THE INFLUENCE OF SOME CONTROLLING PARAMETERS ON THE EFFICIENCY OF AN IMMUNE ALGORITHM

Agnieszka KIMACZYŃSKA[*], Sebastian KIMACZYŃSKI[**], Ryszard ROHATYŃSKI[***]

University of Zielona Góra, Department of the Management,
Podgórna 50, 65-001 Zielona Góra,
[*] A.Kimaczynska@wz.uz.zgora.pl, [**] S.Kimaczynski@wz.uz.zgora.pl, [***] R.Rohatynski@wz.uz.zgora.pl

Summary

In the paper the influence of the probability of the mutation and size of population on the efficiency of the immune algorithm is considered. The influence was investigated on a simple knapsack problem example. A remarkable lack of the results repeatability was observed. To quantitative description of this phenomenon statistical measures were applied. The best combinations of the parameter values of the immune algorithm were found by means of the full plan of numerical experiments.

Keywords: immune algorithm, knapsack problem, efficiency.

WPŁYW PARAMETRÓW STERUJĄCYCH NA EFEKTYWNOŚĆ ALGORYTMU IMMUNOLOGICZNEGO

Streszczenie

W artykule opisano badania wpływu prawdopodobieństwa mutacji i liczebności populacji na skuteczność algorytmu immunologicznego. Jako przykładowe zadanie systemu immunologicznego wybrano problem plecakowy. Zaobserwowano brak powtarzalności wyników. Do opisu tego zjawiska zastosowano ujęcie statystyczne. Doświadczalnie określono najlepszą kombinację wartości parametrów algorytmu immunologicznego.

Słowa kluczowe: algorytm immunologiczny, problem plecakowy, efektywność.

INTRODUCTION

In an earlier paper an investigation of genetic algorithms efficiency in task of a mechanism optimization was presented [1]. Algorithms were tested at different values of their main controlling parameters: crossover and mutation probabilities and the size of population. As a result of numerous computational experiments values of these parameters which assure the best efficiency have been obtained. The iteration numbers to finding the best solution and their statistical dispersion have been accepted as the measures of efficiency. In the present paper the similar concept has been assumed for the immune algorithms investigation.

Artificial immune system performance depends significantly on two parameters: the population size and the mutation probability. Proper selection of these parameters is a prerequisite of the algorithm convergence and achievement of correct results. The large number of population causes long time of calculations (i.e. large numer of iterations), while too small one can result in incorrect solutions. On the other hand, too high probability of the mutation can result in disruption of the search direction, whereas too low one causes the convergence of the evolution process extremely slow.

Consequently, there must be optimum values of the population size and mutation probability for a certain class of optimization problems. The paper aims at the demonstration of this statement on a relative simple example.

1. OUTLINE OF THE ALGORITHM AND PROBLEM TO BE SOLVED

a. Immune algorithm

Artificial immune [2] is relatively new discipline which started about a decade ago. Nevertheless, it has gained many successful applications, e.g. in diagnostics of computer viruses [3], in security systems of offices, in recognizing signals warning about a potential failure in aeroplanes, in expert systems for advising physicians in diagnosing diseases from symptoms [4], and many other. Artificial immune systems have been inspired by natural immune systems of living organisms. The main function of a natural immune system is protection of the organism from toxic cells and pathogens. This is performed by a two-stage process: (i) the detection of pathogens and, subsequently, (ii) their efficient elimination. The detection consists in recognition whether the particular cell is a familiar one or a stranger. The stranger can be undesirable virus, a bacterium, or even a degenerated cell. Lymphocytes of B-type and T-type are responsible for the pathogens identification. After having the pathogens detected the lymphocytes react sending antibodies to neutralize menacing cells. The lymphocytes which have been activated are subjected to cloning. Thanks to this they remember pathogens and the next time they react to the pathogens more quickly. In order to increase the efficiency of the lymphocytes' mission they are also subjected to genetic operations: crossover, inversion, and mutation [2].

b. Knapsack problem

The classical knapsack problem is an abstraction of many real problems occuring in logistics, in digital coding, in aiding managers to undertake decisions, etc. [6]

The conventional formulation of this problem looks as follows [5]:

There is number *n* of objects (parcels, goods, etc.), each having a certain size w_i and a particular value c_i. There is also given a container („the knapsack") of a capacity *W*. The point of the task is to place the objects in the knapsack with some constraints: to maximize the sum of the values of objects packed into knapsack but not exceeding the capacity of the knapsack.

Formally the knapsack problem is recorded as bellow:

$$f(x) = c_i x_i \rightarrow \max$$
$$w_i x_i \le W$$

Where:
$f(x)$ – the objective function,
c_i – value of item of *i*'ts number,
x_i – binary variable which is '1' if an item is packed to the knapsack and '0' otherwise.
w_i – a characteristic feature (size, volume, weight) of the item number *i*,
W – the entire capacity of the knapsack.

2. PROBLEM FORMULATION

The research was aimed at finding the influence of the immune algorithm controlling parameters, i.e. the population size and the mutation probability, on the probability of achieving correct results in various number of iterations. For this purpose a knapsack problem was formulated and for 25 combinations of the parameters with 15 repetitions of each combination, numerical experiments were carried out.

a. The knapsack problem to be solved

The exemplary knapsack problem was formulated as follows:

We have 15 things which have to be packed into a knapsack of the capacity up to 70 units.
Weights of things (in some units) are given in order below:

[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15]

Values of things (in some units) are given in order below:

[8 1 12 5 11 9 4 4 7 15 4 6 10 3 2]

The optimal solution of this problem is known. It is shown in a vector form:

[1 1 1 1 1 1 1 1 1 0 1 1 0 0 0],

where '1' means that the thing has been packed and '0' means the given thing has been ommitted.

Optimal solution of this knapsack problem is known. The optimal value of the knapsack is 54, and its weight is 68 units.

b. Plan of experiments

The full plan of experiments for the considered problem was assumed that resulted in 25 combinations of the population size and mutation probability values. These combinations are listed in table below:

Table 1 Combinations of the population size and mutation probability values.

No. of the combinations	Size of population	Mutation probability
1	10	0,2
2	10	0,3
3	10	0,5
4	10	0,7
5	10	0,9
6	20	0,2
7	20	0,3
8	20	0,5
9	20	0,7
10	20	0,9
11	50	0,2
12	50	0,3
13	50	0,5
14	50	0,7
15	50	0,9
16	75	0,2
17	75	0,3
18	75	0,5
19	75	0,7
20	75	0,9
21	100	0,2
22	100	0,3
23	100	0,5
24	100	0,7
25	100	0,9

3. RESULTS OF COMPUTER SIMULATION

The immune algorithm which was used for the experiment was adapted from [2]. The original algorithm was converted for Pascal on Matlab. A simple procedure of sorting the objective function values was added. Also, there was changed the way the parameter M (which is responsible for number of clones) values was modified. The parameter M decreased by 1, if during 20 iterations there was no improvement of the value of the objective function. When the value of the parameter M reached 0, the algorithm was programmed to stop. This stop condition was meant to prevent repetition of the algorithm iterations which would not bring any improvements of the objective function value. It turned out that often the algorithm stopped prematurely, not reaching the optimum result.

For the statistical evaluation of the algorithm efficiency, calculations for each combination were repeated 15 times.

The following data were recorded:

- the number of the iteration,
- the weight of the knapsack obtained for each member of the population,
- the value of the knapsack obtained for each member of the population.

These data were used for calculation of the average numbers of the iterations, the average weights of the knapsack, the average values of the knapsack, and for evaluation of the scattering of the results (by the standard deviation), for each combination of input data. The results are shown on figures 1 to 5.

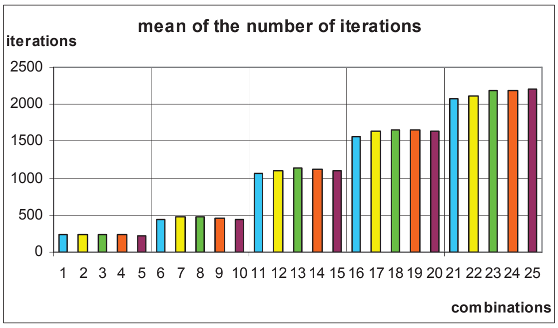

Figure 1. The mean of the number of the iterations

The mean of the number of algorithm iterations is shown on figure 1. The number of the iterations depends significantly on the size of the population. It increases with the population size.

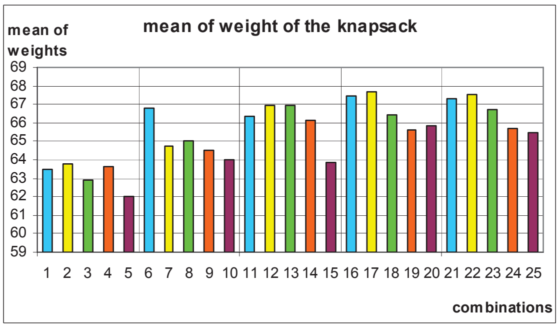

Figure 2. The mean of the weight of the knapsack

The mean of the weight of the knapsack as shown on figure 2, is strongly dependent on quantity and value of things packed to the knapsack: the greater quantity and value of things packed the greater weight of the knapsack. The maximum weight of the knapsack - 70 units, was never violated.

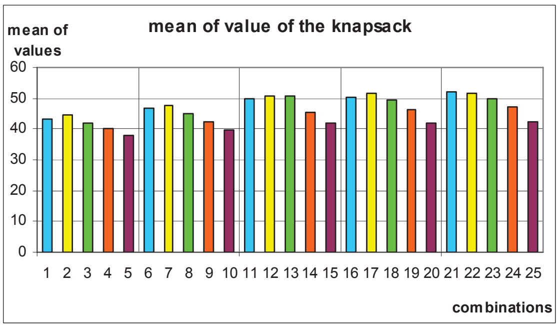

Figure 3. The mean of the value of the knapsack

Figure 3 depicts the mean of the knapsack values. For a given population size all members of the populations, which were close to the optimum, were obtained at the mutation probability 0,3. These parameter combinations which yielded the average value of the knapsack greater than 50 (the best solution is 54 units) deserve special attention. Numbers of such the combinations are 12, 13, 16, 17, 21, 22, 23.

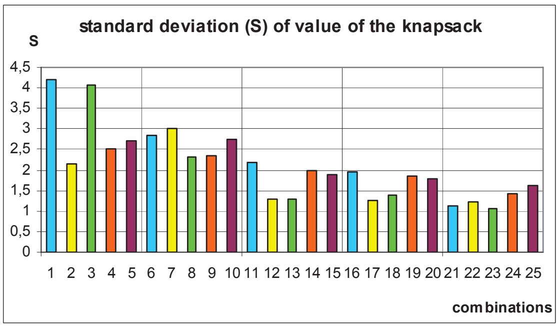

Figure 4. The standard deviation of the knapsack

Except of the mean values of the knapsack, also the standard deviations of the values were evaluated. It is shown on figure 4. It can be seen that the standard deviation of the values is fairly high. This phenomenon of strong randomization of results is a characteristic feature of the immune algorithms. Of course, the lesser deviation, the better. Thus, the special attention deserve these parameter settings which yield the standard deviation below 1.5; namely the combinations 12, 13, 17, 18, 21, 22, 23. These combinations resulted in a relative stability of the results of the investigated immune algorithm. For the user it means a relatively high credibility of the results i.e. it is not necessary to run the algorithm many times.

Because relations between the mean values of the knapsack and the standard deviations are of great importance so they are demonstrated on fig. 5.

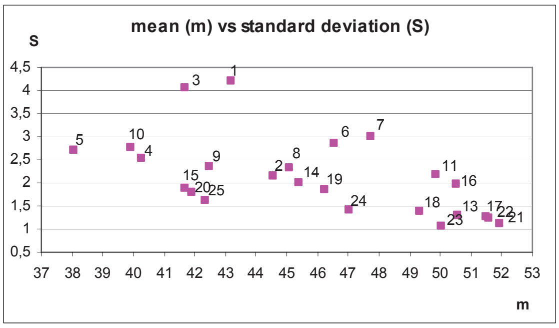

Figure 5. The mean of the results versus their standard deviation

This figure confirms that the best settings are the combinations 21, then 22 and 17. They have population size 100 or 75, and the mutation probability equal to 0.2 or 0.3. For these parameter values, however, amount of the iterations is quite large, which means a long calculation time. For the population of 75 members the algorithm needs about 1600 iterations, whereas at 100 members – as many as 2100 iterations. However, this number of the population members and the probability of mutation equal 0.2 or 0.3 guarantee receiving of the most credible results.

There is explicit need for the trade off between size of population, time of calculation and probability of getting optimal solution when using artificial immune algorithms. Recognition of this fact opens an interesting field for further investigations.

4. CONCLUSIONS

Experiments carried out by means of the immune algorithm on the knapsack problem have shown that there are some combinations of the values of the algorithm controlling parameters ensuring its relatively highest efficiency. Our research confirmed that in the artificial immune systems the population size plays the same role as in natural ones; the greater, however, it is the greater is the number of iterations needed for the algorithm to arrive at the best solution. On the other hand, the probability of the mutation has not shown monotonical impact on the efficiency of calculations; there seem to be a range of the best mutation probability values. Best operation of the immune algorithm was observed when probability of the mutation was about 0.3. It should be noted that this figure is much greater than the one recommended for the genetic and evolutionary algorithms.

Any generalization of the results presented in this paper should be taken with caution. Different kinds of problems have own specificity which certainly would require another set of values of controlling parameters, and perhaps evev another set of controlling parameters. However, it is very probable that for any specific kind problem there exist optimum values of some parameters. The awareness that such values exist creates a promising area for further research on the more efficient use of the artificial immune algorithms in searching for innovative solutions of problems which are difficult for other methods.

5. REFERENCES

[1] R. Rohatyński, T. Klekiel, K. Białas-Heltowski: *Investigation of evolutionary algorithm effectiveness in optimal synthesis of certain mechanism*, *Evolutionary methods in mechanics*, Kluwer Academic Publishers, Cracow 2004, pp. 13 – 22.

[2] S. T. Wierzchoń: *Sztuczne Systemy immunologiczne – teoria i zastosowania*, EXIT, Warszawa 2001.

[3] J. O. Kephard: A *biologically inspired immune system for computers*, Artificial Life IV, MIT Press 1994, pp. 130 – 139.

[4] M. Bereta, T. Burczyński: *Hybrid immune algorithm for feature selection and classification of ECG signals*, AI-METH 2005, Gliwice 2005, pp. 25 – 28.

[5] M. M. Sysło: *Algorytmy*, WSiP, Warszawa 2002;

[6] P. Kamiński: *Formulation objective function of the decision – making problem in ship power plant*, International Conference Design 2006, Dubrovnik, vol 1.

[7] M. Święcicki, W. Wais, P. Wais, L. Nowak: *Metody optymalizacji immunologicznej w rozwiązywaniu problemów kombinatorycznych należących do klasy NP*, Zastosowania teorii systemów, WIMiR AGH, Kraków 2005, pp. 349–363.