

# Conceptual and Practical Aspects of the aiNet Family of Algorithms

*Fabrício O. de França, University of Campinas, Unicamp, Brazil*

*Guilherme P. Coelho, University of Campinas, Unicamp, Brazil*

*Pablo A. D. Castro, University of Campinas, Unicamp, Brazil*

*Fernando J. Von Zuben, University of Campinas, Unicamp, Brazil*

---

## ABSTRACT

*In this paper, a review of the conceptual and practical aspects of the aiNet (Artificial Immune Network) family of algorithms will be provided. This family of algorithms started with the aiNet algorithm, proposed in 2002 for data clustering and, since then, several variations have been developed for data clustering, biclustering and optimization in general. Although the algorithms will be positioned with respect to other pertinent approaches from the literature, the emphasis of this paper will be on the formalization and critical analysis of the set of contributions produced along almost one decade of research in this specific theme, together with the provision of insights for further extensions.*

*Keywords: Artificial Immune Systems, Continuous Optimization, Data Clustering, Discrete Optimization, Dynamic Optimization, Idiotypic Networks, Multi-Objective Optimization*

---

## 1. INTRODUCTION

The study of Artificial Immune Systems on Computer Science (CS) is aimed at mimicking some processes of the vertebrate immune systems on a computational context, trying to solve problems from the most diverse areas, not only from CS but also engineering, medical sciences, and many more (de Castro & Timmis, 2002a).

The first work to apply the known concepts of AIS on CS was the machine learning algorithm of Farmer et al. (1986), in which a dynamic model for the immune system based on Jerne's network hypothesis (Jerne, 1974) was proposed and, as a result, the model presented high similarity with Holland's Classifier Systems (Holland, 1984).

Four years later, Bersini and Varela (1990) published a paper about immune networks, trying to exploit even further the relation of immune systems on machine learning and classifier systems.

DOI: 10.4018/jncr.2010010101

But only in the mid 90s the AIS paradigm gained the status of an active research area, particularly with the publishing of the application of Self-Nonself Discrimination principles to the computer protection problem (Forrest et al., 1994), and also of the Negative Selection algorithm (Dasgupta, 1998).

In late 90s, the work of de Castro and Von Zuben (1999) introduced the concepts and applications of the Clonal Selection principle (Burnet, 1978) to artificial immune systems, not only dealing with machine learning problems but also adapting the immune-inspired mechanisms to perform discrete and continuous optimization. In the trace of this achievement, the popular CLONALG algorithm (de Castro & Von Zuben, 2002a) – *Clonal Selection Algorithm* – was proposed, and followed by several applications and extensions.

The addition of network-based aspects inspired from Jerne's Immune Network Theory (Jerne, 1974) to the CLONALG algorithm led to the *Artificial Immune Network* algorithm, or simply *aiNet* (de Castro & Von Zuben, 2002b), that not only extended the capabilities of CLONALG but also introduced very interesting aspects such as the network interaction among solutions and the dynamic adaptation of the size of the set of candidate solutions. The aiNet algorithm was originally developed for clustering problems, and it is capable of automatically identifying a proper number of arbitrarily shaped clusters.

Following the successful results obtained by the original aiNet algorithm, many extensions appeared in the literature, not only for clustering problems but also for continuous optimization (de Castro & Timmis, 2002b), combinatorial problems (de Sousa et al., 2004; Gomes et al., 2004), bioinformatics (Bezerra & de Castro, 2003; de Sousa et al., 2004; Gomes et al., 2004), prediction systems (Li et al., 2010), biclustering (Castro et al., 2007a,b,c; Coelho et al., 2008; de França et al., 2006b), multi-objective optimization (Coelho & Von Zuben, 2006a; Coelho et al., 2008), dynamic optimization problems (de França et al., 2005a; de França et al., 2006a;

Junqueira et al., 2005; Junqueira et al., 2006), and many others.

With almost ten years of existence and supporting a very active line of immune-inspired algorithms, it has become hard to keep track of all existent variations of the aiNet algorithm and their potential applications. Therefore, the goal of this paper is to critically review the original aiNet algorithm, describing the conceptual aspects associated with it, and to present and discuss its main extensions together with the results obtained so far.

The paper is organized as follows: Section 2 presents the theoretical aspects of natural and artificial immune systems that were applied on the aiNet context; Section 3 is dedicated to draw a parallel between the clustering problem and the immune concepts presented in Section 2, and to explain and discuss the original aiNet algorithm; in Section 4 an overview of the aiNet family of algorithms is presented, together with a discussion of general aspects of the different algorithms and the problems that they are meant to solve; Section 5 is dedicated to optimization algorithms, where the variations of aiNet developed for optimization of continuous, discrete, multimodal, multi-objective and dynamic-environment problems will be detailed; Section 6 discusses and details the extensions of aiNet for biclustering; and, finally, Section 7 concludes this survey summarizing the most relevant issues, presenting the final remarks and discussing some prospects concerning the future of the aiNet family of algorithms.

## 2. NATURAL AND ARTIFICIAL IMMUNE SYSTEMS

As the name of the algorithms in the aiNet family suggests, they are inspired by the natural immune system of vertebrates, more specifically, by the mechanisms associated with B-cells in the adaptive immune response (de Castro & Von Zuben, 2002b). These immune-inspired algorithms are based on a paradigm known as *Artificial Immune Systems* (AIS), originated from attempts to model and apply

immunological principles to problem solving in a wide range of areas such as optimization, data mining, computer security and robotics (de Castro & Timmis, 2002a).

The natural immune system, which is the main inspiration for AIS, can be considered as one of the most important components of vertebrates, as its mechanisms of recognition and combat against infectious foreign agents (*pathogens*) are responsible for keeping the organism healthy. These recognition and combat mechanisms are mainly based on the molecular patterns present on such external invaders, which are known as *antigens*. Once the immune cells identify the presence of antigens from a given pathogen, the immune response is triggered, in order to try to eliminate such potentially dangerous elements from the body. The defense mechanism provided by the immune system can be divided into two front lines: the *innate* and the *adaptive* immune systems (Pinchuk, 2002). The components of the *innate* immune system are capable of combating a wide variety of invading agents, without the need of being previously exposed to them, differently from the components of the *adaptive* immune system, which are triggered in response to specific infections.

There are several theories that try to explain the adaptive immune response, being the aiNet-based algorithms mainly inspired by the *Clonal Selection* principle (Burnet, 1978) and by the *Immune Network* theory (Jerne, 1974). These concepts will be briefly explained in what follows, with a remarkable degree of simplification when compared to the real interactions taking place in the natural immune system.

According to the *Clonal Selection* principle, when a pathogen invades the organism and its antigens are identified by the immune system, some specific immune cells (namely the *B-cells*) begin to proliferate by means of cloning, in order to face the infection. During this proliferation, those cells also differentiate into two distinct types of cells: the majority of them becomes *plasma cells*, capable of secreting *antibodies* at high rates; and the remaining ones into long-living *memory cells*, which allows a

faster immune-response in future infections of similar pathogens. The antibodies secreted by the plasma cells are molecules capable of binding to the antigens of an infectious agent (thus they present complementary molecular patterns), in order to indicate that those structures marked by such antibodies should be eliminated from the organism.

During the proliferation and differentiation of B-cells, a controlled mutation also occurs, together with a selective pressure over the antibodies, in order to increase the overall affinity of such molecules to the antigens of the invading pathogen. In other words, after the cloning phase, the new generated cells suffer a mutation process and those cells capable of producing antibodies that bind better with the antigens than the others (those that present higher *affinity*) are selected to remain in the population of immune cells. Those cells that lead to the production of antibodies that do not present a high affinity with the antigens or that respond to cells of the organism itself are removed from the repertoire.

From the steps described above, it can be seen that the Clonal Selection principle states that the immune system is composed of several components that remain “at rest” in the body, until they are triggered by a foreign antigenic stimulus (de Castro & Timmis, 2002a). A different view of the immune response was proposed by Jerne (1974) with the *Immune Network* theory (or *Idiotypic Network* theory). According to Jerne, the cells of the immune system are not only capable of recognizing antigens, but also each other, defining the immune system as a huge self-organizing network of components that continuously recognize and/or are recognized by other components. When an immune cell recognizes another cell, its response can be positive, in the sense that the cell is activated and starts the process of cloning, mutation and secretion of antibodies, or negative, which leads to tolerance and even suppression. In the immune network theory, the main characteristics of the immune system, such as learning, memory, self-tolerance, size and diversity of the immune repertoire, are

considered *emergent properties*, consequence of regulatory mechanisms aimed at maintaining the network within a specific range of activity (de Castro & Timmis, 2002a).

The immune network theory dominated the immunology field until the nineties, when it began to fade out mainly due to the lack of empirical evidences that could justify the proposed mechanisms. However, it seems that a renewed interest on idiotypic networks may be emerging nowadays, especially in clinical contexts (Behn, 2007).

Although the popularity of Jerne's theory among immunologists varied significantly in the last decade, a different situation occurred among computer scientists and engineers, who adopted its main concepts as inspirations in the development of several different immune-inspired techniques, as this theory seems to provide mechanisms to stimulate diversity among solutions in population-based algorithms.

Besides the diversity maintenance and stimulation mechanisms presented in algorithms inspired by Jerne's immune network theory, artificial immune systems that also seek for inspiration in the clonal selection principle present two additional distinctive characteristics that, in the last years, were responsible for an increase of interest from the scientific community in this paradigm: (i) the size of the population at each generation of these algorithms is automatically defined according to the demands of the application; and (ii) local optimal solutions tend to be simultaneously preserved in the population, once located during the search. As the aiNet family is mainly based on these two immune concepts (clonal selection and immune network), all its algorithms present these three characteristics, as will be seen in the following sections.

### 3. THE ARTIFICIAL IMMUNE NETWORK ALGORITHM – AINET

As it was already mentioned, the aiNet algorithm (de Castro & Von Zuben, 2002b) was proposed

to solve clustering problems. However, what is the parallel devised by de Castro & Von Zuben between the combat against pathogens in the natural immune system and the division of data samples into groups?

One approach to solve clustering problems can be basically divided into two main steps: (i) the positioning of a set of prototypes (each one with a pre-defined radius) that represent the samples (or patterns) in the dataset in a concise form; and (ii) the separation of these prototypes (and, consequently, the samples represented by each one of them) into clusters. This second step can be easily accomplished by traditional mechanisms such as the pruning of inconsistent branches in minimum spanning trees (MST - Zahn, 1971), if you consider the final set of prototypes as an interconnected graph that properly represents the original data structure. Therefore, the first step is crucial to obtain a suitable set of clusters of the problem, to reduce eventual redundancy that may be present in the dataset and to identify the main aspects of the structure and spatial distribution of the dataset.

Let's consider solely the prototype positioning step and assume that such prototypes can be seen as the B-cells of the immune system of an organism, and the data samples as the antigens of the pathogens that invade such organism. According to the clonal selection principle, when an infection occurs, the B-cells of the organism proliferate (through *cloning*) and suffer a controlled mutation (referred just as *hypermutation* from now on) so that they become more capable of recognizing the invading pathogens (the antibodies of such B-cells become more capable of identifying the antigens of the pathogens). Back to the clustering problem, as the B-cells of an infected organism, the population of prototypes (which was initially randomly spread across the data space) must also proliferate and evolve to match the samples of the dataset, so that they can suitably represent subsets of these samples.

It is also desired that each prototype in the clustering problem represents different sets of samples from the dataset, associated with dif-

ferent “regions” of the data space, so that all the important structural aspects are somehow captured. Therefore, it is not desired that several prototypes converge to the same region of the data space and map similar sets of patterns or, in other words, it is not desired that several prototypes become too similar (they should be *diverse*). Returning to the infected organism, if we consider now the immune-network theory, it proposes that the B-cells in the immune system are also capable of recognizing each other and suppressing (or stimulating) the proliferation of the recognized B-cells, which may be interpreted as analogous to the diversity maintenance requirement in the clustering problem.

As we can see, it is possible to identify interesting similarities among the prototype positioning step in clustering problems and what can be produced by algorithms inspired by the clonal selection and immune network theories, which led de Castro & Von Zuben to explore such similarities and propose the *Artificial Immune Network* algorithm (de Castro & Von Zuben, 2002b), or just *aiNet*, which will be described next.

### The Structure of the aiNet Algorithm

As mentioned before, in the aiNet algorithm the patterns in the dataset to be clustered are considered as the antigens of pathogens (in an analogy to the natural immune system) and the prototypes (solutions for the problem) as the B-cells (or simply their antibodies). As both the prototypes and data patterns are represented as multi-dimensional points in the real space, the interaction among these components are considered to occur in the Euclidean shape-space (Perelson & Oster, 1979) and their affinity is evaluated according to their Euclidean distance, although other metrics could have been considered as well.

The main structure of the aiNet algorithm is described in Pseudo-code 1 (Figure 1). The initial population of cells is randomly generated with uniform distribution in the data space, so that the prototypes are as evenly spread

as possible at the beginning. After the initial positioning of the prototypes, the algorithm enters its main loop, which is repeated until a stop condition is satisfied. This stop condition can be a predefined number of iterations, the generation of a predefined number of antibodies, the achievement of a predefined average error between each antigen and its corresponding representative in the population of cells, and so on.

When an antigen is presented to aiNet, it will be matched against all antibodies in the network (population of the algorithm), and those with a maximal degree of similarity with this antigen will be selected to proliferate (cloning). During proliferation, the higher the affinity of the parent cell with the antigen, the more offspring is generated and this offspring then suffer somatic hypermutation proportional to the inverse of its affinity with the antigen.

Mutated clones with a high degree of affinity with other clones (suppression) and those with lower affinity with the antigens (apoptosis) are excluded from the population, and the remaining mutated clones are inserted into the main population of the algorithm.

After all antigens are presented to the cells, the similarity among all the antibodies is evaluated and those with a degree of similarity above a threshold are suppressed. Finally, new randomly generated ones are also introduced in the population, at every iteration (random insertion).

Once the stop criteria are met, another suppression step is performed aiming at eliminating cells that recognize no antigen. This step has not been explicitly considered in the original version of aiNet, but cells that do not represent any sample cannot be considered as prototypes in cluster analysis.

The parameters of aiNet that should be defined by the user are:  $N$ , which is the initial size of the population of antibodies;  $n$ , the number of cells of the network that should be cloned;  $m$ , which corresponds to the percentage of best cells that should be kept in the clone population;  $\sigma_1$ , which is the suppression threshold for the affinity among antibodies;  $\sigma_2$ , which is the

Figure 1. Pseudo code 1

Pseudo-code 1. Main Structure of the aiNet algorithm
<p><b>Input:</b></p> <ul style="list-style-type: none"> <li>- <math>N</math>: the initial size of the population of antibodies;</li> <li>- <math>n</math>: the number of cells that should be cloned;</li> <li>- <math>m</math>: the percentage of cells to be moved to the memory set;</li> <li>- <math>\sigma_1</math>: antibody-antibody suppression threshold;</li> <li>- <math>\sigma_2</math>: antibody-antigen suppression threshold;</li> <li>- <math>d</math>: the percentage of randomly generated cells to be included in the population.</li> </ul> <p>Randomly create the initial population of <math>N</math> antibodies, each one with dimension <math>L</math> (dimension of the data patterns);</p> <p><b>While</b> stop criteria not met <b>do</b></p> <p style="padding-left: 20px;">1 - <b>For</b> each antigen <b>do</b></p> <p style="padding-left: 40px;">1.1 - Determine its affinity to all network cells;</p> <p style="padding-left: 40px;">1.2 - Select the <math>n</math> highest affinity network cells;</p> <p style="padding-left: 40px;">1.3 - Generate <math>N_c</math> clones from these <math>n</math> cells. The higher the affinity, the larger <math>N_c</math>;</p> <p style="padding-left: 40px;">1.4 - Apply hypermutation to the generated clones, with variability inversely proportional to the progenitor's fitness;</p> <p style="padding-left: 40px;">1.5 - Determine the affinity among the antigen and all clones;</p> <p style="padding-left: 40px;">1.6 - Keep only <math>m\%</math> of the highest affinity mutated clones into the clone population;</p> <p style="padding-left: 40px;">1.7 - Eliminate all clones but one whose affinity with the antigen is inferior to a predefined threshold <math>\sigma_2</math> (apoptosis);</p> <p style="padding-left: 40px;">1.8 - Determine the affinity among all the mutated clones and eliminate those whose affinity with each other is above a pre-defined threshold <math>\sigma_1</math> (suppression);</p> <p style="padding-left: 40px;">1.9 - Insert the remaining mutated clones into the population;</p> <p style="padding-left: 20px;">2 - Determine the similarity among all the antibodies and eliminate those with similarity above a threshold <math>\sigma_1</math> (suppression);</p> <p style="padding-left: 20px;">3 - Introduce a percentage <math>d\%</math> of new randomly generated cells (random insertion);</p> <p><b>End While</b></p> <p>Eliminate the network cells, if any, which recognize no antigen.</p>

suppression threshold for the affinity among antibodies and antigens; and  $d$ , that corresponds to the percentage of new randomly generated cells that should be included in the population at each generation, to increase diversity. The number of clones  $N_c$  that should be generated for each cell is automatically defined according to the number  $N$  of cells in the population and the distance from the original cell to the antigens.

After the execution of aiNet to self-organize the prototypes in the data space, a knowledge extraction phase is performed to finally define the clusters of the problem. The aiNet adopts in this phase the pruning of inconsistent branches in minimum spanning trees (Zahn, 1971) to detect inherent separations between clusters, which allows the algorithm to automatically identify the number of clusters of the problem and separate the data patterns associated with

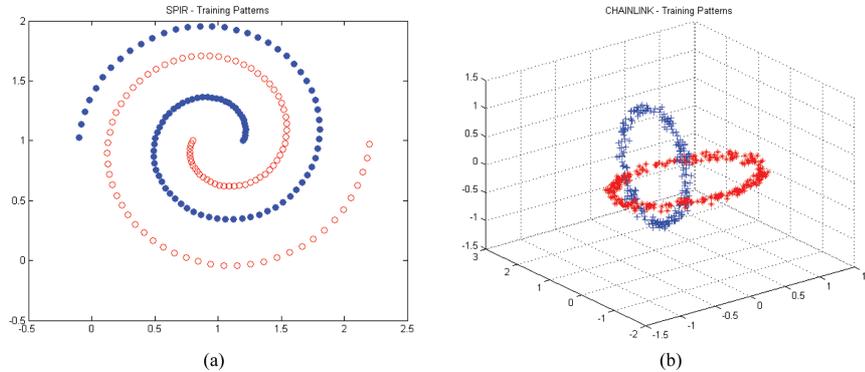
each cluster, even if they present arbitrary shapes (de Castro & Von Zuben, 2002b).

### Illustrative Examples and Applications of the aiNet Algorithm

In order to illustrate the performance of aiNet, de Castro and Von Zuben (2002b) applied the algorithm to two non-linearly separable clustering problems, namely SPIR (Figure 2(a)) and CHAINLINK (Figure 2(b)). The main purpose was to emphasize that the shape of the clusters may be arbitrary. Although Figure 2 presents labeled data, the labels were not considered to produce the results. The SPIR benchmark is composed of 190 samples in two dimensions, whereas the CHAINLINK consists of 1000 samples in three dimensions.

The final structure of the network for each benchmark is presented in Figure 3(a)(b).

Figure 2. Two benchmark problems used to evaluate aiNet: (a) SPIR and (b) CHAINLINK



Besides the proper clustering, another relevant characteristic of aiNet is its capability to reduce data redundancy. The network obtained for SPIR is composed of 124 antibodies. Note the compression was superior in regions where the amount of redundancy is higher, i.e., the centers of the spirals (see Figure 3(a)). The number of antibodies in the network for CHAINLINK is 32 (see Figure 3(b)), leading to a compression rate of 96.8%.

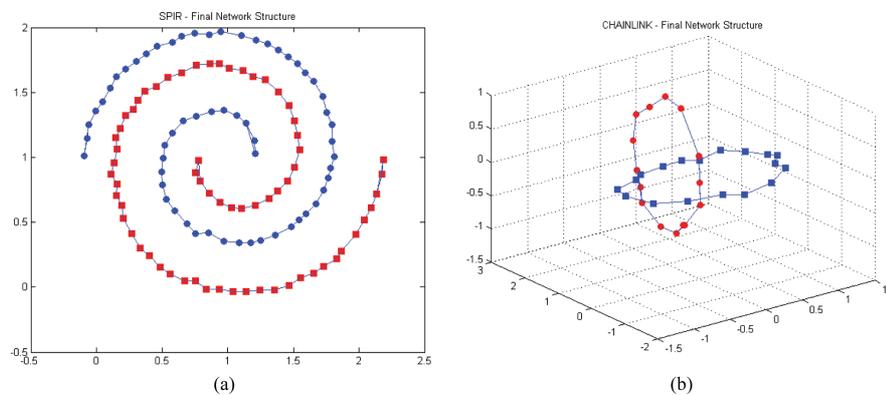
Although these problems are evidently of low dimension and with easily distinguishable clusters, thus not fully reflecting real-world clustering problems, it is possible to find several applications of aiNet-based algorithms to

real world problems, such as gene expression (Bezerra & de Castro, 2003; Bezerra et al., 2004), document clustering (Ciesielski et al., 2006; Xu et al., 2006), compositional timbre design (Caetano et al., 2005), prediction of landslides (Li et al., 2010), and learning of RBF neural networks (de Castro & Von Zuben, 2001).

### Comments About the aiNet Algorithm

A detailed analysis of the data compression property of aiNet was done by Stibor and Timmis (2007), who presented a metric based on Parzen Window (Fukunaga & Hayes, 1989)

Figure 3. Final network structure obtained by aiNet algorithm: (a) SPIR with 124 antibodies and (b) CHAINLINK with 32 antibodies



estimation and the Kullback-Leibler divergence (Kullback, 1959) to evaluate the quality of aiNet's compression capabilities over four different datasets. Essentially, this quality metric first estimates the probability densities of the original patterns in the dataset and of aiNet's final prototypes (antibodies), and then compares the similarity between them.

Stibor and Timmis' (2007) first concern was about the sensitivity of one of aiNet's parameters: the  $\sigma_1$  threshold that controls the maximum allowed similarity between two different antibodies. When this parameter was set to a high value, the final number of antibodies generated by aiNet exceeded the number of samples from the original dataset. Notice that Stibor and Timmis (2007) considered as prototypes cells that represent no antigen. If you have more prototypes than samples, some prototypes will stay in their arbitrary initial position, thus guiding to meaningless results in terms of cluster analysis. It is easy to detect this kind of degeneracy and thus avoid operating with extreme values for threshold  $\sigma_1$ . Another extreme scenario happened when  $\sigma_1$  was set to a low value: the overall similarity between the aiNet's output and the original dataset was unacceptably low, which should be expected as a much higher compression was achieved. So, Stibor and Timmis (2007) provided empirical evidences to properly detect inadequate settings for the threshold parameter  $\sigma_1$ . Therefore, if you are outside both extreme scenarios (which is easily detectable), you are in good condition to choose a proper value to  $\sigma_1$ .

An immediate conclusion that the reader may draw from the study of Stibor and Timmis (2007) is that aiNet requires a proper parameter setting. In this case, it shows the challenge of finding an appropriate suppression threshold, which directly controls the compression rate achieved by aiNet. However, this is actually a common problem for every clustering mechanism with the ability to self-control the number of prototypes. Of course, metrics that estimate the relative quality of the results produced by aiNet could also be adopted to implement self-

tuning procedures to set the parameters, at the cost of incrementing the computational burden of the algorithm.

Stibor and Timmis (2007) also stated that, from their results, the aiNet algorithm may be unsuitable for non-uniformly distributed data sets because of this pre-defined and fixed threshold, as different regions of the data set may require different values for this parameter.

A more elaborate type of immune-inspired network was further proposed to deal with the situation of variable data density from cluster to cluster (Bezerra et al., 2005). This new algorithm, called *Adaptive Radius Immune Algorithm* (ARIA), utilizes mechanisms of clonal selection and immune network suppression together with the density information contained in the data. ARIA makes use of an adaptive suppression radius that is inversely proportional to the local density for each antibody's neighborhood. In this way, for high density portions of data, antibodies are allowed to get closer to each other because of their small radii. For sparse regions, the radii tend to be larger, so antibody distribution tends to be sparser too.

In this context, similar to what happens to aiNet, a new branch of algorithms can be developed following the ARIA framework.

The computational cost associated with each iteration of the aiNet algorithm presented in Pseudo-code 1 is directly proportional to the size of the current population, as the most expensive step is the calculation of the Euclidean distance among all antibodies in the suppression phase. Therefore, for each iteration, the complexity is approximately  $O(N^2 \cdot L)$ , where  $N$  is the current population size and  $L$  the dimension of the antibodies. If we consider that the aiNet algorithm will be repeated for several iterations before the knowledge extraction phase and that this last step will be performed only once, the complexity of the knowledge extraction phase (generally  $O(N^2 \cdot \log(N))$  in the worst case – Martel, 2002) will be approximately equal to the complexity of a single iteration of aiNet, and smaller than the overall complexity of the immune-inspired step.

#### 4. THE AINET FAMILY OF ALGORITHMS

Right after the proposal of the aiNet algorithm by de Castro and Von Zuben (2002b) for clustering, de Castro and Timmis (2002b) proposed its variant for optimization problems (named opt-aiNet) and, since then, several extensions and modifications of both algorithms were proposed in the literature, as can be seen in the diagram shown in Figure 4.

Although these variations follow the same immune-inspired framework introduced in aiNet (as it is possible to see in the list of main mechanisms of each algorithm in Figure 4), the way that each operator is implemented may vary significantly according to the goals of each algorithm, and some proposals even include several mechanisms with no direct immune inspiration (such as some problem-dependent strategies of local search), which makes these algorithms more hybrid approaches than pure artificial immune systems. However, they are still based on the clonal selection and immune network theories, and so they were included in this review.

Chronologically, from the variations of aiNet presented in Figure 4 (which will be detailed later in this paper) the first ones were developed for optimization problems, being opt-aiNet (de Castro & Timmis, 2002b) followed by copt-aiNet (an extension of opt-aiNet for combinatorial optimization – de Souza et al., 2004), dopt-aiNet (an extension of opt-aiNet for continuous optimization in dynamic environments – de França et al., 2005a) and omni-aiNet (an extension of opt-aiNet for *omni optimization* – Coelho & Von Zuben, 2006a), respectively.

The concept of *omni optimization* (Deb & Tiwari, 2005) is associated with the capability of the same algorithm to automatically adapt its mechanisms to solve single and multi-objective optimization problems, both uni and multi-modal. Therefore, in omni-aiNet (and in all *omni optimizers*), the user does not need to modify

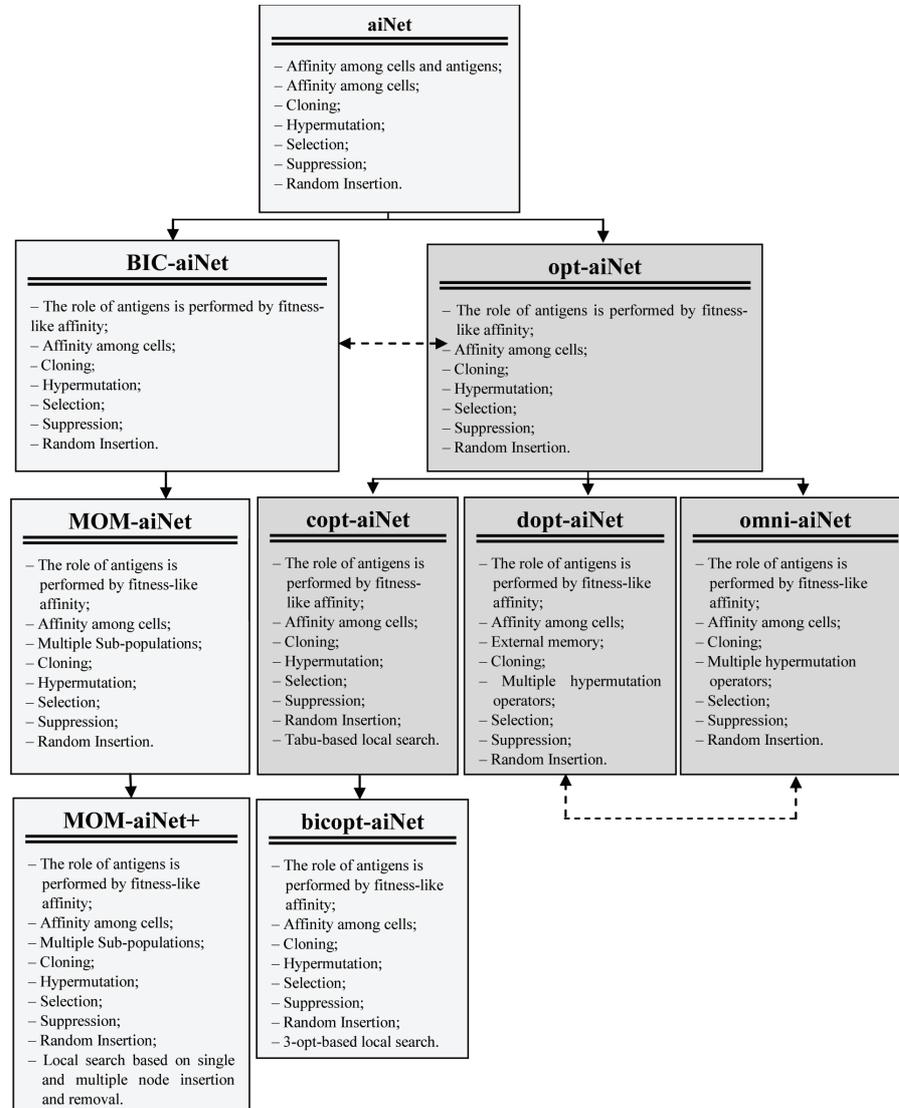
the structure of the algorithm according to the problem being solved, implying that a single and generic technique is capable of solving different types of optimization problems.

In 2006, de França et al. (2006b) proposed bicopt-aiNet, which was the first variation of aiNet to deal with *biclustering* (Hartigan, 1972; Mirkin, 1996), being bicopt-aiNet later followed by BIC-aiNet (Castro et al., 2007a), MOM-aiNet (Coelho et al., 2008) and MOM-aiNet+ (Coelho et al., 2009a), all developed to deal with biclustering. Sometimes also referred in the literature as *co-clustering* and *two way clustering*, biclustering works by simultaneously clustering the objects of the dataset and their attributes, which is useful when the problem requires finding the correlation among subsets of objects under certain subsets of attributes (a kind of local clustering). Also, this technique is able to find relations that may be hidden or suppressed by some strong attributes, as usually happens to standard clustering algorithms.

The biclustering concept can be interpreted in different ways depending on the application. One way (adopted in bicopt-aiNet) is to reorder the rows and columns of the data matrix to create several contiguous regions inside this matrix where the data presents a higher correlation. A different approach (adopted in BIC-aiNet, MOM-aiNet and MOM-aiNet+) is to find subsets of objects from the dataset that correlates with subsets of attributes. This second approach is more powerful and flexible, as it can explore a larger set of potential data relations and these multiple relations among objects and attributes are revealed in parallel and without cross-interference. Also, this second approach makes it possible that one object belongs to more than one bicluster, given a different subset of attributes, thus creating two distinct clusters that hold different knowledge.

In the following sections, these variations of aiNet will be described in detail, being first presented the algorithms developed for optimization and later those for biclustering, according to their chronological order of publication.

Figure 4. Diagram of the evolution of the algorithms derived from aiNet. Dashed arrows indicate algorithms that share some mechanisms. Light gray boxes indicate clustering/biclustering algorithms and dark gray boxes optimization techniques.



## 5. ARTIFICIAL IMMUNE NETWORKS FOR OPTIMIZATION

In this section, the representatives of the optimization branch of the aiNet family of algorithms will be presented. More specifically,

the opt-aiNet (*Artificial Immune Network for Optimization* – de Castro & Timmis, 2002b), copt-aiNet (*Artificial Immune Network for Combinatorial Optimization* – de Souza et al., 2004), dopt-aiNet (*Artificial Immune Network for Dynamic Optimization* – de França et al., 2005a), and omni-aiNet (*Artificial Immune*

*Network for Omni Optimization* – Coelho & Von Zuben, 2006a) algorithms will be detailed, together with results from the literature, examples of applications and a critical analysis of issues and aspects still open in each technique (when it applies).

### Opt-aiNet

The opt-aiNet algorithm (*Artificial Immune Network for Optimization*—de Castro & Timmis, 2002b) was proposed as the first extension of the original aiNet for optimization problems and, due to the presence of diversity maintenance mechanisms introduced in aiNet, this algorithm was developed focusing on multimodal optimization problems.

The opt-aiNet algorithm presents several interesting features: (i) the population size is automatically adjusted during the execution of the algorithm; (ii) opt-aiNet tries to locate multiple optima; and (iii) it is capable of preserving local optimal solutions found throughout the search. When compared to aiNet, the main difference is that opt-aiNet does not present an explicit representation of antigens. Instead, opt-aiNet is based on the direct evaluation of each cell in the population (the affinity to antigens is replaced by the value assumed by the function to be optimized, taking the content of each cell as the argument).

Despite the immune inspiration of the algorithm, opt-aiNet is very similar to evolutionary algorithms, as the main steps of Clonal Selection (cloning, hypermutation and selection) correspond to a micro-evolution that occurs in the immune system and, so, they are modeled in opt-aiNet with operators similar to those already present in evolutionary computing.

One difference of opt-aiNet, when compared to other evolutionary approaches, is that the immune-inspired algorithm is also based on Jerne's network theory, with the presence of suppression mechanisms (and random insertion) among immune cells to stimulate diversity, which allows the algorithm to eliminate redundancy within the population. However, there are also several techniques in the evolutionary

computation literature to introduce/maintain diversity in the population, such as *nicheing* (Sareni & Krähenbühl, 1998), being capable of achieving the same goal of the suppression mechanisms present in opt-aiNet. Nonetheless, the suppression mechanism in opt-aiNet does not affect elitism as most nicheing techniques (the best individuals may have their fitness significantly reduced if they belong to a crowded region), influences the population size, and corresponds to an inherent module born with the original algorithm (and not an appended one). Anyway, a formal and controlled comparison between these techniques is still missing in the literature.

### Structure of the Algorithm

The general structure of opt-aiNet, which works with cells coded as real-valued vectors that represent candidate solutions to the problem, is presented in Pseudo-code 2 (Figure 5). In this algorithm, the user should adjust the following parameters:  $N$ , which is the initial number of individuals in the population;  $N_c$ , the number of clones that should be generated per cell; the suppression threshold  $\sigma_s$ ; and the percentage  $d$  of new randomly generated cells that should be inserted after a suppression step.

From Pseudo-code 2, it is important to highlight the *selection* mechanism, given in step 5. In opt-aiNet, the best cell in each *sub-population* (the set formed by each progenitor cell and their respective mutated clones) is selected to remain in the population, which is an approach different from the one adopted in MOM-aiNet and MOM-aiNet+ algorithms (described later, in Section 6). MOM-aiNet and MOM-aiNet+ maintain several cells for each sub-population.

### Applications

In the original work in which opt-aiNet was proposed (de Castro & Timmis, 2002b), opt-aiNet was applied to several uni- and bi-dimensional multimodal problems, and it was compared to the original CLONALG algorithm (de Castro &

Figure 5. Pseudo code 2

<b>Pseudo-code 2. Main Structure of the opt-aiNet algorithm</b>
<b>Input:</b> - $N$ : the initial number of individuals in the population; - $N_c$ : the number of clones per cell; - $\sigma_s$ : suppression threshold; - $d$ : percentage of new randomly generated cells that should be inserted; - $\beta$ : step of the mutation operator.  Randomly create the initial population, with cells of dimension $L$ (dimension of the problem); <b>While</b> stop criteria not met <b>do</b> 1 - Evaluate each cell in the population; 2 - Generate $N_c$ clones for each cell in the population; 3 - Apply hypermutation to the generated clones, with variability inversely proportional to the progenitor's fitness; 4 - Evaluate each mutated clone; 5 - For each (progenitor + clones), select the fittest individual; <b>If</b> average fitness of the population has improved <b>then</b> 6 - Return to step 2; <b>Else</b> 7 - Determine the affinity among all the cells in the population; 8 - Suppress all but the highest fitness of the cells with affinity higher than a given threshold $\sigma_s$ ; 9 - Introduce a percentage $d\%$ (of the current population size) of new randomly generated cells; <b>End If</b> <b>End While</b>

Von Zuben, 2002a), which does not present the diversity stimulation mechanisms of opt-aiNet. As expected, it was verified that opt-aiNet was able to find many more global and local optima than CLONALG. To illustrate this capability of opt-aiNet, it is shown in Figure 6 the final population of opt-aiNet obtained when applied to a multimodal function given by:

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

and defined in the compact region  $[-1, 2] \times [-1, 2]$ .

Additionally, Figure 7 shows the evolution of the population size throughout the iterations of the algorithm for this problem. It is interesting to notice how the size of the population increases while the algorithm is finding each local optimum of the problem and, after stabilization is achieved, the population size drops to the minimum necessary to represent each local optimum found.

Besides the original application of opt-aiNet to function optimization, this algorithm was also applied to the minimization of mutual information in the blind source separation of post-nonlinear mixtures problem (Duarte et al., 2006), and to several works that deal with *ensemble learning*, in which the diversity maintenance mechanisms of opt-aiNet were studied and explored to generate a set of diverse and accurate artificial neural networks (Coelho & Von Zuben, 2006b; Pasti & de Castro, 2007; Pasti et al., 2009).

## Discussion

As most aiNet-based algorithms, the main drawback of opt-aiNet is the difficulty associated with the adjustment of the suppression threshold  $\sigma_s$ . As it is generally impossible to know a priori what the shape of the search space is, it is hard to predict how the solutions will behave during the search (how close they will be from each other) and so estimate a minimum distance of suppression. Therefore, this parameter must be

Figure 6. Graphical representation of a multimodal function and the proposed solutions obtained by opt-aiNet in a single run

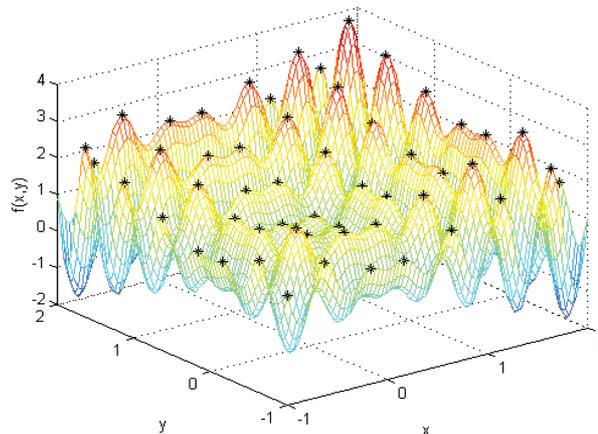
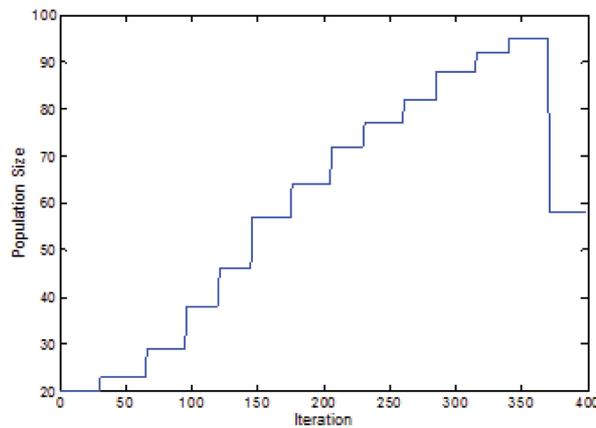


Figure 7. Graphical representation of the evolution of opt-aiNet's population size throughout the iterations, for the problem illustrated in Figure 4



empirically adjusted for each problem, which may be a laborious procedure.

As in the case of all the other aiNet variants, the computational complexity of opt-aiNet is also proportional to the population size in each iteration, or  $O(N.N_c.L)$  when no suppression occurs, where  $N$  is the current population size,  $N_c$  is the number of clones per cell and

$L$  is the dimension of the problem. When the suppression step is applied, the complexity of the algorithm per iteration becomes  $O(N^2.L)$ , whenever  $N > N_c$ .

### Copt-aiNet

Following the aspects introduced by opt-aiNet for continuous optimization, de Souza et al.

(2004) devised an extension of that algorithm to deal with combinatorial optimization problems, which was called *copt-aiNet (Artificial Immune Network for Combinatorial Optimization)*.

### Structure of the Algorithm

The *copt-aiNet* algorithm follows the same structure presented in Pseudo-code 2 with the modifications just on how the inner components are implemented.

First of all, the population of cells (candidate solutions) is modeled as a single vector of integer values, which corresponds to a potential solution to the problem.

The fitness function in *copt-aiNet* depends on the given application, and should be implemented accordingly, to reflect the objective function of the studied problem.

The mutation of the cells is performed by three different permutation operators (de França et al., 2006b), randomly chosen at each iteration:

- **Insertion:** given two elements respectively at positions  $\mathbf{p}_1$  and  $\mathbf{p}_2$ , the element at  $\mathbf{p}_2$  is removed from its original position and inserted before  $\mathbf{p}_1$ , as illustrated in Figure 8a.
- **Exchange:** the two elements at positions  $\mathbf{p}_1$  and  $\mathbf{p}_2$  exchange their positions, as illustrated in Figure 8b.
- **Cross-Exchange:** all the elements between the two points at positions  $\mathbf{p}_1$  and  $\mathbf{p}_2$ , including the end points, are reversed as depicted in Figure 8c.

After mutation, whenever the solutions are not improved for a given number of iterations, it is also performed a *tabu search* (Glover & Kochenberger, 2002) with the moves based on a 2, 3 and 4-exchange of available edges, given a graph representation of the combinatorial problem.

Finally, the last main different internal procedure of *copt-aiNet* (when compared to *opt-aiNet*) is associated with the suppression mechanism, which employs a different strategy

in order to find the similarities between cells. This is performed by counting the minimum number of swap operations that would be necessary to convert a given solution to another (de Souza et al., 2004).

### Applications

In de Souza et al. (2004), the *copt-aiNet* algorithm was applied to perform *Multiple Rearrangements of Gene Expression Data* and some interesting results were found, mainly regarding the diversity of solutions. Table 1 describes the 8 best solutions found by *copt-aiNet* on a biological dataset called *Diffuse Large B-cell Lymphoma*, which comprises 380 genes and 63 experiments. For this problem, the cost to be minimized is associated with the sum of the difference in the expression profile of neighbor genes. Table 1 depicts a distance matrix between solutions, and presents their objective-function values in the last column (cost). As can be seen, despite presenting very different solutions their costs are just slightly different, which means that the algorithm is capable of finding several alternative solutions without significant variation in the quality of the results. Such feature can be very useful in gene expression analysis, as each solution reveals different correlations in the dataset, similarly to what is pursued with biclustering techniques.

Another application was devised in de França et al. (2006a) where *copt-aiNet* was applied to a Dynamic TSP problem where, from time to time (number of iterations), the optimum solution changes to a new location. The experiments not only showed that *copt-aiNet* was capable of finding better local optima than an improved ant colony optimization approach (*IMP.MMAS* – de França et al., 2004a,b; de França et al., 2005b) but also had a faster reactive response to changes, with a prompt detection of the transformation on the environment.

### Discussion

The computational complexity of the *copt-aiNet* algorithm in each iteration is  $O(N \cdot N_c \cdot L)$ ,

Table 1. The difference between the 8 best solutions obtained by *copt-aiNet* on the *Lymphoma* dataset, according to the distance criteria adopted by de Souza et al. (2004)

Solution	1	2	3	4	5	6	7	8	Cost
1	---	102	87	102	99	91	118	107	1827.25
2		---	107	108	102	105	93	115	1827.69
3			---	93	104	108	108	123	1829.07
4				---	99	110	117	117	1829.18
5					---	111	117	100	1829.58
6						---	115	117	1830.44
7							---	130	1830.83
8								---	1831.40

where  $N$  is the current size of the population,  $N_c$  the number of clones and  $L$  the dimension of the problem, and may become  $O(N^2.L^2)$  when suppression is performed and  $N.L > N_c$  (or  $O(N.t)$  when the local search is performed, with  $t$  being the number of local search steps and  $t > N_c.L$ ).

### Dopt-aiNet

In an effort to apply the *opt-aiNet* algorithm to the problem of optimization in dynamic environments, i.e. objective functions that change their optima over time, it was observed that this algorithm presented limitations in some of its mechanisms. To improve the robustness of *opt-aiNet* and to create a fast-reacting algorithm to deal with dynamic problems, the algorithm named *dopt-aiNet* (*Artificial Immune Network for Dynamic Optimization* – de França et al., 2005a) was devised, which not only applies the *aiNet* paradigm to a new scenario, but also improves the inner methods adopted in *opt-aiNet*, like the mutation operator and suppression mechanisms.

### Structure of the Algorithm

The representation of candidate solutions in the population and the initialization of this population work in the same way as in *opt-aiNet*, with cells represented by real coded vectors,

randomly sampled from a uniform distribution in the problem domain at the beginning of the algorithm's execution.

The first drawback identified by de França et al. (2005a) was in *opt-aiNet*'s mutation operator. *opt-aiNet* employs a Gaussian mutation with a fixed parameter ( $\beta$ ) that controls the step size of the mutation (see de Castro & Timmis, 2002b), which is proportional to the value of the objective function and sometimes requires a pre-analysis of the function landscape to be properly set. Therefore, de França et al. (2005a) proposed the dynamic adjustment of this parameter with a linear search known as *golden section* (Bazaraa et al., 1993).

It was also identified that *opt-aiNet*'s Gaussian mutation operator tends to bias the direction of search to only some subset of all possible directions, in high-dimensional problems. So, to deal with this issue, *dopt-aiNet* introduced a second mutation operator, called *One-dimensional Mutation*, which creates a clone population with size equal to the number of dimensions of the problem, and, for each clone, a *Gaussian Mutation* is performed only in one element of the solution, i.e. only in one of the dimensions.

Together with these two mutation mechanisms, a third mutation operator was also proposed, which was called *Gene Duplication*. This operator randomly selects the value in one position of the real-coded solution vector

and replicates it into the remaining positions, only when such copy improves the quality of the solution. This procedure creates a new cell that is inserted into the population.

Another modification included by dopt-aiNet was the replacement of the Euclidean distance adopted in the suppression mechanism of opt-aiNet by what was called *Cell Line Suppression*, as the original metric tends to allow the elimination of solutions in different peaks of the problem's landscape if such problem presents both steep and wide peaks, and the suppression threshold is not suitably adjusted in each case.

With the *Cell Line Suppression*, the distance between two cells **A** and **B** is obtained with the calculation of the distance from the point in the middle of these two solutions to its projection in the straight line that connects **A** and **B**. According to de França et al. (2005a), this metric leads to a higher robustness, considering the suppression threshold, as it can effectively identify when two solutions are on the same optima.

Finally the last modification of dopt-aiNet is the introduction of the multi-population concept. As the dopt-aiNet was created to deal with situations of constantly changing scenarios, it was proposed a memory population where the converged cells (cells that cannot improve anymore) are stored for later usage, which is particularly useful in situations of cyclic changes.

## Applications

A set of eighteen static optimization functions was chosen by de França et al. (2005a) in order to evaluate dopt-aiNet's performance against other traditional evolutionary and immune-inspired algorithms from the literature (the algorithms presented in Angeline (1998), Leung and Wang (2001), Siarry et al. (1997), Timmis et al. (2004) and Yao and Liu (1997)), and the results indicated that dopt-aiNet was able to find the global optimum of each problem, while performing less function evaluations than the contenders (it was evaluated if each algorithm

was able to achieve the optimum of the problem and how many function evaluations were required to accomplish this task). The results of the contenders were taken from the literature.

Considering dynamic optimization problems, in the same work de França et al. (2005a) adopted four well known static optimization problems and modified them so that their optima were displaced at each iteration, with linear, circular and Gaussian movements, while the objective function values of the optima are preserved. In this situation, dopt-aiNet was compared with the *B-Cell Algorithm* (BCA – Timmis et al., 2004) and the PSO proposed by Angeline (1997) (the parameters of the three algorithms were empirically defined) and the results showed that the maximum and average errors with respect to the optimum of the problem remained very low, even lower than those from BCA and PSO.

Additionally, in Junqueira et al. (2005) the dopt-aiNet algorithm was also applied to estimate the coefficients of a FIR equalization filter, on an unsupervised approach with static and time-varying channels, and the obtained results were compared to those produced by a *Genetic Algorithm* with niching and also to those proposed by opt-aiNet. The results showed that dopt-aiNet was able to outperform both algorithms by finding the global optimum 90% of the time, while opt-aiNet found it just 82% of the time and the GA obtained it in less than 50% of the experiments.

Later, in Junqueira et al. (2006), dopt-aiNet was applied to optimize a blind spatial equalization in undermodeled channels, that is, to find the proper parameters of a given spatial filtering technique in order to correctly separate signals associated with multiple users and restore corrupted signals originated from interference and noise. The results showed that, again, dopt-aiNet was able to find solutions very close to the global optimum, both in static and dynamic scenarios.

## Discussion

Besides the dopt-aiNet success on the aforementioned problems and controlled scenarios, this algorithm was recently tested against a proposed set of benchmarking problems

during the *IEEE Congress on Evolutionary Computation* competition on dynamic environments (de França & Von Zuben, 2009). These benchmarking problems tried to comprise several distinct situations that can be faced on dynamic environments. The results, though, showed that dopt-aiNet had some difficulties to deal with such environments, mainly because of its overall computational complexity. In the competition, the number of function evaluations allowed for each scenario was not enough for the algorithm to achieve the moving peaks of the proposed problems. The solution proposed by the authors to deal with such limitations was to work with a very small population size, limiting the number of clones, and reducing the computational effort of the mutation operators. Although the other four competing algorithms also had to limit their operators and population size as well, dopt-aiNet still requires more function evaluations in average than the others to achieve a compatible performance. Indeed, network-based approaches operating with a small population cannot benefit from several aspects of their search power.

Similarly to the opt-aiNet algorithm, dopt-aiNet also has a complexity per iteration of  $O(N \cdot N_c \cdot L)$ , with  $N$  being the current population size,  $N_c$  the number of generated clones and  $L$  the problem dimension and, whenever the suppression step is applied, the complexity becomes  $O(N^2 \cdot L)$  if  $N > N_c$ .

### Omni-aiNet

As mentioned before, the *omni-aiNet* algorithm (*Artificial Immune Network for Omni Optimi-*

*zation* –Coelho & Von Zuben, 2006a) is the extension of aiNet to solve *omni optimization* problems (Deb & Tiwari, 2005) or, in other words, this algorithm is capable of automatically adapting its mechanisms to solve single and multi-objective optimization problems, both uni and multimodal, without requiring any modification in its structure from the user.

### Structure of the Algorithm

As opt-aiNet, omni-aiNet also works with a real-coded population of cells, which corresponds to the candidate solutions for the problem being optimized. The main steps of the algorithm are given in Pseudo-code 3, and the parameters that should be adjusted by the user are:  $N$ , which is the initial number of subpopulations;  $N_c$ , that corresponds to the number of clones that are generated per antibody (and maximum size of each subpopulation);  $N_{gs}$ , the number of iterations between two consecutive suppression and random insertion steps; the suppression threshold  $\sigma_s$ ; the percentage of new random solutions that should be inserted after each suppression step; and  $\eta_{\min}$  and  $\eta_{\max}$ , the minimum and maximum values of the parameter that controls the amplitude of the mutation operator.

The main aspects of the omni-aiNet algorithm, depicted in Pseudo-code 3 (Figure 9), which will be discussed here, are the mutation, selection and suppression mechanisms, which present significant differences when compared to those in the original opt-aiNet.

The main hypermutation operator adopted in omni-aiNet is the *polynomial mutation* mechanism (Deb, 2001), modified so that

Figure 8. The three different mutation operators of *copt-aiNet*: (a) insertion mutation; (b) exchange mutation; and (c) cross-exchange mutation

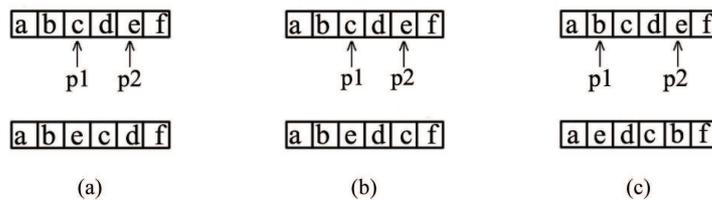


Figure 9. Pseudo-code 3

<b>Pseudo-code 3. Main Structure of the omni-aiNet algorithm</b>
<p><b>Input:</b></p> <ul style="list-style-type: none"> <li>- <math>N</math>: the initial number of individuals in the population;</li> <li>- <math>N_c</math>: the number of clones per cell;</li> <li>- <math>\sigma_s</math>: suppression threshold;</li> <li>- <math>N_{gs}</math>: number of iterations between consecutive suppression steps and random insertions;</li> <li>- <math>d</math>: number of new randomly generated cells that should be inserted;</li> <li>- <math>\eta_{min}</math>: controls the amplitude of mutation (minimum value);</li> <li>- <math>\eta_{max}</math>: controls the amplitude of mutation (maximum value);</li> </ul> <p>Randomly create the initial population with <math>N</math> cells, each one with dimension <math>L</math> (dimension of the problem);</p> <p><b>While</b> stop criteria not met <b>do</b></p> <ol style="list-style-type: none"> <li>1 - Evaluate each cell in the population;</li> <li>2 - Generate <math>N_c</math> clones for each cell in the population;</li> <li>3 - Apply hypermutation to the generated clones, with variability inversely proportional to the progenitor's fitness;</li> <li>4- Combine progenitors and mutated clones into a single population;</li> <li>5 - Perform non-dominance ranking of the population and select the <math>N</math> best cells;</li> <li>6 - Apply the gene duplication operator to the selected cells;</li> </ol> <p style="padding-left: 20px;"><b>If</b> mod(iteration, <math>N_{gs}</math>) == 0 <b>then</b></p> <ol style="list-style-type: none"> <li>7 - Determine the affinity among all the cells in the population (Euclidean distance);</li> </ol> <p style="padding-left: 40px;"><b>If</b> two cells present affinity higher than a given threshold <math>\sigma_s</math> <b>then</b></p> <ol style="list-style-type: none"> <li>8 - Apply a <i>binary tournament</i> between both cells to decide which should be eliminated;</li> </ol> <p style="padding-left: 40px;"><b>End If</b></p> <ol style="list-style-type: none"> <li>9 - Introduce <math>d</math> new randomly generated cells;</li> </ol> <p style="padding-left: 20px;"><b>End If</b></p> <p><b>End While</b></p>

the amplitude of mutation becomes inversely proportional to the quality of each clone (better solutions suffer mutation with smaller variation). Besides polynomial mutation, omni-aiNet also incorporates the *Gene Duplication* mutation operator, already adopted by de França et al. (2005a) in dopt-aiNet. However, differently from dopt-aiNet, as Gene Duplication is computationally expensive (for every individual with  $k$  coordinates, it requires  $k-1$  fitness evaluations in the worst case) and it also requires that all coordinates of each individual are scaled within the same range, Coelho et al. (2010) decided to apply such operator only sporadically, thus reducing the overall computational cost of the algorithm.

Another significant difference, when compared to opt-aiNet, is in the selection

mechanism, performed after each cloning and hypermutation steps. In omni-aiNet, the best solutions from the whole set of parents and mutated clones are kept in the population, even if more than one cell is selected from each set of parent+clones (differently from opt-aiNet, that selects only a single best cell from each set of parents and their mutated clones). However, in this selection mechanism the size of the population is also maintained constant, with the same value that it had before de cloning and hypermutation steps (the mechanisms that allow a dynamic variation of the size of the algorithm's population are the suppression and the random insertion).

The selection mechanism basically ranks all the solutions according to the *constrained  $\epsilon$ -dominance* criterion (Deb & Tiwari, 2005) and

iteratively selects all the individuals of each rank (from the first to the last one) until the desired number of solutions to remain in the population is achieved. If the number of individuals in the rank to be inserted into the population is higher than the number of vacancies, the algorithm employs the *grid procedure* to select the most spread solutions. This grid procedure is similar to the way the NSGA-II algorithm (Deb et al., 2002) calculates the crowding distance metric, and further details can be found in Coelho and Von Zuben (2006a) and Coelho et al. (2010).

Finally, the suppression of similar individuals and insertion of new randomly generated ones are made from  $N_{gs}$  to  $N_{gs}$  generations in omni-aiNet. The value of  $N_{gs}$  should be greater than one to provide enough time for the algorithm to explore the vicinity of each solution before the suppression of similar individuals occurs (this allows the recent inserted cells to improve before suppression). In this suppression phase, the Euclidean distance between every pair of individuals in the population is calculated and scaled with respect to the maximum distance found so far. Two individuals that are closer to each other than a suppression threshold ( $\sigma_s$ ) are subject to a *Binary Tournament*, where the worst cell is eliminated from the population. This Binary Tournament basically adopts the same criteria used in the ranking procedure (constrained  $\varepsilon$ -dominance).

## Applications

The omni-aiNet algorithm was applied to a set of function optimization problems of different types in Coelho and Von Zuben (2006a), to illustrate the capability of the algorithm to adapt its mechanisms to each kind of problem treated. Although no formal and rigorous comparisons were made (the contender algorithms had their parameters empirically adjusted and comparisons were visually made in most of the cases), the results obtained were encouraging and even outperformed those obtained by another proposal of omni optimizer from the literature (Deb & Tiwari, 2005) for two multi-objective optimization problems.

A slightly modified version of omni-aiNet was also applied in a multi-objective approach for phylogenetic reconstruction (Coelho et al., 2007a; Coelho et al., 2007b; Coelho et al., 2010), what led to significant improvements when compared to *Neighbor Joining* (Saitou & Nei, 1987) and *Close-Neighbor-Interchange* (Nei & Kumar, 2000), which are traditional single-objective approaches usually adopted in the literature.

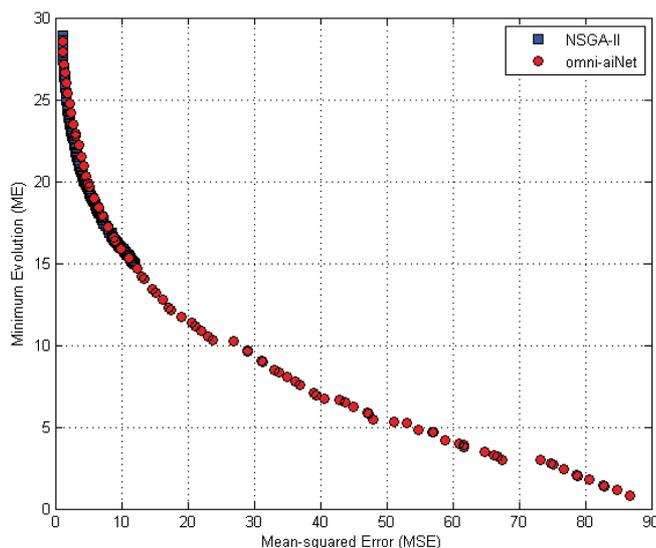
Besides, in Coelho et al. (2010) the NSGA-II algorithm (Deb et al., 2002) was also modified for the same purpose and its results were compared to those obtained by omni-aiNet (both algorithms were run for the same number of objective function evaluations and with the same number of individuals in the initial population). It was observed that, on average, the NSGA-II was capable of obtaining solutions slightly closer to the Pareto front of the problem, but omni-aiNet provided a much broader coverage of the front, as illustrated in Figure 10 (for a single execution of both algorithms, selected randomly from the five repetitions made).

The omni-aiNet algorithm was also applied to *ensemble learning* by Pasti et al. (2009), and its multi-objective mechanisms were explored to evaluate whether the explicit maximization of diversity of the components in *ensembles* (Multi-layer perceptrons for classification problems) resulted in performance gains (smaller classification errors) when compared to those ensembles built with components trained with algorithms that solely try to minimize their individual errors. The obtained results indicate that the multi-objective approach led to ensembles with much more diverse components, but such higher diversity was not directly reflected in a consistent reduction of the ensemble's overall classification error.

## Discussion

Although omni-aiNet presented promising results in the first comparisons made by Coelho and Von Zuben (2006a) and also a great performance in the highly complex phylogenetic reconstruction task (Coelho et al., 2010), further

Figure 10. Illustration of the differences observed in the final results (sets of non-dominated solutions) obtained by omni-aiNet and NSGA-II after a single execution of both algorithms (selected randomly from the five repetitions made) for a phylogenetic reconstruction problem (Coelho et al., 2009). In this problem, the Minimum Evolution and the Mean-squared Error criteria should be minimized.



and deeper analysis of the algorithm's performance and comparisons with state-of-the-art techniques in challenging function optimization problems (single and multi-objective and uni and multimodal) must still be made.

It is possible to notice from Pseudo-code 3 that, differently from opt-aiNet and other variants, omni-aiNet performs a global selection of individuals from the pool of mutated clones and parent cells, which may have some impact in the capability of the algorithm to maintain diversity in the variable space. Therefore, this potential issue may be further investigated and, if it is confirmed, a direct solution would be the adoption of the concept of multiple subpopulations, as it was already made in MOM-aiNet and MOM-aiNet+ algorithms (see Section 6).

The omni-aiNet's computational cost is obviously higher than opt-aiNet's because of the multi-objective operators included into this algorithm, such as the non-dominance ranking and grid procedure. Because of that, the

complexity of each iteration becomes  $O(N^2 \cdot N_c^2 \cdot M)$ , where  $N$  is the current population size,  $N_c$  is the number of clones per solution and  $M$  is the number of objectives of the problem. In the iterations with a suppression step (and given that  $L > N_c^2 \cdot M$ , being  $L$  the dimension of the problem), the complexity of the algorithm becomes  $O(N^2 \cdot L)$ .

## 6. ARTIFICIAL IMMUNE NETWORKS FOR BICLUSTERING

In this section, the extensions of aiNet to deal with the biclustering problem will be presented. More specifically, the biclopt-aiNet (*Artificial Immune Network for Bicluster Combinatorial Optimization* – de França et al., 2006b), BIC-aiNet (*Artificial Immune Network for Biclustering* – Castro et al., 2007a), MOM-aiNet (*Multi-Objective Multipopulation Artificial*

*Immune Network for Biclustering* – Coelho et al., 2008) and MOM-aiNet+ (*Improved Multi-Objective Multipopulation Artificial Immune Network for Biclustering* – Coelho et al., 2009a) algorithms will be detailed, together with results from the literature, examples of applications and discussion of issues and aspects still open in each technique (when it applies).

### Bicopt-aiNet

The first extension of aiNet for biclustering was proposed by de França et al. (2006b) with their *Artificial Immune Network for Bicluster Combinatorial Optimization* (bicopt-aiNet) which, as the name says is based on the copt-aiNet of de Souza et al. (2004).

bicopt-aiNet was proposed to deal with the first type of biclustering problems (see Section 4), those that require the reordering of rows and columns of the data matrix so that regions characterized by high correlation are identified and their elements are grouped.

### Structure of the Algorithm

As bicopt-aiNet is very similar to the *copt-aiNet* algorithm, explained before in Section 5, just the differences between both algorithms will be explained in this section.

First of all, the representation of solutions used in bicopt-aiNet consists of two permutation vectors that represent a possible order of the rows and columns of the data matrix. The mutation operator follows the same procedure of the copt-aiNet algorithm explained in Figure 8, except that it is applied to both permutation vectors of the candidate solution. The suppression and insertion mechanisms are also equivalent to those presented in copt-aiNet.

Regarding the fitness function, de França et al. (2006b) proposed a methodology to reflect the similarities of the region surrounding each element of the reordered data matrix, by calculating the sum of the distances between each element and its surroundings, defined by an  $n \times n$  window frame denoted as *radius of vision* (illustrated in Figure 11).

The main difference between bicopt-aiNet and copt-aiNet is that, in the former, it is applied a *3-opt* local search (as described in Glover & Kochenberger, 2002) at each *local\_it* iterations. This local search performed both in the row and column vectors of the solution, tries to exchange three pairs of consecutive indices of the permutation vectors, in order to reduce the total cost of the objective function.

### Applications

To evaluate the algorithm's capabilities, de França et al. (2006b) built a set of colored bit-map images and presented a scrambled version of them (rows and columns mixed together) to bicopt-aiNet, so that they could be reordered. The goal of these experiments was to find the best image that maximizes the proposed fitness model, based on the Euclidian distance between the RGB values of each pixel of the image according to the "radius of vision". A similar experiment was performed here, and the results are shown in Figure 12. As it is possible to see, the best result found by bicopt-aiNet is visually very similar to the original image.

As a more practical experiment, de França et al. (2006b) proposed the application of bicopt-aiNet to a gene ordering problem. This problem consists in finding the best order of the rows and columns that would result in an easier retrieval of knowledge and object relations from the dataset. In Figure 13 we can visually see the differences between an unprocessed dataset (Figure 13a) and the data reordered by bicopt-aiNet (Figure 13b). The latter can help the data analyst to infer, among other possibilities, which objects are more related with each other and which features are more useful in this data.

### Discussion

Regarding the fitness of the solutions found both by bicopt-aiNet and by a Hierarchical Clustering algorithm for the gene reordering problems in de França et al. (2006b), bicopt-aiNet's solutions seemed to perform much better. However, visually those differences are not

Figure 11. The objective function for the biclustering problem as modeled in bicopt-aiNet. The  $n \times n$  window is defined by a radius of vision: with radius = 1 the calculation is made with the neighbors marked with "B", with radius = 2 the calculation is made with the "B's" and the "C's", and so on.

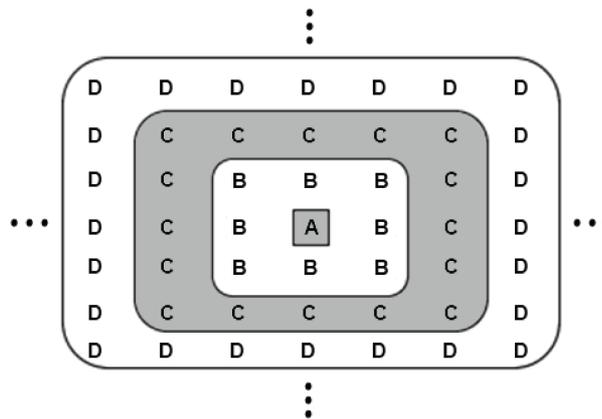
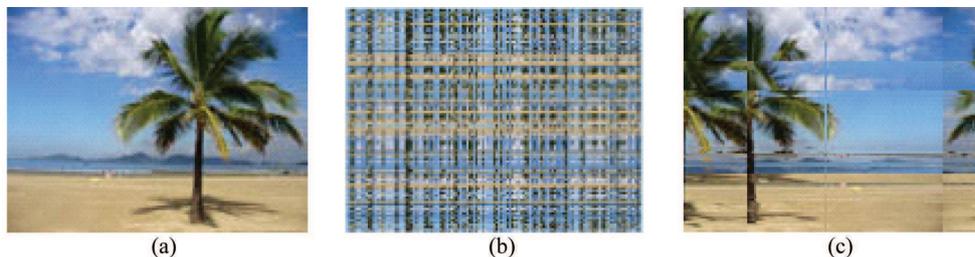


Figure 12. A test image for bicopt-aiNet: (a) original color bitmap image; (b) the scrambled image presented to the algorithm; (c) the best result found



so evident, what may be caused by the fitness function itself that may not be adequate to this particular application.

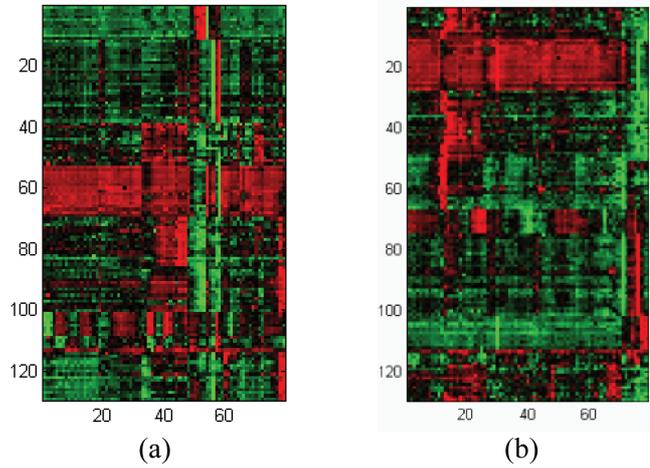
The computational cost at each iteration of bicopt-aiNet also depends on the current population size, so its complexity is  $O(N \cdot N_c \cdot \max(n, m))$ , where  $N$  is the current population size,  $N_c$  is the number of clones that should be generated for each cell and  $n \times m$  is the problem dimension. When considering the suppression step of the algorithm, this complexity may become  $O(N^2 \cdot \max^2(n, m))$  whenever  $N \cdot \max(n, m) > N_c$ .

### BIC-aiNet

In 2007, Castro et al. (2007a) proposed another extension of the aiNet framework for biclustering, which was named *BIC-aiNet (Artificial Immune Network for Biclustering)*. However, differently from bicopt-aiNet, BIC-aiNet was proposed to deal with the second approach of the biclustering problem (see Section 4).

This analysis makes it possible that one object belongs to more than one cluster, given a different set of attributes, or, in other words,

Figure 13. A test case of bicopt-aiNet on a data reordering problem with (a) representing the original data matrix and (b) the reordered data



an object **A** may relate to an object **B** with a given subset of attributes and may relate to an object **C** as well, but with another set of attributes, thus creating two distinct clusters that hold different knowledge.

In short, this biclustering approach requires that the algorithm should be capable of generating sets of biclusters that represent most of the available data (high coverage), that each bicluster has a high similarity among its elements, a maximal volume (denomination of the size of the bicluster, based on the number of elements times the number of attributes) and, when dealing with sparse datasets, a high occupancy (low number of null elements in the bicluster). And these were the goals of Castro et al. (2007a) with BIC-aiNet.

### Structure of the Algorithm

In BIC-aiNet, the similarity metric used to compare elements inside a bicluster was the *Mean Squared Residue* (Cheng & Church, 2000) given by Eq. 2.

$$R = \sum_{i,j} \frac{(a_{ij} - a_{i\cdot} - a_{\cdot j} + a_{\cdot\cdot})^2}{|N| \cdot |M|} \quad (1)$$

where  $N$  and  $M$  are the total number of rows and columns, respectively, in the bicluster,  $a_{ij}$  is the value of the element  $(i,j)$ ,  $a_{i\cdot}$  is the mean value of column  $j$ ,  $a_{\cdot j}$  represents the mean value of row  $i$ , and  $a_{\cdot\cdot}$  is the mean value considering all the elements of the bicluster.

BIC-aiNet follows again the same content described in Pseudo-code 1 and the modifications will be explained in what follows.

Starting with the initialization, BIC-aiNet creates a population of biclusters with just one row and one column, that is, it starts with a small bicluster and tries to improve it by increasing its volume while attending the maximum mean squared residue constraint.

The representation of solutions in the cell population consists of two vectors containing the set of row and column indices belonging to the bicluster. Notice that these vectors have variable size and are constructed in this way in order to make the algorithm more efficient.

BIC-aiNet's representation will only require, for the whole population, a memory size proportional to the size of the whole dataset when the biclusters cover all the data (worst case), with an eventual overhead added by overlaps. The reason is that each cell will hold just a small part of the dataset, generally different

from those of other cells due to different similarities within the dataset and overlap constraints (that will be discussed later). However, this memory reduction increases the computational cost associated with insertions and removals of elements in the solution vectors.

The fitness function adopted in BIC-aiNet is a weighted sum of the objectives of the biclustering problem (mean squared residue and size of the bicluster) and is given by:

$$f(M, N) = \left[ \frac{R}{\lambda} + \frac{w_c \cdot \lambda}{|M|} + \frac{w_r \cdot \lambda}{|N|} \right]^{-1}, \quad (1)$$

where  $\lambda$  is a parameter that penalizes solutions with high residue values,  $w_c$  is the importance of the number of columns and  $w_r$  the importance of the number of rows in the bicluster. The operator  $|\cdot|$  provides the number of elements in a given vector. Parameters  $\lambda$ ,  $w_c$  and  $w_r$  should be defined by the user for each problem.

Considering the mutation operator, according to Castro *et al.* (2007a) it was conceived to incrementally try to improve the bicluster and, for that matter, it is divided into three steps:

- *Insert one row in the bicluster;*
- *Insert one column in the bicluster;*
- *Remove one row or column from the bicluster.*

Only one of these three steps is applied to each clone at each iteration of BIC-aiNet, and they are chosen randomly with the same probability. Also, the element of the dataset that will be inserted or removed is randomly chosen from a set of possible elements.

Differently from aiNet, in BIC-aiNet the diversity maintenance mechanisms (suppression and insertion of new cells) are applied from time to time, and not in all iterations (as in dopt-aiNet and omni-aiNet – see Section 5), in order to allow the solutions to evolve before eventual suppressions are made.

The suppression method of BIC-aiNet calculates the amount of elements that are com-

mon in each pair of biclusters in the population and, when two biclusters have a percentage of intersection above a pre-defined threshold, the one with lower fitness is removed from the population.

## Applications

The BIC-aiNet algorithm was applied in a Collaborative Filtering application (Castro *et al.*, 2007a; Castro *et al.*, 2007b) and to perform Text Mining (Castro *et al.*, 2007c; Castro *et al.*, 2009).

In the first application, the algorithm was tested on a diverse set of item-recommendation problems and compared against other well known approaches to the recommendation problem such as Pearson Correlation (Sarwar *et al.*, 2001), Bayesian Networks (Jensen, 2001), Naïve Bayes (Miyahara & Pazzani, 2000), a Probabilistic Memory-based Collaborative Filtering (Yu *et al.*, 2004) and a biclustering method called Nearest-Biclusters (Symeonidis *et al.*, 2007). The results were evaluated according to the quality of prediction, as each algorithm is based on different frameworks. The comparisons showed that, given the capability to relate a subset of objects to just a subset of attributes, BIC-aiNet presented better performance (smaller prediction error) when compared to the other methods. The identification of these partial relations between objects and attributes that may be present in the datasets showed to be particularly useful for this problem, as most of the time the items in the dataset could be automatically divided into several different categories.

In the text mining application, BIC-aiNet was able to identify how subsets of words are related to different categories of documents. In Castro *et al.* (2007c), the selected words not only could differentiate documents belonging to different subjects like sports, informatics and economy, but could also find terms related to groups of documents from different classes, such as informatics and economy, for example.

In Castro *et al.* (2009), BIC-aiNet was adopted as the basis of a *Query Expansion*

*Algorithm.* In this work, when a user makes a query to the system, it suggests several terms that are related to such query but that may lead the results to a whole different set of documents. This work showed that BIC-aiNet could find and suggest coherent terms and, when compared to a simple term-term similarity approach and to the usage of cosine similarity to find the words correlated to a query term, BIC-aiNet led to alternative sets of suggestions. Therefore, Castro et al. (2009) indicate that a combination of the suggestions made by BIC-aiNet with the suggestions made by the other approaches may lead to an increased performance, although this proposal was not implemented.

## Discussion

Besides the good performance obtained by BIC-aiNet on the experiments performed, this algorithm is highly susceptible to wrong parameter settings. As most of its parameters control the shape of the fitness function, they require a precise tuning in order to lead to the desired characteristics of the set of biclusters (i.e., higher volume, lower residue, better coverage, etc.). Therefore, a more robust fitness function would be certainly beneficial to this algorithm.

The computational cost of each iteration of BIC-aiNet is  $O(N \cdot N_c \cdot \max(n, m))$ , where  $N$  is the current population size,  $N_c$  is the number of clones to be generated per cell and  $n \times m$  is the dimension of the dataset, or  $O(N^2 \cdot \max(n, m))$  whenever suppression occurs and  $N > N_c$ .

## MOM-aiNet and MOM-aiNet+

As the BIC-aiNet algorithm, MOM-aiNet (*Multi-Objective Multipopulation Artificial Immune Network for Biclustering* – Coelho et al., 2008) and MOM-aiNet+ (*Improved Multi-Objective Multipopulation Artificial Immune Network for Biclustering* – Coelho et al., 2009a) are also variants of the original aiNet developed for the second approach of biclustering. However, differently from BIC-aiNet, both MOM-aiNet and MOM-aiNet+ include concepts of

multi-objective optimization (Deb, 2001) and multipopulation search.

In MOM-aiNet and MOM-aiNet+, the biclustering problem is seen as a two-objective optimization problem, where the mean squared residue of the biclusters (Eq. 2) should be minimized while their volume should be maximized.

## Structure of the Algorithm

Besides the interpretation of biclustering as a multi-objective optimization problem, MOM-aiNet and MOM-aiNet+ also further explore the *multipopulation aspect* of aiNet. The aiNet algorithm evolves multiple cells (individuals) in parallel and, associated with each of these cells, there is a population of mutated clones. Each of these subpopulations is stimulated to converge to distinct promising regions of the search space (generally to the closest local optimum) but, at each iteration of the algorithm, only the best individual of each subpopulation is selected to constitute the population of the next generation. In MOM-aiNet and MOM-aiNet+, not only the best individual of each subpopulation is kept in the population, but several locally non-dominated sets of individuals. Also, in the end of their execution, MOM-aiNet and MOM-aiNet+ algorithms return not only the global set of non-dominated individuals, but all the non-dominated solutions within each sub-population. With this approach, according to Coelho et al. (2008) and Coelho et al. (2009a), the final set of solutions may provide a higher coverage of the data matrix and may also contain sub-optimal biclusters (according to the mean-squared residue and volume criteria) that still represent important correlations of the dataset that were not identified by the “optimal” set of solutions (Coelho et al., 2009b).

When compared to each other, MOM-aiNet and MOM-aiNet+ differ on the presence, in the latter, of mechanisms that transform unfeasible biclusters (with mean squared residue above the upper constraint imposed by the user) into feasible ones, and also mechanisms that add

nodes that do not increase the mean squared residue into the biclusters, so that their volume is improved. These two mechanisms added to MOM-aiNet+ are based on mechanisms of Cheng & Church's popular CC algorithm (Cheng & Church, 2000).

The main steps of both MOM-aiNet and MOM-aiNet+ algorithms are given in Pseudo-code 4, being those steps exclusive to MOM-aiNet+ marked with an '(\*)'. The cells in the population of these algorithms are candidate biclusters, and they are encoded in the same way as in BIC-aiNet. The user must adjust the following parameters in MOM-aiNet and MOM-aiNet+:  $N$ , which is the initial number of subpopulations;  $N_c$ , that corresponds to the number of clones that are generated per antibody (and maximum size of each subpopulation);  $N_{gs}$ , the number of iterations between two consecutive suppression and random insertion steps;  $N_{ls}$ , which is the number of iterations between two consecutive multiple node insertions;  $\sigma_s$ , the suppression threshold;  $\delta$ , the upper residue constraint; and  $\gamma$ , which is the lower occupancy constraint (for sparse datasets).

The main aspects of MOM-aiNet and MOM-aiNet+ that should be highlighted from Pseudo-code 4 (Figure 14) are the constraints controlled during the optimization and the selection and suppression mechanisms, which are common to both algorithms.

There are two constraints controlled in MOM-aiNet and MOM-aiNet+: the first one is an upper limit  $\delta$  for the mean squared residue, being every bicluster with a residue higher than  $\delta$  considered dominated by any other feasible bicluster; and the other is the occupancy rate (for sparse datasets), which measures the proportion of non-null values in the bicluster. When an insertion action is chosen by the mutation process, the set of candidate rows/columns to be inserted is reduced to those that keep the bicluster occupancy rate above a threshold  $\gamma$ . With the explicit control of these constraints, both MOM-aiNet and MOM-aiNet+ allow the user to easily define which are the desired characteristics of the set of biclusters to be obtained by the algorithms.

After the mutation step, performed on each clone generated to compose a subpopulation, all the non-dominated biclusters of this subpopulation, considering the mutated clones and original cells, are selected to generate the new subpopulation for the next iteration. If the number of non-dominated elements exceed  $N_c$ , a crowding-distance-based (Deb et al., 2002) suppression is performed, in order to maintain a controlled-size and locally diverse subpopulation.

Finally, in this suppression phase of the algorithms (which is performed from  $N_{gs}$  to  $N_{gs}$  iterations), the largest biclusters of each subpopulation are compared, based on the degree of overlap among them (intersection of nodes – pairs of rows and columns – in each bicluster) and, when a pair of biclusters has a degree of overlapping higher than a given threshold  $\sigma_s$ , the two subpopulations are merged and the non-dominance selection is performed, creating a single subpopulation. Two subpopulations will only be merged when they start to express the same correlations among rows and columns of the data matrix or, in other words, they will only be merged when they start to converge to the same region of the data matrix.

## Applications

Both MOM-aiNet and MOM-aiNet+ have been applied to gene expression and collaborative filtering datasets (Coelho et al., 2008; 2009a; b), in order to have their characteristics explored and understood. Both algorithms were extensively compared to popular biclustering techniques from the literature, such as CC (Cheng & Church, 2000), FLOC (Jiong et al., 2003), the multi-objective-based algorithm of Mitra & Banka (Mitra & Banka, 2006) and even BIC-aiNet (Castro et al., 2007a;b;c).

The results obtained showed that, even though MOM-aiNet explicitly optimizes just the volume and mean square residue of the biclusters, it was also able to maximize the coverage of the dataset obtained by the biclusters, mainly due to its multipopulation aspect and suppression mechanism, which inhibits two

Figure 14. Pseudo-code 4

Main structure of the MOM-aiNet and MOM-aiNet+ algorithms. Steps that are exclusive of MOM-aiNet+ are marked with (\*).

---

**Pseudo-code 4.**

**Input:**

- $N$ : the initial number of subpopulations;
- $N_c$ : the number of clones per antibody (and maximum size of each subpopulation);
- $N_{gs}$ : number of iterations between consecutive suppression steps and random insertions;
- $N_{ls}$ : number of iterations between consecutive multiple node insertions;
- $\sigma_s$ : suppression threshold;
- $\delta$ : residue constraint;
- $\gamma$ : occupancy constraint (for sparse datasets);
- Data matrix of size  $n \times m$ .

Randomly create  $N$  initial subpopulations with a single cell;

**While**  $iteration < \text{maximum number of iterations}$  **do**

**For** each subpopulation  $i$  **do**

1 - Evaluate each cell in subpopulation  $i$ ;

2 - Generate  $N_c$  clones for each cell in the population;

3 - Apply hypermutation to the generated clones (with BIC-aiNet's mutation operator);

(\*)**For** each unfeasible clone  $j$  **do**

(\*)4 - Make clone  $j$  feasible;

(\*)**End For**

5 - Combine progenitors and mutated clones in a single subpopulation;

6 - Select all the non-dominated cells (maximum  $N_c$  cells) to remain in the subpopulation;

**End For**

**If**  $\text{mod}(iteration, N_{gs}) == 0$  **then**

7 - Determine the affinity among the largest cells in each subpopulation (those with the largest volume);

**If** two of these cells present affinity higher than a given threshold  $\sigma_s$  **then**

8 - Merge the two subpopulations;

9 - Select the non-dominated cells (maximum  $N_c$  cells);

**End If**

10 - Introduce new randomly generated cells;

**End If**

(\*)**If**  $\text{mod}(iteration, N_{ls}) == 0$  **then**

(\*)11 - Perform multiple node insertion, to increase the volume of the biclusters;

(\*)**End If**

**End While**

---

distinct populations from exploring similar regions of the search space. Because of the suppression mechanism, MOM-aiNet was also able to maintain the degree of overlap among biclusters (the intersection in their elements) under control.

Considering the comparisons with the other techniques, MOM-aiNet could not always find the maximum volume on dense datasets, which is again due to the effect of the suppression mechanism that inhibits two populations from growing indefinitely and intersecting with each other. The most noticeable advantage of MOM-aiNet was observed on sparse datasets,

where the algorithm was able to find the best trade-off among the objectives. Sparse datasets are usual, for instance, in applications involving recommender systems (Adomavicius & Tuzhilin, 2005).

Considering now a direct comparison between MOM-aiNet+ and MOM-aiNet (Coelho et al., 2009a), the experiments performed have shown that MOM-aiNet+ was able to increase the volume of the final set of solutions (when compared to MOM-aiNet's results) without any significant impact in the mean squared residue. However, the improvement in volume also led to a reduction in the coverage of the

dataset and an increase in the degree of overlap among the biclusters in one of the dense datasets studied.

## Discussion

Although it gives a better control of the desired aspects of the resulting set of biclusters, and also allows an easier adjustment of the set of parameters (i.e., each parameter is very explicit on what feature of the bicluster it influences), the multi-population approach requires a higher computational effort to find a proper set of solutions. However, as its inner procedures can be easily parallelized on a multi-core or multi-processor environment, this algorithm can still be applied without much concern.

Also, because of the multi-objective nature of the biclustering problem, explicitly explored in MOM-aiNet and MOM-aiNet+, it is still difficult to infer right away which algorithm would be more suitable to a given problem, so the development of a controlled framework for experiments and comparison of different algorithms for distinct biclustering problems is still lacking.

When the computational cost per iteration is considered, if no suppression is performed MOM-aiNet tends to present a slightly smaller complexity –  $O(N \cdot N_c \cdot \max(n, m))$ , where  $N$  is the population size,  $N_c$  is the size of each sub-population, and  $n \times m$  is the dataset dimension – when compared to MOM-aiNet+ –  $O(N \cdot N_c \cdot (n + m))$ . However, when suppression is performed, both algorithms present the same complexity, which is  $O(N \cdot N_c^2 \cdot (m+n))$  when all the sub-populations are merged and  $O(N^2 \cdot (m+n))$  when no merging is performed.

## 7. FINAL COMMENTS

Artificial Immune Systems (AISs) are a computational paradigm inspired by the immunological system of vertebrates and designed to solve a wide range of problems, such as optimization, clustering, pattern classification and computational security.

Several immune-inspired algorithms have been proposed in the literature, and they differ mainly on the immunological metaphor used as inspiration. Two important principles from immunology are the Clonal Selection and the Immune Network theories that, when computationally implemented, seem to allow the algorithms to adjust automatically the size of their populations along the execution and to maintain diversity in the population, thus providing a better ratio between the capabilities of exploration and exploitation of the search space. However, besides some empirical evidences observed in the literature, there is still no published work that provides a thorough comparison with other traditional diversity control mechanisms and that theoretically verifies such properties.

Almost one decade ago, de Castro and Von Zuben (2002b) developed the *Artificial Immune Network* algorithm (aiNet), based on both Clonal Selection and Immune Network theories, to solve clustering problems and data compression. Following the empirical indications of interesting properties that was raised by aiNet, many extensions appeared in the literature, not only for clustering but also for continuous optimization, combinatorial optimization, multi-objective optimization, dynamic optimization, biclustering and recommender systems.

In the Computational Intelligence field, aiNet's research area is classified as a general purpose meta-heuristic and it is often compared to similar bio-inspired approaches like Evolutionary Algorithms, Particle Swarm Optimization and Ant Colony Optimization. Although there are significant similarities among these paradigms, as aiNet can also be interpreted as a populational evolutionary approach and a multi-agent environment, the aiNet family of algorithms also present some unique features, like an innate diversity maintenance mechanism imbued into the meta-heuristic, the capability to self-regulate the population size – in the sense of estimating the minimum amount of resources to implement the search – and, finally, the ability to present several high-quality solutions at the end of its execution, which tends to provide to

the user a more robust set of choices. There is also several diversity maintenance mechanisms proposed in the Evolutionary Computational literature, but such approaches are considered as additional mechanisms that should be included in the algorithms whenever diversity is required, differently from aiNet's framework, in which this characteristic is intrinsic.

The goal of this paper was to review and discuss the publications of the original aiNet algorithm and its extensions, describing the conceptual aspects, the drawbacks and open issues associated with each algorithm, the results obtained so far and the applications already made. The algorithms reviewed in this work and their main applications are summarized in Table 2, and the overall computational complexity of each of them is summarized in Table 3.

From this review, it was interesting to notice that the same immune-inspiration led to the development of several general techniques capable of effectively solving a broad set of problems, which may also be the reason that several open issues associated with these techniques (not only specific to each algorithm, but also to the framework itself) were also identified. Therefore, as the results obtained with the application of the aiNet-based techniques indicate that these algorithms present high potential, we believe that the scientific community may significantly benefit from an effort to better understand all the nuances of each technique and to comprehend and solve their open issues, which are summarized in what follows.

To begin with, besides the work of Stibor and Timmis (2007), there are no theoretical studies on the mechanisms of these algorithms, nor deeper controlled experimental comparisons of each specific operator proposed in the algorithms with different approaches from the literature (for example, there are no comparisons of aiNet's diversity maintenance mechanisms with niching, focusing on the diversity issue itself).

Another important aspect that must be verified is whether the good empirical results reported so far in the literature are due to the framework itself or to the operators adopted.

In other words, it should be studied whether a given operator benefits from the context that it is inserted into the meta-heuristic or the meta-heuristic performs well only due to its operators.

Considering the algorithms in general, there are several future perspectives besides the possibility of further improvements. For example, most of them still require a more consistent experimentation, to identify what behavior they present in different situations (problems with distinct characteristics) and to better understand the exact impact that each operator has in its overall performance. Besides, further controlled comparisons with different state-of-the-art techniques from the literature haven't been made yet for some of these algorithms.

As it was observed throughout this review, the aiNet-based algorithms tend to present a large amount of parameters that must generally be adjusted empirically for each problem. Besides that, these algorithms are very sensitive to some of these parameters so that an improper set up may lead to unpredictable behaviors. Therefore, another important future perspective is the development and inclusion, in the algorithms, of mechanisms capable of self adapting to each problem, thus reducing the overhead caused by such large amount of parameters.

Another possibility is associated with the incorporation of different biologically-inspired aspects and mechanisms that may introduce interesting properties to the algorithms and, at the same time, increase their biological relevance. For example, some tendencies in this area seems to be related to how the network of cells are organized, structured and stimulated in order to improve the overall performance of the remaining inner mechanisms of the algorithms, like those responsible for the diversity maintenance and search for local solutions.

Finally, another important perspective of research lies in the parallelization of each algorithm so that their inherent parallel mechanisms can explore the actual availability of multi-core systems, which may optimize the execution of

Table 2. Summary of the algorithms reviewed in this work

Algorithm	Application	Case Studies
aiNet	Clustering/Classification	Prediction of Landslides (Li et al., 2010), Clustering in Bioinformatics (Bezerra & de Castro, 2003; Bezerra et al., 2004), Learning of RBF Networks (de Castro & Von Zuben, 2001), Document Clustering (Ciesielski et al., 2006; Xu et al., 2006), Compositional Timbre Design (Caetano et al., 2005).
opt-aiNet	Numerical Optimization	Blind Source Separation of Post-nonlinear Mixtures (Duarte et al., 2006), Generation of Diversity on Neural Networks (Coelho & Von Zuben, 2006b; Pasti & de Castro, 2007; Pasti et al., 2009)
copt-aiNet	Combinatorial Optimization	Gene Expression Reordering (de Souza et al., 2004)
dopt-aiNet	Numerical Dynamic Optimization	FIR Equalization Filter (Junqueira et al., 2005), Blind Spatial Equalization in Undermodeled Channels (Junqueira et al., 2006)
omni-aiNet	Omni Numerical Optimization	Phylogenetic Reconstruction (Coelho et al., 2007a; Coelho et al., 2007b; Coelho et al., 2010), Ensemble Learning (Pasti et al., 2009)
bicopt-aiNet	Biclustering (reordering)	Gene Expression, Image Reconstruction (de França et al., 2006b)
BIC-aiNet	Biclustering	Gene Expression (Coelho et al., 2008, 2009a, b), Recommender Systems (Castro et al., 2007a; Castro et al., 2007b), Text Mining (Castro et al., 2007c; Castro et al., 2009)
MOM-aiNet MOM-aiNet+	Biclustering	Gene Expression (Coelho et al., 2008; 2009a, b)

Table 3. Summary of the overall computational complexity per iteration of each algorithm reviewed in this work (with and without execution of the suppression step). The same convention adopted in the text is followed here, with  $N$  being the population size,  $N_c$  is the number of clones in the iteration,  $L$  is the problem dimension,  $M$  is the number of objectives (in multi-objective optimization), and  $n$  and  $m$  are the dimensions of the data set.

Complexity per iteration		
Algorithm	Without Suppression	With Suppression
aiNet	–	$O(N^2.L)$
opt-aiNet	$O(N.N_c.L)$	$O(N^2.L)$ , for $N > N_c$
copt-aiNet	$O(N.N_c.L)$	$O(N^2.L^2)$ , for $N.L > N_c$
dopt-aiNet	$O(N.N_c.L)$	$O(N^2.L)$ for $N > N_c$
omni-aiNet	$O(N^2.N_c^2.M)$	$O(N^2.L)$ for $L > N_c^2.M$
bicopt-aiNet	$O(N.N_c.\max(n,m))$	$O(N^2.\max^2(n,m))$ for $N.\max(n,m) > N_c$
BIC-aiNet	$O(N.N_c.\max(n,m))$	$O(N^2.\max(n,m))$ for $N > N_c$
MOM-aiNet	$O(N.N_c.\max(n,m))$	$O(N^2.(m+n))$ (worst case)
MOM-aiNet+	$O(N.N_c.(n+m))$	$O(N^2.(m+n))$ (worst case)

these algorithms so that their overall complexity can be reduced.

## ACKNOWLEDGMENT

This research was supported by CNPq and CAPES. Correspondence concerning this article should be addressed to Fabrício O. de França, Laboratory of Bioinformatics and Bioinspired Computing (LBiC), Department of Computer Engineering and Industrial Automation (DCA), School of Electrical and Computer Engineering (FEEC), University of Campinas (Unicamp), P. O. Box 6101, Campinas, SP, 13083-970–Brazil. E-mail: olivetti@dca.fee.unicamp.br

## REFERENCES

- Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734–749. doi:10.1109/TKDE.2005.99
- Agrawal, R., Gehrke, J., Gunopulus, D., & Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM/SIGMOD International Conference on Management of Data*, Seattle, WA (pp. 94-105).
- Angeline, P. J. (1997). Tracking extrema in dynamic environments. In *Proceedings of the 6th International Conference on Evolutionary Programming VI*, Indianapolis, IN (pp. 335-345).
- Angeline, P. J. (1998). Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. In *Proceedings of the 7th International Conference on Evolutionary Programming VII*, London (pp. 601–610).
- Bazaraa, M. S., Sherali, H. D., & Shetty, C. M. (1993). *Nonlinear Programming: Theory and Algorithms* (2nd ed.). New York: Wiley.
- Behn, U. (2007). Idiotypic networks: toward a renaissance? *Immunological Reviews*, 216(1), 142–152.
- Bersini, H., & Varela, F. (1990). Hints for adaptive problem solving gleaned from immune networks. In *Proceedings of the Parallel Problem Solving from Nature, 1st Workshop 233 PPSN I* (pp. 343-354). Berlin: Springer.
- Bezerra, G. B., Barra, T. B., de Castro, L. N., & Von Zuben, F. J. (2005). Adaptive radius immune algorithm for data clustering. In *Proceedings of the 4th International Conference on Artificial Immune Systems (ICARIS)*, Banff, Canada (pp. 290-303).
- Bezerra, G. B., & de Castro, L. N. (2003). Bioinformatics data analysis using an artificial immune network. In *Proceedings of the 2nd International Conference on Artificial Immune Systems (ICARIS)*, Edinburgh, UK (pp. 22-33).
- Bezerra, G. B., de Castro, L. N., & Von Zuben, F. J. (2004). A Hierarchical Immune Network Applied to Gene Expression Data. In *Proceedings of 3rd International Conference on Artificial Immune Systems (ICARIS)*, Catania, Italy (pp. 14-27).
- Burnet, F. M. (1978). Clonal Selection and After . In Bell, G. I., Perelson, A. S., & Pimgley, G. H. Jr., (Eds.), *Theoretical Immunology* (pp. 63–85). New York: Marcel Dekker.
- Caetano, M., Manzolli, J., & Von Zuben, F. J. (2005). Application of an Artificial Immune System in a Compositional Timbre Design Technique. In *Proceedings of 4th International Conference on Artificial Immune Systems (ICARIS)*, Banff, Canada (pp. 389-403).
- Castro, P. A. D., de França, F. O., Ferreira, H. M., Coelho, G. P., & Von Zuben, F. J. (2009). *Query Expansion Using An Immune-inspired Biclustering Algorithm. Natural Computing*. New York: Springer.
- Castro, P. A. D., de França, F. O., Ferreira, H. M., & Von Zuben, F. J. (2007a). Applying Biclustering to Perform Collaborative Filtering. In *Proceedings of the 7th International Conference on Intelligent Systems Design and Applications (ISDA)*, Rio de Janeiro, Brazil (pp. 421-426).
- Castro, P. A. D., de França, F. O., Ferreira, H. M., & Von Zuben, F. J. (2007b). Evaluating the Performance of a Biclustering Algorithm Applied to Collaborative Filtering – A Comparative Analysis. In *Proceedings of the 7th International Conference on Hybrid Intelligent Systems (HIS)*, Kaiserslautern, Germany (pp. 65-70).

- Castro, P. A. D., de França, F. O., Ferreira, H. M., & Von Zuben, F. J. (2007c). Applying Biclustering to Text Mining: An Immune-Inspired Approach. In *Proceedings of the 6th International Conference on Artificial Immune Systems (ICARIS)*, Santos, Brazil (pp. 83-94).
- Cheng, Y., & Church, G. M. (2000). Biclustering of Expression Data. In *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology*, San Diego, CA (pp. 93-103).
- Ciesielski, K., Wierzchoń, S. T., & Kłopotek, M. A. (2006). An Immune Network for Contextual Text Data Clustering. In *Proceedings of 6th International Conference on Artificial Immune Systems (ICARIS)*, Oeiras, Portugal (pp. 432-445).
- Coelho, G. P., de França, F. O., & Von Zuben, F. J. (2008). A Multi-Objective Multipopulation Approach for Biclustering. In *Proceedings of 7th International Conference on Artificial Immune Systems (ICARIS)*, Phuket, Thailand (pp. 71-82).
- Coelho, G. P., de França, F. O., & Von Zuben, F. J. (2009a). Improving a Multi-Objective Multipopulation Artificial Immune Network for Biclustering. In *Proceedings of the 2009 IEEE Congress on Evolutionary Computation (CEC)*, Trondheim, Norway (pp. 2748 - 2755).
- Coelho, G. P., de França, F. O., & Von Zuben, F. J. (2009b). Multi-Objective Biclustering: When Non-dominated Solutions are not Enough. *Journal of Mathematical Modelling and Algorithms*, 8(2), 175–202. doi:10.1007/s10852-009-9102-8
- Coelho, G. P., Silva, A. E. A., & Von Zuben, F. J. (2007a). Evolving phylogenetic trees: A multiobjective approach. In *Proceedings of the Brazilian Symposium on Bioinformatics (BSB)*, Angra dos Reis, Brazil (pp. 113-125).
- Coelho, G. P., Silva, A. E. A., & Von Zuben, F. J. (2007b). A Multiobjective Approach to Phylogenetic Trees: Selecting the Most Promising Solutions from the Pareto Front. In *Proceedings of the 7th International Conference on Intelligent Systems Design and Applications (ISDA)*, Rio de Janeiro, Brazil (pp. 837-842).
- Coelho, G. P., Silva, A. E. A., & Von Zuben, F. J. (2010). An Immune-Inspired Multi-Objective Approach to the Reconstruction of Phylogenetic Trees. *Journal of Neural Computing and Applications*.
- Coelho, G. P., & Von Zuben, F. J. (2006a). omni-aiNet: An Immune-inspired Approach for Omni Optimization. In *Proceedings of the 5th International Conference on Artificial Immune Systems (ICARIS)*, Oeiras, Portugal (pp. 294-308).
- Coelho, G. P., & Von Zuben, F. J. (2006b). The Influence of the Pool of Candidates on the Performance of Selection and Combination Techniques in Ensembles. In *Proceedings of the 2006 IEEE International Joint Conference on Neural Networks (IJCNN)*, Vancouver, Canada (pp. 10588-10595).
- Dasgupta, D. (Ed.). (1998). *Artificial Immune Systems and their Applications*. Berlin: Springer Verlag.
- de Castro, L. N., & Timmis, J. (2002a). *Artificial immune systems: A new computational intelligence approach*. London: Springer.
- de Castro, L. N., & Timmis, J. (2002b). An Artificial Immune Network for Multimodal Function Optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, Hawaii, USA (pp. 699-674).
- de Castro, L. N., & Von Zuben, F. J. (1999). *Artificial Immune Systems: Part I -Basic Theory and Applications* (Tech. Rep. No. DCA-RT 01/99). Brazil: University of Campinas, School of Electrical and Computer Engineering.
- de Castro, L. N., & Von Zuben, F. J. (2001). Automatic determination of radial basis functions: an immunity-based approach. *International Journal of Neural Systems*, 11(6), 523–535. doi:10.1016/S0129-0657(01)00094-1
- de Castro, L. N., & Von Zuben, F. J. (2002a). Learning and Optimization Using the Clonal Selection Principle. *IEEE Transactions on Evolutionary Computation*, 6(3), 239–251. doi:10.1109/TEVC.2002.1011539
- de Castro, L. N., & Von Zuben, F. J. (2002b). aiNet: An artificial immune network for data analysis. In H. A. Abbass, R. A. Sarker, & C. S. Newton (Eds.), *Data Mining: A Heuristic Approach* (pp. 231-259). Hershey, PA: IGI Global.
- de França, F. O., Bezerra, G. B. P., de Castro, L. N., & Von Zuben, F. J. (2006b). New Perspectives for the Biclustering Problem. In *Proceedings of the IEEE Congress on Evolutionary Computation*, Vancouver, Canada (pp. 2768-2775).

- de França, F. O., de Castro, L. N., & Von Zuben, F. J. (2004a). A Max Min Ant System Applied to the Capacitated Clustering Problem. In *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing*, São Luís, Brazil (pp. 755-764).
- de França, F. O., de Castro, L. N., & Von Zuben, F. J. (2005a). Artificial Immune Network for Multimodal Function Optimization on Dynamic Environments. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2005)*, Washington, DC (pp. 289-296).
- de França, F. O., de Castro, L. N., & Von Zuben, F. J. (2005b). Max Min Ant System and Capacitated p-Medians: Extensions and Improved Solutions. *Informatica*, 29(2), 755–764.
- de França, F. O., Gomes, L. C. T., de Castro, L. N., & Von Zuben, F. J. (2006a). Handling Time-Varying TSP Instances. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, Canada (pp. 9735-9742).
- de França, F. O., & Von Zuben, F. J. (2009). A dynamic artificial immune algorithm applied to challenging benchmarking problems. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, Trondheim, Norway (pp. 423-430).
- de França, F. O., Von Zuben, F. J., & de Castro, L. N. (2004b). Definition of Capacitated p-Medians by a Modified Max Min Ant System with Local Search. In *Proceedings of the ICONIP - 2004 11th International Conference on Neural Information Processing – Special Session on Ant Colony and Multi-Agent Systems*, Calcutta, India (pp. 1094-1100).
- de Sousa, J. S., Gomes, L. C. T., de Castro, L. N., Bezerra, G. B., & Von Zuben, F. J. (2004). An Immune-Evolutionary Algorithm for Multiple Rearrangements of Gene Expression Data. *Genetic Programming and Evolvable Machines*, 5(2), 157–179. doi:10.1023/B:GENP.0000023686.59617.57
- Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. Chichester, UK: John-Wiley & Sons.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197. doi:10.1109/4235.996017
- Deb, K., & Tiwari, S. (2005). Omni-optimizer: A Procedure for Single and Multi-objective Optimization. In C. A. Coello, A. H. Aguirre, & E. Zitzler (Eds.), *Proceedings of the 3rd. International Conference on Evolutionary Multi-Criterion Optimization (LNCS 3410)*, pp. 47-61). New York: Springer.
- DiMaggio, P. A. Jr, McAllister, S. R., Floudas, C. A., Feng, X., Rabinowitz, J. D., & Rabitz, H. A. (2008). Biclustering via optimal re-ordering of data matrices in systems biology: rigorous methods and comparative studies. *BMC Bioinformatics International Journal of Technology and Human Interaction*, 9(458), 1–16.
- Duarte, L. T., Suyama, R., Attux, R. R. F., Von Zuben, F. J., & Romano, J. M. T. (2006). Blind Source Separation of Post-nonlinear Mixtures Using Evolutionary Computation and Order Statistics. In *Proceedings of the 6th International Conference on Independent Component Analysis and Blind Signal Separation (ICA)*, Charleston, SC (pp. 66-73).
- Farmer, J. D., Packard, N., & Perelson, A. (1986). The immune system, adaptation and machine learning. *Physica D. Nonlinear Phenomena*, 2(1-3), 187–204. doi:10.1016/0167-2789(86)90240-X
- Forrest, S., Perelson, A. S., Allen, L., & Cherukuri, R. (1994). Self-nonsel discrimination in a computer. In *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*, Oakland, CA (pp. 202-212).
- Fukunaga, K., & Hayes, R. R. (1989). The reduced parzen window classifier. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(4), 423–425. doi:10.1109/34.19040
- Glover, F. W., & Kochenberger, G. A. (2002). *Handbook of Metaheuristics*. Norwell, MA: Kluwer Academic Publishers.
- Gomes, L. C. T., de Sousa, J. S., Bezerra, G. B., de Castro, L. N., & Von Zuben, F. J. (2004). copt-aiNet and the Gene Ordering Problem. *Revista tecnologia da informação*, 3(2), 27-34.
- Hartigan, J. A. (1972). Direct clustering of a data matrix. [JASA]. *Journal of the American Statistical Association*, 67(337), 123–129. doi:10.2307/2284710
- Holland, J. H. (1984). Genetic algorithms and adaptation . In Selfridge, O. G., Rissland, E. L., & Arbib, M. A. (Eds.), *Adaptive Control of Ill-Defined Systems* (pp. 317–333). New York: Plenum Press.
- Jensen, F. V. (2001). *Bayesian Networks and Decision Graphs*. Statistics for Engineering and Information Science.

- Jerne, N. K. (1974). Towards a network theory of the immune system. *Ann. Immunol. Inst. Pasteur, 125C*, 373–389.
- Jiong, Y., Haixun, W., Wei, W., & Yu, P. S. (2003). Enhanced biclustering on expression data. In *Proceedings of the 3rd IEEE Symposium on Bioinformatics and Bioengineering*, Bethesda, MD (pp. 321-327).
- Junqueira, C., de França, F. O., Attux, R. R. F., Panazio, C. M., de Castro, L. N., Von Zuben, F. J., & Romano, J. M. T. (2006). Immune-inspired Dynamic Optimization for Blind Spatial Equalization in Undermodeled Channels. In *Proceedings of the IEEE Congress on Evolutionary Computation*, Vancouver, Canada (pp. 9801-9808).
- Junqueira, C., de França, F. O., Attux, R. R. F., Suyama, R., de Castro, L. N., Von Zuben, F. J., & Romano, J. M. T. (2005). A Proposal for Blind FIR Equalization of Time-Varying Channels. In *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing*, Mystic, USA (pp. 9-14).
- Kullback, S. (1959). *Information Theory and Statistics*. New York: John Wiley & Sons.
- Leung, Y., & Wang, Y. (2001). An Orthogonal Genetic Algorithm with Quantization for Global Numerical Optimization. *IEEE Transactions on Evolutionary Computation*, 5(1), 41–53. doi:10.1109/4235.910464
- Li, C., Ma, T., & Zhu, X. (2010). aiNet- and GIS-based regional prediction system for the spatial and temporal probability of rainfall-triggered landslides. *Natural Hazards*, 52(1), 57–78. doi:10.1007/s11069-009-9351-x
- Martel, C. (2002). The expected complexity of Prim's minimum spanning tree algorithm. *Information Processing Letters*, 81(4), 197–201. doi:10.1016/S0020-0190(01)00220-4
- Mirkin, B. (1996). *Mathematical Classification and Clustering*. New York: Springer.
- Mitra, S., & Banka, H. (2006). Multi-objective evolutionary biclustering of gene expression data. *Pattern Recognition*, 39(12), 2464–2477. doi:10.1016/j.patcog.2006.03.003
- Miyahara, K., & Pazzani, M. J. (2000). Collaborative Filtering with the Simple Bayesian Classifier. In *Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence (PRICAI)*, Melbourne, Australia (pp. 679-689).
- Nei, M., & Kumar, S. (2000). *Molecular Evolution and Phylogenetics*. New York: Oxford University Press.
- Pasti, R., & de Castro, L. N. (2007). The Influence of Diversity in an Immune-Based Algorithm to Train MLP Networks. In *Proceedings of the 6th International Conference on Artificial Immune Systems (ICARIS)*, Santos, Brazil (pp. 71-82).
- Pasti, R., de Castro, L. N., Coelho, G. P., & Von Zuben, F. J. (2009). *Neural network ensembles: immune-inspired approaches to the diversity of components*. Natural Computing.
- Perelson, A. S., & Oster, G. F. (1979). Theoretical studies of clonal selection: Minimal antibody repertoire size and reliability of self-nonself discrimination. *Journal of Theoretical Biology*, 81(4), 645–670. doi:10.1016/0022-5193(79)90275-3
- Pinchuk, G. (2002). *Theory and problems of immunology*. New York: McGraw-Hill.
- Saitou, N., & Nei, M. (1987). The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4), 406–425.
- Sareni, B., & Krähenbühl, L. (1998). Fitness Sharing and Nicheing Methods Revisited. *IEEE Transactions on Evolutionary Computation*, 2(3), 97–106. doi:10.1109/4235.735432
- Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-Based Collaborative Filtering Recommendation Algorithms. In *Proceedings of 10th World Wide Web Conference (WWW10)*, Hong Kong, China (pp. 285-295).
- Siarry, P., Berthiau, G., Durdin, F., & Haussy, J. (1997). Enhanced simulated annealing for globally minimizing functions of many-continuous variables. [TOMS]. *ACM Transactions on Mathematical Software*, 23(2), 209–228. doi:10.1145/264029.264043
- Stibor, T., & Timmis, J. (2007). An Investigation on the Compression Quality of aiNet. In *Proceedings of the 2007 IEEE Symposium on Foundations of Computational Intelligence (FOCI)*, Honolulu, Hawaii (pp. 495-501).
- Symeonidis, P., Nanopoulos, A., Papadopoulos, A., & Manolopoulos, Y. (2007). Nearest-Biclusters Collaborative Filtering with Constant Values. In *Proceedings of the Advances in Web Mining and Web Usage Analysis*, Chicago, IL (pp. 36-55).

- Timmis, J., Edmonds, C., & Kelsey, J. (2004). Assessing the Performance of Two Immune Inspired Algorithms and a Hybrid Genetic Algorithm for Function Optimisation. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, San Diego, CA (pp. 1044-1051).
- Xu, L., Mo, H., Wang, K., & Tang, N. (2006). Document Clustering Based on Modified Artificial Immune Network. In *Proceedings of the 1<sup>st</sup> International Conference on Rough Sets and Knowledge Technology (RSKT)*, Chongqing, China (pp. 516-521).
- Yao, X., & Liu, Y. (1997). Fast evolution strategies. In *Proceedings of the 6th International Conference on Evolutionary Programming VI*, Indianapolis, IN (pp. 151-161).
- Yu, K., Schwaighofer, A., Tresp, V., Xu, X., & Kriegel, H.-P. (2004). Probabilistic Memory-based Collaborative Filtering. *IEEE Transactions on Knowledge and Data Engineering*, 16(1), 56–59. doi:10.1109/TKDE.2004.1264822
- Zhan, C. T. (1971). Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters. *IEEE Transactions on Computers*, C-20(1), 68–86. doi:10.1109/T-C.1971.223083

*Fabrcio O. de Franca, Guilherme P. Coelho, Pablo A. D. Castro and Fernando J. Von Zuben are with the Laboratory of Bioinformatics and Bioinspired Computing (LBiC), Department of Computer Engineering and Industrial Automation (DCA), School of Electrical and Computer Engineering (FEEC), University of Campinas (Unicamp).*

*Guilherme Palermo Coelho received his Eng. degree in computer engineering and M.Sc degree in electrical engineering from the University of Campinas (Unicamp), Campinas, Brazil, in 2003 and 2006, respectively. He is currently working towards the Ph.D degree in electrical engineering also at the University of Campinas (Unicamp). His main research areas are computational intelligence, multiobjective optimization, artificial immune systems, evolutionary computation, biclustering, collaborative filtering, artificial neural networks and ensembles.*

*Pablo A. D. Castro received his M.S. in Computer Science from Federal University of São Carlos (UFSCar, Brazil) in 2004 and his PhD in Computer Engineering from University of Campinas (UNICAMP, Brazil) in 2009. Pablo is member of the Laboratory of Bioinformatics and Bioinspired Computing (LBiC) and the main topics of his research are bioinspired computing, Bayesian networks, data mining, feature selection, multi-objective optimization, and fuzzy systems.*

*Fernando J. Von Zuben received his Dr.E.E. degree from the University of Campinas (Unicamp), Campinas, SP, Brazil, in 1996. He is currently the header of the Laboratory of Bioinformatics and Bioinspired Computing (LBiC), and he is an Associate Professor at the Department of Computer Engineering and Industrial Automation, University of Campinas (Unicamp). The main topics of his research are computational intelligence, bioinspired computing, multivariate data analysis, and machine learning. He coordinates open-ended research projects in these topics, tackling real-world problems in the areas of decision-making, pattern recognition, and discrete and continuous optimization.*