Chapter

An Immune Multiobjective Optimization with Backtracking Search Algorithm Inspired Recombination

Hamed Ould Sidi, Rachid Ellaia, Emmanuel Pagnacco and Ahmed Tchvagha Zeine

Abstract

We propose a novel hybrid multiobjective (MO) immune algorithm for tackling continuous MO problems. Similarly to the nondominated neighbor immune algorithm (NNIA), it considers the characteristics of OM problems: based on the fitness values, the best individuals from the test population are selected and recombined to guide the rest of the individuals in the population to the Pareto front. But NNIA uses the simulated binary crossover (SBX), which uses the local search method. In our algorithm, the recombination is essentially inspired by the cross used in the backtracking search algorithm (BSA), but the adaptations are found in the immune algorithm. Thus, three variants are designed in this chapter, resulting in new recombination operators. They are evaluated through 10 benchmark tests. For the most advanced proposed variant, which is designed to have global search ability, results show that an improved convergence and a better diversity of the Pareto front are statistically achieved when compared with a basic immune algorithm having no recombination or to NNIA. Finally, the proposed new algorithm is demonstrated to be successful in approximating the Pareto front of the complex 10 bar truss structure MO problem.

Keywords: multiobjective optimization, evolutionary algorithms, backtracking search, hybrid recombination, artificial immune systems, truss optimization

1. Introduction

Planty of the multiobjective (MO) optimization problems lie in the engineering field. The objective functions are contradictory, the optimal solutions of these problems are known by the Pareto front. To get the optimal solutions of these problems, evolutionary algorithms (EAs) have been recognized to be very efficient in solving MO optimization problems by finding a representative Pareto front in one run. State-of-the-art algorithms are presented in [1–5]. These algorithms, which are population-based, are able to simultaneously explore various regions of the Pareto front.

In last past few years, immune systems have inspired new algorithms for resolutions OM problems. The fundamental principles of the artificial immune system (AIS) algorithm are clonal selection, [6] mutation, and more recently, recombination [7–15].

The nondominated neighbor-based immune optimization algorithm (NNIA) is effective to deal with MO problems [9]. NNIA has proved that it is advantageous to incorporate a crossover operator into the algorithm. To do this, it uses simulated binary crossover (SBX). But SBX is a recombination operator, which performs search near the recombination parent [16].

Backtracking Search Optimization Algorithm (BSA) is a new nature-inspired algorithm proposed by [17]. BSA's special mechanism to ensure a trial individual is effective, ability to learn fast solving different numerical optimization problems sequentially and quickly, with a clear structure. Since it was introduced, the BSA has attracted many research studies, and it has been applied to various optimization problems [18–20].

In this chapter, we develop a novel hybrid MO immune method to solve the problems of continuous multiobjective optimization. The NNIA algorithm uses the best individuals in the trial population to drive the search to Pareto front. But NNIA uses SBX, which mainly has local search capability. In our proposal, the recombination is inspired by the cross used in the BSA algorithm, but adaptations are found to fit the immune algorithm. Therefore, three variants are considered in the context of this chapter, which gives rise to new recombination operators for immune algorithm.

This chapter is organized as follows: In Section 2, we introduce the MO problem. In Sections 3 and 4, NNIA algorithm and BSA recombination are presented, and we propose new algorithms to solve the MO problem. The effectiveness of these algorithms is investigated in Section 5 when confronting to various MO test problems. In Section 6, the chosen algorithm is applied to solve the multiobjective topology optimization of truss structures. A short summary is proposed to conclude this paper.

2. Multiobjective optimization problem

The multiobjective optimization problem is formalized in this section. Concepts related to the Pareto front are introduced, firstly from a theoretical point of view and then when considering its approximation through a numerical approach [21].

2.1 Pareto front

Let us consider the following multiobjective optimization (MO) problem:

$$\min_{\mathbf{x}\in\Omega} \mathbf{f}(\mathbf{x}) = \left(f_1(\mathbf{x}), \cdots, f_m(\mathbf{x})\right)^{\mathrm{T}} \tag{1}$$

where m is the number of objective functions, $\mathbf{x} = (\mathrm{x}_1, \cdots, \mathrm{x}_n) \in \Omega$ is the n_x-dimensional decision space where each decision variable x_i is bounded by lower and upper limits $x_{li} \le x_i \le x_{ui}$ for $i = 1, \cdots, n_x$.

Pareto-front-related concepts are [22]:

1. *Pareto dominance: Suppose $\mathbf{x}_a$ and $\mathbf{x}_b$ are two different feasible solutions to the MO problem. Then $\mathbf{x}_a$ dominates $\mathbf{x}_b$ if and only if*

$$f_i(\mathrm{x}_a) \le f_i(\mathbf{x}_b))\forall i \in \{1, \cdots, m\} \tag{2}$$

and:

$$\exists k \in \{1, \dots, m\} \ f_k(\mathbf{x}_a) < f_k(\mathbf{x}_b) \tag{3}$$

1. *Pareto-optimal solution:* A solution $\mathbf{x}^*$ is said to be Pareto-optimal if there does not exist another solution that dominates it.

2. *Pareto-optimal set:* The Pareto-optimal set is the set X^* of all Pareto-optimal solutions.

3. *Pareto-optimal front:* The Pareto-optimal front is the set F^* of values or outcomes of all the objective functions, which corresponds to the solutions:

$$F^* = \left\{ \mathbf{f}(\mathbf{x}^*) = \left(f_1(\mathbf{x}^*), \dots, f_m(\mathbf{x}^*) \right)^{\mathrm{T}} \text{ such that : } \ \mathbf{x}^* \in X^* \right\} \tag{4}$$

2.2 Multiobjective solution

Otherwise, the following terms are cited in the reference [9]:

1. *Antibody*: An antibody refers to the coding of a decision variable **x**. In this study, a real-valued representation is adopted, being **x** itself.

2. *Crowding distance*: The crowding distance (CD) is a measure for diversity maintenance [1]. It reads:

$$CD\left(\widehat{\mathbf{X}}\right) = \sum_{j=1}^{m} \frac{D_j\left(\widehat{\mathbf{X}}\right)}{f_j^{\max} - f_j^{\min} + \varepsilon_D} \tag{5}$$

where $f_j^{\max}$ and $f_j^{\min}$ are maximal and minimal values of the j-th objective respectively, ε_D is a small number and:

$$D_j\left(\widehat{\mathbf{X}}\right) = \begin{cases} \infty & \text{if} \quad \hat{\mathbf{x}}_k \text{ is a boundary point of } \widehat{\mathbf{X}} \\ \min\left| f_k\left(\widehat{\mathbf{X}}\right) - f_l\left(\widehat{\mathbf{X}}\right) \right| & \text{otherwise} \end{cases} \tag{6}$$

for $k, l \in \{1, \cdots, n_x\}$ such that $k \neq l \neq j$.

3. Immune optimization algorithm and recombination operator

3.1 Nondominated neighbor immune optimization algorithm

Nondominated neighbor immune algorithm (NNIA), using the nondominated neighbor-based selection and proportional cloning, pays more attention to the less-crowded regions of the current trade-off front.

We denote by $\widehat{\mathbf{X}}$ the dominant population, $\mathbf{X}_A$ the active population and $\mathbf{X}_C$ the clone population at time t are stored by time-dependent variable matrices, Their sizes are $n_{\hat{X}}$, n_A, and n_C respectively.

The NNIA algorithm is presented in 1 where: = is the update operator. n_D, $n_{\hat{X}\max}$, $n_{A\max}$, n_C, and n_{it}, the number of iterations.

Algorithm 1 Pseudo code of NNIA

```
Function $\widehat{\mathbf{X}} = NNIA\left(n_x, m, \mathbf{f}(\mathbf{x}), \mathbf{x}_l, \mathbf{x}_u, n_{\hat{X}\max}, n_{A\max}, n_C, n_{it}\right)$
1: Generate a uniform random initial population $\widehat{\mathbf{X}}$ of size $n_C \times n_x$ in respect to $\mathbf{x}_l$ and $\mathbf{x}_u$;
2: $\widehat{\mathbf{X}} \coloneqq Find\_Non\_Dominated\left(\widehat{\mathbf{X}}|\mathbf{f}(\mathbf{x})\right)$;
3: for $t = 0 : n_{it}$, do
4:    $\mathbf{X}_A \coloneqq CD\_Truncation\left(\widehat{\mathbf{X}}, n_{A\max}\right)$;
5:    $\mathbf{X}_C \coloneqq Cloning(\mathbf{X}_A, n_C)$;
6:    $\mathbf{X}_C \coloneqq Recombination(\mathbf{X}_C, \mathbf{X}_A, \mathbf{x}_l, \mathbf{x}_u)$;
7:    $\mathbf{X}_C \coloneqq Hypermutation(\mathbf{X}_C, \mathbf{x}_l, \mathbf{x}_u)$;
8:    $\widehat{\mathbf{X}} \coloneqq Find\_Non\_Dominated\left(\left[\widehat{\mathbf{X}}; \mathbf{X}_C\right]|\mathbf{f}(\mathbf{x})\right)$;
9:    $\widehat{\mathbf{X}} \coloneqq CD\_Truncation\left(\widehat{\mathbf{X}}, n_{\widehat{X}\max}\right)$;
end for
```

3.2 Recombination and crossovers

3.2.1 NNIA recombination

For a recombination, operation of NNIA has been adopted in many MO EAs [1, 4], an antibody of the cloning population and an antibody of the active population are selected and modified as:

$$\{\mathbf{X}_C\}_{ij} \coloneqq \begin{cases} \frac{1-\beta}{2}\{\mathbf{X}_C\}_{ij} + \frac{1+\beta}{2}\{\mathbf{X}_A\}_{kj} & \text{if} \quad a = 1 \ \& \ b = 0 \\ \frac{1+\beta}{2}\{\mathbf{X}_C\}_{ij} + \frac{1-\beta}{2}\{\mathbf{X}_A\}_{kj} & \text{if} \quad a = 1 \ \& \ b = 1 \\ \{\mathbf{X}_C\}_{ij} & \text{if} \quad a = 0 \end{cases} \quad |a \sim \mathcal{U}(0,1), \ \ b \sim \mathcal{U}(0,1) \tag{7}$$

for $i \in \{1, \dots, n_C\}, j \in \{1, \dots, n_x\}$, and k a random integer uniformly chosen in $\{1, \dots, n_A\}$. Above, $\mathcal{U}$ is the uniform discrete distribution, and β is a sample from a random distribution having the density:

$$p(\beta) = \begin{cases} 0 & \text{if} \quad \beta < 0 \\ \frac{\eta+1}{2}\beta^\eta & \text{if} \quad 0 \le \beta \le 1 \\ \frac{\eta+1}{2\beta^{\eta+2}} & \text{if} \quad \beta > 1 \end{cases}$$

where η is a real nonnegative distribution. Hence, four independent random variables are involved in this recombination operation: a, b, k, and β. A boundary control is performed by:

$$\{\mathbf{X}_C\}_{ij} := \begin{cases} x_{li} & \text{if} \quad \{\mathbf{X}_C\}_{ij} < x_{li} \\ \{\mathbf{X}_C\}_{ij} & \text{if} \quad x_{li} \le \{\mathbf{X}_C\}_{ij} \le x_{ui} \\ x_{ui} & \text{if} \quad \{\mathbf{X}_C\}_{ij} > x_{ui} \end{cases}$$

3.2.2 Crossover operator of backtracking search optimization algorithm

Crossover strategy of BSA [17]. It consists in mixing two input populations $\mathbf{X}_P$ and $\mathbf{X}_Q$ to form a new output population $\mathbf{X}_R$, of equal sizes: $n_X \times n_x$. Then, BSA' crossover reads:

$$\{\mathbf{X}_R\}_{ij} := \begin{cases} \{\mathbf{X}_P\}_{ij} & \text{if} \quad \mathbf{T}_{ij} = 0 \\ \{\mathbf{X}_Q\}_{ij} & \text{if} \quad \mathbf{T}_{ij} = 1 \end{cases} \tag{8}$$

for $i \in \{1, \ldots, n_X\}, j \in \{1, \ldots, n_x\}$ and where $\mathbf{T}$ is a boolean matrix of sizes: $n_X \times n_{x,}$ which is formed by following the algorithm 2.

To control the amount of mixing between the two populations $\mathbf{X}_P$ and $\mathbf{X}_Q$, we must define the parameter η such that $0 < \eta \le n_x$. Moreover, we perform a random permutation on the lines of the $\mathbf{X}_P$ population before applying the relation (8).

Algorithm 2 Algorithm for the generation of the T matrix used in the BSA crossover

```
Initialize $\mathbf{T} := \mathbf{0}$ and $a := \mathcal{U}(0, 1)$;
if $a = 0$ then
    for $i = 1 : n_X$ do
        $u := Permuting(1 : n_x)$;
        $b := \mathcal{U}(0, \eta)$;
        for $k = 1 : b$ do
            $j = u_k$;
            $T_{ij} = 1$;
        end for
    end for
else
    for $i = 1 : n_X$ do
        $j := \mathcal{U}(0, n_x)$;
        $T_{ij} = 1$;
    end for
end if
```

4. Recombination propositions for an hybrid algorithm

To get a more efficient immune algorithm, we will propose a hybridization method, which consists of exchanging the crossover operator used for recombination in NNIA with a new recombination operator inspired by BSA [23].

To find this new algorithm, we have to use two ideas:

1. The first idea consists of expanding the active population in order to obtain an extended active population, having its size equal to the clonal population size. The simplest way to achieve this consists of duplicating the active population;

2. The second idea consists of replacing the active population for the crossover by the clonal population, leading finally to a crossover that uses only the clonal population.

5. Experiments

In this section, we study the performance of the hybridization when solving some well-known MO techniques including five ZDT problems [24] and five DTLZ problems [25].

An optimization problem is typically written as:

$$\min_{\mathbf{x}\in\Omega} \quad \mathbf{f}(\mathbf{x}) = \left(f_1(\mathbf{x}), \cdots, f_m(\mathbf{x})\right)^{\mathrm{T}} \tag{9}$$

where m is the number of objective functions, $\mathbf{x} = (\mathrm{x}_1, \cdots, \mathrm{x}_n) \in \Omega$ is the n_x-dimensional decision space where each decision variable x_i is bounded by lower and upper limits $x_{li} \le x_i \le x_{ui}$ for $i = 1, \cdots, n_x$.

5.1 Performance metrics

Approximate Pareto front solution of MO algorithms must achieve these two goals:

1. Convergence toward the true Pareto front; and

2. Diversity of solutions: the Pareto front must be uniformly distributed and spread over the entire feasible objective space to adequately capture the trade-offs.

For benchmark test problems, the true Pareto front is known, allowing to exploit performance metrics that used it.

We opted for two performance metrics for assessing algorithms efficiency. To measure the extent of the convergence to the true set of Pareto-optimal solutions and the spread of the Pareto front set, a normalized version of the inverted generational distance (IDG) metric proposed by [26] is adopted, while a normalized version of the spacing metric introduced by [27] enables to measure the uniformity of the obtained solutions.

5.1.1 Normalized inverted generational distance

The normalized inverted generational distance (NIGD) is based on a proposition of [26]. For measuring of the distance between the true Pareto front F^*, which is known at n^* discrete values—and stored in the matrix $\mathbf{F}(\mathbf{X}^*)$—and approximate solutions of the Pareto-optimal front $\mathbf{F}\left(\widehat{\mathbf{X}}\right)$:

$$\mathrm{NIGD}\left(\mathbf{F}(\mathbf{X}^*), \mathbf{F}\left(\widehat{\mathbf{X}}\right)\right) = \frac{1}{n^*}\sqrt{\sum\nolimits_{j=1}^{n^*} \mathbf{c}_j^2}, \tag{10}$$

for:

$$\mathbf{c}_j = \min_{i\in\{1,\ \dots,\ n_X\}} \left(\sqrt{\sum\nolimits_{k=1}^{m} \left(\overline{\mathbf{F}}_k\left(\mathbf{X}_i^*\right) - \overline{\mathbf{F}}_k\left(\widehat{\mathbf{X}}_j\right)\right)^2}\right), \qquad j \in \{1,\ \dots,\ n^*\}$$

where $\bar{\bullet}$ denotes a normalized objective function, ranging from 0 to 1 and defined by:

$$\overline{\mathbf{F}}_k(\mathbf{x}) = \frac{\mathbf{F}_k(\mathbf{x}) - \min(\mathbf{F}_k(\mathbf{X}^*))}{\max(\mathbf{F}_k(\mathbf{X}^*)) - \min(\mathbf{F}_k(\mathbf{X}^*))}. \tag{11}$$

To obtain smaller values of this measure, the approximated set $\mathbf{F}\left(\widehat{\mathbf{X}}\right)$ must be very close to the Pareto front and cannot miss any part of the whole Pareto front at the same time.

5.1.2 Normalized spacing measure

The spacing metric introduced by [27] is modified by taking normalized objectives functions. This leads to the normalized NSP measure, defined by:

$$\mathrm{NSP}(\mathbf{F}(\hat{\mathbf{X}})) = \sqrt{\frac{1}{n_X - 1}\sum\nolimits_{j=1}^{n_X}(\mathbf{d}_j - \overline{d})^2} \tag{12}$$

for:

$$\mathbf{d}_j = \min_{\substack{i\in\{1,\dots,n_X\}\\ i\neq j}}\left(\sqrt{\sum\nolimits_{k=1}^{m}\left(\overline{\mathbf{F}}_k(\hat{\mathbf{X}}_i) - \overline{\mathbf{F}}_k(\hat{\mathbf{X}}_j)\right)^2}\right), \qquad j\in\{1,\dots,n_X\}$$

where $\overline{d}$ is the mean of $\mathbf{d}$.

5.2 Empirical comparison

In this section, performance of five NNIA variants are evaluated. The five variants are:

1. NNIA-X: the NNIA algorithm without crossover;

2. NNIA: the algorithm proposed by [9];

3. NNIA+X1: the hybridization of the NNIA algorithm with the BSA crossover by using the first strategy proposed in the Section 4. Inputs of the BSA crossover function are (1) the clonal population and (2) a random permutation of an extended active population obtained by duplicating individuals;

4. NNIA+X2: the hybridization of the NNIA algorithm with the BSA crossover by using the second strategy proposed in the Section 4. Inputs of the BSA crossover function are (1) the clonal population and (2) a random permutation of the clonal population;

5. NNIA+X3: the hybridization of the NNIA algorithm with the BSA crossover by using the third strategy proposed in the Section 4. Inputs of the BSA crossover function are (1) the clonal population and (2) a random permutation of an extended active population obtained by duplicating individuals with a proportion of random individuals.

For NNIA, parameters proposed in Ref. [9] are set:

- maximum size of dominant population: $n_{\hat{X}\max} = 100$,
- maximum size of active population: $n_{A\max} = 20$, and
- size of clone population $n_C = 100$,

with the distribution index for SBX that is 15, the distribution index for polynomial mutation that is 20 and the mutation probability of $1/n_x$ and the number of iterations stopped at 250. For NNIA+X3, the proportion of random individuals is chosen to be equal to n_A and their distribution is uniform.

Figures 1 and **2** show the statistic box plots for NIGD and NSP obtained for 1000 independent runs performed on each test problems ZDT and DTLZ that are chosen by [9][1].

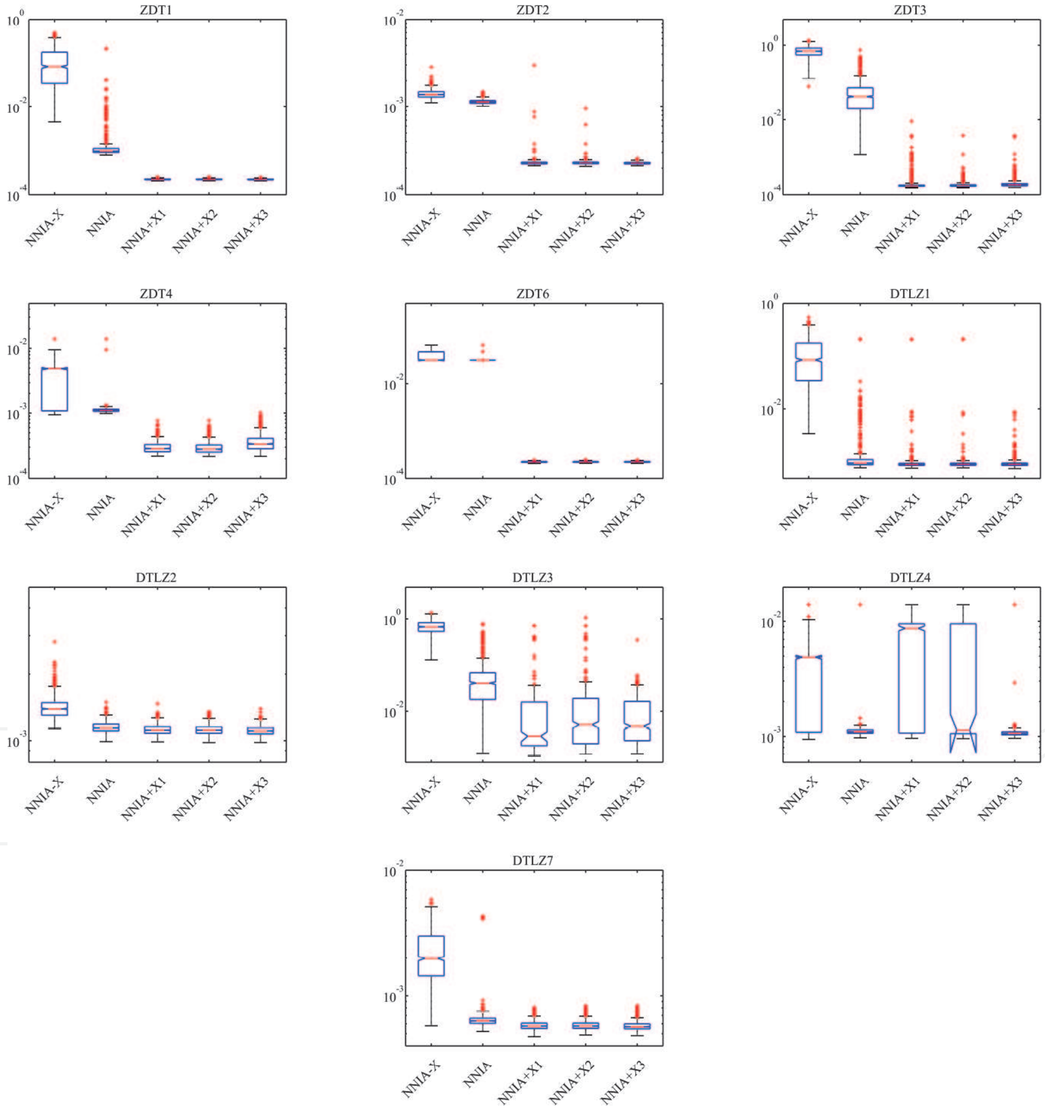

Figure 1.
NIGD obtained from 1000 independent runs of problems ZDT1, ZDT2, ZDT3, ZDT4, ZDT6, DTLZ1, DTLZ2, DTLZ3, DTLZ4, and DTL7.

[1] From informations given in [25], it is believed that the problem denoted DTLZ6 in [9] is in fact the DTLZ7 problem.

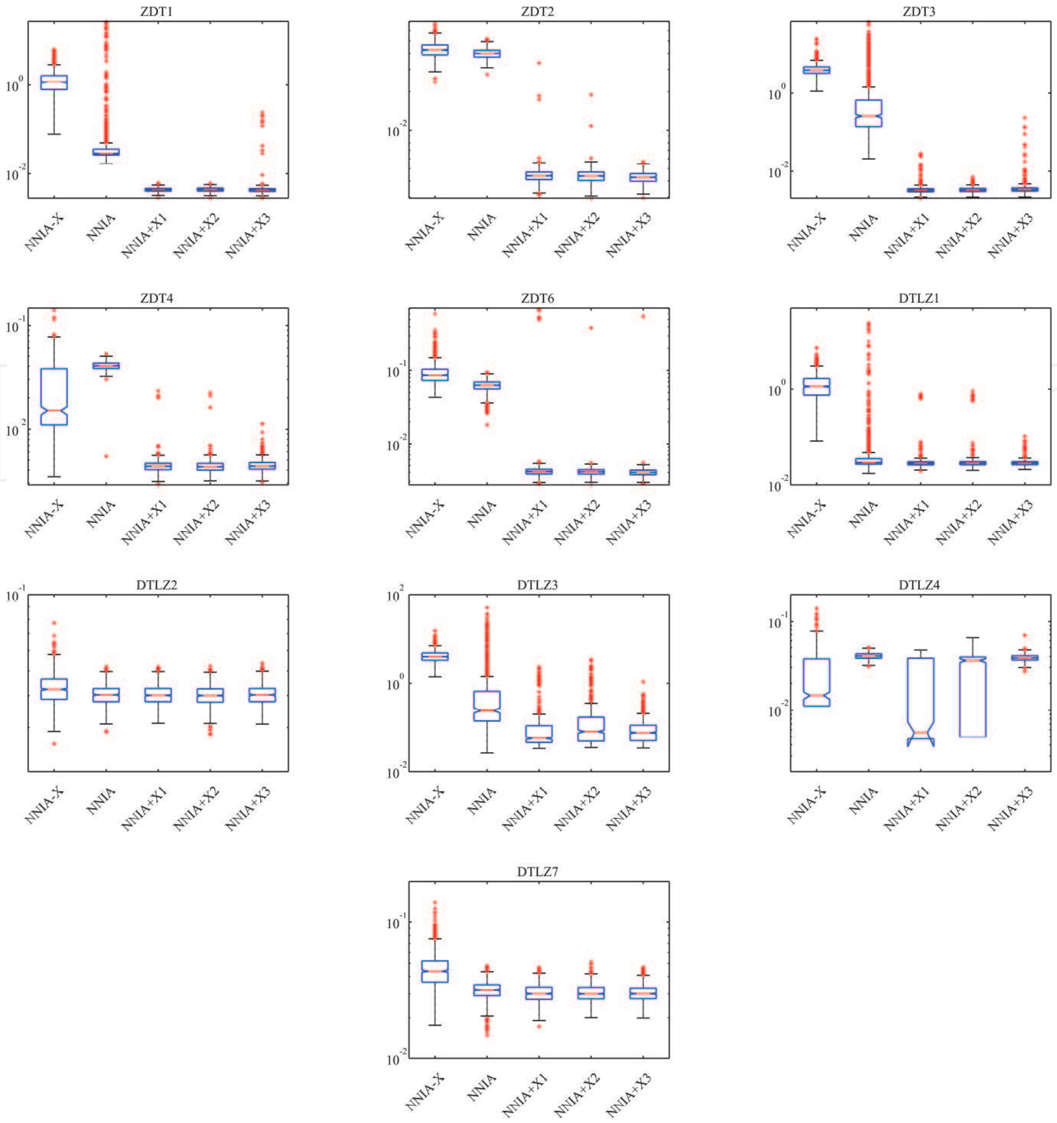

Figure 2.
Statistics box plots of NSP obtained from 1000 independent runs of benchmark test problems ZDT1, ZDT2, ZDT3, ZDT4, ZDT6, DTLZ1, DTLZ2, DTLZ3, DTLZ4, and DTL7.

NIGD's statistical results show that the NNIA algorithm is better than NNIA-X for the problems addressed. We also observe the efficiency of the NNIA+X1 and NNIA +X2 algorithms compared with NNIA with the exception of the difficult DTLZ4 test problem. For ZDT4 problem, NNIA+X3 is lower than NNIA+X1 and NNIA+X2. But we can always notice that our proposed algorithm NNIA+X3 remains superior to NNIA for the problems treated. Except for the two issues ZDT4 and DTLZ4, NSP shows the superiority of NNIA over NNIA-X. In all treated cases, NNIA+X1, NNIA +X2, and NNIA+X3 appear to be equal to or greater than NNIA. But for all these algorithms, the DTLZ4 problem seems to be the most difficult, since there are runs for which the Pareto front is approximated by a single, unique, point.

Table 1 shows the percentage of results showing a single point for the Pareto front of the DTLZ4 test problem when a sequence of 1000 runs is performed with

NNIA-X	NNIA	NNIA+X1	NNIA+X2	NNIA+X3
2%	11%	13%	11%	0.2%

Table 1.
Percentage of results exhibiting a single point for the Pareto front of the DTLZ4 test problem when 1000 runs are carrying out.

each algorithm. Generally, we conclude that NNIA+X3 retains better population diversity, and its convergence is faster than NNIA for these two and three objective test problems.

6. Experiments on the 10 bar truss design problem

In this section, we address the multi-objective sizing optimization of truss-like structures which is a continuous subject of researches in structural design [28–30].

6.1 Problem formulation

In this study, we consider the 10 bar truss test, ketch in the **Figure 3**. Two objective functions have to be minimized: the mass and the displacement; and one objective function has to be maximized: the first flexible natural frequency of the structure.

Denoting $\mathbf{x} \in \Omega$ the vector of the topological and sizing optimization parameters, such that $0 \leq x_i \leq 1$ for $i \in \{1, \dots, n\}$ where $n = 10$ is the number of elements, the three individual objectives are:

1. The mass w of the structure

$$w(\mathbf{x}) = \sum_{i=1}^{n} \rho A l_i x_i,$$

where l_i is the length of the i-th element, $\rho = 2768$ kg/m^3 is the density of the material and $A = 0.01419352$ m^2 is the element cross-section area.

2. The maximum displacement u of the structure

$$u(\mathbf{x}) = \max\left(\mathbf{u}^* = \arg\min_{\mathcal{S}}\left(\frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathbf{K}(\mathbf{x})\mathbf{u} - \mathbf{u}^{\mathrm{T}}\mathbf{F}\right)\right),$$

where:

- **F** is the vector of loads

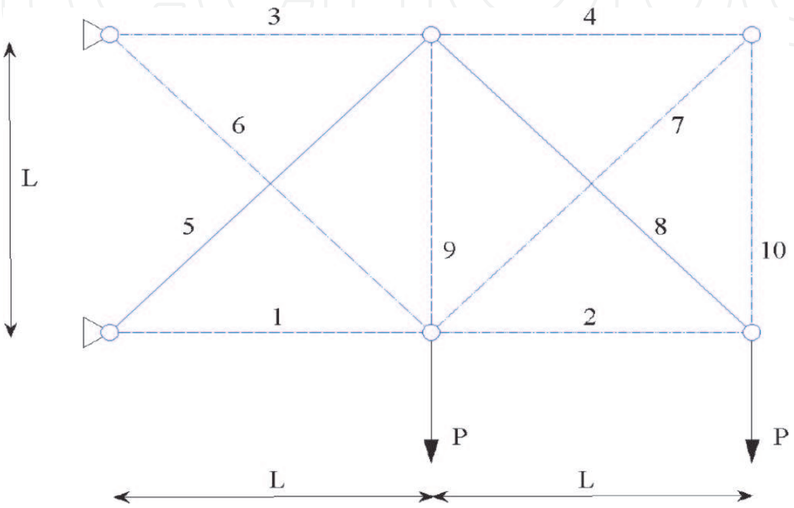

Figure 3.
Sketch of the 10 bar truss.

- **K** is the stiffness matrix of the finite element (FE) model, having the Young' modulus $E = 68.95$ GPa.

The set $\mathcal{S}$ refers to the kinematic admissible space, *i.e.* the one that satisfies the imposed boundary conditions given by the supports while carrying all the prescribed loads, where $P = 448.2$ kN.

3. The function (minimum flexible natural frequency f) to maximize it

$$-f(\mathbf{x}) = -\min\left(\frac{1}{2\pi}\omega^*\right),$$
$$\text{where}: \quad \{\omega^{*2}, \mathbf{u}^*\} = \arg\min_{\mathbf{u}\in\mathcal{S}}\left(\omega^2 = \frac{\mathbf{u}^T\mathbf{K}(\mathbf{x})\mathbf{u}}{\mathbf{u}^T\mathbf{M}(\mathbf{x})\mathbf{u}}\right), \quad \|\mathbf{u}\| \neq 0$$

where **M** is the mass matrix of the FE model[2].

Moreover, this MO problem is subjected to constraints for the mechanical stress σ_i for each element i:

$$|\boldsymbol{\sigma}_i(\mathbf{x})| \leq \overline{\sigma} \qquad i \in \{1, \dots, n\}$$

where $\overline{\sigma} = 172.4$ MPa is the yield strength.

As designs with local rigid body modes or kinematic modes are not of interest, constraints are added to the MO problem formulation:

$$\frac{|\boldsymbol{\sigma}_i(\mathbf{x})|}{\overline{\sigma}} > \varepsilon, \qquad i \in \{1, \dots, n\}\text{such that} x_i > 0$$

where $\varepsilon = 0.001$.

Since the optimal Pareto front is unknown for this problem, unnormalized metric indicators are to assess for the MO algorithm performance. Thus, in practice, we introduce an *a priori* scaling of the three objectives, by defining:

$$f_1(\mathbf{x}) = \frac{w(\mathbf{x})}{7,000}, \qquad f_2(\mathbf{x}) = \frac{u(\mathbf{x}) - 0.016}{20}, \qquad f_3(\mathbf{x}) = \frac{22,500 - (2\pi f(\mathbf{x}))^2}{5,000}$$

Moreover, in order to handle constraints of this MO problem, we use the penalty method. This technique consists of replacing the constrained optimization problems by an optimization problems without constraints, when introducing new objective functions to be optimized:

$$\phi_k(\mathbf{x}) = f_k(\mathbf{x}) + r\varphi(\mathbf{x}) \tag{13}$$

where the penalty function chosen here is:

$$\varphi(\mathbf{x}) = \sum_{i=1}^{n}\left(\max\left\{0, \frac{|\sigma_i(\mathbf{x})|}{\overline{\sigma}} - 1\right\}\right)^2 + \sum_{i=1}^{n}\left(\max\left\{0, \varepsilon - \frac{|\sigma_i(\mathbf{x})|}{\overline{\sigma}}\right\}\right)^2 \tag{14}$$

and where r is a positive penalty parameter. We have chosen here $r = 10^{10}$.

[2] To obtain the best numerical efficiency for the FE analysis, the FE disassembly strategy proposed in Ref. [31] is involved.

Finally, the MO problem definition for the 10 bar truss of this work is:

$$\min_{\mathbf{x}\in\Omega}\left(f_1(\mathbf{x}) + r\varphi(\mathbf{x}),\, f_2(\mathbf{x}) + r\varphi(\mathbf{x}),\, f_3(\mathbf{x}) + r\varphi(\mathbf{x})\right)$$

6.2 Numerical simulations for two objective functions

In this subsection, we will subdivide and transform the 10 bar MO problem from the previous section into three MO subproblems. Objective functions are considered two by two: (w, u), $(w,f)_,$ and (u,f) To solve each of these 10 bar MO problems, we use the NNIA algorithms and the NNIA+X3, keeping the parameters to those of the previous subsection 5.2.

After 250 and 750 iterations, we obtain the two **Figures 4** and **5** (respectively), which show Pareto fronts of a typical execution, if the two algorithms start from the same initial population. In these figures, we observe that the NNIA+X3 algorithm shows better diversity for each subproblem, and that NNIA+X3 gives better convergence for the subproblems (w,f) and (u,f). Since each iteration of one of these algorithms requires $n_c = 100$ evaluations of the mechanical problem, 25,000 function evaluations are performed when 250 iterations are performed, and 75,000 function evaluations are performed when 750 iterations are performed.

Figure 6 shows the evolution of two metric indicators along the number of iterations for one typical run. Metric indicators chosen here are spacing and hyper-volume of Pareto fronts. Spacing evolution is presented in log-log scale in the figure. Each evaluation of the hyper-volume is achieved by using the same anti-utopia point and utopia point for results consistency. Moreover, in order to compare the three MO results on the same graph, a relative hyper-volume is plotted: the graph corresponds to the hyper-volume obtained divided by its maximum value. These graphs show a better diversity and convergence for NNIA+X3 compared with NNIA when early number of iterations are considered.

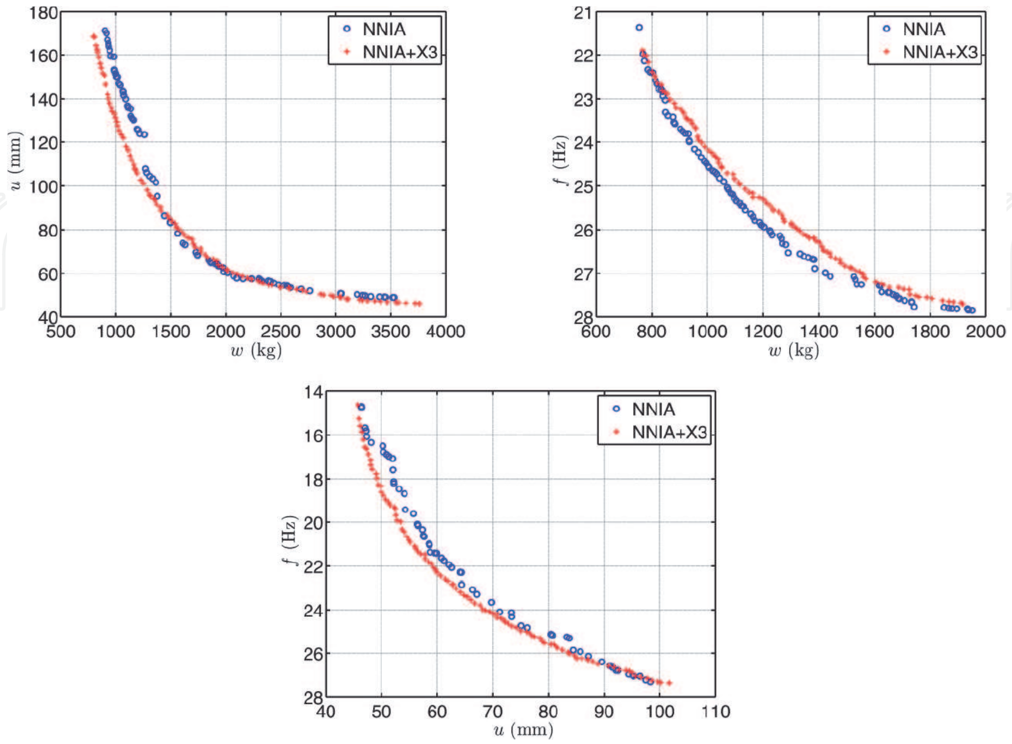

Figure 4.
Pareto fronts of the 10 bar truss MO problem two by two: (w, u) (up-left), (w,f) (up-right), (u,f) (down), after 250 iterations.

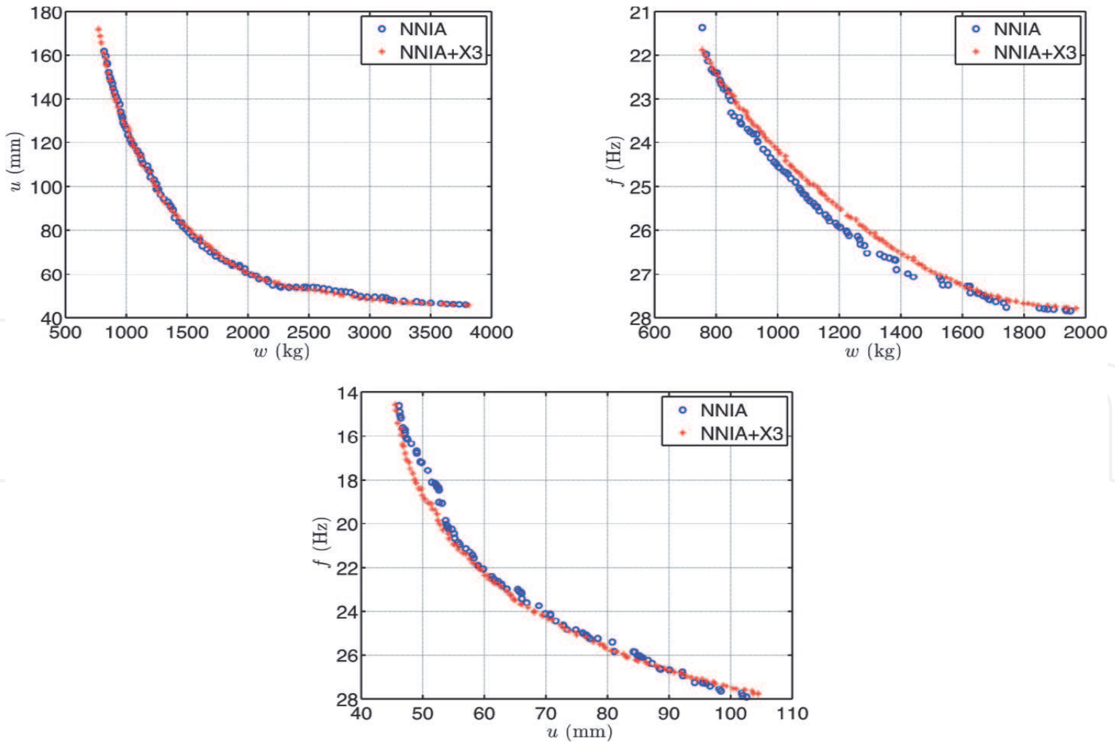

Figure 5.
Pareto fronts of the 10 bar truss MO problem, after 750 iterations for two by two: (w, u) *(up-left),* (w,f) *(up-right),* (u,f) *(down).*

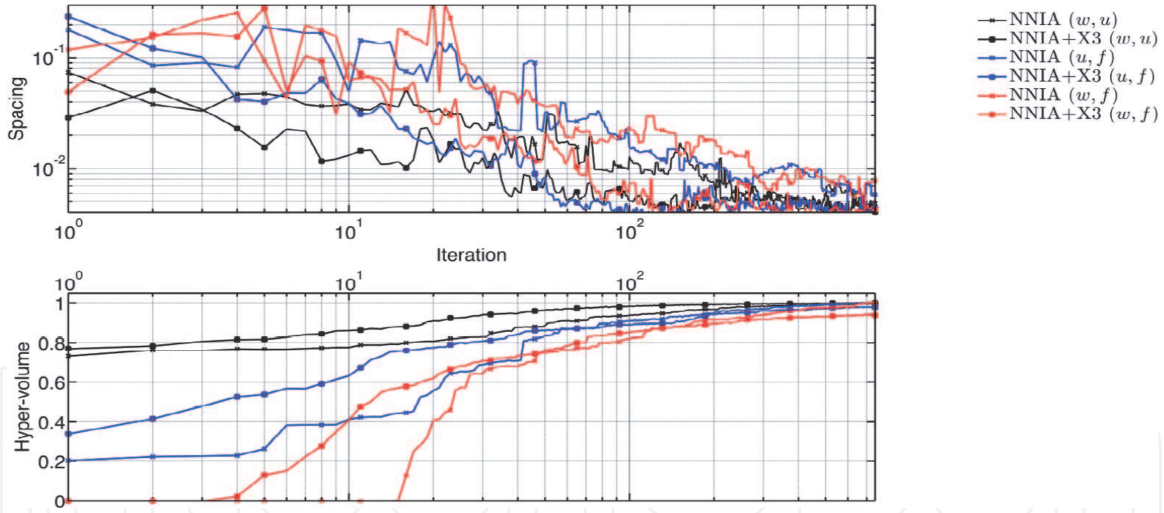

Figure 6.
Metrics indicators of the 10 bar truss MO problem for the three objectives functions two by two: spacing (up) and relative hyper-volume (down).

Tendency observed in the previous figure is confirmed by statistical results of **Figure 7** and **Figure 8**. These figures show box plots statistic for spacing and hyper-volume (respectively) when 300 runs stopped at 250 iterations are carried out. For spacing, means and variance are clearly better for NNIA+X3. Hyper-volume statistic results are also better for the NNIA+X3 when considering the (w, u) and (u,f) subproblems, while they are almost identicals for the (w,f) supbproblem, although the mean and variance are also better for the NNIA+X3. From results for the hyper-volume of the (w,f) supbproblem, it is assessed that this subproblem is the most difficult to solve since a wide spread is observed in data for both algorithms.

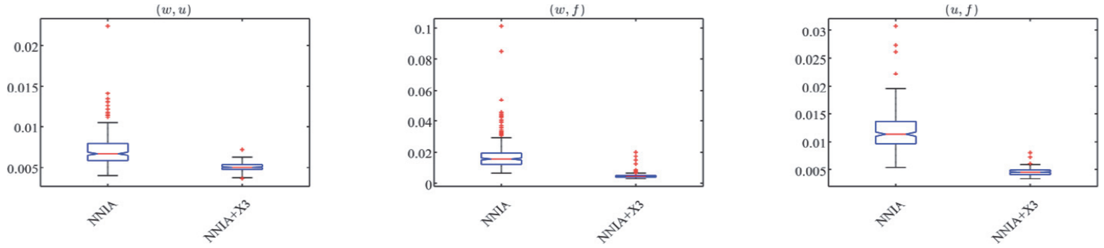

Figure 7.
Statistics box plots of spacing for 300 runs of the two-by-two MO 10 bar subproblems: (w,u) (left), (w,f) (middle), (u,f) (right).

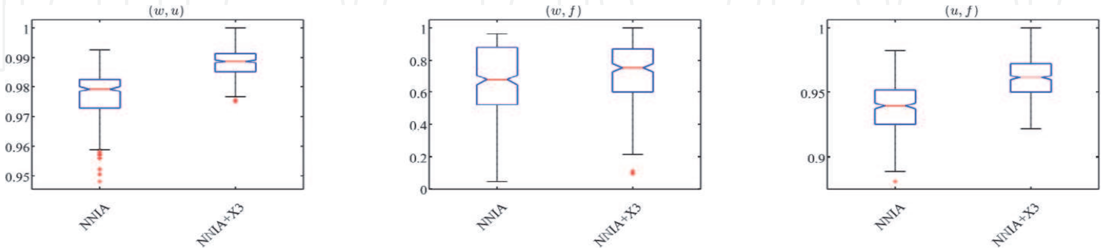

Figure 8.
Statistics box plots of relative hyper-volume for 300 runs of the two-by-two MO 10 bar subproblems: (w,u) (left), (w,f) (middle), (u,f) (right).

6.3 Numerical simulation for three objective functions

Figure 9 shows different views of the Pareto front obtained for one typical run when solving the three objectives 10 bar truss problem, using NNIA+X3 with the following parameter values: size of active population 30, clonal scale 150 and 750 iterations. In this case, the size of the dominant population is not limited to any number and all Pareto points found are kept. **Figure 10** shows the evolution of the

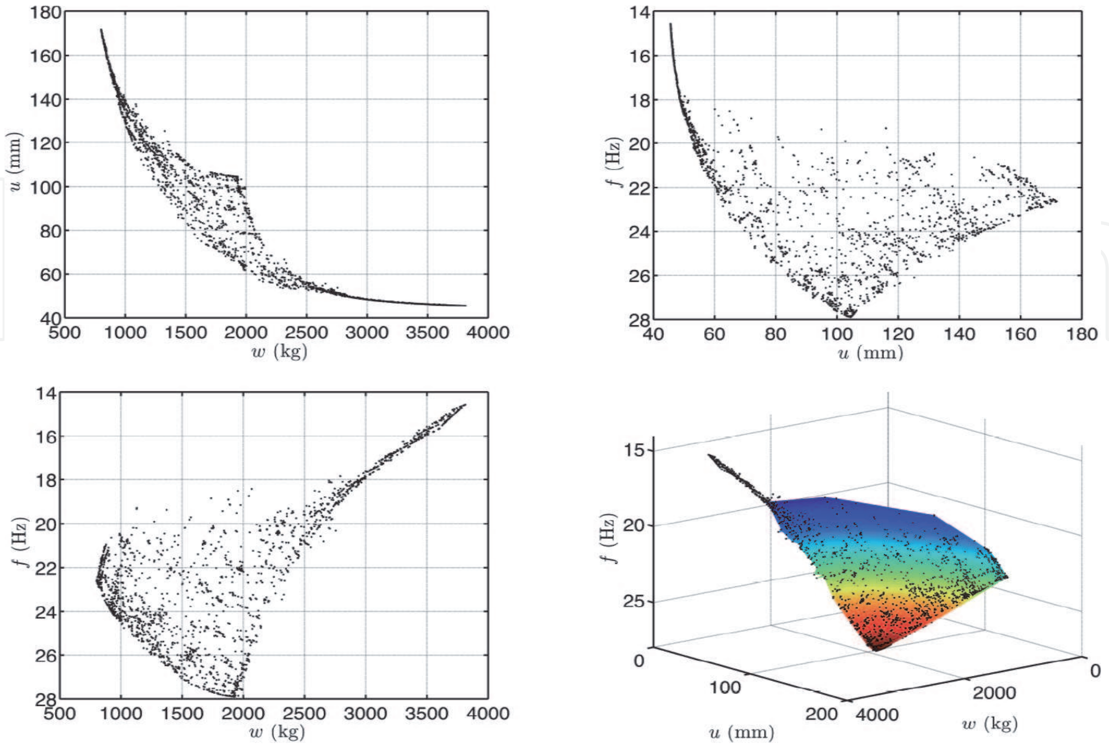

Figure 9.
Four different views of the Pareto front obtained for solving the three objective functions of the 10 bar truss problem; Colorized surface of the down-right subfigure is added for a better visualization, and the color corresponds to the frequency objective f.

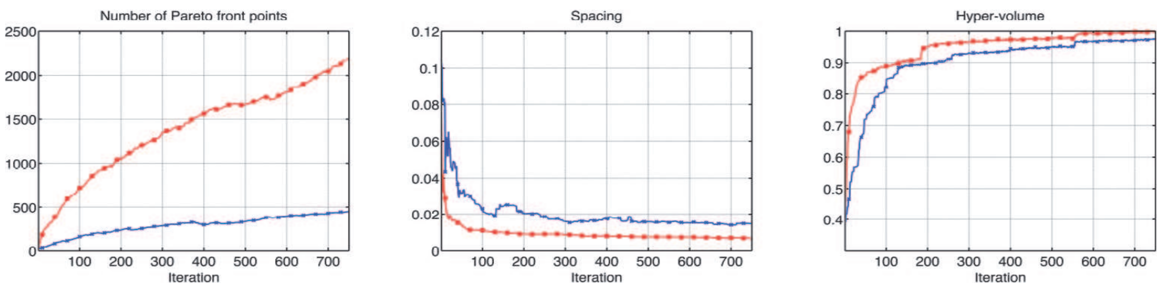

Figure 10.
Evolution of results for a typical run of the MO 10 bar problem: Number of points for the Pareto front (left), spacing (middle), and relative hyper-volume (right); Blue line with cross markers: NNIA; Red line with squared markers: NNIA+X3.

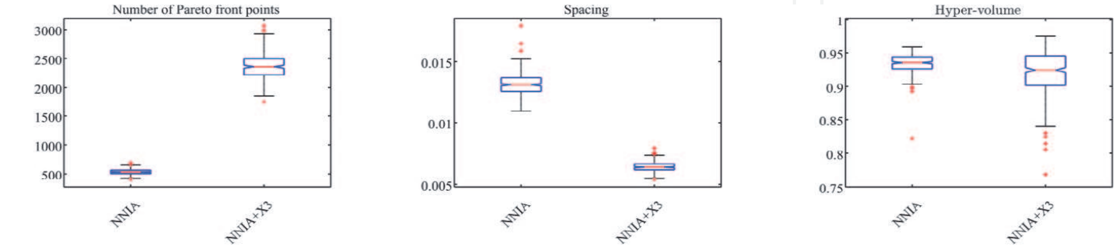

Figure 11.
Statistics box plots for 300 runs of the three objectives 10 bar problem with NNIA and NNIA+X3 with random initial population: Number of Pareto front points (left), spacing (middle), and relative hyper-volume (right).

number of points in the dominant population for the Pareto front given in **Figure 9**. It ends at 2216 Pareto points for this run.

Figure 11 shows box plot statistics when 300 runs are carried out with NNIA and NNIA+X3. It is observed that the number of points for the Pareto front is higher for NNIA+X3, with a better spacing. But the hyper-volume is better for NNIA. Detailled analysis of results has revealed that bad results for hyper-volume are due to a slow convergence to an extreme Pareto front point: the individual optima for the frequency objective. For this problem, the individual minima found for the frequency objective are most of the time better for NNIA than for NNIA+X3. However, it is also found that individual minima of the three objectives are rarely found in the Pareto front by both algorithms.

For better results, the idea is to handle the three individual minima found by a mono-objective optimization into the random initial population of both NNIA and NNIA+X3. This simple modification greatly improves performance results. **Figure 12** shows statistics box plots when 300 runs of MO problem are carried out when the three individual optima are given in the initial population. In such a situation and for each of the 300 runs done, NNIA+X3 appears to be superior to NNIA for all performance aspects, including the computed hyper-volume.

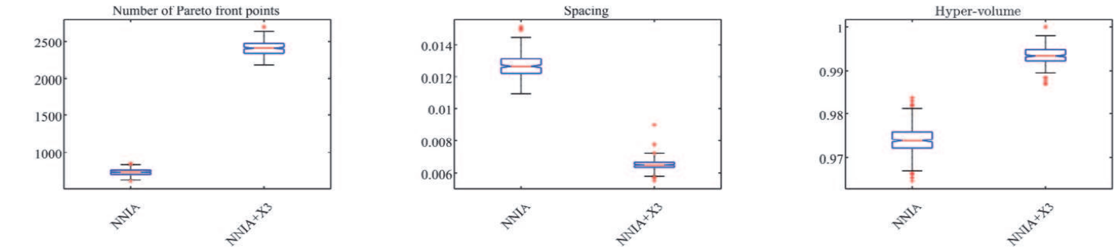

Figure 12.
Statistics box plots for 300 runs of the three objectives 10 bar problem with NNIA and NNIA+X3 when individual optima are handled in the initial population: Number of Pareto front points (left), spacing (middle), and relative hyper-volume (right).

7. Summary

This work is devoted to recombination for an NNIA algorithm. We propose three recombinations, inspired by the BSA algorithm crossing operator when adapting input populations.

In the first NNIA+X1 algorithm, the clonal population and an extended active population are concerned, when the extended active population is founded by duplicating individual antibodies.

In the second algorithm, NNIA+X2, recombination is achieved by using the clonal population and itself.

The NNIA+X3 algorithm uses the clonal population and an extended working population, which finds by duplicating individual antibodies and a proportion of random individuals. From this algorithm, a certain degree of mutation is carried out. The results obtained for the benchmark, ZDT, and DTLZ functions show that our proposed algorithm NNIA+X3 can accelerate the speed of convergence and maintain the desirable diversity, especially when solving problems with many local Pareto-optimal fronts. The experimental results of this algorithm to solve the problems of bi-objectives and three-objectives of optimization of 10 bar trellis structure indicate that the proposed NNIA+X3 surpasses the NNIA algorithm in terms of convergence rate and of course of quality of the solution.

Author details

Hamed Ould Sidi[1], Rachid Ellaia[2], Emmanuel Pagnacco[3]
and Ahmed Tchvagha Zeine[1]*

1 URMCD, University of Nouakchott Al Aasriya, Nouakchott, Mauritania

2 LERMA-EMI, Mohammadia School of Engineers, Rabat, Morocco

3 LMN, INSA de Rouen Normandie, St. Etienne du Rouvray, France

*Address all correspondence to: nouneahmed@gmail.com

References

[1] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation. 2002;**6**(2): 182-197

[2] Sierra MR, Coello CAC. Multi-objective particle swarm optimizers: A survey of the state-of-the-art. International Journal of Computational Intelligence Research. 2006;**2**:287-308

[3] Song MP, Gu GC. Research on particle swarm optimization: A review. In: Proceedings of the International Conference on Machine Learning and Cybernetics. Vol. 4. 2004. pp. 2236-2241

[4] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm. In: Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems. Athens, Greece; 2002. pp. 95-100

[5] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation. 1999;**3**(4): 257-271

[6] Omkar SN, Khandelwal R, Yathindra S, Naik GN, Gopalakrishn S. Artificial immune system for multi-objective design optimization of composite structures. Engineering Applications of Artificial Intelligence. 2008;**21**: 1416-1429

[7] Coello C, Cortes N. Solving multiobjective optimization problems using an artificial immune system. Genetic Programming and Evolvable Machines. 2005;**6**(2):163-190

[8] Gao J, Wang J. WBMOAIS: A novel artificial immune system for multiobjective optimization. Computers and Operations Research. 2010;**37**(1): 50-61

[9] Gong M, Jiao L, Du H, Bo L. Multiobjective immune algorithm with nondominated neighbor-based selection. Evolutionary Computation. 2008;**16**(2):225-255

[10] Jiao L, Liu F, Ma W. A novel immune clonal algorithm for MO problems. IEEE Transactions on Evolutionary Computation. 2012;**16**(1): 35-50

[11] Luh GC, Chueh CH, Liu WW. MOIA: Multi-objective immune algorithm. Engineering Optimization. 2003;**35**:143-164

[12] Shang R, Jiao L, Liu F, Ma M. A novel immune clonal algorithm for MO problems. IEEE Transactions on Evolutionary Computation. 2012;**16**(1): 35-50

[13] Shi J, Gong M, Ma W, Jiao L. A multiobjective immune algorithm based on a multiple-affinity model. European Journal of Operational Research. 2010; **202**(1):60-72

[14] Zinflou A, Gagn C, Gravelc M. GISMOO: A new hybrid genetic/ immune strategy for multiple-objective optimization. Computers and Operations Research. 2012;**9**(9): 1951-1968

[15] Qi YT, Hou ZT, Yin ML, Sun HL, Huang JB. An immune multi-objective optimization algorithm with differential evolution inspired recombination. Applied Soft Computing. 2015;**547**(29): 395-410

[16] Deb K, Beyer HG. Self-adaptive genetic algorithms with simulated binary crossover. Evolutionary Computation. 2001;**9**(2):197-221

[17] Civicioglu P. Backtracking search optimization algorithm for numerical optimization problems. Applied Mathematics and Computation. 2013; **219**(15):8121-8144

[18] Sheoran Y, Kumar V, Rana KPS, Mishra P, Kumar J, Nair SS. Development of backtracking search optimization algorithm toolkit in LabVIEW. Procedia Computer Science. 2015;**57**:241-248

[19] Civicioglu P. Circular antenna array design by using evolutionary search algorithms. Progress In Electromagnetics Research B. 2013;**54**: 265-284

[20] Chaib AE, Bouchekara HREH, Mehasni R, Abido MA. Optimal power flow with emission and non-smooth cost functions using backtracking search optimization algorithm. International Journal of Electrical Power and Energy Systems. 2016;**81**:64-77

[21] Fang S, Yunfang C, Weimin W. Multi-objective optimization immune algorithm using clustering. Computing and Intelligent Systems. 2011;**234**: 242-251

[22] Bosman PAN, Thierens D. The balance between proximity and diversity in multiobjective evolutionary algorithms. IEEE Transactions on Evolutionary Computation. 2003;**7**(2): 174-188

[23] Chen J, Lin Q, Ji Z. A hybrid immune multiobjective optimization algorithm. European Journal of Operational Research. 2010;**204**(2): 294-302

[24] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation. 2000;**8**(2):173-195

[25] Deb K, Thiele L, Laumanns M, Zitzler E. Scalable Multi-Objective Optimization Test Problems. Technical Report 112. Zurich, Switzerland: Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH); 2001. epagnacc, 2016.07.21

[26] Sierra MR, Coello CAC. Improving PSO-based multi-objective optimization using crowding, mutation and ε-dominance. In: Proceedings of the Evolutionary Multi-Criterion Optimization. Vol. 3239. 2005. pp. 263-276

[27] Schott JR. Fault tolerant design using single and multicriteria genetic algorithm optimization [masters thesis]. Dept. Aeronautics and Astronautics, Massachussets Institue of Technology; 1995

[28] Arkadiusz M. Geometrical aspects of optimum truss like structures for three-force problem. Structural and Multidisciplinary Optimization. 2012; **45**(1):21-32

[29] Richardson JN, Adriaenssens S, Bouillard P, Coelho RF. Multiobjective topology optimization of truss structures with kinematic stability repair. Structural and Multidisciplinary Optimization. 2012;**46**:513-532

[30] Kaveh A, Laknejadi K. A hybrid evolutionary graph-based multi-objective algorithm for layout optimization of truss structures. Acta Mechanica. 2013;**224**(2):343-364

[31] Hemez FM, Pagnacco E. Statics and inverse dynamics solvers based on strain-mode disassembly. Revue Européenne des Eléments Finis. 2000; **9**(5):511-560