A Distributed Immune Algorithm for Solving Optimization Problems

Mariusz Oszust[1] and Marian Wysocki[2]

[1] Rzeszow University of Technology, Department of Computer and Control Engineering
W. Pola 2, 35-959 Rzeszow, Poland
moszust@prz-rzeszow.pl
[2] Rzeszow University of Technology, Department of Computer and Control Engineering
W. Pola 2, 35-959 Rzeszow, Poland
mwysocki@prz-rzeszow.pl

Abstract. The mammal immune system is a distributed multiagent system. Its properties of distributive control and self organization have created interest in using immune principles to solve complex engineering tasks such as decentralized robot control, pattern recognition, multimodal and combinatorial optimization. In this paper a new immunity-based algorithm for solving optimization problems is proposed. The algorithm differs from the representative immune algorithm CLONALG. The agents participating in distributed problem solving enrich their knowledge about the solution via communication with other agents. Moreover they are decomposed into groups of specialists that can modify only some decision variables and/or use their own method of local improvement of the solution. The empirical results confirming usability of the algorithm and its advantage over CLONALG are presented. Obtained estimates of the global optima of multimodal test functions and traveling salesperson problem (TSP) are closer to the theoretical solutions and require fewer tentative computations.

1 Introduction

The mammal immune system contains a set of tissues and cells protecting the body from foreign structures. The foreign structures that activate cells of the immune system (the *lymphocytes*) are called *antigens*. The lymphocytes cooperate to defense the organism against antigens. Part of them called *B lymphocytes* proliferate after the contact with the antigen and differentiate themselves during the *clonal selection* process. In the differentiation stage the B lymphocyte provides accelerated somatic mutations (*hypermutation*) on its *antibodies* in order to acquire better binding (*affinity*) to the antigen. After successful elimination of the antigen the best lymphocytes are maintained as memory cells. According to Jerne's idiotypic network (immune network) hypothesis

Please cite this paper as follows: Oszust M., Wysocki M.: A Distributed Immune Algorithm for Solving Optimization Problems, Intelligent. Distributed Computing, Systems and Applications, Studies in Computational Intelligence Springer-Verlag Berlin / Heidelberg vol. 162, pp. 147–155, 2008. The final publication is available at http://link.springer.com/chapter/10.1007%2F978-3-540-85257-5_15

[5,3] antibodies are stimulated not only by antigens but also by other antibodies. The basic idea of this hypothesis is that the immune system is constructed of a lymphocyte network. These lymphocytes interact with each other. In this way, the immune system is a parallel distributed system.
There are many algorithms based on immune metaphors. The authors of [2] propose an immune algorithm CLONALG based on the clonal selection principle. The algorithm operates on a population of lymphocytes. The lymphocytes with best antibody-antigen affinity (best stimulated by the antigen) are cloned and the clones are hypermutated. Next the best clones replace the worse ones. The process ends after predefined number of generations. Wierzchon in [10,11] proposed a modified clonal selection algorithm extending CLONALG by preselection and crowding mechanisms.
Due to the similarities observed between the immune system and the multi-agent system, (e.g.: distributed or decentralized structure, autonomous entities with individual and global goals, ability of communication and coordination, adaptability, knowledge with which they make intelligent decisions) various artificial immune systems are created using agents. The paper [8] provides an agent system AISIMAM solving a mine detection problem. The proposed model of the system defines two types of agents: antigens and lymphocytes. The lymphocytes (robots) cooperate trying to detect and diffuse the antigens (mines). More complex example of a mutlitagent system is described in [6]. Its main goal is to provide an integrated solution to control and coordinate distributed systems with large number of autonomous agents. The lymphocytes in this system identify the antigen, use communication with other lymphocytes to generate new capabilities, gain knowledge and make distributed decisions.
This paper presents an algorithm inspired by the concept of agents participating in distributed problem solving. The proposed distributed immune algorithm (DIA) does not directly use clonal selection but it rather resembles the Jerne's idiotypic network. The lymphocytes in DIA enrich their knowledge about the solution of the problem via communication with other lymphocytes. Moreover, they are decomposed into groups of specialists that can modify (mutate) only some decision variables and/or use their own method of mutation. The paper is organized as follows. Section 2 presents the new algorithm. Section 3 presents the results provided by the algorithm using a set of benchmark functions and compares them with the results obtained by other immune algorithms. It also contains discussion about the parameters of the algorithm and introduces the results of implementation 100 cities TSP. Finally, section 4 concludes this paper and suggests future research directions.

2 The new Algorithm

The aim of this algorithm is to solve optimization problems. Without loss of generality we consider maximizing a scalar objective function $J(x)$ with respect to the vector x of decision variables from a feasible region X. Like in other immune algorithms, two types of agents appear in the proposed method, i.e. the antigen and a set of lymphocytes. The antigen represents unknown global solution of the problem. The lymphocytes, identified here with antibodies, represent candidates of the solution. They are decomposed into groups of specialists. The specialist can modify (mutate) only some decision vari-

ables in x and/or it uses its own method of mutation, both defined at the initialization stage. At the beginning of the algorithm the antigen sends its estimation of the solution (the starting point) to a randomly chosen lymphocyte. This lymphocyte tries to improve the obtained solution through a mutation (hypermutation) process to get higher value of the objective function J. Next it sends the result to some set of randomly chosen lymphocytes. Each receiver repeats the activities of the sender and the process is continued. Each transferred solution is provided with a token, which says how many transfers have been performed or, equivalently, how many agents have participated in building this solution. When the token related to a solution reaches the given threshold, this solution is sent to the antigen which determines the final result. The algorithm is presented below. The SEQ (sequential) construct causes all of the following processes indented by "-" to execute in the listed order, the PAR (parallel) statement defines a set of processes which execute concurrently (or in parallel).

Algorithm DIA(N, Ns, n, T, M, S)

N - population size
Ns - number of specializations
n - number of lymphocytes a tentative solution is directly transferred to, $n \leq N$ (optionally, the number of transfers is randomly selected from the set 1, 2, ... , n in each iteration)
T - threshold value of the token (it determines the number of main iterations)
M - number of iterations in the mutation process
S - number of solutions used by the antigen in determining the final estimation of the global optimum, $S \leq n^{T-1}$

Initialization

- Create a population of N lymphocytes. Each lymphocyte has its individual, randomly generated initial solution x_{mem}.
- Decompose the population into Ns sub-populations, each containing specialists of the same type.
- Create the antigen with an initial solution (the starting point of the algorithm), provided with the token = 0.

PAR

- *Antigen*
  SEQ
  - **send**(starting point, token, value of the objective function, to a randomly chosen lymphocyte)
  - **receive**(S solutions with related values of the objective function, from lymphocytes), determine the best solution
  - **broadcast**(end)
- *Lymphocytes*
  PAR
  - **if received**(end, from the antigen) **then** terminate the computation

- **if received**(solution, token, value of the objective function, from a lymphocyte or from the antigen) **then** place the triple [solution, token, value of the objective function] into FIFO
- **while** FIFO not empty **than do**
  SEQ
  - take [solution, token, value of the objective function], token ← token +1
  - x_{mut_a} ← solution, perform the mutation process on the received solution:
    for $m = 1$ **until** M **do**
    x_{mut_old} ← x_{mut_a}, modify x_{mut_old} accordingly to the specialization of the lymphocyte, if the result is better, retain it as x_{mut_a}
    end
  - x_{mut_b} ← x_{mem}, perform the mutation process on x_{mem}:
    for $m = 1$ **until** M **do**
    x_{mut_old} ← x_{mut_b}, modify x_{mut_old} accordingly to the specialization of the lymphocyte, if the result is better retain it as x_{mut_b}
    end
  - set x_{mem} ← better of two solutions x_{mut_a}, x_{mut_b}.
- **if** token < T **then**
  send(x_{mem}, token, value of the objective function, to n randomly chosen lymphocytes)
  else
  send(x_{mem}, value of the objective function, to the antigen).

The lymphocytes can be identified with distributed autonomous agents that try to cooperatively solve a decision problem. The antigen can be defined as the separate entity or its function can be assigned to a lymphocyte. Looking for similarities of the proposed algorithm with the clonal selection, we can consider the receivers as the clones of the sender. The process of improving the solution by successive groups of lymphocytes can be compared with the creation of the generations of lymphocytes. Furthermore, we can speak about mutual stimulation, because the lymphocyte that sends its result to other lymphocytes, influences (stimulates) them. The lymphocyte sends and stores its best solution. To find this solution it uses the stored result from the previous iteration. This is comparable with storage and cloning of the best lymphocytes, important for the convergence of the algorithm based on clonal selection [9]. Furthermore, such a procedure extends the search region and makes it possible to escape from local optima. Mutation performed on x_{mem} allows to explore a local area around it. Because mutations with lower values of the objective function J are lost, the x_{mut_b} tends to go up the hill, leading to a local optimum. Occasionally, the solution received from other lymphocyte will be on the side of the hill where the climbing region (thus the mutation process performed on x_{mut_a}) is more promising, which means that it will lead to the global solution. The loops concerning x_{mut_a} and x_{mut_b} are independent and they do not have to be performed sequentially. The parameter S of the algorithm requires a comment. If the number n of transfers remains constant in each iteration the user can set S equal to or less than n^{T-1}. The second alternative may be useful when communication failures are

considered. Similar situation arises if the number of transfers is randomly selected from the set $1, 2, ..., n$ in each iteration. Alternatively, one can modify the algorithm introducing a parameter representing the maximal allowed computation time. Independently, one can allow to stop computations after the antigen receives satisfactory solution.

3 Evaluation of the algorithm

In this chapter we show the results of experiments. First we compare the algorithm DIA with CLONALG [2] and with the modified CLONALG proposed in [11], with respect to finding the global optima of some typical multimodal test functions. Next we discuss the influence of the parameters of the DIA. At the end we apply the algorithm to the 100 European cities TSP. The experiments have been performed using an agent-based system and the MadKit platform [4].

3.1 Finding global optima of multimodal functions

We consider three multimodal functions which turn out to be difficult for any search algorithm because they have numerous peaks [10].

$$F_A(x) = x_1 sin(4\pi x_1) - x_2 sin(4\pi x_2 + \pi) + 1, x \in [-1, 2] \quad (1)$$

$$F_B(x) = An + \sum_{i=1}^{n} (x_i^2 - Acos(2\pi x_i)), A = 10, n = 20, x \in [-5.12, 5.12] \quad (2)$$

$$F_C(x) = \alpha \sum_{i=1}^{n} \frac{x_i^2}{4000} + i - \prod_{i=1}^{n} cos(\frac{x_i}{\sqrt{i}}), \alpha = 0.1, n = 10, x \in [-600, 600] \quad (3)$$

Function F_A of two variables has the unique global maximum equal to 4.254. Function F_B is the Rastrigin function of 20 variables, F_C is the Grievank function of 10 variables. Both functions take their unique global minima equal to zero at the origin of the system of coordinates.

The mutation process in the DIA is performed only on decision variables determined by the specialization of the lymphocyte. In our experiments the specialists in the i-th of Ns groups modify the i-th set of p decision variables, where $p =$ (number of decision variables)/Ns. First, a decision variable is randomly selected from the set assigned to the specialist. Then one of two mutation variants is selected with the probability 0.5. In the first variant the result of the mutation is drawn from the domain $[lb, ub]$ of the decision variable, in the second variant the result is taken as $\min(ub, lb + (v - lb) * l)$ where v is the actual value of the decision variable and l is randomly selected from [0, 2]. Other parameters used in the experiments are given in table 1. Table 2 compares the results obtained with three algorithms: (i) DIA, (ii) CLONALG [2], and (iii) modified CLONALG [11] denoted here as MCLONALG.

As we can see, the results obtained using the DIA outperform the remaining results. They are closer to the theoretical solutions in the sense of the best value, as well as in the sense of the mean and dispersion of the best results obtained in repeated executions of the algorithm. Moreover the DIA uses significantly fewer tentative solutions to find the final result.

Table 1. Parameters of the DIA used in the experiments

Function	N	Ns	n	T	M	S
F_A	4	2	2	4	40	8
F_B	5	5	2	8	100	128
F_C	5	5	2	10	100	512

Table 2. Comparison of the results obtained with various immune algorithms

Function	Algorithm	Number of execution	Average number of generated tentative solutions	The maximal value of the objective function	The minimal value of the objective function	The mean of the best values of the objective function	Standard deviation of the best values of the objective function
F_A	**DIA**	**30**	**1200**	**4.25379**	**3.98556**	**4.22618**	**0.05114**
	CLONALG	30	50100	4.13	3.209	3.711	0.281
F_B	**DIA**	**30**	**51000**	**0.00226**	**1e-15**	**0.00018**	**0.00055**
	CLONALG	10	800100	35.0729	12.67206	22.72131	6.30522
	MCLONALG	10	not reported	not reported	0.00646	0.066525	not reported
F_C	**DIA**	**30**	**204600**	**0.00420**	**0.00009**	**0.00130**	**0.00120**
	CLONALG	10	800100	0.07838	0.01205	0.0408	0.01752
	MCLONALG	10	not reported	not reported	0.01667	0.020942	not reported

Influence of the parameters of the algorithm In the DIA the best solution is determined by the antigen on the basis of the solutions received from lymphocytes. These solutions arise as the effect of the process in which many tentative solutions are generated. The number of solutions generated by the DIA is equal to $2 * M * n^{T-1}$. So it is proportional to the number of mutations executed by the lymphocyte during the main iteration and to the $(T-1)$-th power of the number of lymphocytes a tentative result is transferred to. Increasing the values of the parameters M, n, T, we expect that the final solution will be closer to the global optimum. Exemplary relations concerning the function F_C are shown in fig. 1.

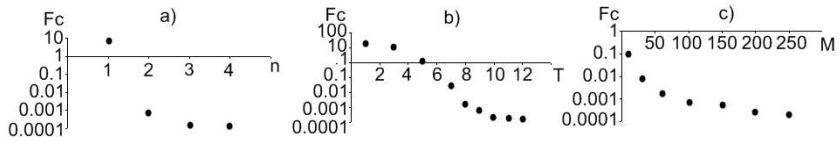

Fig. 1. The minimal value of the objective function F_C (averaged on the base of 10 executions of the algorithm) vs.: a) n, for $M = 100$, $T = 9$, b) T, for $M = 100$, $n = 2$, c) M, for $n = 2$, $T = 9$

Fig. 2. illustrates how the performance of the algorithm is influenced by the parameter S.

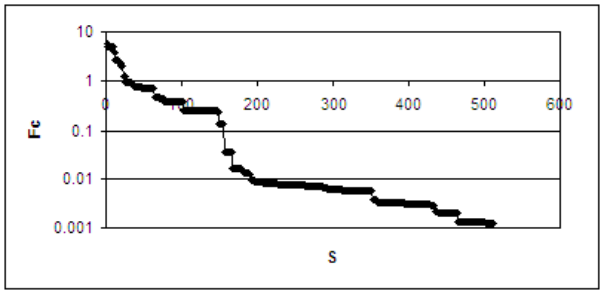

Fig. 2. The minimal value of the objective function F_C (averaged on the base of 30 executions of the algorithm) vs. the number S of successive solutions received by the antigen and used by it in determining the actual estimation of the optimum ($n = 2$, $M = 100$, $T = 10$)

3.2 TSP

In this known NP-hard problem a collection of cities resides on a plane, and we have to find the shortest tour starting in some city, visiting every other city exactly once, and returning to the starting point. In our experiments the tour of the salesperson is represented as the integer valued vector of numbers representing the cities. The population of the lymphocytes is divided into groups of specialists. All specialists in the group use the same method of mutation. The following known variants have been used [1,7]: (1) *switching* - randomly selecting two cities and switching them in the tour, (2) *translation* - removing a section of the tour and then replacing it in between two randomly selected

consecutive cities, (3) *inversion* - removing a section of the tour and then replacing it with the same cities running in the opposite order, (4) combination of the methods (2) and (3).
The DIA algorithm has been tested on the 100 cities TSP [1]. The following values of the parameters have been used: $N = 15$, $Ns = 5$ (each group with 3 specialists, translation of single city is considered as the fifth specialization), n = randomly selected from the set 1, 2, ... , 15 in each iteration, $T = 16$. The length of the shortest route is equal to 21134 km. After 15 executions of the DIA the following results have been obtained: the best result = 21224 km (i.e. 0.43% worse than the global optimum), the mean and the standard deviation of the best solutions from each execution: 21691 km (i.e. 2.64% worse than the global optimum), and 398 km, respectively. Fig. 3. illustrates how the performance of the algorithm is influenced by the parameter S.

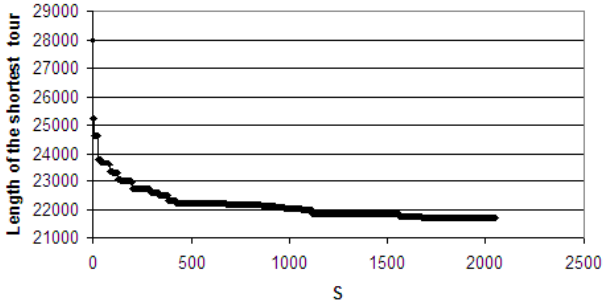

Fig. 3. The minimal length of the tour (averaged on the base of 15 executions of the algorithm) vs. the number S of successive solutions received by the antigen and used by it in determining the actual estimation of the optimum

The same problem has been solved with the slightly modified CLONALG. As opposed to the original version of CLONALG only the best lymphocytes are cloned. The lymphocytes representing tours that are more than α percent longer than the best solution in the population are removed. The number of clones of the lymphocyte remains constant, but the population size changes from generation to generation, because it depends on the numbers of the best and of the worst lymphocytes in the previous generation. All lymphocytes use identical method of mutation, determined at the start of the algorithm (no specialization). The tests have been performed with the following parameters: population size in the first generation = 2, number of generations = 2000, number of clones produced by the lymphocyte = 220, $\alpha = 0.2$. The computations have been executed 10 times for each mutation method. The best results are as follows: (1) for translation 22249 (the best result in the experiment, 5.27% worse than the global optimum), (2) for combination of inversion and translation 22424, (3) for inversion 22924, (4) for switching 28976, (5) the mean of the 10 results 23244 (obtained for translation, 10% worse than the global optimum) - with the standard deviation 796.
The results obtained with the DIA are significantly better although this algorithm generated fewer tentative solutions. The DIA uses specialization. So the result obtained by a lymphocyte is improved by agents with other specializations. Moreover, in CLONALG

the best solution is not improved until the next generation. In the DIA it is directly sent for improvement to other lymphocytes. The additional important factor influencing the quality of the final result is that the best tentative solution generated by the lymphocyte is stored in its memory. After receiving a solution from other agent the lymphocyte tries to improve two solutions, i.e. the stored and the received. The better of two solutions is retained in the local memory and transferred to other lymphocytes.

4 Conclusion and further work

A new immune-based algorithm DIA for solving optimization problems has been proposed. The algorithm has been compared with two known immune-based algorithms CLONALG [2] and modified CLONALG [11]. The results concerning optimization of the typical multimodal test functions as well as the 100 cities TSP outperform the related solutions obtained with the remaining methods. They are closer to the exact solutions in the sense of the best value, as well as in the sense of the mean and dispersion of the best results obtained in repeated executions of the algorithm. Moreover, these results required fewer tentative solutions.
The DIA uses specialization. So the result obtained by a lymphocyte is improved by agents with other specializations. Moreover, in CLONALG the best solution is not improved until the following next generation. In the DIA it is directly sent for improvement to other lymphocytes. The additional important factor influencing the quality of the result is that the lymphocyte retains its tentative best solution. After receiving new solution, it tries to improve both results. Then the better solution is retained in the local memory and transferred to other lymphocytes. This procedure helps to escape from local optima.
The algorithm is faster than the remaining two algorithms, even if comparable numbers of tentative solutions are generated. This is because the lymphocytes in the DIA are focused on local improvement of its solutions and thus the elements of the population are not ranked. The lymphocytes in the DIA are not replaced with new individuals. They enrich their knowledge by the information from other lymphocytes. Thus they can generate good solutions based on communication with other lymphocytes, without direct contact with the antigen. This resembles the idiotypic network of Jerne [5].
The lymphocytes in the DIA can be identified with distributed autonomous agents (e.g. robots, controllers) that cooperatively solve a decision problem in a decentralized manner. In further research we will use the algorithm for energy management via negotiation among networked intelligent appliances.

References

1. Aarts, E., Korst, J.: Simulated Annealing and Boltzmann Machines. John Willey & Sons (1989).
2. De Castro, L. N., Von Zuben, F. J.: Learning and Optimization Using the Clonal Selection Principle. *IEEE Transactions on Evolutionary Computation*, (2002) 3:239–251.
3. Carter, J. H.: The Immune System as a Model for Pattern Recognition and Classification. *Journal of the American Medical Informatics Association*, (2000) 1:28–41.

4. Gutknecht, O., Ferber, J., Michel, F.: The MadKit Agent Platform Architecture. *Rapport De Recherche*, LIRM, Universite Montpellier, France (2000).
5. Jerne, N. K.: The Immune System, *Scientific American*, (1973) vol. 229, No. 1, 52-60.
6. Lau, H. Y. K., Wong, V. W. K.: An Immunity-Based Distributed Multiagent-Control Framework. *Transactions on Systems, Man, and Cybernetics part A: Systems and Humans*, (2006) 1:91–108.
7. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolutionary Programs, Springer Verlag, Berlin, Heidelberg (1996).
8. Sathyanath, S., Sahin, F.: AISIMAM An Artificial Immune System Based Intelligent Multi Agent Model and its Application to a Mine Detection Problem. *Proc. ICARIS*, (2002).
9. Villalobos-Arias, M., Coello Cello, C. A., Hernandez Lerma, O.: Convergence Analysis of Multiobjective Artificial Immune Algorithm. *Proc. ICARIS* (2004) 226–235.
10. Wierzchon, S. T.: Artificial Immune Systems. Theory and Applications. AOW EXIT, Warszawa(in Polish) (2001).
11. Wierzchon, S. T.: Multimodal optimization with artificial immune system. *In: Klopotek, M. A., Michalewicz, Z., Wierzchon, S. T. (eds) Intelligent Information Systems*. Physica-Verlag (2001).