Implementation of Immunological Algorithms in Solving Optimization Problems

Petar Čisar

Academy of Criminalistic and Police Studies, Cara Dušana 196, 11080 Zemun, Serbia, petar.cisar@kpa.edu.rs

Sanja Maravić Čisar

Subotica Tech, Marka Oreškovića 16, 24000 Subotica, Serbia, sanjam@vts.su.ac.rs

Branko Markoski

Technical Faculty "Mihajlo Pupin", University of Novi Sad, Đure Đakovića bb, 23000 Zrenjanin, Serbia, markoni@uns.ac.rs

Abstract: This paper gives a global review of artificial immune systems in computer science and their implementation. The performance of the immunological algorithm in solving optimization problems is analysed using the Optimization Algorithm Toolkit, with emphasis on determining the impact of parameter values. It is shown that these types of algorithms are particularly sensitive to the choice of parameters that affect the functioning of the algorithm.

Keywords: immunological algorithm; implementation; Optimization Algorithm Toolkit; travelling salesman problem; function optimization, parameters

1 Introduction

The contribution of this paper is its use of the Optimization Algorithm Toolkit (OAT) environment to examine the impact of immunological algorithm parameters on accuracy and speed of calculations in solving optimization problems.

The paper consists of three sections. The first section is the introduction, which shows the analogy between biological and artificial immune systems (AIS) and explains the categorization of immunological algorithms. The second section presents an overview of implementation possibilities of artificial immune systems. The third section concentrates on one specific type of implementation, which is optimization, and its various aspects in an adequate software environment. Lastly, this paper offers a conclusion based on the analysed cases.

We begin with a detailed background necessary for understanding the immunological algorithms.

Humans, similar to other living organisms, are constantly exposed to a wide range of micro-organisms such as bacteria, viruses, parasites and other harmful molecules (called antigens) that can damage the human body. In order to prevent this, the human body has developed an immune system. The immune system is a very complex defence system that is composed of different cells (B and T lymphocytes) that prevent foreign objects from damaging the body. The T-cell is a special type of white blood cell that is of key importance to the immune system. It has so-called T-cell receptors on its surface with which it can detect antigens. Normally, the receptors of a T-cell do not match the body's own substances.

In the above context, it is possible to formulate a definition of a clone. A clone is a propagating population of organisms, either single cell or multi cellular, derived from a single progenitor cell. Such organisms should be genetically identical, though mutation events may abrogate this [1].

Each human body has an innate immune system. This system has a major role in the development of a complete immune system. Namely, over the years as the body is attacked by certain antigens, the immune system does not just have the goal of destroying these antigens, but also to memorize them. It has the important property of pattern recognition that may be used to differentiate between outer cells entering the body (antigens) and the body cells. Immune systems have several crucial characteristics such as uniqueness, autonomy, memory and recognition of foreigners, distributed detection, and noise tolerance [2].

Artificial Immune Systems (AIS) are adaptive systems, inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving [3]. The general principle of functioning of the immune algorithm is presented in the following figure.

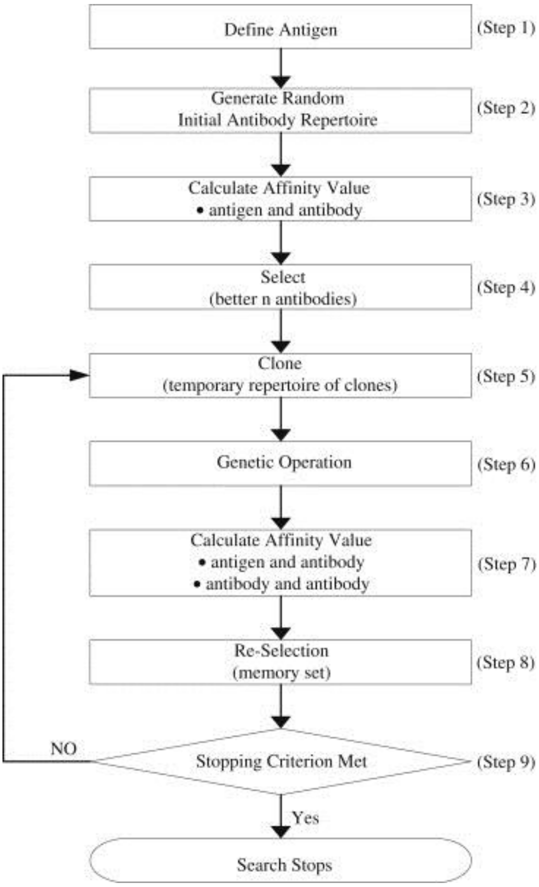


Figure 1

Flowchart of the immune algorithm [4]

The research of the artificial immune system began in the 1990s, and until now a significant number of algorithms have been developed which have different applications. Existing algorithms can be divided into four categories, according to the functioning principle of the used immune system: negative selection, clonal selection, immune network algorithm and dendritic cell.

Negative Selection Algorithm – The inspiration for this type of algorithm can be found in the mechanism in the thymus whose task it is to generate a set of mature T-cells that can only bind to non-self antigens. The thymus, which is a lymphoid organ, can be found in the middle of the upper chest right behind the sternum (breastbone). The thymus is where lymphocytes mature, multiply and become T-cells. Forrest et al [5] introduced the first negative selection algorithm in order to detect data manipulation which was caused by a computer virus. The algorithm's primary task is to produce a set of self strings that are characteristic of the system's normal state. The following task is to generate so-called detectors that exclusively recognize the complement of these strings. Then the given detectors are applied to new data in order to categorize them as being self or non-self, specifying the situation that data was changed.

Clonal Selection Algorithm - The clonal selection approach inspired the development of the AIS that executes optimization and pattern recognition problems. This stems from the antigen controlled maturation of B-cells, with associated hyper mutation process. These immune systems also implement the concept of memory cells to continue to provide excellent solutions to the problem in question. De Castro and Timmis [3] highlight two important features of affinity maturation in B-cells, the first being that the increase of B-cells is in proportion to the affinity of the antigen which binds to it, therefore the higher the affinity, the more clones created. Furthermore, the mutations resulting through the antibody of a B-cell are inversely proportional to the affinity of the antigen to which it binds. De Castro and Von Zuben [6] developed a frequently used clonal selection based-AIS (CLONALG) by implementing these two features, which has been applied to executing pattern matching and multi-modal function optimization tasks.

Immune Network Algorithms – These algorithms originated from Jerne [7] who described the immune network theory in order to provide an explanation for some of the observed vital characteristics of the immune system (learning and memory). The basis of the network approach states that any lymphocyte receptor can be identified by a subset of the receptors. The receptors of this recognizing set have their own similar set, creating a network of interactions [8]. Immune networks are usually called idiotypic networks. If there is no foreign antigen, Jerne summarized that the immune system has to present a behaviour or activity resulting from interactions with itself, resulting in tolerance and memory, among others, from the immunological behaviour of these reactions.

Dendritic Cell Algorithms - Matzinger [9] created the danger theory, which has met with widespread acceptance in the circle of immunologists as a manner of explaining the development of peripheral tolerance (tolerance to external agents). This algorithmic approach states that antigen – presenting cells (APC) are actually activated by signals of danger. The activated APCs can then provide the needed co-stimulatory signal to the T cells which in turn control the immune reaction. The danger signals originate from normal cells of the human body which have been injured by an attack.

The performance of the immunological algorithm in solving optimization problems is analysed in this paper using the Optimization Algorithm Toolkit (OAT). The OAT is a workbench and toolkit for developing, evaluating and experimenting with classical and state-of-the-art optimization algorithms on standard benchmark problem domains. This open source software includes reference algorithm implementations, graphing, visualizations and other options. The OAT provides a functional computational intelligence library for investigating existing algorithms and problems, as well as implementing new problems and algorithms. Built on top of this library is a simple explorer and experimenter graphical user interface that provides a basic understanding of the functionality in the library. The goal of the library is to facilitate the best practice of algorithm, problem, and experiment design and implementation, as well as software engineering principles. The graphical user interface provides non-technical access for configuring and visualizing existing techniques on standard benchmark problem instances [10].

2 Implementation of Artificial Immune Systems

Using the historical chronology of research and existing division of applications [11], the basic implementations of artificial immune systems can be classified into the following main areas and corresponding sub-areas [12]:

Anomaly Detection

- Fault detection
- Computer and network security

Recognition and detection of anomalies in data include, among other areas, discriminative analysis, error estimation, feature extraction, grammatical inference, image analysis, sign recognition, speech recognition and identification mechanisms. Applications are extremely diverse and range from the recognition of spectrum reactants in chemical analysis, prediction of infectious diseases and analysis of medical data. The area of detecting anomalies in time data is based on the detection of new or unexpected patterns. Their applications include the area of early detection of potential hardware failures (for example, the detection of temperature and pressure fluctuations and detection of defects in different devices). Fault detection in software and hardware is extended to distributed systems such as sensor networks and nodes of wireless networks, where it can affect the flexibility and routing of traffic.

Machine Learning

- Clustering and classification
- Robotics
- Pattern recognition
- Control (planning)

The domain of machine learning is very general and includes various forms of pattern recognition, conceptual learning, controlled and uncontrolled learning, clustering and classification data. The earliest applications were in the area of unmonitored learning to identify clusters of data while further implementations are in the area of supervised learning in systems with limited resource classifiers, the model of immune network for clustering and filtering unmarked sets of numerical data, adaptation of intelligent systems and conceptual learning.

In the field of robotics, there are several directions of implementation of artificial immune systems, which can generally be divided into the collective management of robot groups and adaptive management of an individual robot.

The domain of control also contains examples of artificial immune systems in processes of adaptive, sequential and feedback control, which are applicable in manufacturing systems, planning tasks and workflow, process planning and planning of software and hardware support.

Optimizations

- Optimization of numerical functions
- Combinatorial optimization

Numerous algorithms have been developed in this area, mainly based on the principle of clonal selection, for example, the CLONALG clonal selection algorithm [6], the immunological network optimization opt-aiNET [13], an algorithm with the B-cells [14] and the immune optimization algorithm opt-IA [15]. In order to display the elements, the popular and often exploited CLONALG algorithm uses binary strings, implements affinity maturation and is also suitable for pattern recognition. The application of these algorithms and systems is explained in numerous papers involving optimization functions, multi-target optimization, optimization of multimodal functions, combinatorial and time-independent optimization, and offer a range of solutions in the field of route management (e.g. the travelling salesman problem), planning tasks, and storage optimization. The clonal algorithm and the algorithm with B-cells are applied to the optimization of dynamic function, which is analogous to the task of the human immune system.

Other Implementation Areas

- Computational immunology
- Data and Web mining

One of the applications of the AIS lies in computer immunology and the areas of bio-informatics and immuno-informatics, where the first computer simulations of immune networks have evolved into modern stochastic models of the immune system. However, they do not necessarily use an immune network and are used for educational and scientific purposes. Although this area is often related to the problem of clustering, further research is progressing in different directions, from the study of evolution of antibody libraries to the behaviour of ecosystems.

The area of data mining, which involves classification and anomaly detection, covers a wide spectrum of applications from data and web mining, but also different applications, such as the detection of fraud in financial systems, detection of potential customers, discovering disjunctive rules, data filtering, computer supported cooperative work, and information processing.

3 Examples - Optimization

In the fields of mathematics, statistics, empirical sciences, computer science, mathematical optimization refers to the selection of the best element (with regard to some criteria) from the set of available alternatives [16]. An optimization problem involves finding the minimum and maximum of a real function by systematically changing the input values from within a pre-defined set and calculating the corresponding values of the function. More generally, optimization includes finding extreme values of some objective function given a defined domain, including a variety of objective functions and domains.

Solving optimization problems by means of appropriate software will be examined through several examples: route management and function optimization.

The optimization problem as a route management problem can be considered as follows. If one takes a set of cities and the travel costs between each pair of them, the Travelling Salesman Problem (TSP) comprises the problem of identifying the shortest possible route that passes through every city exactly once and returns to the initial city. The standard version of this problem assumes symmetric travel costs, meaning that the travel cost from city X to city Y is equal to the travel cost from the city Y to X. A large problem in graph theory is to minimize the total distance and determine the route for the salesman so that he travels through each city only once. This problem is categorized as NP (nondeterministic polynomial)-hard and NP-complete.

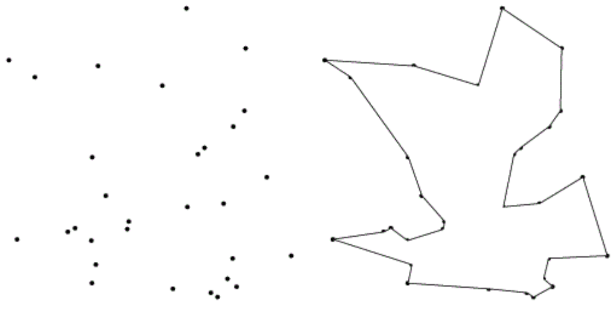


Figure 2

Travelling salesman problem - illustration

The dots in the figure above represent geographic locations of given cities, with known distances between each pair of them.

The simplicity of the statement of the problem is apparent. This makes the TSP one of the most closely examined problems in computational mathematics, though a method for a suitable solution for the general case has yet to be found [17]. It is also used as a benchmark for many optimization methods.

The practical part of this paper will present the testing of immune algorithm in solving the TSP using the OAT as a working environment.

Within this category, the CLONALG algorithm was chosen. The basic principle of operation of the CLONALG algorithm is shown in the following figure.

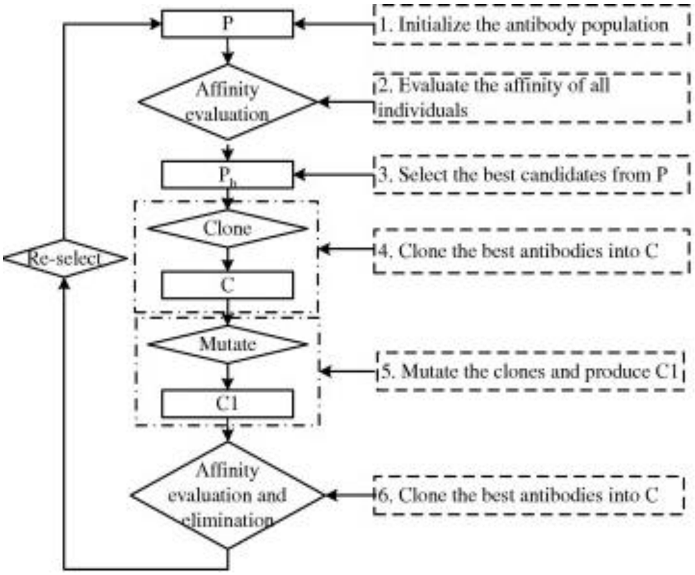
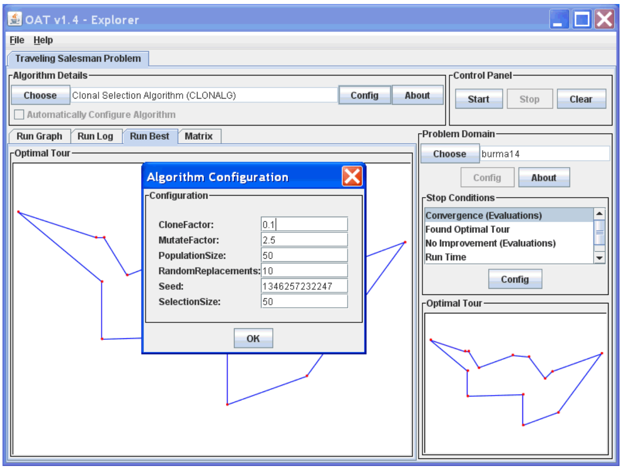

P
1. Initialize the antibody population
Affinity evaluation
2. Evaluate the affinity of all individuals
P_h
3. Select the best candidates from P
Clone
C
4. Clone the best antibodies into C
Re-select
Mutate
C1
5. Mutate the clones and produce C1
Affinity evaluation and elimination
6. Clone the best antibodies into C

Figure 3
Steps of clonal selection method [18]

As shown in the figure above, the clonal selection algorithm contains three basic parameters:

- antibody population size
- clone factor
- mutate factor

As stated before, the aim of this paper is to present the implementation of the OAT environment, to examine the influence of these parameters on accuracy in determining the optimal tour and speed of calculations. In this sense, the ability to configure the algorithm will be used, as shown in the figure below.



Clonal Selection Algorithm (CLONALG)

burma14

CloneFactor: 0.1
MutateFactor: 2.5
PopulationSize: 50
RandomReplacements: 10
Seed: 1346257232247
SelectionSize: 50

Figure 4

CLONALG algorithm configuration

The problem domain with 14 dots and the known optimal tour (shown in the bottom right corner of the figure above) were analysed. The optimization is considered finished when the same graphic form is obtained as the optimal tour. The influence of the analysed parameters (population size, clone and mutate factor) on the solving time is given in the following table.

Table 1

Influence of Parameters

Population size	20	30	40	50	60	70	80
Solving time (sec)	25	95	1	80	2	20	37

Clone factor	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Solving time (sec)	83	1	3	1	1	1	1

Mutate factor	2	2.5	3	3.5	4	4.5	5
Solving time (sec)	< 1	80	< 1	420	170	< 1	> 600

After thoroughly analysing the above values obtained for the solving time, it can be concluded that the CLONALG algorithm is extremely sensitive to the choice of parameters. This means that the performance of this algorithm must be checked with several different parameters, and based on the results, the optimal value will be chosen. This procedure is known in the literature as tuning parameters and directly affects the speed of convergence of the algorithm (i.e. solving time) to the solution. Also, it can be noticed that the CLONALG maintains a diverse set of local optimal solutions, which is mainly due to the selection and reproduction schemes. Another important characteristic is the fact that CLONALG takes into account the cell affinity, corresponding to an individual's fitness, in order to define the mutation rate applied to each member of the population.

The optimization problem as a continuous function optimization problem can be summarized as follows. It is logical that the increase of the population size directly influences the algorithm to find the optimum quickly, but it also increases the processor computation time. These parameters, the clone and mutate factor, are important because they affect the algorithm to find more optima or a faster global optimum. These parameters also affect the CPU time so they need to be precisely set in order to achieve the desired result in a satisfactory time. The function that was tested in research [19] is:

$$f = x \cdot \sin(4\pi x) - y \cdot \sin(4\pi y + \pi) + 1 \qquad (1)$$

The mutate factor affects the number of clones of each individual. In this way it affects the CPU time of execution of the algorithm.

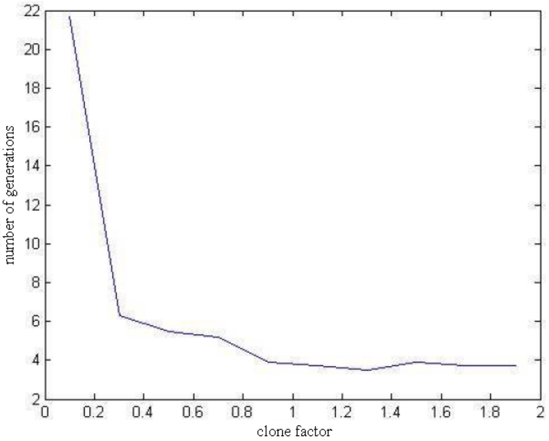


Figure 5

Influence of clone factor on convergence of algorithm [19]

Figure 5 shows that for very small values of clone factor, the algorithm is not able to find the global optimum, whereas increasing the parameter above 1 does not contribute to the improvement of convergence. Also, large values of mutate factor inhibit the algorithm to find the global optimum (Figure 6).

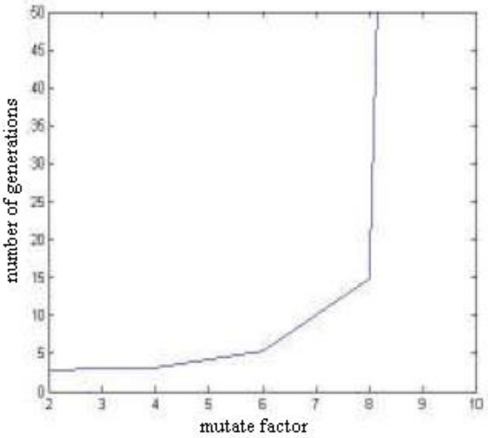


Figure 6
Rate of convergence [19]

Another example of function optimization will be shown in the OAT environment. The Shubert function was analysed, with the help of the optimized artificial immune network (opt-aiNET) algorithm. This is a multimodal test function given by the following relation:

$$f(x_1, x_2) = -\sum_{i=1}^{5} i\cos((i+1)x_1 + 1)\sum_{i=1}^{5} i\cos((i+1)x_2 + 1) \quad (2)$$

The test range is usually limited to: $-5.12 \leq x_1 \leq 5.12$, $-5.12 \leq x_2 \leq 5.12$. This function has several local minima and 18 global minima. In this case, the optimization implies minimization of the function, i.e. finding the minima.

The convergence of solutions around extremes (global minima) is easy to recognize in the next figure – the left side. Single solutions are shown in the form of numerous dots deployed around the points of the minima. The right side of Figure 7 represents the Shubert function in 3D.

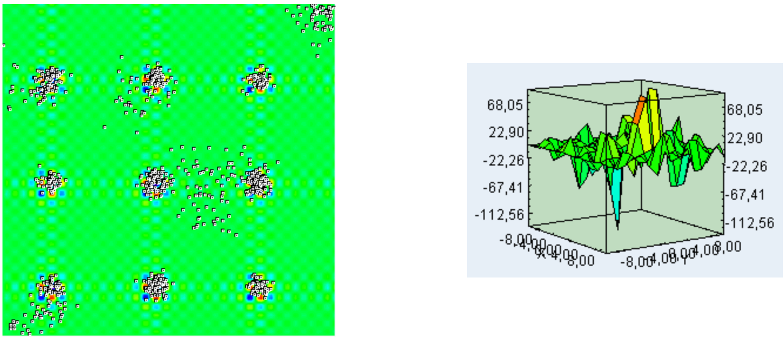


Figure 7

Shubert function optimization – optimized artificial immune network algorithm

The parameter of this algorithm that predominantly affects the final result is the total number of clones per cell. It is important to emphasize that the performed testing has shown that when this number is greater than 90, 1-2 minima always remain unidentified (unclustered).

Also, it was found that using other immunological algorithms offered by the OAT (adaptive clonal selection (ACS) [20] and optimization immune algorithm (opt-IMMALG) [21]) do not give a satisfactory result, i.e. it does not generate clustering solutions.

The next example of optimization is the optimization of binary functions. A binary function, or function of two variables, is the type of function which takes two inputs. A binary function may also be defined as a function from *X* × *Y* to *Z* (X, Y inputs, Z output). Out of the offered functions in the OAT environment, a simple trapdoor function is chosen. The trapdoor function is a function that is easy to compute in one direction, but it is difficult to compute in the opposite direction without the secret trapdoor information (for given *f(x)* and *y* it is easy to compute *x*). Trapdoor functions are widely used in cryptography. An example of a trapdoor one-way function is the factorization of a product of two large primes. While selecting two large primes and multiplying them is relatively easy, factoring the resulting product is rather difficult. This is the basis of the RSA encryption, which is conjectured to be trapdoor one-way. The graphical result of the optimization is shown in the figure below. The black dot represents the logical zero, while the white dot is the logical one. Binary string length was set to 100, the number of total evaluations was 1000, and the run time 33 ms.

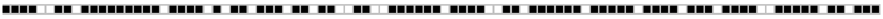


Figure 8

Trapdoor function optimization – optimization immune algorithm

The algorithm used in the above case was the optimization immune algorithm. It has several parameters: maximum age, mutation (static or inversely proportional), number of clones and population size. It has been shown that these parameters do not affect the final result, but only the run time.

It was also found that other offered algorithms in the OAT (B-cell algorithm (BCA), Cloning, Information Gain, Aging (CLIGA), CLONALG and simple immune algorithm (SIA)) also generate correct results. They differ only in run time.

Conclusions

The practical part of the paper is focused on examining the impact of parameters of immunological algorithms on processing time needed to determine the optimal values. In all of the analysed cases of route management, only the spent times of the algorithms that led to the optimal solution were different for the different parameter values. In this sense, the existence of multiple optimal values was found, as well as extremely unacceptable values. This means that great attention must be paid to the selection of parameter values (it is not to be disregarded which values are applied) and it is vital to test the behaviour of the algorithm in practice with several different parameters. Also, it was confirmed that all of the immune algorithms are not equally suitable for certain situations of function optimization. Namely, it has been shown that some of the analysed algorithms did not identify local extremes of examined test function.

References

[1] Biology Online, www.biology-online.org

[2] L. de Castro, F. Zuben: Artificial Immune Systems: Part I – Basic Theory and Applications, TR – DCA 01/99, 1999

[3] L. de Castro, J. Timmis: Artificial Immune Systems: A New Computational Intelligence Approach, Springer, pp. 57–58. ISBN 1-85233-594-7, 9781852335946, 2002

[4] C. Chu, M. Lin, G. Liu, Y. Sung: Application of immune algorithms on solving minimum-cost problem of water distribution network, Mathematical and Computer Modelling, Volume 48, Issues 11-12, pp. 1888-1900, 2008

[5] S. Forrest, A. Perelson, L. Allen, R. Cherukuri: Self-nonself discrimination in a computer, in Proceedings IEEE Symposium on Research in Security and Privacy, pp. 202–212, 1994

[6] L. de Castro, F. Von Zuben: Learning and optimization using the clonal selection principle, IEEE Transactions on Evolutionary Computation, 2002 Jun, 6(3):239-251. ISSN: 1089-778X

[7] N. Jerne: Towards a network theory of the immune system, Annals of Immunology (Inst.Pasteur). 125C. pp. 373-389, 1974

[8] AISWeb, The Online Home of Artificial Immune Systems, http://www.artificial-immune-systems.org/algorithms.shtml

[9] P. Matzinger: Tolerance, Danger and the Extended Family, Annual Review of Immunology. 12, pp. 991-1045, 1994

[10] Optimization Algorithm Toolkit, http://optalgtoolkit.sourceforge.net/

[11] E. Hart, J. Timmis: Application areas of AIS: The past, the present and the future, Journal of Applied Soft Computing, vol. 8, No. 1, pp.191-201, 2008

[12] B. Mihaljević: Područja primjene umjetnih imunoloških sustava i teorije opasnosti, Fakultet elektrotehnike i računarstva, Zagreb, www.fer.unizg.hr/_download/repository/Kvalifikacijski_doktorski_ispit_-_Branko_Mihaljevic.pdf

[13] L. de Castro, J. Timmis: An artificial immune network for multimodal function optimization, Proceedings of the 2002 Congress on Evolutionary Computation (CEC '02), Honolulu, HI, USA, USA: IEEE Computer Society, 2002, 699-704. ISBN: 0-7803-7282-4

[14] J. Timmis, C. Edmonds, J. Kelsey: Assessing the Performance of Two Immune Inspired Algorithms and a Hybrid Genetic Algorithm for Function Optimisation, Proceedings of the Congress on Evolutionary Computation (CEC04), Potland, Oregon. USA. USA: IEEE Press; 2004: 1044-1051

[15] V. Cutello, G. Narzisi, G. Nicosia, M. Pavone, G. Sorace: How to Escape Traps using Clonal Selection Algorithms, The First International Conference on Informatics in Control, Automation and Robotics, ICINCO 2004,Setubal, Portugal. INSTICC Press; 2004: 322-326

[16] "The Nature of Mathematical Programming", Mathematical Programming Glossary, INFORMS Computing Society, http://glossary.computing.society.informs.org

[17] Travelling Salesman Problem, www.tsp.gatech.edu

[18] I. Aydin, M. Karakose, E. Akin: Chaotic-based hybrid negative selection algorithm and its applications in fault and anomaly detection, Expert Systems with Applications, Volume 37, Issue 7, pp. 5285–5294, 2010

[19] K. Đuretec: Umjetni imunološki sustavi, Projekt: Algoritmi zasnovani evolucijskom računanju, Fakultet elektrotehnike i računarstva, Zagreb, www.zemris.fer.hr/~golub/ga/studenti/projekt2008/ais/umjetni_imunoloski_sustavi.pdf

[20] S. Garrett: Parameter-free, adaptive clonal selection, Congress on Evolutionary Computing (CEC 2004), Portland Oregon, USA. USA: IEEE Press; 2004: 1052-1058. ISBN: 0-7803-8515-2

[21] V. Cutello, G. Nicosia, M. Pavone, G. Narzisi: Real Coded Clonal Selection Algorithm for Unconstrained Global Numerical Optimization using a Hybrid Inversely Proportional Hypermutation Operator, 21st Annual ACM Symposium on Applied Computing (SAC), Dijon, France, 2006: 950-954. ACM