

An Artificial Immune System with Partially Specified Antibodies

Ramin Halavati, Saeed Bagheri Shouraki, Mojdeh Jalali Heravi, Bahareh Jafari Jashmi
Computer Engineering Department, Sharif University of Technology, Tehran, Iran
halavati@ce.sharif.edu, bagher-s@sharif.edu, m_jalali@ce.sharif.edu, b_jafari@ce.sharif.edu

ABSTRACT

Artificial Immune System algorithms use antibodies which fully specify the solution of an optimization, learning, or pattern recognition problem. By being restricted to fully specified antibodies, an AIS algorithm can not make use of schemata or classes of partial solutions. This paper presents a symbiotic artificial immune system (SymbAIS) algorithm which is an extension of CLONALG algorithm. It uses partially specified antibodies and gradually builds up building blocks of suitable sub-antibodies. The algorithm is compared with CLONALG on multimodal function optimization and combinatorial optimization problems and it is shown that it can solve problems that CLONALG is unable to solve.

Categories and Subject Descriptors

I.2. [Artificial Intelligence]: Genetic Algorithms

General Terms

Algorithms

Keywords

Artificial Immune System, Optimization, Symbiogenesis.

1. INTRODUCTION

Over the last few years, there has been an ever increasing interest in the area of Artificial Immune Systems and their applications in pattern recognition and optimization such as [5], [8], [9], [20], [22], [23], [35], [36] and many more.

Although the original idea of AIS is taken from vertebrates' immunity system which has several hundred million years of evolution on its back, it has been shown that augmenting it with ideas that distance it from its natural form sometimes increases its performance on digital computers [7], [10], [11] and [23].

One such idea can be taken from the comparison of AIS with Symbiotic Evolution [26], [27] as follows: To solve a problem using AIS, the generic solution is coded as an antibody and the system gradually matures its antibodies to find the best possible solution(s). During the maturation process, the antibodies are

always assumed as rivals and the only cooperation between antibodies happens in cross-reactive responses [1], [19], [25], [30] and [33]. In contrast to this minimal cooperation, in symbiotic evolution, each individual represents only one part of the solution and the combination of these partial solutions make up a complete answer. This idea is used in symbiotic based optimization and learning such as [38], [39] and genetic algorithms which deal with linkage problem [18] such as messy genetic algorithms [12], [17], and cooperative coevolutionary algorithms [28].

Based on this idea, this paper presents an artificial immune system with partially specified antibodies, in which the antibodies have a cumulative effect on the antigens and through some symbiotic combinations, complete antibodies emerge. This process is presented in the next sections as follows: Section 2 presents a brief introduction of CLONALG algorithm [10] as our base AIS algorithm. Section 3 introduces our algorithm, Symbiotic AIS. Section 4 will be on experimental results and comparisons and at last come the conclusions and future steps.

2. CLONALG ALGORITHM

The immunological process has been used for inspiration in AIS in several general purpose algorithms such as negative selection algorithm [16], positive selection algorithm [29], clonal selection algorithm [10], continuous immune models [14], [37] and discrete immune network models [11], [34] and many special purpose contributions such as multi-objective optimization [2], [3], [4], [24] and [40] and multimodal optimization [15], [31], [32].

The most common abstraction in clonal selection algorithms is CLONALG [10]. The authors demonstrated empirically that this algorithm is capable of learning a set of input patterns by selecting, reproducing and mutating a set of "artificial immune cells". In [10] the authors showed the suitability of the algorithm for multimodal search and presented empirical results where it could outperform a fitness sharing strategy. All the steps involved in CLONALG are also seen in an evolutionary algorithm, allowing it to be characterized as an evolutionary algorithm inspired in the immune system. Note that there is an important conceptual difference between the clonal selection algorithm and an evolutionary algorithm. In the former, the theory of evolution is used to explain the behavior of the system, while in the latter it inspired its development [6].

CLONALG starts by generating a population of N antibodies, each specifying a random solution for the optimization process. In each iteration of the algorithm, some percentage of the best existing antibodies are selected, cloned and mutated to construct a new candidate population. All new members are evaluated and a certain percentage of the best members are added to the original population. At last, a percentage of worst members of previous generation of antibodies are replaced with new randomly created ones.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'07, July 7–11, 2007, London, England, United Kingdom.

Copyright 2007 ACM 978-1-59593-697-4/07/0007...\$5.00.

The main idea of the algorithm is depicted in Figure 1. In this sample, the optimum solution is a 7-sided star with chessboard fill pattern. The initial population includes a six sided star with black fill pattern and a square with chessboard fill along with two other less similar shapes. Among these 4 initial antibodies, the most similar ones are selected in step 2 to be cloned and mutated in step 3, producing 6 other shapes. The most similar shapes of the clones are selected in section 4 and the next generation is selected from the combination of these shapes and previous generation. Figure 2 presents the diagram of this process.

As stated above, the antibodies of CLONALG algorithm specify all properties of a possible solution. For example, in Figure 1's sample, an antibody expresses both the shape and the filling pattern of the solution. Using this representation, if an antibody includes a good property surrounded by bad properties, it will have a low affinity value and low chance of selection, cloning, and mutation. But if the antibodies would be allowed to have unspecified properties, for example an antibody just represents the shape and no filling pattern in Figure 1 example, a good property has a higher chance of selection and replication for the next generation, as it can combine with other antibodies which specify good values for other properties. This idea is similar to the natural process of Symbiogenesis [26], [27] in which individuals merge to compose bigger and more powerful organisms. The next section will produce Symbiotic Artificial Immune System that combines AIS and Symbiosis ideas together.

3. SYMBIOTIC ARTIFICIAL IMMUNE SYSTEM ALGORITHM (SYMBAIS)

To add the idea of symbiosis to CLONALG, the new algorithm uses partially specified antibodies as follows:

Initiation: The algorithm starts with a set of partially specified antibodies, each having just one specified property.

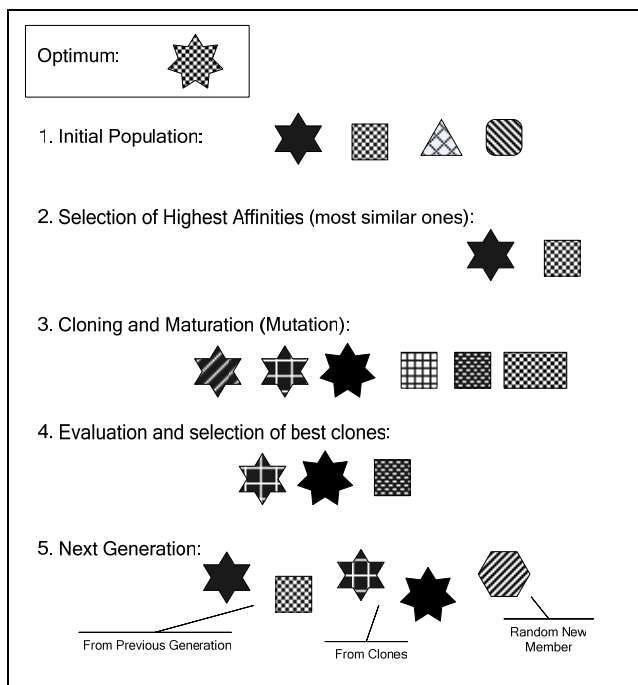


Figure 1: A sample for the main idea of CLONALG.

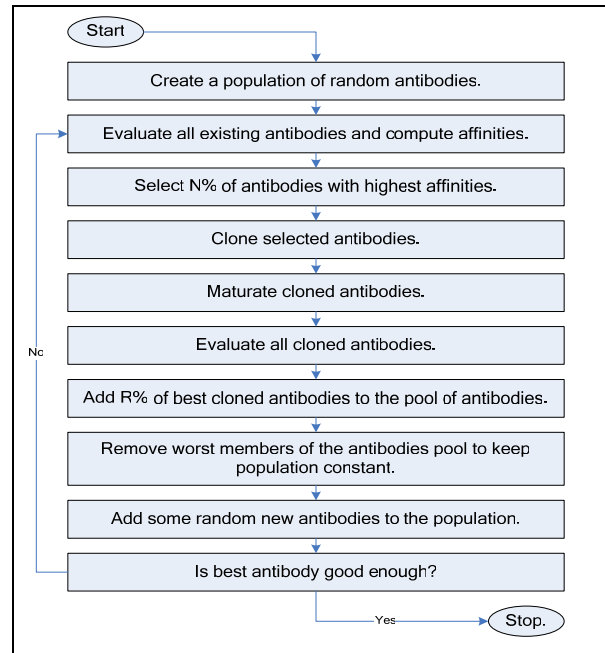


Figure 2: Diagram of CLONLG optimization algorithm

Evaluation: Partially specified antibodies may not have all required data to be evaluated as a solution. Therefore, we must build an assembly of several antibodies to make a complete solution. To do so, the algorithm picks a random antibody and creates a new assembly using it. As long as the assembly does not specify all required properties, other antibodies are randomly selected from the population so that they have no value for the specified properties of the assembly, we will call each member of an assembly a symbiont. There exist situations in which an antibody can not be completed using currently existing antibodies, for example if all antibodies that specify a certain position would have conflicts with the generated assembly. In such cases, we create some antibodies with random values for all missing positions of the created assembly and add them to the population.

Selection: Once an assembly is completed, its affinity value can be computed. To select some antibodies to be cloned, we generate a number of assemblies and pick a certain percent of the best assemblies for cloning and mutation.

Cloning and Mutation: Similar to CLONALG, the cloning rate of each selected assembly is computed using its affinity value and its affinity rank by equation 1. Once an assembly of antibodies is cloned, two symbionts of each clone are randomly chosen and merged to form a bigger antibody. Then, all symbionts (including the merged ones) enter the mutation phase. Mutation is quite similar to CLONALG.

$$\left(\frac{\text{Affinity}}{(1+i) \times \sum_{\text{allSelectedAntibodies}} \text{Affinity}} \right) \times \text{Clones_Count} \times \text{Assemblies_Count}$$

Equation 1: The cloning rate for the i^{th} best assembly.

To avoid fast emergence of fully specified antibodies, we limit the maximum size of possible antibodies to a parameter that gradually

increases from the beginning of the process. This will be called cooling process.

Moving to the Next Generation: As the last step, all cloned and maturated antibodies are added to the pool. Also, all symbionts of a certain percent of worst assemblies are removed from the pool, except those which also have been a member of the cloned assemblies. At last, if the population goes above a certain level, some antibodies are randomly killed to control the population.

The entire process is depicted in Figure 3 as a pseudo code.

PARAMETERS:

- AC** (Assemblies Count): The number of assemblies which are created in each iteration for evaluation.
 - SP** (Selection Percentage): The percent of assemblies that are chosen for cloning.
 - CC** (Clones Count): The number of clones created from the chosen assemblies.
 - MR** (Mutation Rate): The mutation possibility of each bit of a cloned assembly.
 - RP** (Replacement Percentage): The Percent of worst assemblies that are removed from the population.
 - CS** (Cooling Speed): The convergence speed of the bigger antibodies.
1. Initialize the population with antibodies having just one specified property.
 2. Repeat until stopping condition is met:
 - 2.1. Create **AC** assemblies from antibodies: For each assembly, randomly choose one antibody from the population and initialize the assembly by that. As long as the assembly does not have antibodies that specify all required properties, randomly choose antibodies from the population so that they won't have any common specified properties with former members of the assembly. If an assembly can not be completed using currently existing antibodies, create some antibodies with random values for all missing positions. Call each antibody of an assembly a symbiont.
 - 2.2. Compute the affinity value of all assemblies.
 - 2.3. Choose **SP** percent of the best assemblies for cloning and maturation. For each selected assembly (the i^{th} one from best to worst):
 - 2.3.1. Compute the number of clones for this assembly by Equation 1, call it **N**.
 - 2.3.2. Create **N** clones, for each clone:
 - 2.3.3. Mutate the clone with $\text{MR} \times (1 - \text{Affinity of the Clone})$ probability.
 - 2.3.4. Randomly choose two symbionts of the clone. If their size is smaller than $\text{CS} \times \text{Current_Generation_Number}$, combine them.
 - 2.3.5. Add all symbionts of the clone to the antibodies population.
 - 2.4. Remove all symbionts of the worst **RP** percent of assemblies from the pool, except those that have taken part in the cloned assemblies as well.
 - 2.5. If there is more than one copy of any antibody in the pool, remove it.
 - 2.6. If the population of antibodies is above the maximum allowed limit, randomly remove some of the antibodies.
-

Figure 3: Pseudo Code of Symbiotic Artificial Immune System (SymbAIS) algorithm

4. WHY SYMBIOTIC-AIS?

Symbiotic AIS algorithm performs its search process using partially specified antibodies. The main advantage of this algorithm in compare with its base algorithm, CLONALG, is in adding schemata [21] to AIS approach: the partially specified antibodies can represent classes (schemata) of good solutions. Combination of these partial solutions can be a larger step towards optimum solutions in compare with small mutations of conventional AIS. This is most useful in problems that can be broken into several sub-problems, so that the good solutions of

these sub-problems may compose a general good solution for the main problem.

By using the formerly specified *Cooling* parameter, the process is limited to try small antibody building blocks first and search for good solutions using them and gradually build more specified antibodies. After the *Cooling* process is completed (the maximum allowed size of antibody exceeds the size of a fully specified antibody), fully specified antibodies are created and the process converges to a normal CLONALG algorithm. Therefore, if there would be a possibility of finding a solution by combination of smaller building blocks, Symbiotic AIS may find it faster than CLONALG. Also, if there will be a situation in which gradual moving towards the global optimum would not be possible, for example due to the surrounding valleys of bad solutions around the global pick, Symbiotic AIS may find schemata whose combination can find the global optimum. If none of these cases happen, the algorithm becomes a conventional CLONALG at last. Hence, it can be concluded that Symbiotic AIS is a more general approach in compare with CLONALG and can at least find all solutions that CLONALG can find.

5. EXPERIMENTAL RESULTS

We compared SymbAIS algorithm with CLONALG on two problem sets. The first set was the optimization functions taken from [10] and the second set was concatenation of multiple 8-Queen problems [13].

In the first set, we had 3 functions from [10] and created 3 more by combinations of some of them, they are all presented in Equation 2. In all cases, each variable was coded using 22 bits as specified in [10]. All control parameters were also taken from [10] for all functions that were used there.

$$\begin{cases} f(x) = \sin^6(5\pi x) \\ g(x) = 2^{-2((x-0.1)/0.9)^2} \sin^6(5\pi x) \\ h(x, y) = x \cdot \sin(4\pi x) - y \cdot \sin(4\pi y + \pi) + 1 \\ f2(x, y) = f(x) \times f(y) \\ g3(x, y, z) = g(x) \times g(y) \times g(z) \\ h2(x, y, z, t) = h(x, t) \times h(y, z) \end{cases}$$

Equation 2: The Functions used for Benchmarking

In the second set, we used a concatenation of multiple instances of the well known 8-Queen combinatorial optimization problems. In each instance, M separate problems of putting 8 queens on an 8x8 chessboard must be solved, so that no two queens on a board can attack each other. The antibody includes the rows of the queens and columns are all assumed distinct and fixed.

Tables 1 and 2 present the parameters we used for each problem by CLONALG and SymbAIS algorithms. Each problem has been tested 10 times and the number of runs which resulted in global maxima and the average time taken to reach the global maxima are depicted. All algorithms were implemented, compiled, and optimized using Microsoft VC++ 7.0 and run on Pentium 4, 3.0 GHZ PCs under Windows XP operating system.

As it is depicted in Table 1 and Figures 5 and 6, CLONALG could solve the first 4 functions in all trials, faster than SymbAIS, but failed to solve the last 2 functions (g3 and h2) while SymbAIS solved those two as well. Thus, noting that the number of function parameters gradually increase from the first to the last function, it

can be seen that SymbAIS has been more successful than CLONALG on problems with more parameters. Figure 7 also depicts the performance log of the two algorithms on g3 function. As it is represented there, CLONALG has found a near optimal solution in a few seconds while SymbAIS has found that solution in about 500 seconds, but CLONALG has not been able to leave that local maximum and has stayed there for the rest of time, while SymbAIS has proceeded to the global optimum.

Figures 8 and 9 present similar results for Multiple 8-Queen Problems. As depicted there, again CLONALG has been faster than SymbAIS on smaller problems (1 and 5 boards) but it has totally failed on bigger problems (10 and 15 boards) while SymbAIS has successfully solved all.

Table 1. Parameters of CLONALG for each problem set. f, g, h, f2, and g3 are functions stated in Equation 1; Q-1..Q-15 are instances of Multiple 8-Queen problems, respectively 1 board to 15 boards. The empty cells represent experiments in which we could not find a set of parameters that will result in a complete solution.

Problem	f, g	h	f2	g3, h2	Q-1	Q-5	Q-10	Q-15
N_{gen}	50	50	500		500	1000		
n	50	100	100		50	50		
N	50	100	100		300	300		
d	0.0	0.0	0.0		0.36	0.36		
β	0.1	0.1	0.1		0.05	0.34		

Table 2. Parameters of SymbAIS for each problem set. f, g, h, f2, and g3 are functions stated in Equation 1; Q-1..Q-15 are instances of Multiple 8-Queen problems, 1 board to 15 boards, respectively.

Problem	f, g	h	f2	g3, h2	Q-1	Q-5	Q-10	Q-15
AC	20	25	15	25	25	25	25	25
CS	4E-4	4E-4	4E-4	4E-4	4E-4	4E-4	4E-4	4E-4
SP	0.1	0.1	0.1	0.2	0.2	0.2	0.2	0.2
CC	1.5	1.5	1.4	1.8	1.7	1.5	1.5	1.5
MR	1.0	1	0.9	1	0.1	0.1	0.2	0.2
RP	0.3	0.2	0.4	0.2	0.5	0.5	0.4	0.4

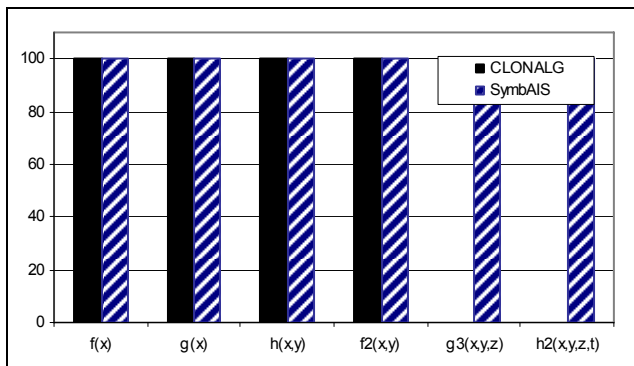


Figure 5. Success Rate Comparison of SymbAIS and CLONALG on Function Optimization problems. Vertical Axis: Percentage of Success, Horizontal Axis: Problem size.

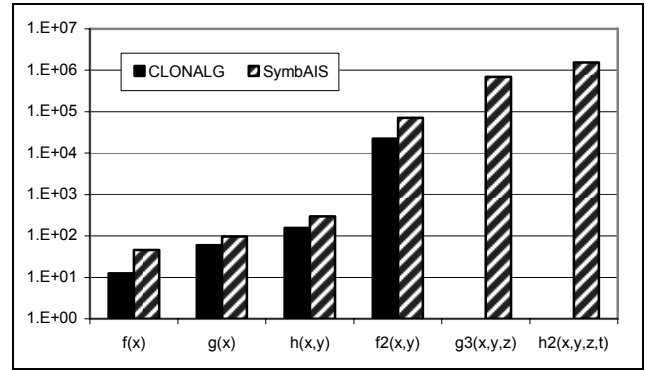


Figure 6. Performance Comparison of CLONALG and SymbAIS on Function Optimization problems. Vertical Axis: Time in Seconds, Horizontal Axis: Function.

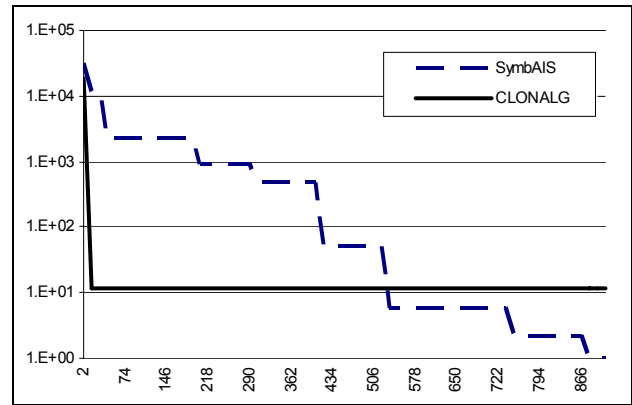


Figure 7. Performance Comparison of CLONALG and SymbAIS on g3 Function Optimization Problem. Vertical Axis: (1-Affinity)*10⁻⁶, Horizontal Axis: Time in seconds.

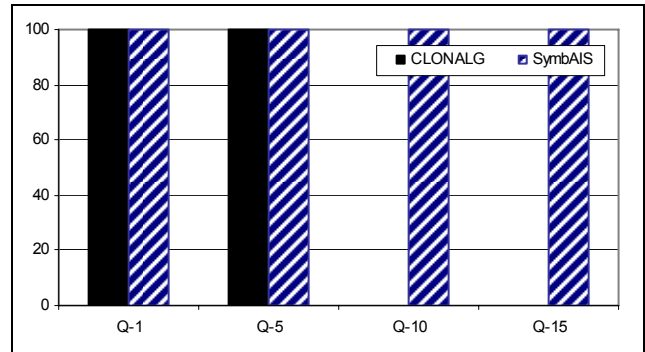


Figure 8. Success Rate Comparison of SymbAIS and CLONALG on Multiple 8-Queen problems. Vertical Axis: Percentage of Success, Horizontal Axis: Number of Boards

6. CONCLUSIONS AND FUTURE WORKS

We introduced Symbiotic Artificial Immune System (SymbAIS) as a general purpose optimization algorithm which is a more general version of CLONALG algorithm. SymbAIS performs its search using partially specified antibodies and gradually builds up building blocks of the classes of possible solutions till it reaches fully specified antibodies.

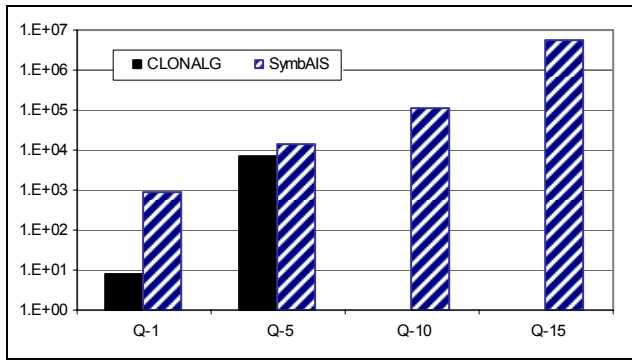


Figure 9. Performance Comparison of SymbAIS and CLONALG on Multiple 8-Queen problems. Vertical Axis: Time in Seconds, Horizontal Axis: Number of Boards

In the worst case, if SymbAIS would not be able to find the global maximum while the antibodies are partially specified, when the algorithm is cooled enough it continues the search with fully specified antibodies, quite similar to CLONALG. Therefore, it can be stated that SymbAIS is at least able to find all solutions that CLONALG can find.

We compared the two algorithms on two test sets. The first one was 6 optimization functions and the second one was concatenations of multiple instances of 8-Queen problems. As presented in section 5, SymbAIS had higher success rates in compare with CLONALG but the longer times to reach the best answer in compare with CLONALG in cases that CLONALG has also found the answer.

Being slower can be a direct result of being limited to search through partial answers at the beginning of the process (Cooling Limitation) but better results on bigger problems show that at this cost, we can expect a higher chance of finding global optima.

7. ACKNOWLEDGEMENTS

This research was in part supported by Iranian Telecommunication Research Center by grant number T/500/5987. Our sincerest thanks to Prof. Caro Lucas for his brilliant comment throughout this task.

8. REFERENCES

- [1] Ada G. L., and Nossal G. The Clonal Selection Theory. *Scientific American*, 257, 2, 1987, 50-57.
- [2] Coello C., Cortes N. Solving Multiobjective Optimization Problems Using an Artificial Immune System. *Genetic Programming and Evolvable Machines*, 6, 2 (June 2005), 163-190.
- [3] Cort'es N. C., and Coello Coello C. A. Multiobjective Optimization Using Ideas from the Clonal Selection Principle. *GECCO 2003*, Chicago, IL, USA
- [4] Cui X., Li M., and Fang T. Study of population diversity of multiobjective evolutionary algorithm based on immune and entropy principles. in *Proceedings of the Congress on Evolutionary Computation 2001 (CEC'2001)*, IEEE Service Center: Piscataway, New Jersey, Vol.2, (May 2001), 1316–1321.
- [5] Dasgupta D. *Artificial Immune Systems and Their Applications*. Ed., Springer-Verlag, 1999.
- [6] de Castro, L.N., and Timmis J., An Artificial Immune Network for Multimodal Optimisation. In *Proceedings of the Congress on Evolutionary Computation. Part of the 2002 IEEE World Congress on Computational Intelligence.*, Honolulu, Hawaii, USA. IEEE, 699-704.
- [7] de Castro, L.N., and Timmis J. Artificial Immune Systems as a Novel Soft Computing Paradigm. *Soft Computing*, 2003.
- [8] de Castro L. N., and Von Zuben F. J. *Artificial Immune Systems: Part I – Basic Theory and Applications*, EEC/Unicamp, Campinas, SP, Tech. Rep. – RT DCA 01/99, p. 95. 1999.
- [9] de Castro, L. N., and Von Zuben, F. J. *Artificial Immune Systems: Part II – A Survey of Applications*. Tech. Rep. – RT DCA 02/00, p. 65. 2000.
- [10] de Castro, L. N., and Von Zuben, F. J. Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, 6, 3, 2002, 239–251.
- [11] de Castro, L. N., and Von Zuben, F. J. aiNet: An Artificial Immune Network for Data Analysis. *Data Mining: A heuristic Approach*, Chapter XII, 2002, 231-259.
- [12] Deb, K. *Binary and floating point function optimization using messy genetic algorithms* (IlliGAL Report No. 91004). Urbana: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, 1991.
- [13] Eiben, A.E., Raué, P.E. and Ruttkay, Z. GA-easy and GA-hard Constraint Satisfaction Problems. *Constraint Processing*, Manfred Meyer, Eds. Springer-Verlag LNCS 923, 1995, 267-283.
- [14] Farmer, J., Packard, N., and Perelson, A. The immune system, adaptation and machine learning. *Physica D*, 22:187–204.
- [15] Forrest, S., and Perelson, A. Genetic algorithms and the immune system. in *Parallel Problem Solving from Nature*, H.-P. Schwefel and R. Manner Eds. Lecture Notes in Computer Science, Springer-Verlag: Berlin, Germany, 1991, 320–325.
- [16] Forrest, S., Perelson, A., Allen, L., and Cherukuri, R. (1994). Self-nonsel self discrimination in a computer. In *Proceedings of 1994 IEEE Symposium on Research in Security and Privacy*. 1994, 132–143.
- [17] Goldberg, D.E., Korb, B., and Deb, K. Messy Genetic Algorithms: Motivation, analysis, and first results. *Computer Systems*, 3, 5, 1989, 493-530.
- [18] Harik, R. *Learning Gene Linkage to Efficiently Solve Problems of Bounded Difficulty Using Genetic Algorithm*. Ph.D. Thesis, University of Illinois at Urbana-Champaign, Urbana, Illinois, 1997.
- [19] Hodgkin, P. D. Role of Cross-Reactivity in the Development of Antibody Responses. *The Immunologist*, 6, 6, 1998, 223-226.
- [20] Hofmeyr, A., and Forrest . Immunity by Design: An Artificial Immune System. in *Proc. of GECCO'99*, 1999, 1289-1296.
- [21] Holland, J.H. *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press, 1975.

- [22] Hunt, J., and Cooke, D. An adaptive and distributed learning system based on the Immune system. *Proceedings of IEEE International Conference on Systems Man and Cybernetics (SMC)*. IEEE, 1995, 2494-2499.
- [23] Hunt, J. E., and Cooke, D. E. Learning Using an Artificial Immune System. *Journal of Network and Computer Applications*, 19, 1996, 189-212.
- [24] Kurpati, A., and Azarm. Immune network simulation with multiobjective genetic algorithms for multidisciplinary design optimization. *Engineering Optimization*, 33, 2000, 245-260.
- [25] Mason, D. Antigen Cross-Reactivity: Essential in the Function of TCRs. *The Immunologist*, 6, 6, 1998, 220-222.
- [26] Maynard Smith, J., and Szathmary, E. *The Major Transitions in Evolution*, WH Freeman: Oxford UK, 1995.
- [27] Merezhkovsky, K. The Theory of Two Plasms as the Basis of Symbiogenesis, a New Study or the Origins of Organisms. *Proceedings of the Studies of the Imperial Kazan University*, Publishing Office of the Imperial University, (In Russian), 1909.
- [28] Potter, M. A., and De Jong, K. A. A Cooperative Coevolutionary Approach to Function Optimization. in *Parallel Problem Solving from Nature (PPSN III)*, Y. Davidor, H.-P. Schwefel and R. Manner, Eds. Berlin: Springer-Verlag, 1994, 249-257.
- [29] Seiden, P. E. and Celada, F. A Model for Simulating Cognate Recognition and Response in the Immune System. *J. theor. Biol.*, 158, 1992, 329-357
- [30] Smith, D. J., Forrest ., Hightower, R. R., and Perelson, A. Deriving Shape Space Parameters from Immunological Data. *J. Theor. Biol.*, 189, 1997, 141-150.
- [31] Smith, R. E., Forrest ., and Perelson, A. Searching for diverse, cooperative populations with genetic algorithms. *Technical Report TCGA No. 92002*, University of Alabama, Tuscaloosa, AL, 1992.
- [32] Smith, R. E., Forrest ., and Perelson, A. Population diversity in an immune system model: Implications for genetic search. in *Foundations of Genetic Algorithms*, L. D. Whitley, Eds. Morgan Kaufmann Publishers: San Mateo, CA, 1993, vol. 2, pp. 153-165.
- [33] Sprent, J. T and B Memory Cells. *Cell*, 76, 1994, 315-322.
- [34] Timmis, J. *Artificial Immune Systems: A Novel Data Analysis Technique Inspired by the Immune Network Theory*. Ph.D. Dissertation, Department of Computer Science, University of Wales, 2000.
- [35] Timmis J, Knight T., de Castro L. N., and Hart E. An overview of artificial immune systems. In *Computation in Cells and Tissues: Perspectives and Tools for Thought*, R. Paton, H. Bolouri, M. Holcombe, J. H. Parish and R. Tateson, Eds. Natural Computation Series, Springer, November 2004, 51-86.
- [36] Timmis, J., Neal, M., and Hunt, J. An Artificial Immune System for Data Analysis. *Biosystems*. 55, 1, 2000, 143-150.
- [37] Varela, F. J., and Coutinho, A. Second Generation Immune Networks, *Imm. Today*, 12, 5, 1991, 159-166.
- [38] Watson, R.A., Pollack, J.B. Incremental Commitment in Genetic Algorithms, *Proceedings of GECCO 1999*. Banzhaf, et al. eds., Morgan Kaufmann, 1999, 710-717.
- [39] Watson, R.A., Pollack, J.B. Symbiotic Combination as an Alternative to Sexual Recombination in Genetic Algorithms, *Proceedings of Parallel Problem Solving from Nature (PPSN VI)*, 2000.
- [40] Yoo, J., and Hajela, P. Immune network simulations in multicriterion design. *Structural Optimization*, 18, 1999, 85-94.