Multiobjective Optimization by a Modified Artificial Immune System Algorithm

Fabio Freschi[1] and Maurizio Repetto[1]

Dept. of Electrical Engineering, Politecnico di Torino, corso Duca degli Abruzzi, 24, 10129 Torino, Italy
{fabio.freschi, maurizio.repetto}@polito.it
http://www.polito.it/cadema

Abstract. The aim of this work is to propose and validate a new multiobjective optimization algorithm based on the emulation of the immune system behavior. The *rationale* of this work is that the artificial immune system has, in its elementary structure, the main features required by other multiobjective evolutionary algorithms described in literature. The proposed approach is compared with the NSGA2 algorithm, that is representative of the state-of-the-art in multiobjective optimization. Algorithms are tested versus three standard problems (unconstrained and constrained), and comparisons are carried out using three different metrics. Results show that the proposed approach have performances similar or better than those produced by NSGA2, and it can become a valid alternative to standard algorithms.

1 Introduction

Many real world applications involve the simultaneous optimization of various and often conflicting objectives. Traditional approaches for solving the Multiobjective Optimization Problem (MOP) aggregate all objectives into one function, then a single objective problem is solved by using standard optimization techniques. Several optimization runs with different parameter settings are performed, in order to achieve a set of solutions.

In the middle of the '80s Schaffer published the first attempt to solve the MOP by using evolutionary algorithms [1, 2]. The use of population-based techniques is preferable with respect to aggregating approaches, because multiple solutions can be found in one single run. From this work, several Multi Objective Evolutionary Algorithms (MOEAs) have been proposed in the last two decades. Coello Coello maintains an updated Evolutionary Multiobjective Optimization repository (`http://delta.cs.cinestav.mx/~ccoello/EMOO/`) in which the references of almost all the proposed algorithms can be found.

Despite the considerable efforts to extend Evolutionary Algorithms for solving MOPs, very few direct approaches to the MOP using the emulation of the Immune System behavior have been proposed. Most of the work concerns the use of Artificial Immune System (AIS) as a tool for maintaining diversity in the population of a Genetic Algorithm (see for example [3]) or for handling constraints in Evolutionary Algorithms [4]. In literature, one of the first reported approaches which uses AIS for solving MOPs is proposed in [5], but also in this case AIS is coupled with GA. Recently Coello Coello and Cruz Cortes develop a MOEA directly based on the emulation of the immune system [6]. The resulting algorithm, called Multiobjective Immune System Algorithm (MISA), can be considered the really first attempt to solve the general MOP directly with AIS. The performances of MISA have been improved in a further work of the same authors [6].

In this paper we propose a new approach for solving MOPs, based on the multimodal AIS optimization algorithm proposed by De Castro and Timmis [7]. The aim is to show that AIS intrinsically include some common features required by classical MOEAs, and that the extension to multiobjective optimization can be done by introducing only few modifications into the standard algorithm. The resulting algorithm is then tested on standard problems and results are compared with the ones obtained by NSGA2 algorithm [8], universally considered as representative of the state-of-the-art in multiobjective optimization.

2 Multi Objective Optimization Problem

Generally the MOP requires to optimize the vector function

$$\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x})]^T \tag{1}$$

subject to inequality and equality constraints

$$\begin{array}{l} g_i(\boldsymbol{x}) \geq 0 \;\; i = 1, 2, \ldots, k \\ h_i(\boldsymbol{x}) = 0 \;\; i = 1, 2, \ldots, p \end{array} \tag{2}$$

where $\boldsymbol{x} = [x_1, x_2, \cdots, x_n]^{\mathrm{T}} \in \Omega$ is the vector of decision variables and Ω is the feasible region. Because of the presence of several objective functions, the aim of a MOEA is to find compromise solutions rather than a single optimal point as in scalar optimization problems. In this case the trade-off solutions are usually called Pareto optimal solutions.

Considering, without loss of generality, a minimization problem for each objective, it is said that a decision vector $\boldsymbol{x}_P$ *dominates* another vector $\boldsymbol{x}_Q$ (denoted by $\boldsymbol{x}_P \prec \boldsymbol{x}_Q$) if

1. $\boldsymbol{x}_P$ is no worse than $\boldsymbol{x}_Q$ in all objectives, AND
2. $\boldsymbol{x}_P$ is strictly better than $\boldsymbol{x}_Q$ in at least one objective.

Mathematically:

$$\forall i = 1, \ldots, m \quad f_i(\boldsymbol{x}_P) \leq f_i(\boldsymbol{x}_Q) \quad \wedge \quad \exists i = 1, \ldots, m \quad f_i(\boldsymbol{x}_P) < f_i(\boldsymbol{x}_Q) \tag{3}$$

If there is no solution $\boldsymbol{x}_Q$ that dominates $\boldsymbol{x}_P$, then $\boldsymbol{x}_P$ is a *Pareto optimal solution*. The set P

$$P \triangleq \{\boldsymbol{x} \in \Omega : \neg\exists \boldsymbol{x}^* \in \Omega, \boldsymbol{f}(\boldsymbol{x}^*) \prec \boldsymbol{f}(\boldsymbol{x})\} \tag{4}$$

of all feasible Pareto optimal decision vectors is referred to as *Pareto optimal set*, while the corresponding image PF

$$PF \triangleq \left\{ \boldsymbol{f}(\boldsymbol{x}) = \left[f_1(\boldsymbol{x}), \ldots f_m(\boldsymbol{x}) \right]^{\mathrm{T}} : \boldsymbol{x} \in P \right\} \tag{5}$$

of objective vectors is called *Pareto optimal front*. Pareto optimal solutions are also called *noninferior* or *nondominated* solutions.

In this work we distinguish between the actual Pareto front, termed PF_{true}, and the final set of nondominated solutions returned by a MOEA, termed PF_{known} as defined in [9].

3 Algorithm

3.1 Artificial Immune system: brief overview

The main characteristic of the Immune System (IS) is that it must fight against external intruders (nonself) but must be tolerant with body cells (self). The main characters of IS are

- antigen (Ag): any substance capable of triggering an immune response;
- antibody (Ab): molecule (lymphocytes) that can match and counteract Ag.

Once a lymphocyte shows a high *affinity* toward an Ag, it is activated that is it undergoes an affinity maturation, a process that is aimed at improving the binding with Ag. New cells are *clones* of the older ones, diversity of new cells is ensured by a somatic *hypermutation* where genes of new cells are pieced together from widely scattered bits of DNA. This process is called *clonal selection* principle. The higher the affinity of the new cells with Ag, the higher their possibility to generate new clones. Despite its efficiency to increase affinity with Ag, somatic hypermutation has the risk of generating autoimmune cells. IS must inhibit new cells which are not self-tolerant (*suppression* of similar cells). Ag recognition does not start every time from scratch; after being stimulated some of the lymphocytes become *memory cells* of the system.

The behavior of the Immune System can be artificially emulated for optimization or, more generally, for machine learning [10]. An algorithm based on emulation of the IS behavior is referred to as Artificial Immune System (AIS). A deep investigation of the AIS can be found in [11, 12].

In the optimization field, AIS has shown to have a great ability for searching multiple optimal solutions [7]. In this case Ags are represented by the optimal points of a function, while Abs are the test configurations. Basically, the optimization algorithm is structured into two nested levels (Fig. 1). The inner one takes into account the Ab-Ag affinity relations, stimulating most promising cells, while the outer level manages the network of cells of the system, eliminating the similar ones. Cardinality of the population can be fixed or dynamic, but new cells are generated throughout the process in order to explore as much as possible the space of configurations. Deep details of the multimodal single objective optimization algorithm are provided in [7].

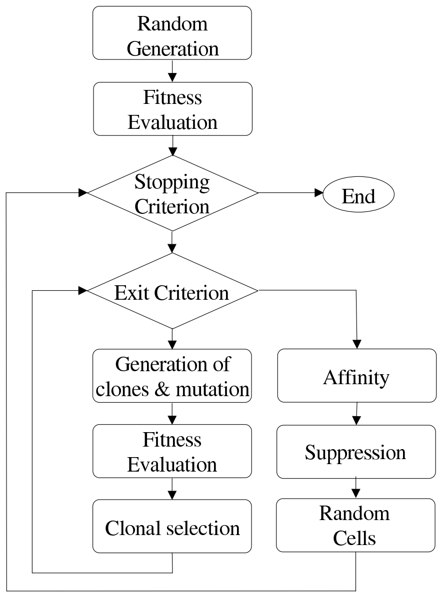


Fig. 1. AIS optimization algorithm flowchart

3.2 Multi Objective Algorithm

Following the structure of the AIS optimization algorithm, we propose a new multiobjective optimization algorithm. The algorithm, called Vector Artificial Immune System (VAIS) has the same structure as the one for the single objective shown in Fig. 1.

1. Initially a random uniformly distributed population is created and the fitness is assigned to each solutions, as it will be described in the next section. The memory is initialized to be empty.
2. Each cell is reproduced in N_{clones} copies of the original one and each clone is locally mutated by a random perturbation. The amplitude of mutation decreases when the fitness of the original parent cell increases, according to Eqs. (6)
$$\begin{aligned} \boldsymbol{x}_{\text{new}} &= \boldsymbol{x}_{\text{old}} + \alpha\ \boldsymbol{x}_{\text{random}} \\ \alpha &= \beta \exp(-f^*) \end{aligned} \tag{6}$$
where $\boldsymbol{x}_{\text{random}}$ is a vector of Gaussian random numbers of mean 0 and standard deviation 1, f^* is the normalized value of fitness from the values $[f_{\min}, f_{\max}]$ into the range $[0, 1]$. The value of the parameter β is chosen to set the maximum amplitude of mutation. $\boldsymbol{x}_{\text{new}}$, $\boldsymbol{x}_{\text{old}}$ and $\boldsymbol{x}_{\text{random}}$ are real valued vectors defined in a normalized parameter space.
3. For each clone the values of the objective functions and Pareto dominance relations are evaluated. Because the fitness depends on the actual population, its value is assigned to the clones and recalculated for the parent cells. The nondominated individuals are copied into the memory.

4. The best (with respect to the fitness value) mutated clone for each cell replaces the original parent (clonal selection).
5. Steps 2-4 (which represent the inner loop) are repeated for N_{in} times.
6. The affinity operator is applied to the memory: the Euclidean distance between memory cells is measured; despite of the traditional AIS algorithm, the distance is evaluated in the objective space, in order to obtain an uniformly distributed Pareto front.
7. All but the highest fitness cells whose distances are less than a threshold are suppressed. The threshold value must be related to the number of solutions desired on the PF_{known} (N_{memory}); if the objectives are normalized into the range $[0,1]$, the value of $\frac{\sqrt{m}}{N_{\text{memory}}}$ (m: number of objectives) represents the distance among solutions uniformly distributed on a straight continuous Pareto front. We choose this value as threshold for suppression.
8. The memory is copied into the original population. New randomly generated cells fill the remaining population, in order to maintain the diversity of solutions. A minimum percentage of newcomers is guaranteed at each iteration to obtain a good exploration of the solution space.
9. The process is repeated N_{out} times from step 2.

The *rationale* of this work is that AIS has, in its elementary structure, the main characteristics required by MOEAs described in literature. One of the main characteristic of classical MOEAs is that they present selection pressure (genetic drift) phenomenon [13] and some tricks must be adopted for enhancing diversity in solutions and space exploration. Instead, AIS makes parallel searches of optimal solutions, leaving the management of the network of cells to the *suppression* operator in the upper level of the algorithm. This operator gives another advantage: when defining the fitness assignment, several MOEAs require information about crowding (density) of solutions [14], while AIS does not need them because similar solutions are suppressed. There are at least two other characteristics intrinsically defined in AIS which are usually needed by other multiobjective algorithms. AIS do not need an additional memory for storing nondominated solutions (like, for example, the MultiObjective Particle Swarm Optimization, MOPSO, algorithm [15]), because this feature is already defined. Finally the clonal selection is always elitist, so AIS does not present *backward* effects during the iteration [16].

Fitness Assignment In literature there are several Pareto-based fitness assignment strategies for MOPs. All non-aggregating techniques require the evaluation of the Pareto dominance among the individuals of the population [17]. This approach has the advantage that it is insensitive to the nonconvexity of the Pareto Optimal Set [18]. In their famous algorithm NSGA2 [19, 8], Deb *et al.* apply a pure Pareto ranking for assigning the fitness value to the population. At each iteration all the nondominated solutions are assigned rank 1 and they are temporary removed from the assignment. Then rank 2 is assigned to the new set of nondminated solutions and so on. In SPEA [17] algorithm and in its evolution SPEA2 [14], instead of calculating the standard Pareto ranking, Zitzler *et* *al.* assign to the population a fitness value which incorporates both dominance and density information. In particular all nondominated solutions have a fitness, called *strength*, proportional to the number of individuals dominated by each of them: let N_i denote the number of individuals dominated by the nondominated i-th cell and N_{dom} the total number of dominated solutions, then the strength of i is

$$s_i = \frac{N_i}{1 + N_{\text{dom}}} \tag{7}$$

The fitness of a dominated solution j is calculated from the strength of the solutions i which dominate it

$$f_j = \sum_{i:i \prec j} s_i \tag{8}$$

The NSGA2 fitness assignment approach does not distinguish among nondominated solutions and the hierarchical classification of solution can become computationally intensive if the population is large. On the other side, the SPEA2 approach includes density information that are not required by the VAIS algorithm described in the previous section, because AIS has in itself operators which preserve diversity (such as affinity and suppression) and prevent the crowding of solutions. For these reasons we have adopted a simpler fitness assignment, which overcome these problems called Simple Strength Approach, SiSA. For each nondominated individual the fitness is equal to the strength, as defined in SPEA2, while for a dominated cell, the fitness is the number of individuals which dominate it. The resulting fitness guarantees a partial ranking, because all nondominated solutions have fitness values lower than 1, while the dominated ones always greater than 1.

Constraint Handling Constraints can be classified into two different types:

- constraints on objectives;
- constraints on variables.

This classification comes from the consideration that in real world problems the evaluation of objectives is the most time consuming operation in the optimization process (think, for example, to objective functions evaluated by Finite Element Analisys software) so constraints on objectives must be carefully treated in order to avoid wasting time and resources. Constraints on decision variables can be treated more easily because they can be managed before evaluating the objective functions.

In literature constraints are usually handled by using penalty functions techniques. Reference [20] gives a good survey of these strategies. Another approach, based on the definition of *constrained dominance* is developed by Deb *et al.* [8]. This technique does not require the definition of penalty functions, but simply modify the definition of dominance given in Eq. (3) including infeasible solutions.

In this work we propose a technique for handling inequality constraints on variables preserving the feasibility of solution [21]. For what concerns equality constraints on variables, these can be often rearranged decreasing the dimension of the search space. Sometimes equality constraints can be transformed into inequality one [22]. The VAIS algorithm can generate infeasible solutions in two cases. The first one is when a new random individual is generated. In this case any infeasible solution is simply discarded. An infeasible solution can also occur after applying the mutation operator to a cell close to the constraint. In this case the feasibility of the solution is maintained by progressively reducing the mutation amplitude with the bisection rule. This process stops when the mutated clone becomes feasible (Fig. 2). This technique can be applied without any hypothesis on the type of constraints (linearity, convexity, ...).

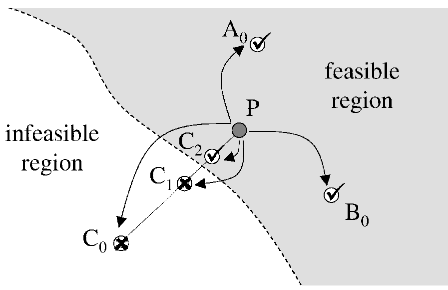


Fig. 2. Constraint handling: from a parent cell P, three clones are generated and mutated. Clones A_0 and B_0 are in the feasible region. Given that clone C_0 falls in the infeasible region, the amplitude of mutation is progressively reduced with a bisection rule, until a feasible clone, C_2, is obtained.

4 Experiments

The proposed algorithm, called Vector Artificial Immune System (VAIS) is compared versus NSGA2. This algorithm has achieved the largest attention in the multiobjective optimization literature and has been used as reference algorithm in various studies. For all tests NSGA2 has been run using a population size of 100; other parameters are set according to the values suggested by the developers in [8]. The results of VAIS are obtained using the following parameters: population size = 100, number of clones for each cell = 4, number of inner iteration = 5, percentage of random cells at each outer iteration = 20%, $\beta = 0.05$. These values have been determined after an intensive preliminary test phase of the algorithm on different test functions. The number of generations for both algorithms is set depending on the maximum number of function evaluations allowed in the test.

Three different measures have been used for numerical comparisons of the trade-off fronts produced by the algorithms, each of them takes into account a particular desired characteristic of the PF_{known}.

1. **Spacing** (S): first introduced by Schott [23], this metric measures how well the solutions throughout the PF_{known} are distributed. This metric is mathematically defined as

$$S \triangleq \sqrt{\frac{1}{N_{\text{known}}-1}\sum_{i=1}^{N_{\text{known}}}\left(\bar{d}-d_i\right)^2} \tag{9}$$

where, for each i in the set of N_{known} solutions of the PF_{known},

$$d_i \triangleq \min_j \sum_{k=1}^{m}\left|f_k^i(\boldsymbol{x})-f_k^j(\boldsymbol{x})\right| \tag{10}$$

and $\bar{d}$ is the mean value of all d_i. A value of 0 for this metric states that the solutions on the PF_{known} are equally spaced and the representation of the front is as smooth and uniform as possible.

2. **Reverse Generational Distance** (RGD): one of the main issue for measuring the performance of a MOEA is the ability to produce solutions on the PF_{known} as near as possible to the PF_{true}. In order to evaluate this characteristic, Van Veldhuizen and Lamont [24] have introduced a particular metric called *Generational Distance* (GD). It is defined as

$$GD \triangleq \frac{1}{N_{\text{known}}}\sqrt{\sum_{i=1}^{N_{\text{known}}} d_i^2} \tag{11}$$

where N_{known} is the number of nondominated vectors in the PF_{known} and d_i is the Euclidean distance measured in the objective space between each of them and the nearest member of the PF_{true}. Obviously $GD = 0$ means $PF_{\text{known}} \equiv PF_{\text{true}}$. As noted by Bosman and Thierens [25] a PF_{known} consisting on only a single solution can have a low value for this indicator. In order to include the goal of diversity, they propose to compute for each j solution in the PF_{true} the distance $\tilde{d}_j$ to the closest solution in the PF_{known} set

$$RGD \triangleq \frac{1}{N_{\text{true}}}\sqrt{\sum_{j=1}^{N_{\text{true}}} \tilde{d}_j^2} \tag{12}$$

where N_{true} is the cardinality of the PF_{true} set. We refer to this metric as *Reverse Generational Distance.*

3. **Error ratio** (ER): presented by Van Veldhuizen in [26] this metric measures the number of nondominated vectors of the PF_{known} that are not member of the PF_{true}

$$ER \triangleq \frac{\sum_{i=1}^{N_{\text{known}}} e_i}{N_{\text{known}}} \tag{13}$$

where $e_i = 1$ if solution i is not on the PF_{true}, $e_i = 0$ otherwise.

In their analysis, Knowles and Corne [27] have noted that the use of these metrics can not draw final conclusions on outperformances among MOEAs. However these indicators are commonly used in standard evolutionary multiobjective optimization literature [22].

The MOEA community has developed several test functions, that have become a standard reference for testing new algorithms. We choose three representative problems which point out some difficulties for the optimization algorithms. The following results are evaluated after having performed 20 independent runs of both algorithms.

4.1 Test Function 1

The first test is performed using the problem proposed by Tanaka [28]:
Minimize

$$\begin{aligned} f_1(\boldsymbol{x}) &= x_1 \\ f_2(\boldsymbol{x}) &= x_2 \end{aligned} \tag{14}$$

subject to

$$\begin{aligned} g_1(\boldsymbol{x}) &= x_1^2 + x_2^2 - 1 - 0.1\cos\left(16\arctan\frac{x_1}{x_2}\right) \geq 0 \\ g_2(\boldsymbol{x}) &= \left(x_1 - \frac{1}{2}\right)^2 + \left(x_2 - \frac{1}{2}\right)^2 \leq \frac{1}{2} \end{aligned} \tag{15}$$

and $x_1, x_2 \in [0, \pi]$. The final number of fitness function evaluations in this case has been set to 12000. The function presents a discontinuous and concave Pareto front which entirely lies on the first constraint. It has been proved that some MOEAs can have difficulties in finding Pareto optimal solutions with discontinuous and concave segments [29]. Fig. 3 shows the PF_{true} (continuous line) and the PF_{known} (circles) found by VAIS and NSGA2. The solutions shown correspond to the median result with respect to the RGD metric. It can be seen that the the average performances of VAIS are better than NSGA2 with respect to the spacing and the reverse generational distance (Table 1); the opposite happens with respect to the error ratio. It must be noticed that in this case differences are very small and not statistically significant.

Table 1. Results of the metrics for the Tanaka test function.

	S		RGD		ER	
	VAIS	NSGA2	VAIS	NSGA2	VAIS	NSGA2
Best	0.00144	0.00479	1.82779E-4	4.56689E-4	0.00552	0.00000
Worst	0.00260	0.00857	3.82790E-4	7.29166E-4	0.03191	0.05000
Average	0.00201	0.00640	2.56854E-4	5.65629E-4	0.02009	0.01700
Median	0.00204	0.00642	2.54257E-4	5.30819E-4	0.02139	5.30819E-4
Std. Dev.	2.77208E-4	8.40962E-4	8.28494E-5	4.25030E-5	0.00640	8.28494E-5

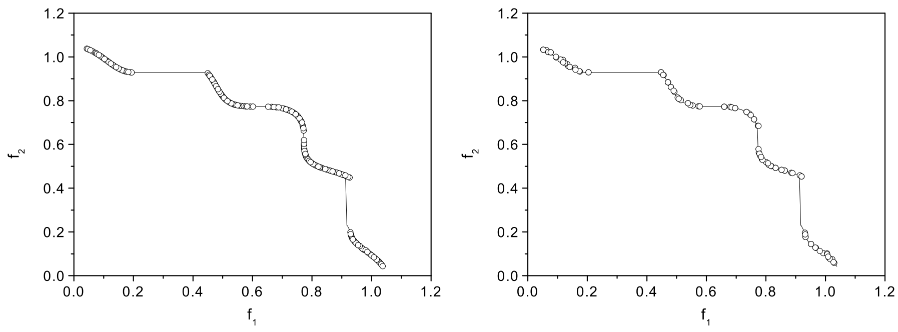


Fig. 3. Pareto Front produced by VAIS (left) and NSGA2 (right) for the Tanaka test function.

4.2 Test Function 2

The second function has been proposed by Viennet [30]:
Minimize

$$\begin{aligned} f_1(\boldsymbol{x}) &= \frac{1}{2}\left(x_1^2 + x_2^2\right) + \sin\left(x_1^2 + x_2^2\right) \\ f_2(\boldsymbol{x}) &= \frac{\left(3x_1 - 2x_2 + 4\right)^2}{8} + \frac{\left(x_1 - x_2 + 1\right)^2}{27} + 15 \\ f_3(\boldsymbol{x}) &= \frac{1}{\left(x_1^2 + x_2^2 + 1\right)} - 1.1 \exp\left(-x^2 - y^2\right) \end{aligned} \tag{16}$$

with $x_1, x_2 \in [-3, 3]$. The final number of fitness function evaluations in this case has been set to 6000. This function presents several challenging characteristics, such as a high dimensional objective space, discontinuous Pareto optimal set and several local minima in objective functions. Because of the PF_{true} has not an analytical expression, in this case it is obtained by enumeration of all possible solutions. By looking at the Pareto fronts produced in this case (Fig. 4), it can be seen that VAIS has a better representation of the PF_{true}. This fact is confirmed by the analisys of the numerical results presented in Table 2 which shows a better behavior of VAIS for all metrics.

Table 2. Results of the metrics for the Viennet test function.

	S		RGD		ER	
	VAIS	NSGA2	VAIS	NSGA2	VAIS	NSGA2
Best	0.01150	0.03009	5.06375E-4	0.00172	0.00000	0.00000
Worst	0.03868	0.04599	0.00388	0.01022	0.01765	0.04000
Average	0.01526	0.04028	8.67047E-4	0.00308	0.00345	0.01650
Median	0.01284	0.04098	5.84390E-4	0.00190	0.00217	0.01000
Std. Dev.	0.00640	0.00408	7.76958E-4	0.00287	0.00406	0.01226

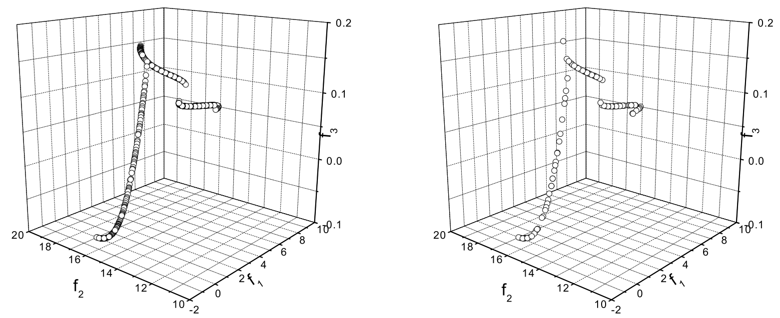


Fig. 4. Pareto Front produced by VAIS (left) and NSGA2 (right) for the Viennet test function.

4.3 Test Function 3

The last test is performed on a function proposed by Zitzler [31] and characterized by a high dimensional decision space and local Pareto fronts in the objective space. The problem is defined as:
Minimize

$$\begin{aligned} f_1(\boldsymbol{x}) &= 1 - \exp(-4x_1)\sin^6(6\pi x_1) \\ f_2(\boldsymbol{x}) &= w(\boldsymbol{x})\left(1 - \frac{f_1(\boldsymbol{x})}{w(\boldsymbol{x})}\right)^2 \end{aligned} \tag{17}$$

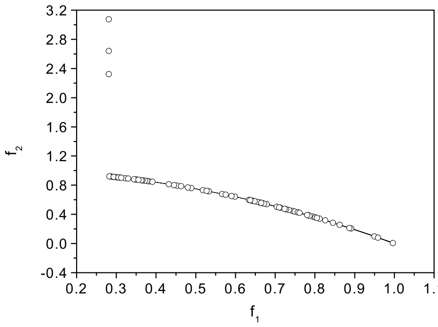
where

$$w(\boldsymbol{x}) = 1 + 9\left(\frac{\sum_{i=2}^{5} x_i}{4}\right)^{0.25} \tag{18}$$

with $x_i \in [0,1]$ and $i = 1,\ldots,5$. The true Pareto front is obtained when $w(\boldsymbol{x}) = 0$, that is with $x_1 \in [0,1]$ and $x_2 = \cdots = x_5 = 0$. Another challenging characteristic of this function is that the Pareto optimal front is not uniformly represented because the function f_1 is non linear (for more details in problem difficulties for MOP see [32]). For this test function both algorithms stop after 40000 fitness function evaluations. The comparison between the algorithms with respect to the spacing measure shows that NSGA2 has a more uniform spread of solutions than VAIS. But VAIS has better performance with respect to the other two metrics (Table 3). This result can be explained looking at Fig. 5: NSGA2 has difficulties in finding the global Pareto front, getting stuck at a local one.

5 Conclusion and Further Work

In this paper it has been shown that AIS has in its elementary structure the main characteristics of MOEA described in literature. Following this idea, a new MOEA based on the clonal selection principle, has been developed. First comparisons with another state-of-the-art algorithm show that performances



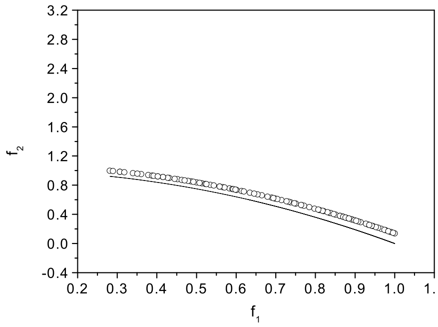


Fig. 5. Pareto Front produced by VAIS (left) and NSGA2 (right) for the Zitzler test function.

Table 3. Results of the metrics for the Zitzler test function.

	S		RGD		ER	
	VAIS	NSGA2	VAIS	NSGA2	VAIS	NSGA2
Best	0.03140	0.00570	8.58313E-5	1.48321E-4	0.06818	0.10000
Worst	0.55028	0.00712	0.00935	0.00613	0.41463	1.00000
Average	0.21017	0.00651	0.00186	0.00473	0.14747	0.95500
Median	0.16552	0.00669	8.11857E-4	0.00493	0.12162	1.00000
Std. Dev.	0.16224	4.24203E-4	0.00227	0.00137	0.07991	0.20125

of VAIS are similar or better than those produced by NSGA2. These results encourage the authors to continue the research and tests on the algorithm.

Some improvements will be done in order to produce a competitive, general purpose algorithm for MOPs, that can become a valid alternative to standard MOEAs:

- some other strategies for constraint handling will be tested, especially for managing constraints on objectives;
- the possibility of including problems with integer and mixed-integer decision variables will be added;
- other tests will be performed with other multiobjective optimization algorithms which represent the state-of-the-art in evolutionary multiobjective optimization; in this study other performance measures will be implemented;
- finally the algorithm will be tested on some high dimensional real world problems, especially in the field of electromagnetism.

6 Acknowledgements

The authors would like to thank Ms Eloisa Salina Borello for her preliminary studies about Multiobjective Optimization and Artificial Immune Systems.

References

1. Schaffer, J.D.: Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. PhD thesis, Vanderbilt University (1984)
2. Schaffer, J.D.: Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In Erlbaum, L., ed.: Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms, Hillsdale, New Jersey (1985) 93–100
3. Smith, R.E., Forrest, S., Perelson, A.S.: Population Diversity in an Immune System Model: Implication for Genetic Search. In: Foundation of Genetic Algorithm 2. Morgan Kaufmann, San Mateo, CA (1993) 153–165
4. Kurpati, A., Azarm, S.: Immune Network Simulation with Multiobjective Genetic Algorithms for Multidisciplinary Design Optimization. Engineering Optimization **33** (2000) 245–260
5. Yoo, J., Hajela, P.: Immune Network Simulations in Multicriterion Design. Structural Optimization **18** (1999) 85–94
6. Coello Coello, C.A., Cruz Cortes, N.: An Approach to Solve Multiobjective Optimization Problems Based on an Artificial Immune System. In Timmis, J., Bentley, P.J., eds.: First International Conference on Artificial Immune Systems (ICARIS'2002), University of Kent, Canterbury, England (2002) 212–221 ISBN 1-902671-32-5.
7. De Castro, L., Timmis, J.: An Artificial Immune Network for Multimodal Function Optimization. In: Proceedings of the 2002 Congress on Evolutionary Computation, CEC'02. Volume 1. (2002) 699–704
8. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: AFast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6** (2002) 182–197
9. Van Veldhuizen, D.A., Lamont, G.B.: On Measuring Multiobjective Evolutionary Algorithm Performance. In: 2000 Congress on Evolutionary Computation. Volume 1., Piscataway, New Jersey, IEEE Service Center (2000) 204–211
10. De Castro, L.N., Von Zuben, F.J.: Learning and Optimization Using the Clonal Selection Principle. IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems **6** (2002) 239–251
11. De Castro, L.N., Von Zuben, F.J.: Artificial Immune Systems: Part I Basic Theory and Applications. Technical Report RT DCA 01/99, Universidade Catolica de Santos, Coordenação de Pos-Graduação e Pesquisa (COPOP) (1999)
12. De Castro, L.N., Von Zuben, F.J.: Artificial Immune Systems: Part II A Survey of Applications. Technical Report RT DCA 02/00, Universidade Catolica de Santos, Coordenação de Pos-Graduação e Pesquisa (COPOP) (2000)
13. Tan, K. C. and, Y.Y.J., Goh, C.K., Lee, T.H.: Enhanced Distribtion and Exploration for Multiobjective Evolutionary Algorithms. In: Congress on Evolutionary Computation (CEC'2002). Volume 4. (2003) 1521–2528
14. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. TIK Report 103, Computer Engineering and Networks Lab (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Switzerland (2001)
15. Coello Coello, C.A., Toscano Pulido, G., Salazar Lechuga, M.: Handling Multiple Objectives with Particle Swarm Optimization. IEEE Transactions on Evolutionary Computation **8** (2004) 256–279
16. Lu, H., Yen, G.G.: Rank-Density-Based Multiobjective Genetic Algorithm and Benchmark Test Function Study. IEEE Transactions on Evolutionary Computation **7** (2003) 325–343

17. Zitzler, E., Thiele, L.: Multiobjective Evolutionary Algorithms: a Comparative Study and the Strength Pareto Approach. IEEE Transactions on Evolutionary Computation **3** (1999) 257–271
18. Fonseca, C.M., Fleming, P.J.: An Overview of Evolutionary Algorithms in Multiobjective Optimization. Evolutionary Computation **3** (1995) 1–16
19. Deb, K., Agrawal, S., Pratab, A., Meyarivan, T.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J.J., Schwefel, H.P., eds.: Proceedings of the Parallel Problem Solving from Nature VI Conference, Springer (2000) 849–858
20. Aguirre, A.H., Botello Rionda, S., Coello Coello, C.A., Lizarraga Lizarraga, G., Mezura Montes, E.: Handling Constraints Using Multiobjective Optimization Concepts. International Journal for Numerical Methods in Engineering **59** (2004) 1989–2017
21. Michalewicz, Z., Schoenauer, M.: Evolutionary Algorithms for Constrained Parameter Optimization Problems. Evolutionary Computation **4** (1996) 1–32
22. Coello Coello, C.A., Van Veldhuizen, D.A., Lamont, G.B.: Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Publishers, New York (2002) ISBN 0-3064-6762-3.
23. Schott, J.R.: Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Master's thesis, Dept. Aeronautics and Astronautics, Massachussets Institue of Technology (1995)
24. Van Veldhuizen, D.A., Lamont, G.B.: Multiobjective Evolutionary Algorithm Research: a History and Analysis. Technical report tr-98-03, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH (1998)
25. Bosman, P.A.N., Thierens, D.: The Balance between Proximity and Diversity in Multiobjective Evolutionary Algorithms. IEEE Transactions on Evolutionary Computation **7** (2003) 174–188
26. Van Veldhuizen, D.A.: Multiobjective Evolutionary Algorithms: Classifications, Analysis, and Innovations. Ph.d. dissetation, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH (1999)
27. Knowles, J., Corne, D.: On Metrics for Comparing Non-Dominated Sets. In: Congress on Evolutionary Computation (CEC'2002). Volume 1., Piscataway, New Jersey, IEEE Service Center (2002) 711–716
28. Tanaka, M., Watanabe, H., Furukawa, Y., Tanino, T.: GA-Based Decision Support System for Multicriteria Optimization. In: Proceedings of the International Conference on Systems, Man, and Cybernetics. Volume 2., Piscataway, NJ, IEEE (1995) 1556–1561
29. De Jong, K.A.: An Analysis of Behavior of a Class of Genetic Adaptive Systems. PhD thesis, Dept. of Computer Science, University of Michigan, Ann Arbor, MI (1975)
30. Viennet, R., Fontiex, C., Marc, I.: New Multicriteria Optimization Method Based on the Use of a Diploid Genetic Algorithm: Example of an Industrial Problem. In Alliot, J.M., Lutton, E., Ronald, E., Schoenauer, M., Snyers, D., eds.: Proceedings of Artificial Evolution (European Conference, selected papers), Brest, France, Springer-Verlag (1995) 120–127
31. Zitzler, E., Deb, K., Thiele, L.: Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Evolutionary Computation **8** (2000) 173–195 MIT Press.
32. Deb, K.: Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. Evolutionary Computation **7** (1999) 205–230