On the Diversity Mechanisms of opt-aiNet: a Comparative Study with Fitness Sharing

Fabrício O. de França, *Student Member, IEEE*, Guilherme P. Coelho, *Student Member, IEEE* and Fernando J. Von Zuben, *Member, IEEE*

Abstract— **Immune-inspired algorithms based on the Immune Network theory have been frequently claimed to be capable of maintaining diversity among the candidate solutions in their population. However, no specific study on this aspect to verify how the intrinsic diversity mechanisms of such immune algorithms behave, when compared to other approaches from the literature, has been made yet. Therefore, in this work we have addressed this issue, by taking the opt-aiNet algorithm (a popular immune-inspired algorithm developed for real-parameter optimization) and comparing its results with those of a modified version, in which the mechanisms associated with diversity maintenance were replaced by a traditional fitness sharing approach. Besides, two distance metrics were also considered for both algorithms: the traditional Euclidean distance, and the Line Distance, a metric proposed in the literature as capable of identifying whether two solutions belong to distinct local optima. The experiments were performed on six benchmark problems from the literature, each of them with distinct characteristics, and the results have shown that the original immune-inspired mechanisms of opt-aiNet are indeed more capable of stimulating the diversity of solutions, and also requiring a smaller amount of computational resources.**

I. Introduction

In the last years, several metaheuristics based on the *Artificial Immune System* (AIS) [1] framework were proposed in the literature to deal with a wide range of applications, from optimization to data mining [1] [2] [3]. One of the main justifications for the adoption of such immune-inspired metaheuristics, specially those inspired in Jerne's *Immune Network* theory [4], is that they contain intrinsic mechanisms capable of not only stimulating the generation of diverse sets of local optimal solutions during execution, but also of maintaining those distinct solutions.

This characteristic of immune-inspired algorithms has been empirically observed in most of the results reported in the literature (see, for example, the review in [3]), but, to our knowledge, there is no attempt up to now to provide an explicit comparison of the diversity maintenance mechanisms of AISs with other traditional approaches from the literature, such as fitness sharing [5] [6]. Therefore, the main goal of this paper is to make such comparisons, by taking a popular immune-inspired network-based algorithm from the literature (namely opt-aiNet [7]) and applying it to six benchmark problems [8], together with the same algorithm (with identical operators) but with the diversity mechanisms replaced by a traditional fitness sharing approach [6].

Most of the immune-inspired algorithms presented in the AIS literature, including opt-aiNet, are based on two theories from immunology: *Clonal Selection* [9], [10] and *Immune Network* [4]. According to the Clonal Selection principle, when the antigens (molecular signatures) of a pathogen are recognized by the immune system, some specific cells start to proliferate (by cloning) and differentiate into plasma cells (those capable of secreting antibodies that attach to the antigens and signal that they should be eliminated) and memory cells (to provide faster future responses) to face the infection. During this process, such cells also suffer high rates of mutation (*hypermutation*) followed by a selective pressure, in order to become even more capable of recognizing (and combating) the invading pathogens. Complementarily, the Immune Network theory states that, besides being capable of recognizing antigens, the immune cells are also capable of recognizing each other, which may trigger two different responses: the stimulation or suppression of the immune cells.

In Artificial Immune Systems, the way that the Immune Network theory is modeled basically consists in the periodical evaluation of all cells in the population (candidate solutions) to identify which of those cells have become more similar than a given threshold during evolution (the Clonal Selection steps of cloning, hypermutation and selection) and, if two cells are highly similar, the one with the worst antigen recognition capability is eliminated (suppressed) from the population, while new cells are introduced.

The diversity mechanisms of network-based AISs are intrinsic to the algorithms, while approaches such as fitness sharing generally constitute appended mechanisms to previously developed metaheuristics. Also, those mechanisms of AISs do not affect elitism (as fitness sharing do) and are fundamentally related to the dynamic variation of the population size during execution. However, what remains to be answered is: are these mechanisms actually more capable of maintaining diversity than fitness sharing? This is what we intend to verify in the following sections, for a selected set of benchmark problems.

This paper is structured as follows. In Section II, the implementation of the opt-aiNet algorithm adopted in this work will be presented in details, together with the modified version that replaces the diversity maintenance mechanisms with fitness sharing. The experimental methodology followed here, the benchmark problems adopted and the obtained results will be presented in Section III, while the final

Fabrício Olivetti de França, Guilherme Palermo Coelho 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), P. O. Box 6101 Campinas - SP - 13083-970, Brazil (emails: {olivetti, gcoelho, vonzuben}@dca.fee.unicamp.br).

978-1-4244-8126-2/10/$26.00 ©2010 IEEE

comments will be given in Section IV.

II. The opt-aiNet algorithm and its Fitness-Sharing variant

The opt-aiNet algorithm (*Artificial Immune Network for Optimization*) was proposed by de Castro & Timmis [7] as an optimization-aimed extension of aiNet (*Artificial Immune Network*), originally proposed by de Castro & Von Zuben [11] for clustering problems. All the algorithms in the aiNet family, which includes opt-aiNet and all extensions proposed in the literature in the last decade [3], are based on the Clonal Selection [9] and Immune Network [4] principles, which provides them with mechanisms capable of automatically adjusting the population size according to the demands of the applications[1]. Besides, these mechanisms also allow the algorithms to maintain a high diversity of solutions during their executions, which contributes to a broader exploration of the search space.

As one of the most popular diversity stimulation approach from the literature, fitness sharing [5] [6] is a *niching* technique that was originally proposed to minimize the occurrence of *genetic drift* [12] caused by elitist selection operators in traditional genetic algorithms (GA's). This phenomenon makes traditional GA's suitable for unimodal problems, as all the individuals in the population tend to converge to a single solution, but they usually face problems in real-world optimization, as these problems are generally based on multimodal domains and require the identification of multiple distinct optima. With niching, which is based in the dynamics of natural ecosystems [6], the population diversity is maintained, thus allowing the algorithm to explore several peaks in parallel. More specifically, fitness sharing modifies the search landscape of the problem being optimized by penalizing individuals that inhabit densely populated regions, reducing their fitness.

In this section, the details of the opt-aiNet algorithm implemented in this work will be discussed, together with a modification of opt-aiNet (named *opt-aiNetFS – opt-aiNet with Fitness Sharing*) that replaces the original immune-inspired mechanisms of diversity stimulation by a traditional fitness sharing approach.

A. opt-aiNet

In opt-aiNet, the candidate solutions of the problem are modeled as a population of *antibodies* (encoded as real-valued vectors) and, differently from other immune-inspired algorithms such as aiNet, opt-aiNet does not explicitly adopt the concept of *antigens*: the quality of each antibody in the population is evaluated in a *fitness-like* way [1], as given in Eq. 1:

$$f_i^{Ag}(t) = \begin{cases} \frac{f_i - \min_j(f_j)}{\max_j(f_j) - \min_j(f_j)} & \text{if maxim. prob.} \\ 1 - \frac{f_i - \min_j(f_j)}{\max_j(f_j) - \min_j(f_j)} & \text{otherwise} \end{cases}, \quad (1)$$

[1]This behavior is deeply influenced by the proper adjustment of parameters, as will be seen later.

where $f_i^{Ag}(t) \in [0,1]$ is the fitness of antibody i at iteration t and f_k is the value of the objective function for antibody k. This fitness $f_k^{Ag}(t)$ is basically the value of the objective function f_k for antibody k, normalized in $[0,1]$ in iteration t.

The main steps of the opt-aiNet algorithm are given in Alg. 1. The user must properly set seven parameters to run opt-aiNet, besides the definition of the algorithm's stop criterion (which could be the maximum number of iterations, maximum number of function evaluations, etc.). These parameters are: nAB, which corresponds to the initial population size; the number of clones nC that should be generated for each cell in the population, in the cloning phase; β_{worst} and β_{best} which is a mutation parameter (described later in this section); the suppression threshold σ_s, which is the minimum allowed distance between two antibodies that does not trigger a suppression; d which corresponds to a percentage of the population size (in the current iteration) that defines the number of new randomly generated individuals that should be inserted into the population; and $supp_{it}$, that corresponds to the number of iterations that the algorithm should run between two consecutive suppression steps.

Algorithm 1 Main steps of opt-aiNet.

Parameters:
- nAB: initial number of antibodies;
- nC: number of clones per antibody;
- β_{worst}: mutation parameter for the worst solutions;
- β_{best}: mutation parameter for the best solutions;
- σ_s: suppression threshold;
- d: percentage of new individuals to be inserted;
- $supp_{it}$: number of iterations between suppressions;

1- Randomly create the initial population;
2- $it = 0$;
while stop criteria not satisfied **do**
    3- Clone individuals in the population;
    4- Apply hypermutation to the clones;
    5- For each set of parent and mutated clones, select the best cell to survive to the next generation;
    if mod$(it, supp_{it}) == 0$ **then**
        6- Suppress similar individuals;
        7- Introduce d (% of the actual population size) new randomly generated cells;
    end if
    8- $it = it + 1$;
end while
9- Suppress similar individuals;

From the steps given in Alg. 1, a special attention must be given to the hypermutation (step 4) and suppression (step 6) operators. In the hypermutation mechanism, each clone suffers mutation with genetic variability inversely proportional to its quality (fitness), so the new antibody is generated according to [7]:

$$Ab_{t+1}^i = Ab_t^i + \beta_t^i \cdot e^{-f_i^{Ag}(t)} \cdot \mathcal{N}(0,1), \quad (2)$$

where Ab_{t+1}^i is the new antibody i, Ab_t^i is the original antibody i, $f_i^{Ag}(t)$ is the fitness of antibody i and $\mathcal{N}(0,1)$ is a random value with Gaussian distribution (mean 0 and variance 1). In [7], the parameter β (that controls the amplitude of the random variation) is a static parameter that should be defined by the user, which is often a complex task as β is dependent on the problem being solved. In our implementation, we have decided to variate β according to the quality of the solution, as given in Eq. 3, to increase the effects of small genetic variation for best solutions and higher variations for the worst ones.

$$\beta_t^i = \beta_{worst} + f_i^{Ag}(t) \cdot (\beta_{best} - \beta_{worst}), \qquad (3)$$

where β_t^i is the value of parameter β at iteration t for antibody i, $f_i^{Ag}(t)$ is the fitness of antibody i at iteration t, and β_{worst} and β_{best} are the values of β (defined by the user) to be assigned to the worst and best individuals.

Finally, in the suppression step, all the individuals in the population are compared against each other (the distance $d(i, j)$ among antibodies i and j is calculated $\forall i, j$) and, if two antibodies are more similar than the threshold σ_s (if $d(i, j) < \sigma_s$), the worst of the two is eliminated from the population.

The suppression threshold σ_s should also be adjusted according to the problem being solved. If σ_s is too large, the neighborhood of each cell in opt-aiNet's population may enclose more than a single peak (considering a maximization problem), and only a single cell will be kept in this region. If σ_s is too small, the size of the population may grow significantly during execution, and several solutions may be kept in each peak (specially sub-optimal ones).

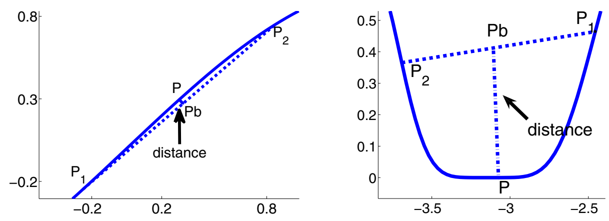
This difficulty in determining the radius of each peak is not an exclusive problem of opt-aiNet (and all aiNet-based algorithms). In fitness sharing, one of the most difficult task is also to determine the threshold of dissimilarity [6] (or the niche radius – see Sect. II-C), which has an equivalent role of the suppression threshold in opt-aiNet.

In this work, we have chosen to adopt two distance metrics to evaluate the similarity among the antibodies in the population. The first one was the traditional Euclidean distance, normalized by the maximum possible distance in the search space. And the second was the *Line Distance*, proposed in [13] and briefly explained next. The main difference of the Line Distance is that it tends to better identify individuals within the region of the same peak, independently of the conformation of the search space.

B. The Line Distance Metric

In [13] it was pointed out that the adoption of the Euclidean distance as a metric of similarity among antibodies in opt-aiNet leads to a lack of robustness associated with the adjustment of the suppression threshold σ_s, as a pre-analysis of the search space is required in order to find a correct value for this parameter. And, even with this pre-analysis, sometimes it is impossible to find a threshold capable of dealing with different regions of the search space.

In order to make the suppression process more robust, de França *et al.* [13] proposed the *Line Distance* metric, which basically consists in the creation of a straight line between a pair of points P_1 and P_2 on the search space (the antibodies to be compared) and then in the location of the point P_b that



(a) (b)

Fig. 1. Two examples of the Line Distance between points P_1 and P_2 on the search space. P_b is the middle point of the line formed by P_1 and P_2 and P is the closest point to P_b, on the function surface. In (a) P_1 and P_2 are on the same peak, while in (b) they are on different peaks.

is the closest one to the middle point P of the given line, as depicted on Fig. 1. Therefore, the line distance between P_1 and P_2 will be equivalent to the Euclidean distance between P and P_b, which corresponds to the distance between a simple function approximation using two points, and the real function surface.

C. opt-aiNet with Fitness Sharing

The opt-aiNet algorithm with fitness sharing (*opt-aiNetFS*) adopts exactly the same mechanisms presented in Alg. 1, except for steps 6 and 7, which were eliminated. Besides, the fitness of each antibody was modified, so that the crowding of the region in which each solution is placed is also considered. In this approach, individuals placed in the neighborhood of other solutions from the population are penalized, according to Eq. 4.

$$f_i^{shared}(t) = \begin{cases} f_i - m_i & \text{if maximization prob.} \\ f_i + m_i & \text{otherwise} \end{cases}, \qquad (4)$$

where $f_i^{shared}(t)$ is the shared fitness for individual i at iteration t, m_i is the niche count (given by Eq. 5), which is associated with the number of individuals within the neighborhood of i, and f_i is the value of the objective function for antibody i.

$$m_i = \sum_{j=1}^{N} sh(d(i, j)), \qquad (5)$$

where $d(i, j)$ is the distance between antibodies i and j, and sh is the sharing function.

This sharing function (given in Eq. 6) indicates the level of similarity between two elements i and j distant $d(i, j)$ of each other. It returns 1 if the individuals are identical ($d(i, j) = 0$), zero if their distance $d(i, j) > \sigma_s$, being σ_s a threshold of dissimilarity (analogous to opt-aiNet's suppression threshold), and an intermediate value if $0 < d(i, j) < \sigma_s$.

$$sh(d(i, j)) = \begin{cases} 1 - \left(\frac{d(i,j)}{\sigma_s}\right)^{\alpha} & \text{if } d(i, j) < \sigma_s \\ 0 & \text{otherwise} \end{cases}, \qquad (6)$$

where α is a constant parameter that regulates the shape of the sharing function (in this work, $\alpha = 1.0$ for all problems).

One important aspect that must be highlighted in Eq. 4 is that, differently from the original fitness sharing approach [6], here the penalty term m_i is added or subtracted (according to the problem) to the original value of the objective function. The original proposal states that $f_i^{shared}(t) = \frac{f_i}{m_i}$ and assumes that f_i is non-negative and should be maximized, which is not the case here, as the problems were supposed to be minimized and cannot be normalized (the maximum and minimum values of the objective functions are unknown *a priori*), so the original approach could not be adopted. Also, the multiplication $f_i^{shared}(t) = f_i \cdot m_i$ (which would be natural, as we are dealing with minimization problems) instead of division, cannot be used as well, as the fitness of some of the studied problems ranges from positive to negative values along the search space. Therefore, by subtracting/adding the term m_i, we still penalize individuals in highly populated regions of the search space, while stimulate the proliferation of isolated individuals.

III. Experimental Results

To properly verify the performance of the studied diversity mechanisms, six benchmark (minimization) functions with different features were taken from [8] and adopted here, in $\Re^2$. In what follows, the properties of such problems will be described briefly and their choice will be justified. In this work, we will keep their original names, as used in [8] for further reference.

The first two functions, named F_2 and F_4, are unimodal (with the addition of noise on F_4) and will be used to verify if the opt-aiNet suppression mechanism can correctly resize the population to a single cell. Notice here that, as the fitness sharing mechanism does not control the size of population, its diversity test on these two functions will be reported but not used for performance comparison.

The third function, named F_9, is highly multimodal and will serve to assess the algorithms' capabilities of locating and maintaining several local optima that present a small distance among each other.

The fourth function, F_{12}, is in the opposite situation as it is a multimodal function but with just a few local optima. The purpose of this problem is to evaluate the robustness of the algorithms, as the valleys of F_{12} are not equidistant.

In order to challenge the algorithms even further, the fifth and sixth functions chosen, F_{15} and F_{20}, have common features like a huge number of local optima, some plateaus along the search space and distinct properties for each group of local optima, like separation distance, location and number of optima per group. These two problems will further evaluate how the algorithms studied here deal with extreme situations.

In order to measure the performance of the studied algorithms it was performed 20 experiments (repetitions) for each algorithm on each function. On each group of experiments, four different aspects were analyzed. The first one was the Mean-Squared Error (MSE) of the final best solutions found by each algorithm at each experiment, which measures the overall quality of obtained results. The second aspect studied was the average size of the final population of each algorithm (notice that this result only applies to the opt-aiNet algorithm, as the Fitness Sharing variant has a fixed population size).

The third aspect analyzed was the ratio (R_{opt}) between the number of local optima found by each algorithm and the population size, as described on Eq. 7. The higher the value of R_{opt}, the better the performance of the algorithm regarding diversity. It is easy to notice that this ratio has a maximum value of 1, as it is impossible to an algorithm to return a population size smaller than the number of local optima found. The number of global/local optima found by each algorithm was counted manually, by plotting the final solutions obtained together with a *contour plot* of each function, for each repetition made. Also, for clarification, the average number of optima found will also be reported.

$$R_{opt} = \frac{\#\text{ of local optima}}{||Ab||}, \quad (7)$$

where $||Ab||$ is the size of the final population of antibodies.

Finally, it will also be considered the average time required by each algorithm to complete the optimization process. As a basis of comparison the experiments were all run on the same machine, an Intel Core2QUAD Q9550 @ 2.83GHz with 2GB of RAM (each method running on a single core).

The parameters adopted for the studied algorithms for all the problems are given in Tab. I. The only parameters that were set differently for each problem were the suppression threshold of original opt-aiNet variants (given in Tab. II) and the population size of the Fitness Sharing counterparts (shown in Tab. III). All these parameters were intensively tested and verified, so that the best set for each algorithm on each function could be obtained. The population size for the Fitness Sharing versions of opt-aiNet were set as the rounded largest average number of solutions maintained by the original opt-aiNet for each function.

TABLE I

Set of parameters adopted for the four algorithms studied.

Parameter	Value
Max. Number of Fitness Evaluations	$40,000$
nAB	100
nC	5
β_{best}	0.8
β_{worst}	1.1
d	40%
$supp_{it}$	10

Table IV presents the results obtained for each algorithm for problem F_2. From these results, it is possible to see that each method has performed equally regarding the final mean squared error and the number of optima located. The main noticeable difference lies in the final population size obtained by the algorithms with the two distance measures. When using the Euclidean distance, the opt-aiNet algorithm

TABLE II

SUPPRESSION THRESHOLD/THRESHOLD OF DISSIMILARITY σ_s ADOPTED FOR EACH PROBLEM. THE VALUES FOR THE FITNESS SHARING COUNTERPART *opt-aiNetFS* WITH EUCLIDEAN DISTANCE IS SHOWN IN PARENTHESIS WHEN IT DIFFERS FROM THE ORIGINAL VERSION.

Function	σ_s	
	Euclidean Distance (FS)	Line Distance
F_2	0.200	50
F_4	0.500	50
F_9	0.020(0.050)	0.3
F_{12}	0.015	50
F_{15}	0.020	10
F_{20}	0.020(0.001)	50

TABLE III

POPULATION SIZE ADOPTED FOR THE FITNESS SHARING ALGORITHMS.

Function	F_2	F_4	F_9	F_{12}	F_{15}	F_{20}
Size	10	10	45	10	30	15

ended up with an average of 3.95 solutions while its version with the *Line Distance* metric was able to detect the unimodality of the problem and converged to an average of 1.1 solutions. Figure 2 illustrates this difference among the different approaches. In Fig. 2, it is noticeable how opt-aiNet with Euclidean distance fails to correctly identify that the extra three points belong to the same optima and should be suppressed, which is due to the use of Euclidean distance with a fixed suppression threshold.

TABLE IV

QUALITY AND DIVERSITY RESULTS FOR BENCHMARK FUNCTION F_2. IN THIS TABLE, **MSE** STATES FOR THE MEAN-SQUARED ERROR OF THE BEST SOLUTION, **POP. SIZE** IS THE AVERAGE SIZE OF THE FINAL POPULATION, **# OF OPT.** CORRESPONDS TO THE AVERAGE NUMBER OF OPTIMA FOUND BY EACH ALGORITHM AND $\mathbf{R_{opt}}$ IS THE RATIO BETWEEN THE NUMBER OF LOCAL OPTIMA FOUND AND THE POPULATION SIZE (THIS NOTATION WILL BE KEPT IN THE FOLLOWING TABLES).

Method	**MSE**	**Pop. Size**	**# of opt.**	$\mathbf{R_{opt}}$
Euclidean	$4.04 \cdot 10^{-5}$	3.95	1.00	0.25
FS + Eucl.	$1.68 \cdot 10^{-5}$	10.00	1.00	0.10
Line Dist.	$0.65 \cdot 10^{-5}$	1.10	1.00	0.91
FS + Line Dist.	$0.86 \cdot 10^{-5}$	10.00	1.00	0.10

Table V shows the results of the algorithms for the noisy unimodal function F_4. Apart from the opt-aiNetFS with Line Distance method (*FS + Line Dist.*), which suffered with some outliers, all other methods were able to obtain high quality solutions in all experiments. In this particular problem, the average population size of opt-aiNet with Euclidean distance was closer to one, when compared to the results for F_2, and, when coupled with the Line Distance metric, it was able to obtain one individual in all experiments, thus reinforcing the robustness of this metric. In Figure 3 the contour plots point out that the opt-aiNetFS methods were deceived by the

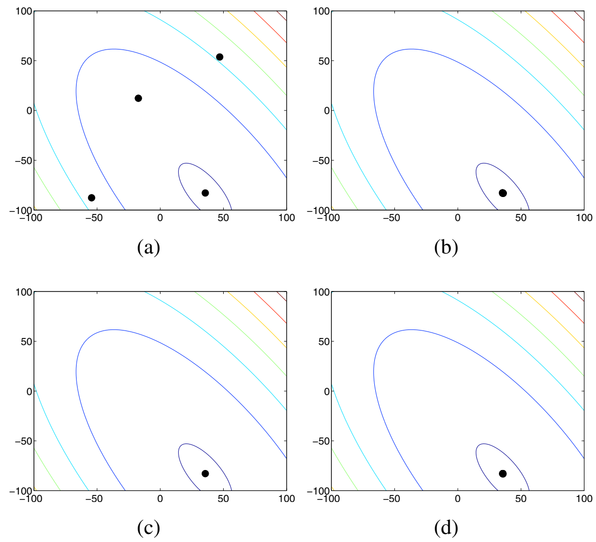


Fig. 2. A contour plot of a single experiment on function F_2 with methods: (a) opt-aiNet with Euclidean distance; (b) opt-aiNetFS with Euclidean distance; (c) opt-aiNet with Line Distance; (d) opt-aiNetFS with Line Distance.

noisy landscape and, in opposite to what happened in F_2, the population was not able to converge to a single point.

TABLE V

QUALITY AND DIVERSITY RESULTS FOR BENCHMARK FUNCTION F_4.

Method	**MSE**	**Pop. Size**	**# of opt.**	$\mathbf{R_{opt}}$
Euclidean	$1.45 \cdot 10^{-5}$	1.55	1.00	0.65
FS + Eucl.	$173.19 \cdot 10^{-5}$	10.00	1.00	0.10
Line Dist.	$1.03 \cdot 10^{-5}$	1.00	1.00	1.00
FS + Line Dist.	55.94	10.00	0.90	0.09

For the first multimodal function, F_9, the results given in Tab. VI show that each method was able to obtain good quality solutions (with low MSE) but now, as this function contains a huge number of optima, the differences among the studied methods are noticeable. First of all, although with a sufficiently large number of individuals, the opt-aiNetFS methods (both with Euclidean and Line distances) found on average a smaller number of local optima. Regarding the two distance measures, the Line Distance seems to be more robust, as it allowed opt-aiNet to keep a smaller number of individuals per local optima (generally just a single one), which led to a higher R_{opt} score. However, the Euclidean distance led to a larger set of local optima, which indicates that this metric provides a higher exploration capacity to the algorithm. Figure 4 illustrate the tendency of opt-aiNet to spread the individuals in population into every optimum of the search space, while opt-aiNetFS only tends to focus on the region closer to the global optimum.

The next multimodal function studied here, F_{12}, presents just 3 visible local optima in the problem's domain. From the results presented in Tab. VII it is possible to notice that again the MSE shows a similar quality of the final solutions obtained by the algorithms with different diversity

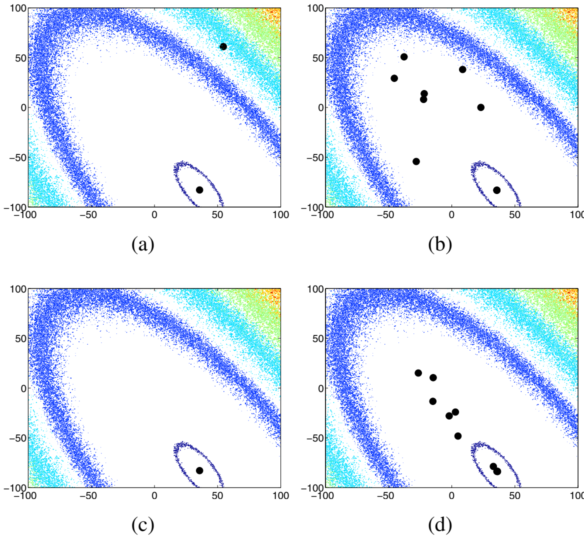


Fig. 3. A contour plot of a single experiment on function F_4 with methods: (a) opt-aiNet with Euclidean distance; (b) opt-aiNetFS with Euclidean distance; (c) opt-aiNet with Line Distance; (d) opt-aiNetFS with Line Distance.

TABLE VI
QUALITY AND DIVERSITY RESULTS FOR BENCHMARK FUNCTION F_9.

Method	MSE	Pop. Size	# of opt.	$\mathbf{R_{opt}}$
Euclidean	0.02	40.8	35.95	0.88
FS + Eucl.	0.01	45	21.05	0.47
Line Dist.	0.01	27.85	25.85	0.93
FS + Line Dist.	0.04	45	19.75	0.44

methods. Regarding the average number of optima found, both opt-aiNetFS methods presented values below 3, as in some experiments they could only locate 2 optima. The Euclidean and Line Distance versions of opt-aiNet both were capable of finding all the 3 optima in all the experiments, but with the latter maintaining much fewer solutions at the end, which so far indicates a tendency of the Line Distance being more conservative regarding the final population size. Also, Fig. 5 shows the difficulty faced by opt-aiNet with Euclidean distance to detect what were the local optima of the problem, which can be associated with the sensibility of its suppression threshold. In this situation, if the suppression threshold is set to a high value, the algorithms will not be able to find all the local optima, as the neighborhood of each antibody will enclose several peaks/valleys. Therefore, the choice of the suppression threshold corresponds to the most adequate trade-off between the number of possibly identified optima and the compactness of the population.

The obtained results follow the same trend on the next benchmark function, F_{15}, as shown in Tab. VIII. The MSEs are again quite similar for all the algorithms, although the opt-aiNetFS counterparts were slightly better here. As this problem presents a larger number of optima than F_{12}, it depicts again how the opt-aiNet with Euclidean distance tends to find a higher number of local optima while the opt-aiNet with Line Distance tends to keep a more conservative

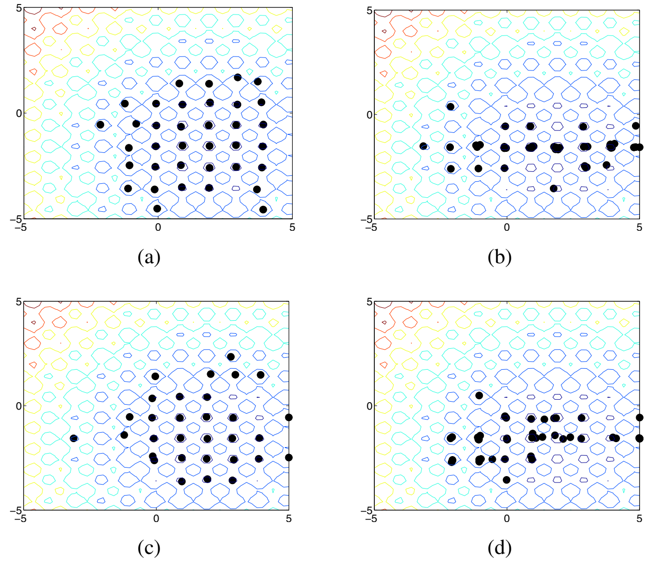


Fig. 4. A contour plot of a single experiment on function F_9 with methods: (a) opt-aiNet with Euclidean distance; (b) opt-aiNetFS with Euclidean distance; (c) opt-aiNet with Line Distance; (d) opt-aiNetFS with Line Distance.

TABLE VII
QUALITY AND DIVERSITY RESULTS FOR BENCHMARK FUNCTION F_{12}.

Method	MSE	Pop. Size	# of opt.	$\mathbf{R_{opt}}$
Euclidean	0.02	10.45	3.00	0.29
FS + Eucl.	0.03	10.00	2.75	0.27
Line Dist.	0.04	4.65	3.00	0.64
FS + Line Dist.	0.05	10.00	2.65	0.26

approach and maintain, in average, a single individual per optimum. The opt-aiNetFS counterparts tend to cluster some individuals together in the same optima which leads to these poor results regarding diversity but, by doing this, it tends to explore the region of the global optimum with more individuals, which allows the algorithm to obtain solutions closer to the optimum (thus resulting in a smaller MSE). The contour plot depicted in Fig. 6 shows a clear advantage of opt-aiNet with Euclidean distance as it was able to identify more optima than opt-aiNet with Line Distance and much more than the opt-aiNetFS methods. This is mainly due to the uniform distribution of the optima presented by this function, which simplifies the adjustment of the suppression threshold.

TABLE VIII
QUALITY AND DIVERSITY RESULTS FOR BENCHMARK FUNCTION F_{15}.

Method	MSE	Pop. Size	# of opt.	$\mathbf{R_{opt}}$
Euclidean	0.38	26.65	21.10	0.79
FS + Eucl.	0.15	30.00	5.75	0.19
Line Dist.	0.42	14.80	13.80	0.93
FS + Line Dist.	0.13	30.00	5.60	0.19

For problem F_{20}, it was obtained different results regarding MSE, as shown on Tab. IX. In this benchmark function, the global optimum is on the boundary of the search

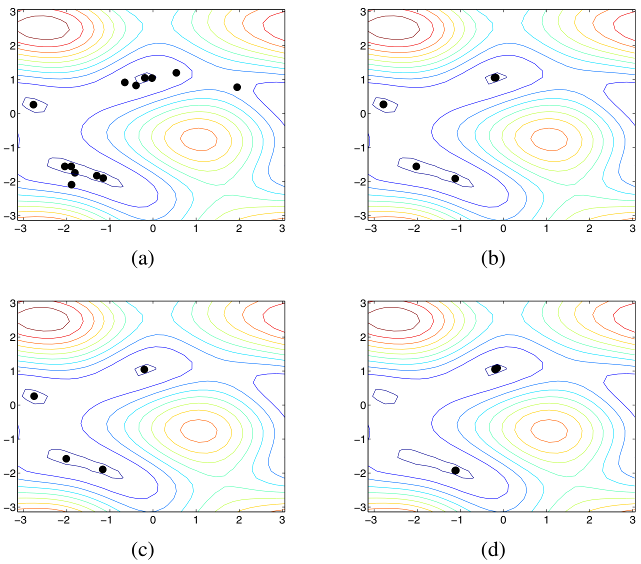


Fig. 5. A contour plot of a single experiment on function F_{12} with methods: (a) opt-aiNet with Euclidean distance; (b) opt-aiNetFS with Euclidean distance; (c) opt-aiNet with Line Distance; (d) opt-aiNetFS with Line Distance.

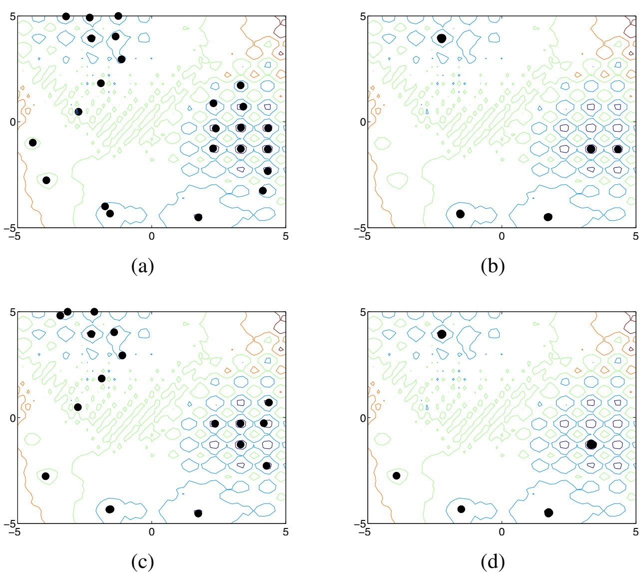


Fig. 6. A contour plot of a single experiment of function F_{15} with methods: (a) opt-aiNet with Euclidean distance; (b) opt-aiNetFS with Euclidean distance; (c) opt-aiNet with Line Distance; (d) opt-aiNetFS with Line Distance.

space and surrounded by low quality local optima, which can deceive most algorithms. This situation gives an advantage to the algorithms that can maintain a better diversity, as it tends to further explore the search space and has a better chance of finding the global optimum. Regarding the diversity, the opt-aiNet suppression and insertion mechanisms were capable of outperforming fitness sharing, by finding more optima and converging to a smaller final population. Comparing the two distance metrics in opt-aiNet, the Line Distance presented a little advantage in this function, as it led to a smaller population size with a similar number of identified optima.

TABLE IX

QUALITY AND DIVERSITY RESULTS FOR BENCHMARK FUNCTION F_{20}.

Method	MSE	Pop. Size	# of opt.	R_{opt}
Euclidean	0.22	12.65	8.10	0.64
FS + Eucl.	19.02	15.00	5.25	0.35
Line Dist.	0.15	7.70	7.20	0.93
FS + Line Dist.	28.37	15.00	5.65	0.38

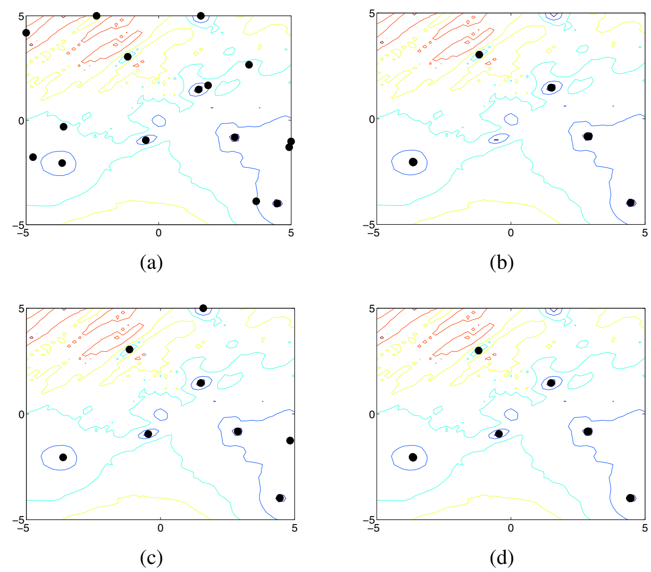


Fig. 7. A contour plot of a single experiment of function F_{20} with methods: (a) opt-aiNet with Euclidean distance; (b) opt-aiNetFS with Euclidean distance; (c) opt-aiNet with Line Distance; (d) opt-aiNetFS with Line Distance.

In Fig. 7, we can see a situation contrary to that observed in the previous function (illustrated in Fig.6). As the regions around local optima are larger than those in the previous problem, the opt-aiNet with Euclidean distance showed a tendency of keeping several individuals in these regions, which led to a higher population size, as noticed before on Tab. IX. On the contrary, opt-aiNet with Line Distance was able to properly maintain a single solution per optimum and suppress all other solutions that were in the same basis of attraction. The opt-aiNetFS methods were also able to maintain a good distribution of solutions, although they found a slightly smaller number of local optima in average.

Another aspect of the algorithms that should be noticed is the computational complexity of each method. The Euclidean distance is much less expensive than the Line Distance, and the fitness sharing procedure also tends to be much more complex as it calculates the distance at every iteration. On Tab. X the average time consumed by each method is shown to give an idea of how complex each method is, compared to the others. At the first four problems, which present a lower cost per function evaluation, opt-aiNet with both distance metrics presented equivalent computational costs, while the opt-aiNetFS counterparts spent much more time per experiment. Concerning the last two functions, the results show that the opt-aiNet with Euclidean distance is about 3 times less expensive than its variant with Line Distance. The opt-aiNetFS algorithms were again more expensive than the two opt-aiNet, specially the opt-aiNetFS with Line Distance, which presented a much higher cost than the others.

TABLE X
AVERAGE TIME, IN SECONDS, REQUIRED BY EACH METHOD ON EACH BENCHMARK FUNCTION.

	opt-aiNet + Eucl.	**opt-aiNetFS + Eucl.**	**opt-aiNet + Line Dist.**	**opt-aiNetFS + Line Dist.**
F_2	0.95	53.37	1.13	245.52
F_4	1.09	53.66	1.37	338.29
F_9	1.50	237.80	3.50	1154.20
F_{12}	1.50	53.39	1.96	303.84
F_{15}	37.00	195.00	151.00	$55,365.00$
F_{20}	36.00	115.00	98.00	$28,314.00$

IV. Final Comments

In this paper, the diversity mechanisms of opt-aiNet were studied and compared against a modified fitness sharing method on continuous optimization problems. Also, two different distance metrics were analyzed: the well known Euclidean distance and the Line Distance, introduced in [13]. The modifications introduced in the fitness sharing approach were necessary as it was originally proposed to deal with the minimization of nonnegative functions, which was not the case of the problems studied here.

The experimental part of this work consisted on the application of the four variations of opt-aiNet (the original algorithm, with both Euclidean and Line distances, and the algorithm with the inclusion of fitness sharing, also with both distance metrics) to six benchmark problems with a diverse set of properties. Two of these problems are unimodal (being one of them with noise in fitness), while the remaining four are multimodal, with differences on the number of local optima, the distribution of such optima and their separability. First, the algorithms were analyzed regarding the quality of the final set of solutions found, to ensure that all methods could achieve a minimum solution quality. Then, the diversity of the final solutions returned by each algorithm was analyzed, by evaluating the ratio between the number of local optima found and the final population size (the ideal behavior would be to return a single solution per local optimum).

The obtained results showed that opt-aiNet's diversity mechanism, with its ability to dynamically adjust the population size, was responsible not only for the obtainment of better solutions but also for the maintenance of more diverse solutions. With its suppression mechanism, opt-aiNet was capable of reducing the population size to a minimum, in order to save computational resources. Considering the overall computational costs, opt-aiNet also has the advantage of just using its suppression method from time to time, while in fitness sharing the shared fitness must be recalculated for all the individuals in the neighborhood of a solution.

Regarding the two distance metrics adopted in this work, each one showed to be favorable in distinct situations. The Euclidean distance seemed to be more robust when the optimization problem presents a more uniform distribution of local optima, while the Line Distance tends to perform better on the opposite situation, in search spaces with oddly distributed local optima. Also, the suppression threshold parameter seemed more robust with the Line Distance, as its variation did not present significant impact on the overall performance of the algorithms. The proper adjustment of the suppression threshold when the Euclidean distance was adopted was more challenging.

Overall, in the experiments performed here, the opt-aiNet diversity mechanism showed evidences that it is robust enough to be incorporated to state-of-the-art optimization methods in order to reduce costs and increase the exploratory behavior of such algorithms.

However, a more extensive set of experiments should still be performed, with more benchmark problems of distinct characteristics and, specially, with other diversity maintenance techniques from the Evolutionary Computation literature. These experiments that aim at drawing a more complete conclusion on this matter will be the next step of this study.

Acknowledgment

The authors would like to thank CAPES and CNPq for the financial support.

References

[1] L. N. de Castro and J. Timmis, *Artificial immune systems: a new computational intelligence approach*. Springer Verlag, 2002.

[2] D. Dasgupta, "Advances in artificial immune systems," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 40–49, 2006.

[3] F. O. de França, G. P. Coelho, P. A. D. Castro, and F. J. Von Zuben, "Conceptual and practical aspects of the aiNet family of algorithms," *Intl. Journal of Natural Computing Research*, vol. 1, no. 1, pp. 1–35, 2010.

[4] N. K. Jerne, "Towards a network theory of the immune system." in *Annales d'immunologie*, vol. 125, no. 1-2, 1974, pp. 373–389.

[5] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Proc. of the Second International Conference on Genetic Algorithms*, J. J. Grefensette, Ed. L. Erlbaum Associates Inc. Hillsdale, NJ, USA, 1987, pp. 41–49.

[6] B. Sareni and L. Krähenbühl, "Fitness sharing and niching methods revisited," *IEEE Transactions on Evolutionary Computation*, vol. 2, no. 3, pp. 97–106, 1998.

[7] L. N. de Castro and J. Timmis, "An artificial immune network for multimodal function optimization," in *Proc. of the 2002 IEEE Congress on Evol. Computation, CEC*, vol. 2, 2002, pp. 699–704.

[8] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," Nanyang Technological University, Singapore, Tech. Rep., May 2005.

[9] F. M. Burnet, "Clonal selection and after," in *Theoretical Immunology*, G. I. Bell, A. S. Perelson, and G. H. Pimgley Jr, Eds. Marcel Dekker Inc., 1978, pp. 63–85.

[10] L. N. de Castro and F. J. Von Zuben, "Learning and optimization using the clonal selection principle," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 3, pp. 239–251, 2002.

[11] L. N. de Castro and F. J. Von Zuben, *aiNet: An artificial immune network for data analysis*. Idea Group Publishing, 2001, vol. 1, pp. 231–259.

[12] A. Rogers and A. Prügel-Bennett, "Genetic drift in genetic algorithm selection schemes," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 298–303, 1999.

[13] F. O. de França, F. J. Von Zuben, and L. N. de Castro, "An artificial immune network for multimodal function optimization on dynamic environments," in *Proc. of the 2005 Conference on Genetic and Evolutionary Computation*. ACM, 2005, pp. 289–296.