) 19-44
11. Schaefer, R., Kołodziej, J.: Genetic Search Reinforced by the Population Hierarchy. In De Jong K. A., Poli R., Rowe J. E. (eds): *Foundation of Genetic Algorithms*, Morgan Kaufman Publisher (2003) 383-399.
12. Weglarz, J., Nabrzyski, J., Schopf, J.: *Grid Resource Management: State of the Art and Future Trends*. Kluwer Academic Publishers, Boston (2003)
13. Wierzchon, S. T.: Immune-based Recommender System. In O. Hryniewicz, J. Kacprzyk, J. Koronacki and S. T. Wierzchon (eds.) *Issues in Intelligent Systems. Paradigms*. Exit, Warsaw (2005) 341-356
14. Zitzler, E., Deb, K., and Thiele, L.: Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, Vol. 8, No. 2 (2000) 173-195