

# MOAISDX: A New Multi-objective Artificial Immune System based on Decomposition

Estefania A. Aguilar Arroyo<sup>✉</sup> and Carlos A. Coello Coello<sup>✉</sup>

CINVESTAV-IPN, Departamento de Computación  
Av. IPN No. 2508, Col. San Pedro Zacatenco, Mexico City 07360, MEXICO.

**Abstract.** From among the many techniques currently available to solve multi-objective optimization problems (MOPs), an alternative is the use of multi-objective artificial immune systems (MOAISs). This sort of meta-heuristic emulates immune processes using computational resources, with the aim of solving MOPs. MOAISs have mechanisms such as the clonal selection principle as well as positive and negative selection, that make them powerful search tools. In recent years, there have been proposals of MOAISs that adopt selection schemes that are more appropriate to deal with many-objective problems (i.e., problems having more than 3 objectives), from which decomposition has been a popular choice. We propose here a new MOAI called “Multi-objective Artificial Immune System based on Decomposition” (MOAISDX). The performance of our proposed approach is compared with respect to that of NSGA-II and MOEA/D, as well as with respect to four recent MOAISs. The results obtained from this comparative study show that MOAISDX outperforms NSGA-II and obtains results similar to those of MOEA/D in most of the adopted test instances. Furthermore, MOAISDX has better performance than that of the other MOAISs compared, particularly as we increase the number of objectives.

**Keywords:** Multi-objective optimization · Artificial immune systems · Decomposition

## 1 Introduction

Multi-objective optimization involves the solution of problems that consist of two or more (often conflicting) objective functions. Assuming minimization, a multi-objective optimization problem (MOP) is defined as:

$$\text{minimize } \vec{f}(\vec{x}) := [f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x})], \quad (1)$$

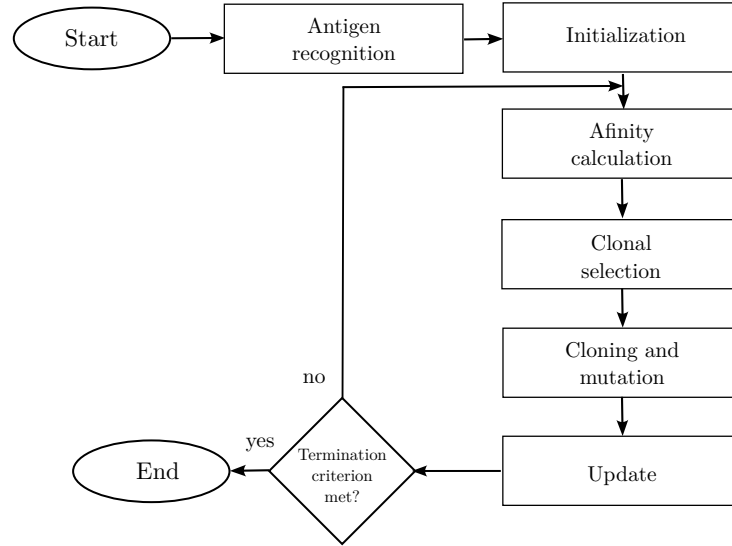
subject to:

$$g_i(\vec{x}) \leq 0 \quad i = 1, 2, \dots, m \quad (2)$$

$$h_i(\vec{x}) = 0 \quad i = 1, 2, \dots, p \quad (3)$$

where  $\vec{x} = [x_1, x_2, \dots, x_n]^T$  is known as the decision vector,  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $i = 1, \dots, k$  are the objective functions and  $g_i, h_j : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $i = 1, \dots, m$ ,  $j = 1, \dots, p$  are the constraint functions. The goal in multi-objective optimization is to find a set of solutions that represent the best trade-offs among the objective functions. To identify the quality of such solutions we utilize the concept of Pareto dominance. Given two vectors  $\vec{x}, \vec{y} \in \mathbb{R}^n$ ,  $\vec{x}$  is said to dominate  $\vec{y}$ , denoted as  $\vec{x} \prec \vec{y}$ , if  $\vec{f}(\vec{x}) \leq \vec{f}(\vec{y})$  and  $\vec{f}(\vec{x}) \neq \vec{f}(\vec{y})$  (given that  $\vec{x} \leq \vec{y}$  if  $x_i \leq y_i$  for  $i = 1, \dots, n$ ). A vector  $\vec{x}$  is said to be nondominated with respect to  $\mathcal{X} \subset \mathbb{R}^n$  if there is no other vector  $\vec{y}$  such that  $\vec{y} \prec \vec{x}$ . We say  $\vec{x}$  is a Pareto optimal solution, if  $\vec{x}$  is nondominated with respect to the feasible region  $\mathcal{F} \subset \mathbb{R}^n$ . The set of all Pareto optimal solutions is known as the Pareto optimal set. The set of all  $\vec{f}(\vec{x})$  such that  $\vec{x}$  is in the Pareto optimal set is called Pareto front.

Multi-objective artificial immune systems (MOAISs) are a metaheuristic inspired on our biological immune system. MOAISs emulate immune processes using computational resources and are oriented to the solution of multi-objective problems. Akin to a multi-objective evolutionary algorithm (MOEA), a MOAIS maintains a population of potential solutions (called antibodies) along the optimization process. The aim is to simulate the immune response subjecting the antibodies to a series of immune operators. In general, a MOIAS undergoes a series of common steps with an indistinct model [14]:



**Fig. 1.** Steps of a generic MOIAS.

- **Antigen recognition:** The multi-objective problem is recognized as an antigen; in constrained problems the constraint functions are sometimes identified as antigens.

- **Initialization:** A set of randomly initialized potential solutions are designated as the main population, known as antibodies in decision space.
- **Affinity calculation:** The affinity between a pair of antigens and antibodies is calculated at each generation.
- **Clonal selection:** The antibodies with the highest affinities among the population are selected to undergo a proliferation process. In other words, to speed up the convergence of the population, more resources are allocated to more promising areas in the search space.
- **Update:** The parents with the lowest affinities are replaced by antibodies with higher affinities and the resulting population becomes the parents for the next generation.

When the termination criterion is met, the output of a MOAIS is the final population, which contains an approximation of the Pareto optimal set.

MOAISs are called hybrid algorithms when the design includes non immunological operators (e.g., a crossover operator). Hybrid MOAISs are far more popular than pure MOAISs because the lack of crossover severely limits their exploitation capabilities.

MOAISs based on the clonal selection principle [19] aim to identify antibodies with high affinities (i.e., the antibodies that are able to bind with a certain antigen with a high precision) and propagate them, producing a (pre-defined) number of identical copies or clones. The clones are then mutated according to a specific metric, which is normally based on their affinity [4]. This allows to guide the search, changing at higher rates antibodies which are not suitable potential solutions, while keeping or slightly changing the ones that are. In this way, there is no need for an explicit control of the relationship between exploration and exploitation in the search.

Most of the currently available MOAISs are based on Pareto optimality. These approaches use a procedure called nondominated sorting to rank solutions such that all the Pareto optimal solutions found at a certain generation have the same probability of being selected [6]. These approaches normally adopt an additional mechanism (called density estimator) that allows them to maintain diversity in the population over time. This sort of approach quickly loses effectiveness as we increase the number of objectives, which makes them unsuitable for problems having more than three objectives (the so-called many-objective optimization problems). In recent years, there has been some research on the use of decomposition-based approaches into MOAISs [14], obtaining good results. In a decomposition-based approach, the main idea is to transform a MOP into several single-objective problems, which are then solved simultaneously and in a collaborative manner using neighborhood search [21]. A set of search directions (weighted vectors) are used together with a scalarizing function to guide the search [18]. These approaches can be used to solve MOPs with any number of objectives. Although the effectiveness of decomposition-based approaches is based on the scalarizing function adopted, there are several choices which are normally very effective.

Here, we propose a new decomposition-based MOAIS, adopting a cloning operator and a selection mechanism for sub-problem optimization. Our proposed approach is called Multi-objective Artificial Immune System based on Decomposition (MOAISDX).

## 2 Previous related work

Here, we will briefly review some MOAISs representative of the state of the art in the area.

The Nondominated Neighbor Immune Algorithm (NNIA) [9] draws inspiration from the clonal selection principle. The decision of which individuals are to be selected for cloning is made according to the value of their crowding distances. Because of this mechanism, the resulting approximation sets have a good distribution along the Pareto front because the focus is on selecting the nondominated antibodies that lie in less populated areas of the Pareto front. NNIA also maintains an external archive with nondominated solutions found during the search. However, it tends to lose population's diversity and deteriorates its convergence when increasing the number of objectives due to the use of a selection mechanism based on Pareto optimality. NNIA was originally tested on problems with only two and three objectives. Its computational complexity is  $O(N^2)$ , where  $N$  is the population size.

The Novel Immune Clonal Algorithm (NICA) [20] uses an approach similar to NNIA, and it also adopts Pareto optimality to filter out solutions in the population. However, unlike NNIA, NICA incorporates Pareto optimality in its cloning mechanism, since only nondominated solutions undergo the cloning process. In this MOAIS, the number of clones assigned to each selected solution is the same regardless of its affinity. NICA also employs the crowding distance to suppress a nondominated individual at each generation in order to maintain a good distribution in the resulting set. NICA has the same limitations as NNIA because it is also based on Pareto optimality. NICA was evaluated on problems with three objectives, obtaining competitive results. Its computational complexity is  $O(N^3)$ , where  $N$  is the population size.

A Hybrid Evolutionary Immune Algorithm for multiobjective optimization problems (HEIA) [16] is a hybrid framework for artificial immune systems oriented to solve MOPs. HEIA combines immune mechanisms with evolutionary operators to improve the search capabilities of a pure MOAIS. It adopts recombination and mutation schemes in different randomly generated sub-populations trying to find a compromise between proximity and diversity in the final approximation set. Its selection strategy involves categorizing the solutions as dominated and nondominated, as well as removing dominated solutions and sorting the remaining solutions according to their crowding distances. HEIA also keeps an external archive with the nondominated solutions found so far. HEIA is also based on Pareto optimality. HEIA was tested on the ZDT [24], WFG [10], UF [22] and DTLZ [7] test suites. It was tested on bi-objective and three-objective

problems yielding competitive results. It was also reported that HEIA has a computational complexity of  $O(N^2)$ , where  $N$  is the population size.

A Multi-objective Immune Algorithm with Dynamic Population Strategy (MOIAS-DPS) [17] is a hybrid MOAIS which includes a mechanism to dynamically control the population size based on the current state of its external archive of nondominated solutions. The aim is to intensify the exploration capacity by gradually rising the population size when the archive is not full, and decreasing it when it is. MOIAS-DPS was tested on problems having two and three objectives. MOAIS-DPS is also based on Pareto optimality. Its computational complexity was not provided.

The Multi-Objective Immune Algorithm with a Decomposition-based Clonal Selection (MOIA-DCSS) [13] is based on decomposition, so the solutions are associated to a sub-problem and a weighted vector. The number of clones assigned to each individual depends on the improvement of the associated sub-problem through generations. To update the population, the largest relative improvement with regard to the scalarizing function is used to determine if a new solution replaces the current solution. MOIA-DCSS was validated using the F [11], UF and WFG test suites with two and three objectives. Its computational complexity was not provided.

The Vertical Distance-based clonal selection mechanism for MOIAs (VD-MOAI) [12] utilizes a decomposition approach coupled with the Tchebychev scalarizing function. The algorithm makes use of clonal selection based on the vertical distance between a solution and the weighted vector associated to a sub-problem. This method assigns the number of clones to each solution proportional to the vertical distance and it promotes convergence, focusing also on preserving diversity by assigning more clones to solutions with lower vertical distances. VD-MOAI also uses a differential evolution crossover operator. VD-MOAI was originally tested only with problems having two and three objectives. VD-MOAI was tested on the WFG, UF and F test suites. Its computational complexity was not provided.

The Balancing Convergence and Diversity in Multiobjective Immune Algorithm (BCD-MOIA) [15] is a hybrid MOAIS which introduces a cloning operator to balance population convergence and diversity throughout the search. The cloning operator has two parts. Convergence is maintained taking into account the individual's relative improvement. Diversity is kept by establishing which individuals are closer to the associated weighted vector. Both metrics are then combined with a penalty factor which aims to regulate the effect that each value has on the search. BCD-MOIA was tested on the UF and F test suites, which consist of bi-objective and three-objective problems. Its computational complexity was not provided.

### 3 Our proposal

Here, we present the details of our proposed algorithm called *Multi-objective Artificial Immune System based on Decomposition* (MOAISDX). The main focus

in the design of the algorithm is the role of the clonal selection principle combined with additional mechanisms used to regulate the population's state. Thus, selection is divided into positive selection and negative selection.

**Decomposition** The design of our proposed MOAISDX follows the MOEA/D framework [21]. At each generation, MOAISDX maintains:

- a population of antibodies (**Ab**) of size  $n$ , which represent potential solutions,
- a set of weighted vectors  $(\lambda_1, \lambda_2, \dots, \lambda_n)$
- a scalarizing function  $g^{sch}(\cdot)$
- a neighborhood  $B(\cdot)$  of size  $T$
- a population of  $N_C$  of clones  $C$
- an affinity function  $Aff(\cdot)$

In MOEA/D, each sub-problem is solved in a simultaneous and collaborative manner with the sub-problems within its neighborhood. From the neighboring solutions, parents are chosen for crossover. The offspring  $y$  is then mutated and used to update the neighborhood or the general population. The criteria used to decide if the offspring  $y$  will replace the current solution  $x$  associated to a sub-problem relies on their scalarizing function values. That is if  $g^{sch}(y) < g^{sch}(x)$ ,  $y$  will replace  $x$  as the solution associated to a sub-problem. To avoid filling the entire population with the same solution, a number of replacements is fixed by the user. MOAIS keeps a population of clones along the optimization process. The clone population is being continuously filled with clones, which are copies of antibodies with the highest affinities. Once this population is full, clones undergo *somatic mutation*. The mutation scheme used in MOAISDX is polynomial-based mutation [5].

Keeping a population of clones is a recurrent mechanism in MOAIS' designs, but the clones are used as a pool of parents that are adopted for crossover [14]. Instead of using a single solution (offspring) for the update process, MOAIS uses the whole population of clones to perform a local or global update. This way, we take advantage of all mutated clones and, to a lesser extent, of identical clones. If  $g^{sch}(c_i) < g^{sch}(Ab_i)$  then a clone  $c_i$  replaces the solution  $Ab_i$  associated to the sub-problem  $i$ .

**Cloning operator** Given that we are aiming to keep a solution set with good convergence and diversity at each generation, we are faced with the challenge of selecting solution members that will reflect it. In a decomposition approach, we select solutions with the lowest possible aggregation value given a reference vector and a scalarizing function. This selection method alone will ensure that we keep good solutions (according to the scalarizing function of our choice). To make sure that we build a good repertoire of candidate solutions for the selection, we propagate solutions that are close to the related weighted vector. In this paper, we use cosine similarity as an affinity measure. The cosine similarity of two vectors  $\vec{x}$  and  $\vec{y}$  is given by:

$$\cos(\alpha) = \frac{\vec{x} \cdot \vec{y}}{\sqrt{||\vec{x}|| ||\vec{y}||}} \quad (4)$$

The cosine similarity value of two vectors reflects how close they are in terms of the angle between them. This value is in the range  $[0, 1]$  where 1 indicates the two vectors are in the same direction and 0 that they are orthogonal to each other. Cosine similarity is invariant to scale and works well in higher dimensional spaces.

Linge Li *et. al.* [12] used the following expression to assign clones to each individual:

$$c_i = \left\lceil |C| \times \frac{affinity}{\sum_{j=0}^{|C|} affinity} \right\rceil$$

where  $c_i$  is the number of assigned clones,  $|C|$  is the size of the clone population and *affinity* is the affinity measure used. In our case, we adopt the value of cosine similarity of the individual and its corresponding weighted vector as the affinity measure, to propagate individuals closely related. Once the number of clones is calculated, the cloning operator is applied to the selected antibodies. The cloning operator is given by:

$$C = \bigcup_{i=1}^T [Ab_i \otimes c_i]$$

where  $\otimes$  is the cloning operation, that is, the process used to generate or to replicate an antibody keeping all of its characteristics intact.

**Positive and negative selection** Once clones are generated, the population of clones undergoes mutation. The role of the affinity function is to generate mutation percentages that correspond to the affinities. Namely, the goal is that the least fit antibodies are subject to a more intensive mutation rate than those who are more fit according to the following expression  $pm = \exp(-5.0 * \text{affinity value})$  [4]. To avoid loss of diversity, MOAISDX inspects all clones to identify which ones are identical (clones that have not been mutated) and make them candidates to be removed from the population. This is called Negative Selection. The identical clones can be either ignored (become anergic) or replaced by new randomly generated antibodies. The goal is to keep the population of clones as diverse as possible and to take care of convergence by setting a limit on the number of discarded clones.

Positive selection can be interpreted as the process of local/global improvement. That is, each active element in the population of clones  $C$  is a candidate to replace elements of the neighborhood or the main population. A clone can become the associated solution to a sub-problem if  $g^{scl}(c_i) < g^{scl}(Ab_j)$  and  $affinity(c_i) \geq affinity(Ab_j)$  for  $i = 1, 2, \dots, |C|$  and  $j = 1, 2, \dots, N$ .

### 3.1 Our proposed algorithm

MOAISDX begins with the initialization where the  $T$  closest sub-problems to each sub-problem are computed (neighborhood). At this stage, the main population is initialized by randomly generating  $N$  antibodies  $Ab$  and their objective values are calculated. In the main loop, we start by updating the reference vector  $z^*$ . Subsequently, cloning takes place by selecting antibodies with the highest affinities. For each selected antibody, a fixed number of clones is produced according to its affinity. When cloning is finalized, the clones are mutated. The mutation rate is inversely proportional to their affinity value. Negative selection is then triggered to decrease the number of identical clones, replacing some of them by new randomly generated antibodies. The last stage is related to local/global improvement, where highly fit clone population members take the place of the associated solution to a particular sub-problem. The whole process can be viewed in Algorithm 1. In MOAISDX, recombination is included in a straightforward way: the parents are chosen from the neighborhood and the offspring are incorporated in the population of clones. The computational complexity of MOAISDX depends on the main loop; cloning, negative selection, and mutation each take  $O(C)$ , where  $C$  is the size of the clone population, and the update takes  $O(CT)$  where  $T$  the size of the neighborhood. The computational complexity of MOAISDX is  $O(NCT)$  in each generation, where  $N$  is the size of the population.

## 4 Validation of our proposed approach

The performance of our proposed MOAISDX was compared with respect to that of other state-of-the-art MOAISs and with respect to two MOEAs.

### 4.1 Experimental settings

16 test problems were used in our experiments, including 7 problems from the DTLZ test suite (DTLZ1-DTLZ7) and 9 problems from the WFG test suite (WFG1-WFG9). The number of variables for each problem are given in Table 1. In the first experiment, all 16 instances were tested with 3 objectives. In the second experiment, only DTLZ1 was used to explore the capabilities of the algorithm in a high dimensional objective space. Four recent MOAISs were used in our experiments: BCD-MOIA, HEIA, MOIA-DCSS and VD-MOIA. The parameters for each algorithm are those suggested in their original articles. For the MOAISs based on decomposition, the niche size and other decomposition related parameters are given in Table 1. For MOAISDX,  $N_c$  was set equal to the niche size and  $N_R$  was set to one. The weighted vectors were generated using the Das-Dennis approach [3] for 3, 4 and 7 objectives. For 5, 6, 8, 9 and 10 objectives, the weighted vectors were generated using the Riesz s-Energy method [2].

We adopted the hypervolume indicator [23] to assess performance:

$$I_{HV}(A : z^{\vec{ref}}) = \{\cup \text{volume}(v : z^{\vec{ref}}) | v \in A\} \quad (5)$$



**Input:** MOP, a stopping criterion,  $N$ : number of sub-problems, set of weighted vectors uniformly distributed  $\lambda^1, \dots, \lambda^N$ ,  $T$ : size of the neighborhood,  $N_c$ : or  $|C|$  clone population size,  $N_R$ : maximum number of replacements

```

begin
  Initialization;
  EP =  $\emptyset$ ;
  Compute Euclidean distances between any two weighted vectors and
  determine the  $T$  closest ones for each vector. For each  $i = 1, \dots, N$ , find
   $B(i) = \{i_1, \dots, i_N\}$ , where  $\lambda^{i_1}, \dots, \lambda^{i_T}$  are the  $T$  closest weighted
  vectors to  $\lambda^i$ ;
  Randomly generate an initial population  $Ab^1, \dots, Ab^N$  or by a problem
  specific method.  $FV^i = F(x^i)$ ;
  Update;
  while termination criterion not met do
    for  $i = 0$  to  $N$  do
      Reference point update  $z^*$ : for each  $j = 1, \dots, k$  if  $z_j < f_j(y)$ ,
      then  $z_j = f_j(y)$ ;
      Clone ;
      Mutate ;
      Negative selection;
      Local/global update (positive selection): For each
       $i = 1, \dots, N_C$  and for each index  $j \in B(i)$ , if
       $g^{scl}(C_i|\lambda, z) \leq g^{scl}(Ab_j|\lambda, z)$  and  $affinity(C_i) > affinity(Ab_j)$ 
      then  $Ab_j = C_i$  y  $FV^j = F(C_i)$ ;
      EP update;
    end
  end
end

```

Algorithm 1: MOAISDX algorithm

where  $A$  is the approximation set and  $z^{ref}$  is a reference point. For DTLZ1-DTLZ2, DTLZ4-DTLZ6, the reference point was set to  $(2, 2, \dots, 2)$ , for DTLZ3  $(4, 4, \dots, 4)$ , for DTLZ7  $(2, 2, \dots, 2, 8)$  and for the WFG test suite, it was set to  $(3, 5, \dots, 2 * k + 1)$ .

For each experiment, we performed 20 independent runs for each algorithm, test instance and number of objectives. MOAISDX, MOEA/D and NSGA-II were implemented in C/C++ in the EMO project framework [1], while BCD-MOIA, HEIA, MOIA-DCSS and VD-MOIA were implemented in Java in the jMetal framework [8].

## 4.2 Experimental results

The results of all our experiments were assessed using the Wilcoxon rank sum test with a 95% confidence. In the first experiment, we investigated the performance of MOAISDX with respect to 4 recent MOAISs: BCD-MOIA, HEIA, MOIA-DCSS and VD-MOIA. BCD-MOIA, MOIA-DCSS and VD-MOIA are

**Table 1.** Parameters values chosen for the experiments

Parameters/ Number of objectives	3	4	5	6	7	8	9	10
Population size (N)	136	166	175	203	210	240	270	290
Niche size	27	33	35	40	42	54	58	60
SBX parameters	$p_c = 1.0, \eta_c = 20, \eta_m = 20$							

decomposition-based algorithms while HEIA is a hybrid framework which uses sub-populations, each of which is evolved with a different evolutionary strategy. BCD-MOIA, HEIA and VD-MOIA require fixed mutation rates which were set at  $1/n$ , where  $n$  is the number of variables. BCD-MOIA, MOAI-DCSS and VD-MOIA use differential evolution in the optimization process, while MOAISDX uses SBX and polynomial-based mutation.

In Table 2, we show the comparison of results between our proposed MOAISDX and the other MOAISs and MOEAs adopted for both test suites with three objectives. For the DTLZ test suite, MOAISDX outperformed BCD-MOIA, MOIA-DCSS and VD-MOIA in four out of seven instances and it performed slightly better than HEIA in three out of seven instances. BCD-MOIA, MOIA-DCSS and VD-MOIA outperformed MOAISDX in three out of seven instances and HEIA outperformed MOAISDX in three out of seven instances. MOAISDX shows better performance when dealing with problems with multi-modal and disconnected Pareto fronts, while other MOAISs tend to perform better when solving uni-modal problems.

When comparing results with respect to MOEAs in the DTLZ test suite, our proposed MOAISDX outperformed MOEA/D and NSGA-II in 5 and 4 out of 7 instances. NSGA-II outperformed MOAISDX in 3 out of 7 instances, and MOEA/D performed similarly to MOAISDX in 2 out of 7 instances. In the WFG test suite, MOAISDX outperformed MOEA/D in 4 out of 9 instances, and it outperformed NSGA-II in 4 out of 9 instances. MOEA/D outperformed MOAISDX in 2 out of 9 instances and NSGA-II outperformed MOAISDX in 3 out of 9 instances. MOAISX shows that it is able to deal with different problem characteristics with moderate success, although its performance is best when dealing with concave, non-separable, parameter-dependent problems.

In the second experiment, as shown in Table 3, we studied the performance of our proposed MOAISDX with respect to other MOAISs and with respect to MOEAs in a high dimensional objective space. Although we performed experiments using the same test problems as before, due to space limitations, we will present here only one test instance with 3 to 10 objectives, but these results are representative of the behavior of the algorithms adopted in our experimental study. We selected DTLZ1. The parameters settings for this experiment are shown in Table 1.

With three, five, six and seven objectives, our proposed MOAISDX outperformed all the other algorithms in the comparison. With four objectives, MOAISDX outperformed all but one algorithm (HEIA), with respect to which it had a similar performance. With eight and ten objectives, MOAISDX outper-

formed 5 out of 6 algorithms, showing a similar performance to that of MOEA/D. With nine objectives, MOAISDX outperformed BCD-MOIA, HEIA, MOIA-DCSS, VD-MOIA and NSGA-II. MOAISDX was outperformed by MOEA/D. In general, MOAISDX performs better when dealing with problems with more than 3 objectives in the test problem selected.

**Table 2.** Comparison of the hypervolume values obtained by MOAISDX and four recent MOIAs and two MOEAs using the DTLZ and WFG test suites with three objectives. The symbols +, -, ~ indicate whether a result is better, worse or similar.

Test problem/ Algorithm	BCD-MOIA	HEIA	MOIA-DCSS	VD-MOIA	MOEA/D	NSGA-II	MOAISDX
DTLZ1	<b>7.64E+00</b> - 1.05E+00	<b>7.97E+00</b> ~ 5.40E-04	<b>7.72E+00</b> - 9.17E-01	<b>7.74E+00</b> - 8.20E-01	<b>7.97E+00</b> ~ 1.40E-04	<b>7.97E+00</b> + 6.07E-04	<b>7.97E+00</b> 2.61E-04
DTLZ2	<b>7.42E+00</b> + 4.47E-04	<b>7.39E+00</b> - 3.47E-03	<b>7.42E+00</b> + 5.61E-04	<b>7.42E+00</b> + 4.73E-04	<b>7.39E+00</b> - 1.15E-03	<b>7.37E+00</b> - 2.16E-02	<b>7.39E+00</b> 2.11E-03
DTLZ3	<b>5.06E+01</b> - 2.37E+01	<b>6.32E+01</b> - 9.59E-01	<b>4.90E+01</b> - 2.40E+01	<b>4.98E+01</b> - 2.32E+01	<b>6.34E+01</b> - 9.42E-03	<b>6.34E+01</b> - 1.10E+00	<b>6.34E+01</b> 1.79E-03
DTLZ4	<b>7.42E+00</b> + 7.43E-04	<b>7.39E+00</b> + 4.26E-03	<b>7.42E+00</b> + 1.41E-02	<b>7.42E+00</b> + 1.33E+00	<b>6.47E+00</b> - 1.06E-02	<b>7.38E+00</b> + 1.42E-02	<b>7.15E+00</b> 4.34E-01
DTLZ5	<b>6.08E+00</b> - 7.37E-05	<b>6.10E+00</b> + 3.27E-04	<b>6.08E+00</b> - 7.82E-05	<b>6.08E+00</b> - 1.03E-04	<b>6.09E+00</b> ~ 1.87E-04	<b>6.10E+00</b> + 3.88E-04	<b>6.09E+00</b> 2.07E-03
DTLZ6	<b>6.08E+00</b> + 9.14E-06	<b>6.11E+00</b> + 5.43E-05	<b>6.08E+00</b> + 1.17E-05	<b>6.08E+00</b> + 1.13E-05	<b>5.79E+00</b> - 7.58E-02	<b>5.51E+00</b> - 2.55E-01	<b>5.88E+00</b> 9.23E-02
DTLZ7	<b>1.70E+01</b> - 9.88E-01	<b>1.72E+01</b> - 7.16E-01	<b>1.67E+01</b> - 1.48E+00	<b>1.50E+01</b> - 2.79E+00	<b>1.73E+01</b> - 2.14E+00	<b>1.73E+01</b> - 3.27E-02	<b>1.73E+01</b> 1.37E-02
WFG1	<b>4.30E+01</b> - 1.02E+00	<b>8.03E+01</b> + 2.57E+00	<b>4.31E+01</b> - 1.02E+00	<b>4.34E+01</b> - 9.65E-01	<b>5.42E+01</b> ~ 1.49E+00	<b>5.19E+01</b> - 1.82E+00	<b>5.41E+01</b> 1.47E+00
WFG2	<b>9.67E+01</b> - 5.39E-01	<b>9.97E+01</b> + 2.78E-01	<b>9.64E+01</b> - 6.44E-01	<b>9.65E+01</b> - 5.81E-01	<b>9.67E+01</b> - 1.04E+00	<b>9.96E+01</b> + 2.45E-01	<b>9.75E+01</b> 6.68E-01
WFG3	<b>7.20E+01</b> - 6.51E-01	<b>7.46E+01</b> + 2.70E-01	<b>7.15E+01</b> - 7.42E-01	<b>7.16E+01</b> - 5.50E-01	<b>7.35E+01</b> - 6.27E-01	<b>7.49E+01</b> + 3.15E-01	<b>7.39E+01</b> 5.54E-01
WFG4	<b>6.98E+01</b> - 6.99E-01	<b>7.36E+01</b> - 4.07E-01	<b>6.96E+01</b> - 6.81E-01	<b>6.98E+01</b> - 2.47E+09	<b>7.37E+07</b> - 4.09E-01	<b>7.35E+01</b> - 6.13E-01	<b>7.39E+01</b> 2.63E-01
WFG5	<b>7.03E+01</b> - 5.34E-01	<b>7.24E+01</b> + 3.04E-01	<b>7.02E+01</b> - 5.87E-01	<b>7.04E+01</b> - 5.44E-01	<b>7.07E+01</b> - 2.92E-01	<b>7.20E+01</b> + 3.79E-01	<b>7.09E+01</b> 4.21E-01
WFG6	<b>7.15E+01</b> - 4.52E-01	<b>7.14E+01</b> - 1.14E+00	<b>7.16E+01</b> - 3.65E-01	<b>7.17E+01</b> - 7.83E-01	<b>7.16E+01</b> - 6.16E-01	<b>7.15E+01</b> - 6.54E-01	<b>7.17E+01</b> 4.78E-01
WFG7	<b>7.24E+01</b> - 3.66E-01	<b>7.42E+01</b> ~ 2.49E-01	<b>7.21E+01</b> - 4.22E-01	<b>7.23E+01</b> - 3.39E-01	<b>7.42E+01</b> ~ 1.22E-01	<b>7.42E+01</b> ~ 3.29E-01	<b>7.42E+01</b> 2.21E-01
WFG8	<b>6.58E+01</b> - 6.94E-01	<b>6.94E+01</b> - 4.39E-01	<b>6.52E+01</b> - 9.41E-01	<b>6.55E+01</b> - 9.21E-01	<b>7.12E+01</b> ~ 5.26E-01	<b>7.00E+01</b> - 4.42E-01	<b>7.12E+01</b> 2.54E-01
WFG9	<b>6.99E+01</b> - 1.82E+00	<b>6.92E+01</b> - 9.07E-01	<b>6.92E+01</b> - 1.04E+00	<b>6.88E+01</b> - 1.04E+00	<b>7.14E+01</b> + 1.63E+00	<b>7.08E+01</b> + 1.18E+00	<b>7.03E+01</b> 1.21E+00

## 5 Conclusions

In this work, we introduced a new multi-objective artificial immune system algorithm based on decomposition (MOAISDX), in which we preserved the immune components through specialized operators and mechanisms. Our proposed cloning operator uses cosine similarity to compute the number of clones or replicas assigned to each member of the population, aiming to produce more copies of those with the highest affinities, that is, the ones that are close to the reference

**Table 3.** Comparison of the hypervolume values between MOAISDX and four recent MOAISs and two MOEAs on DTLZ1 with three to ten objectives. The symbols +, -, ~ indicate whether a result is better, worse or similar.

Dimensionality/ Algorithm	BCD-MOIA	HEIA	MOIA-DCSS	VD-MOIA	NSGA2	MOEA/D	MOAISDX
3D	<b>7.64E+00</b> - 1.05E+00	<b>7.97E+00</b> ~ 5.40E-04	<b>7.72E+00</b> - 9.17E-01	<b>7.74E+00</b> - 8.20E-01	<b>7.97E+00</b> + 6.07E-04	<b>7.97E+00</b> ~ 1.40E-04	<b>7.97E+00</b> 2.61E-04
4D	<b>9.46E+00</b> - 5.22E+00	<b>1.60E+01</b> ~ 2.45E-02	<b>1.12E+01</b> - 4.33E+00	<b>9.46E+00</b> - 5.22E+00	<b>1.55E+01</b> - 2.19E-02	<b>1.60E+01</b> - 4.82E-05	<b>1.60E+01</b> 9.85E-04
5D	<b>2.70E+01</b> - 6.96E+00	<b>3.17E+01</b> - 2.15E+00	<b>2.93E+01</b> - 4.95E+00	<b>2.77E+01</b> - 5.96E+00	<b>0.00E+00</b> - 0.00E+00	<b>3.19E+01</b> - 4.28E-02	<b>3.19E+01</b> 2.15E-02
6D	<b>4.46E+01</b> - 1.58E+01	<b>5.73E+01</b> - 1.29E+01	<b>5.63E+01</b> - 1.07E+01	<b>4.34E+01</b> - 1.26E+01	<b>0.00E+00</b> - 0.00E+00	<b>6.32E+01</b> - 4.23E-01	<b>6.37E+01</b> 1.32E-01
7D	<b>3.85E+01</b> - 4.20E+01	<b>9.28E+01</b> - 4.21E+01	<b>1.18E+02</b> - 1.73E+01	<b>3.54E+01</b> - 4.15E+01	<b>0.00E+00</b> - 0.00E+00	<b>1.27E+02</b> - 1.07E+00	<b>1.28E+02</b> 1.99E-01
8D	<b>1.77E+02</b> - 6.65E+01	<b>1.19E+02</b> - 9.25E+01	<b>2.40E+02</b> - 2.61E+01	<b>2.03E+02</b> - 5.80E+01	<b>0.00E+00</b> - 0.00E+00	<b>2.56E+02</b> ~ 4.82E-01	<b>2.56E+02</b> 2.14E-01
9D	<b>3.12E+02</b> - 1.32E+02	<b>1.14E+02</b> - 1.53E+02	<b>4.45E+02</b> - 8.76E+01	<b>3.19E+02</b> - 1.36E+02	<b>0.00E+00</b> - 0.00E+00	<b>5.12E+02</b> + 7.34E-02	<b>5.11E+02</b> 2.74E-01
10D	<b>5.78E+02</b> - 2.61E+02	<b>2.72E+02</b> - 3.09E+02	<b>8.94E+02</b> - 1.76E+02	<b>6.54E+02</b> - 2.49E+02	<b>0.00E+00</b> - 0.00E+00	<b>1.02E+03</b> ~ 2.18E-01	<b>1.02E+03</b> 1.65E-01

vectors when filling up the population of clones. Positive and negative selection attempt to overcome the loss of diversity that cloning introduces by regulating the number of identical individuals along the search.

Our experimental results showed that our proposed MOAISDX is capable of dealing with complex test problems. MOAISDX shows, in general, a similar or even better performance than NSGA-II and MOEA/D and it outperforms state-of-the-art MOAISs in most of the test problems adopted.

As part of our future work, we are interested in exploring different cloning schemes in which we take into consideration other metrics for clone assignment. We are also interested in studying the role of negative selection in the context of available clone solutions. The role of mutation in MOAISDX is clearly, very important, since it is the main source of diversity in the algorithm and, therefore, studying alternative mutation schemes is also an interesting path for future research.

## References

1. Hernández Gómez, R.: Parallel Hyper-Heuristics for Multi-Objective Optimization. Ph.D. thesis, Department of Computer Science, CINVESTAV-IPN, Mexico City, México (March 21 2018)
2. Blank, J., Deb, K., Dhebar, Y., Bandaru, S., Seada, H.: Generating Well-Spaced Points on a Unit Simplex for Evolutionary Many-Objective Optimization. IEEE Transactions on Evolutionary Computation **25**(1), 48–60 (February 2021)
3. Das, I., Dennis, J.E.: Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. SIAM Journal on Optimization **8**(3), 631–657 (1998)
4. De Castro, L.N., Von Zuben, F.J.: Learning and Optimization Using the Clonal Selection Principle. IEEE transactions on evolutionary computation **6**(3), 239–251 (2002)

5. Deb, K., Goyal, M., et al.: A Combined Genetic Adaptive Search (GeneAS) for Engineering Design. *Computer Science and informatics* **26**, 30–45 (1996)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6**(2), 182–197 (April 2002)
7. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable Test Problems for Evolutionary Multi-Objective Optimization. Tech. Rep. 112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland (2001)
8. Durillo, J.J., Nebro, A.J.: jMetal: A Java framework for multi-objective optimization. *Advances in Engineering Software* **42**(10), 760–771 (October 2011)
9. Gong, M., Jiao, L., Du, H., Bo, L.: Multiobjective immune algorithm with non-dominated neighbor-based selection. *Evolutionary Computation* **16**(2), 225–255 (Summer 2008)
10. Huband, S., Barone, L., While, L., Hingston, P.: A Scalable Multi-objective Test Problem Toolkit. In: Coello, C.A.C., Aguirre, A.H., Zitzler, E. (eds.) *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*. pp. 280–295. Springer. Lecture Notes in Computer Science Vol. 3410, Guanajuato, México (March 2005)
11. Li, H., Zhang, Q.: Multiobjective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation* **13**(2), 284–302 (April 2009)
12. Li, L., Lin, Q., Li, K., Ming, Z.: Vertical Distance-Based Clonal Selection Mechanism for the Multiobjective Immune Algorithm. *Swarm and Evolutionary Computation* **63** (June 2021), article Number: 100886
13. Li, L., Lin, Q., Liu, S., Gong, D., Coello Coello, C.A., Ming, Z.: A Novel Multi-Objective Immune Algorithm with a Decomposition-Based Clonal Selection. *Applied Soft Computing* **81** (August 2019), article number: UNSP 105490
14. Li, L., Lin, Q., Ming, Z.: A Survey of Artificial Immune Algorithms for Multi-Objective Optimization. *Neurocomputing* **489**, 211–229 (June 7 2022)
15. Li, L., Lin, W., Lin, Q., Ming, Z.: Balancing Convergence and Diversity in Multi-objective Immune Algorithm. In: *2020 12th International Conference on Advanced Computational Intelligence (ICACI)*. pp. 102–109 (2020)
16. Lin, Q., Chen, J., Zhan, Z.H., Chen, W.N., Coello, C.A.C., Yin, Y., Lin, C.M., Zhang, J.: A Hybrid Evolutionary Immune Algorithm for Multiobjective Optimization Problems. *IEEE Transactions on Evolutionary Computation* **20**(5), 711–729 (October 2016)
17. Lin, Q., Zhu, Q., Wang, N., Huang, P., Wang, W., Chen, J., Ming, Z.: A Multi-objective Immune Algorithm with Dynamic Population Strategy. *Swarm and Evolutionary Computation* **50**, 100477 (December 2019)
18. Miettinen, K.: *Nonlinear Multiobjective Optimization*, vol. 12. Springer Science & Business Media (1999)
19. Perelson, A.S., Oster, G.F.: Theoretical Studies of Clonal Selection: Minimal Antibody Repertoire Size and Reliability of Self-non-self Discrimination. *Journal of Theoretical Biology* **81**(4), 645–670 (1979)
20. Shang, R., Jiao, L., Liu, F., Ma, W.: A Novel Immune Clonal Algorithm for MO Problems. *IEEE Transactions on Evolutionary Computation* **16**(1), 35–50 (February 2012)
21. Zhang, Q., Li, H.: MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation* **11**(6), 712–731 (December 2007)

22. Zhang, Q., Liu, W., Li, H.: The Performance of a New Version of MOEA/D on CEC09 Unconstrained MOP Test Instances. In: 2009 IEEE Congress on Evolutionary Computation (CEC'2009). pp. 203–208. IEEE Press, Trondheim, Norway (May 2009)
23. Zitzler, E.: Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. Ph.D. thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland (November 1999)
24. Zitzler, E., Deb, K., Thiele, L.: Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Tech. Rep. 70, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriasstrasse 35, CH-8092 Zurich, Switzerland (December 1999)