10th World Congress on Structural and Multidisciplinary Optimization
May 19 - 24, 2013, Orlando, Florida, USA

A New Accelerated Multi-objective Particle Swarm Algorithm. Applications to Truss Topology Optimization

R. Ellaia[1], A. Habbal[2] and E. Pagnacco[3]

[1] Mohammed V University - Agdal, Mohammadia School of Engineering, LERMA,
BP. 765, Ibn Sina avenue, Agdal, Rabat, Morocco.
ellaia@emi.ac.ma

[2] J.A. Dieudonné Laboratory, Nice Sophia-Antipolis University and INRIA Sophia-Antipolis, France,
habbal@polytech.unice.fr

[3] LOFIMS, EA 3828, INSA Rouen BP 8, 76801 Saint-Etienne du Rouvray, France,
Emmanuel.Pagnacco@insa-rouen.fr

Abstract. We propose a new algorithm of computation using particle swarm in order to solve multi-objective problems more quickly and effectively. This approach, called accelerated multi-objective particle swarm, is partially based on our previous work [4] and incorporates a vector function as objective function and it uses matrix computation to develop the Pareto front. Unlike all these studies which use inertia weight to develop Pareto front and an external archive to save non-dominated solution, we will modify this algorithm for causing it to use matrix computation, then this algorithm incorporates function vector as objective function and uses Pareto dominance for selecting best solutions and updating Pareto set. In addition, we also propose a new strategy of initialization that contributes too to the acceleration of the algorithm. The resulting algorithm is applied to multi-objective topology optimization of truss structures.
The results produced by such a strategy illustrate that the algorithm is competitive with NSGA-II and MISA in terms of converging to the true Pareto front. It maintains the diversity of the population, generates better trade-offs and demonstrates that the matrix computation PSO can be used as a reliable numerical optimization tool.

Keywords. Topological optimization, Multi-objective optimization, Particle swarm optimization, Non-dominated solutions, Pareto-optimal front, Matrix computation, Initialization technique, Truss structures.

1 Introduction

In this paper, a novel multi-objective particle swarm optimization (MOPSO) technique is proposed and implemented. The proposed approach extends the single objective PSO by proposing a matrix computation to develop the Pareto-optimal front.
There have been several proposals to extend multi-objective PSO. The most important of these are the Dynamic Neighborhood PSO proposed by Hu and Eberhart [16]; in this algorithm, only one objective is optimized at a time using a scheme similar to lexicographic ordering. Lexicographic ordering tends to be useful only when few objective functions are used (two or three), and it may be sensitive to the ordering of the objectives. Praveen et al. [25] proposes multi-objective PSO with time variant inertia and acceleration coefficients where inertia weight and PSO algorithm parameters expressions depend to iteration number. Other study developed in [32] proposes multi-objective PSO with dynamic population size. Peng and Zhang [24] proposed the multi-objective particle swarm optimizer based on decomposition MO-PSO/D, this approach uses the framework adopted by MOEA/D, but replaces the genetic operators (crossover and mutation) by the inertia flight equations used in traditional PSO. Moubayed et al. [21] proposed a smart multi-objective particle swarm optimizer using decomposition, this algorithm is also based on MOEA/D, and adopts an external archive based on $\epsilon-$dominance. Fieldsend and Singh [13] incorporates an unconstrained elite archive (in which a special data structure called *dominated tree* is adopted)

to store the non-dominated individuals found along the search process. The archive interacts with the primary population in order to define local guides. This approach also uses a turbulence operator. Coello Coello, Pulido and Lechuga [8] uses a global repository in which every particle deposits its flight experiences. Additionally, the updates to the repository are performed considering a geographically based system defined in terms of the objective function values of each individual; this repository is used by the particles to identify a leader that will guide the search. It also uses a mutation operator that acts both on the particles of the swarm, and on the range of each design variable of the problem to be solved. Mostaghim and Teich [20] proposed a sigma method in which the best local guides for each particle are adopted to improve the convergence and diversity of a PSO approach used for multi-objective optimization. They also use a turbulence operator, but applied on decision variable space. The use of the sigma values increases the selection pressure of PSO (which was already high). This may cause premature convergence in some cases.
The remainder of the paper is organized as follows. In Section 1, briefly reviews the general formulation of PSO. In Section 2, the proposed multi-objective optimization algorithm is presented. Experimental results are discussed in Section 3. Finally, Section 4 presents some applications in structural optimization and gives some conclusion.

2 The Particle Swarm Algorithm

In Particle Swarm Algorithm, each particle i is treated as a point in a space with dimension d, a position $\boldsymbol{X_i}$, a velocity $\boldsymbol{V_i}$ and personal best position $\boldsymbol{X_{besti}}$. The personal best position associated with a particle i is the best position that the particle has visited. The best position of all particles in the swarm is represented by the vector $\boldsymbol{X_{gbest}}$. After finding the best values, the particle updates its velocity and positions with the following equations

$$\boldsymbol{V_i(t+1)} = \omega \boldsymbol{V_i(t)} + \rho_1 \cdot rand(1)[\boldsymbol{X_{besti}}(t) - \boldsymbol{X_i}(t)] + \rho_2 \cdot rand(2)[\boldsymbol{X_{gbest}}(t) - \boldsymbol{X_i}(t)] \tag{1}$$

$$\boldsymbol{X_i}(t+1) = \boldsymbol{X_i}(t) + \boldsymbol{V_i}(t+1) \tag{2}$$

ρ_1 and ρ_2 are the balance factors between the effect of self-knowledge and social knowledge in moving the particle towards the target. Usually the value 2 is suggested for both factors in the literature. $rand(1)$ and $rand(2)$ are independent random number in the rang $[0, 1]$.
We modified the velocity function by using a new term $\boldsymbol{X_{Nbest}}$ in the equation (1), which was introduced by Bochenek and Fory [5] defined as:
$\boldsymbol{X_{Nbest}} = (p_{n1}, p_{n2}, ..., p_{nd})$ The best position of the neighborhood. The equation (1) becomes :

$$\begin{aligned}\boldsymbol{V_i}(t+1) = \omega \boldsymbol{V_i}(t) + \rho_1 \cdot rand(1)\big[\boldsymbol{X_{besti}}(t) - \boldsymbol{X_i}(t)\big] \\ +\rho_2 \cdot rand(2)\big[\boldsymbol{X_{gbest}}(t) - \boldsymbol{X_i}(t)\big] + \rho_3 \cdot rand(3)\big[\boldsymbol{X_{Nbest}}(t) - \boldsymbol{X_i}(t)\big]\end{aligned} \tag{3}$$

where the third term of equation (3) represents the distance between the particle position and a position of the particle neighbors leader, i.e. the best particle among its neighbors. This provides a complementary information of swarm member behavior and therefore influences and improves the swarm performance.
ρ_3 is the positive acceleration components called social parameter and $rand(3)$ is the independent random number in the rang $[0, 1]$.

3 Accelerated Multi-objective Particle Swarm Optimization

Multi-objective optimization involves the simultaneous optimization of several incommensurable and often competing objectives. In the absence of any preference information, a non dominated set of solutions is obtained, instead of a single optimal solution. These optimal solutions are termed as Pareto optimal solutions. Simply put, Pareto optimal sets are the solutions that cannot be improved in one objective function without deteriorating their performance in at least one of the rest [9]. In general, a multi-objective problem consists of a vector-valued objective function to be minimized, and of some equality or inequality constraints, i.e.,

$$\begin{cases} \min \quad F(\boldsymbol{x}) = (f_1(\boldsymbol{x}), ..., f_{Nobj}(\boldsymbol{x}))^T \\ \text{subject to} \qquad g_i(\boldsymbol{x}) \leq 0 \ , \ i = 1, ..., m, \end{cases} \tag{4}$$

where $\boldsymbol{x} \in I\!R^d$ is the vector of decision variables, $f_1, ..., f_{Nobj}$ are objective functions, $g_1, .., g_m$ are possible sets of inequality constraints, which represent process model. The set S of constraints defines the feasible space, while the set of all possible values of the objective function constitutes the objective space.
Many algorithms have been suggested for generating the Pareto optimal set, for examples, Weighted sum method, The concept of Pareto optimality can be defined as follows: the decision vector $\boldsymbol{x}$ is said to dominate the decision vector $\boldsymbol{y}$, and we denote $F(\boldsymbol{y}) \preceq F(\boldsymbol{x})$, if and only if :

$$\forall\ i = 1, 2, \cdots, Nobj:\ f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{y}) \ \text{ and } \ \exists\ j = 1, 2, \cdots, Nobj:\ f_j(\boldsymbol{x}) < f_j(\boldsymbol{y}). \tag{5}$$

The decision vector $\boldsymbol{x}^* \in S$ is said to be Pareto-optimal if and only if there is no $\boldsymbol{x} \in S$ for which dominates $\boldsymbol{x}^*$. The set of all Pareto-optimal decision vectors is called the Pareto-optimal, efficient, or admissible set of the problem. The corresponding set of objective vectors is called the non-dominated set.

3.1 Opposition-Based Population Initialization

Good initialization of the population can increase the convergence speed and sometimes improve the final results. But if no information about the solution is available, then random initialization is the most commonly used method to generate initial population.
The concept of opposition-based learning was introduced by Hamid R. Tizhoosh [29], and its applications has proven to be an effective Method for some evolutionary algorithm in some optimization problems. The opposite solution $\breve{x}$ in $[a, b]$ can be calculated as follows [29]:
Let x be a real number in an interval $[a, b]$ ($x \in [a, b]$); the opposite number $\breve{x}$ is defined by $\breve{x} = a + b - x$. For $a = -b$ we receive $\breve{x} = -x$, and for $a = 0$ and $b = 1$ we receive $\breve{x} = 1 - x$. Similarity, this definition can be extended to higher dimensions as follows :
Let $P(x_1, x_2, \cdots, x_d)$ be a point in d-dimensional space, where $x_i \in [a_i, b_i] \forall i \in \{1, 2, \cdots, d\}$. The opposite point of P is defined by $OP(\breve{x}_1, \breve{x}_2, ..., \breve{x}_d)$ where : $\breve{x}_i = a_i + b_i - x_i$
Now, let $P(x_1, x_2, \cdots, x_d)$ a point in an n-dimensional space with $x_i \in [a_i, b_i]\ \forall i \in 1, 2, \cdots, d$, be a candidate solution. Assume $F(x)$ is a fitness multi-objective function which is used to measure non-dominated optimality. According to opposite point definition, at iteration k, the point $OP_k(\breve{x}_1, \breve{x}_2, ..., \breve{x}_d)$ is the opposite of $P_k(x_1, \cdots, , x_d)$. The opposition-based optimization is defined as follow, if OP_k dominates P_k, then the point P_k can be replaced by OP_k; if P_k dominates OP_k, we continue with P_k, otherwise we continue with P_k. Hence, the point and its opposite point are evaluated simultaneously to continue with the fitter one. This lead to an acceleration during the execution of this algorithm and may change the either the concentration of the values of the positions in one region, or the uniformity of the Pareto graph.

3.2 The proposed approach

The Accelerated Multi-objective Particle Swarm Optimization (AMOPSO) approach updates all the best solutions at each iteration. AMOPSO approach is based on the improvement of the population of the best positions, denoted $\boldsymbol{X_{mobest}}$, which converges toward the Pareto optimal set. The population in developing $\boldsymbol{X_{part}}$ benefit from the experiences of the entire developed population. We use the matrix representation of these two populations $\boldsymbol{X_{mobest}}$ and $\boldsymbol{X_{part}}$. Let $N = PopSize$ be the population size and K be the population size of nondominated points in $\boldsymbol{X_{part}}$, then

$$\boldsymbol{X_{mobest}} = (\boldsymbol{X_{mobest_1}}, \boldsymbol{X_{mobest_2}}, \cdots, \boldsymbol{X_{mobest_K}}) \ \text{ and } \ \boldsymbol{X_{part}} = (\boldsymbol{X_{part_1}}, \boldsymbol{X_{part_2}}, \cdots, \boldsymbol{X_{part_N}}) \tag{6}$$

The evaluation of these two populations will use a matrix based computation. We apply the vector function $F = \left(f_1, f_2, \cdots, f_{Nobj}\right)^T$ on the matrix of population, then:

$$\begin{cases} \boldsymbol{F_{mobest}} & = & F(\boldsymbol{X_{mobest}}) = \left(F(\boldsymbol{X_{mobest_1}}), F(\boldsymbol{X_{mobest_2}}), \cdots, F(\boldsymbol{X_{mobest_K}})\right) \\ F(\boldsymbol{X_{mobest_i}}) & = & \left(f_1(\boldsymbol{X_{mobest_i}}), f_2(\boldsymbol{X_{mobest_i}}), \cdots, f_{Nobj}(\boldsymbol{X_{mobest_i}})\right)^T \end{cases} \tag{7}$$

In this approach, the developed population doesn't follow a single best position but it moves toward each non dominated position. The main objective of AMOPSO is then to develop the Pareto front of multi-objective problems

using matrix computation of these functions. Let $N = PopSize$ be the size of the population studied, $\boldsymbol{X_{mobest}}$, $\boldsymbol{X_{pmobest_i}}$ and $\boldsymbol{X_{part}}$ are three vectors where $\boldsymbol{X_{mobest}}$ represent the best position that the entire population has seen, $\boldsymbol{X_{pmobest_i}}$ the best position of the particle i has seen, and $\boldsymbol{X_{part}}$ is the best position that a particle belonging to this population has seen. In the accelerated particle swarm optimization, the population velocity, at iteration t, is generated by the following formula:

$$\boldsymbol{V_i(t)} = \omega \cdot \boldsymbol{V_i(t-1)} + \rho_1 \cdot rand(1)\big(\boldsymbol{X_{pmobest_i}} - \boldsymbol{X_{part_i}}\big)(t-1) + \frac{\rho_2 \cdot rand(2)}{K} \sum_{k=1}^{K} \big(\boldsymbol{X_{mobest_k}} - \boldsymbol{X_{part_i}}\big)(t-1) \quad (8)$$

where ω is the inertia weight, introduced by Eberhart Shi, (see [12]). The optimal strategy is to initially set ω to 0.9 and reduce it linearly to 0.5, allowing initial exploration followed by acceleration toward an improved global Pareto optimum, ρ_1, ρ_2 are the the acceleration coefficients.
The update of the position of the developing population is simply formatted as follow:

$$\boldsymbol{X_{part_i}}(t) = \boldsymbol{X_{part_i}}(t-1) + \boldsymbol{V_i}(t) \quad (9)$$

3.3 The pseudo-code

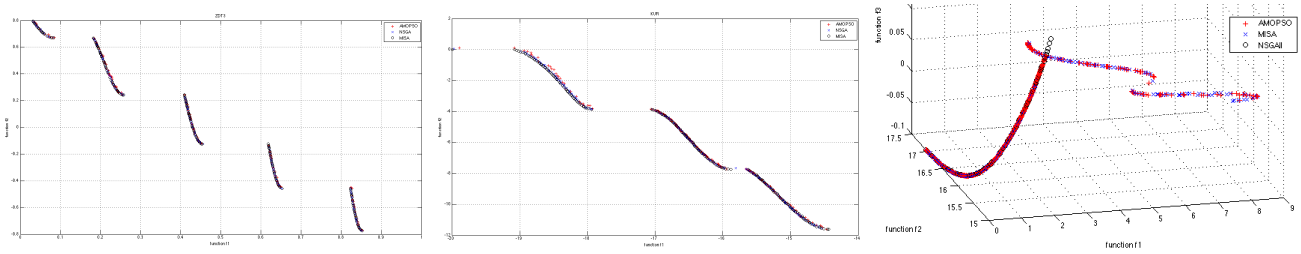
We start the algorithm by initializing the swarm which includes both the positions and velocities of the two populations $\boldsymbol{X_{mobest}}$ and $\boldsymbol{X_{part}}$ using the initialization techniques that we explained above.
We evaluate the objective functions that optimize the problem we are dealing with. We try to compare between improvement of the population of the best positions, denoted $\boldsymbol{X_{mobest}}$, which will converge toward the Pareto optimal set and the population in developing $\boldsymbol{X_{part}}$ that will benefit from the experiences of the entire developed population, we update both populations. The algorithm stops if the criteria is satisfied if not then we update the population $\boldsymbol{X_{part}}$ and we restart.
We keep following these steps until we get efficient positions or solutions to the multi-objective problem suggested. The following pseudo-code illustrate the mechanism of the algorithm :

Algorithm 1 Accelerated Multi-Objective Particle Swarm Optimization

for each particle $i = 1, \cdots, N$: initialize **do**
  The particle's position with an opposition-based optimization method;
  The population's best known position to its initial position: $\boldsymbol{X_{mobest}} \leftarrow \boldsymbol{X_{part}}$;
  $\boldsymbol{F_{part}}$, $\boldsymbol{F_{mobest}}$ and the particle's velocity $\boldsymbol{V_i}$;
  $Pareto \leftarrow 1$;
end for
repeat
  for each particle $i = 1, \cdots, N$ **do**
    Update the particle's velocity using equation (8)
    Update the particle's position using equation (9)
    if $\boldsymbol{F_{part}}(:,t) \preceq \boldsymbol{F_{mobest}}(:,r)$ **then**
      $\boldsymbol{F_{mobest}}(:,r) \leftarrow \boldsymbol{F_{part}}(:,t)$, $\boldsymbol{X_{mobest}}(:,r) \leftarrow \boldsymbol{X_{part}(:,t)}$;
      $flag \leftarrow 1$;
    end if
    if $\boldsymbol{F_{part}}(:,t) \succ \boldsymbol{F_{mobest}}(:,r)$ **then**
      $flag \leftarrow 2$;
    end if
  end for
  if $flag == 0$ **then**
    $\boldsymbol{X_{mobest}} \leftarrow [\boldsymbol{X_{mobest}}, \boldsymbol{X_{part}}(:,t)]$, $\boldsymbol{F_{mobest}} \leftarrow [\boldsymbol{F_{mobest}}, \boldsymbol{F_{part}}(:,t)]$;
    $Pareto \leftarrow Pareto + 1$;
  end if
until a termination criterion is met;
Return $\boldsymbol{X_{mobest}}$ and plot the function $\boldsymbol{F_{mobest}}$;

3.4 Matlab results

We use only three academic test functions reported in the standard evolutionary multi-objective optimization literature [10, 30, 31] to valuate the performance of AMOPSO. The benchmark functions include two and three-objective functions without any inequality and equality constraints. Definitions of test functions can be found in Zitzler et al. [30] and Deb et al. [11] respectively. To better understand the importance of AMOPSO it's necessary to compare this algorithm with others, we take as an example NSGA II and MISA using the functions ZDT3, KUR and Viennet 3 problem in the three dimensional cases.



Test function ZDT3 Test function KUR Test function Viennet 3

Figure 1: The Pareto front for the tests functions ZDT3, KUR and Viennet 3

4 Multi-objective of truss structures sizing and topology optimization

In this section, we address the multi-objective sizing and topology optimization of truss-like structures which is a continuous subject of researches in mechanical design ([19, 23, 27, 3, 2]...). Let the design domain comprises a set of nodes with fixed spatial coordinates, a set of supports and a set of loads, as we can observed for the 10-bar and the 14-bar trusses examples displayed in the figure 2. It is assumed that the structures will be modelised by linear, two nodes, bar elements in linear elasticity, subjected only to axial forces and free from imperfections. The geometric and material parameters used are $L = 9.144$ m, $A = 0.01419352$ m^2 , $P = 448.2$ kN, $E = 68.95$ GPa, $\rho = 2,768$ kg/m^3 and $\overline{\sigma} = 172.4$ MPa.
Denoting $\boldsymbol{x} \in \mathbb{R}^n$ the vector of the topological and sizing optimization parameters, such that $0 \leq x_i \leq 1$ for $i = \{1, \ldots, n\}$ where n is the number of elements, three individual objectives are of interested:

1. the mass w of the structure:

$$f_1(\boldsymbol{x}) = w = \sum_{i=1}^{n} \rho A l_i x_i,$$

   ρ being the density of material, l_i being the length of the i-th element, A being the maximum for the element cross-section area *;

2. the maximum displacement u of the structure:

$$f_2(\boldsymbol{x}) = u = \max\left(\boldsymbol{u}^* = \arg\min_{\mathcal{S}}\left(\frac{1}{2}\boldsymbol{u}^{\mathrm{T}}\boldsymbol{K}(\boldsymbol{x})\,\boldsymbol{u} - \boldsymbol{u}^{\mathrm{T}}\boldsymbol{F}\right)\right),$$

   where $\boldsymbol{K}$ is the stiffness matrix and $\boldsymbol{F}$ the vector of loads of the finite element (FE) model

3. the opposite of the minimum natural frequency f of the structure (in order to maximize it):

$$f_3(\boldsymbol{x}) = -f = -\min\left(\frac{1}{2\pi}\boldsymbol{\omega}^*\right), \qquad \text{where}: \ \{\boldsymbol{\omega}^{*2}, \boldsymbol{u}^*\} = \arg\min_{\boldsymbol{u}\in\mathcal{S}}\left(\boldsymbol{\omega}^2 = \frac{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{K}(\boldsymbol{x})\,\boldsymbol{u}}{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{M}(\boldsymbol{x})\,\boldsymbol{u}}\right), \quad \|\boldsymbol{u}\| \neq 0$$

   where $\boldsymbol{M}$ is the mass matrix of the FE model † (see ref. [14]);

*This constraint is implicitly satisfy thanks to the adopted problem formulation.
†To obtain the best numerical efficiency for the FE analysis, the FE disassembly strategy proposed in ref. [14] is involved.

Notice that the sizing and the topology of the structure are optimized concurrently thanks to the introduced formulation and the domain of definition for $\boldsymbol{x}$. In addition, all these objectives are subjected to mechanical stress constraints σ_i foreach element i:

$$|\sigma_i| \leq \overline{\sigma} \qquad i = \{1, \ldots, n\}$$

where $\overline{\sigma}$ is the yield strength. Moreover, the set $\mathcal{S}$ refers to the kinematic admissible space. It is the one that satisfies the imposed boundary conditions given by the supports while carrying all the prescribed loads. Hence, kinematic instabilities that would be activated by loads are forbidden as they can not carried the loads (these situations are such that $\|\boldsymbol{K}(\boldsymbol{x})\ \boldsymbol{u} - \boldsymbol{F}\| \neq 0$). This constitutes an additional constraint. In practice, we chose to solve mechanical problems by using temporaries boundary constraints (by using a pseudo-inverse in the numerical method), and a check is done *a posteriori* to verify this constraint.
Considering the three objectives two by two, the Pareto fronts are sought using three optimization methods: NSGA-II, AMOPSO and MISA. NSGA-II algorithm parameters are 0.8 for the value of the crossover fraction, a value of 0.2 for the migration fraction and a value of 20 for the migration interval for a forward direction of migration, and a value of 0.35 for the Pareto fraction with an elite count of 2. MISA parameters are a clone scale of 100 and a dominant population size of 100. Constraints are treated in the same way for all the methods by using a penalty factor of 10^{10} within a simple exterior penalty strategy. All the optimization procedures are stopped when the limit value of the number of iterations is reached.
Population size and performance of the algorithms are shown in Table 1. In this table, the number of function evaluations refers to the number of calls to the FE model. Results of the Pareto fronts obtained by the three methods for a typical run are shown in the Figure 3 for the 10-bar truss and in the figure 5 for the 14-bar truss. In both cases, we can observed that:

- the NSGA-II method failed to produce the true Pareto front for the minimization of $F = (f_1,\ f_3)^{\mathrm{T}}$ when the limit for the number of generations is fixed to 3.10^4;
- the AMOPSO method gives not a wide spread of points on the Pareto front nor for the minimization of $F = (f_1,\ f_2)^{\mathrm{T}}$ nor for the minimization of $F = (f_2,\ f_3)^{\mathrm{T}}$;
- the MISA method has some difficulties to produce the true Pareto front for the minimization of $F = (f_2,\ f_3)^{\mathrm{T}}$ in the region near to the minimum of f_2.

However, these are satisfactory results for a designer: they help to decide for a specific solution. For example, if we request for the minimum weight to satisfy *a posteriori* a maximum displacement of 5.07 cm, we can learned from the weight-displacement front that 2824 kg is the optimal value for the 10-bar truss, while it is 2297 kg for the 14-bar truss. As we can see from the shapes of the Pareto fronts, all these objectives are contradictory. Starting from the minimum weight topology design that involves the minimum amount of material, a higher stiffness is obtained by increasing the effective area of the element members in order to minimize the maximum displacement. However, the natural frequency objective is slightly more difficult to analyze since it is both a function of stiffness and a function of mass, *i.e.* increasing the effective area of an element increased both its stiffness and mass. We can see however a meaningful result again: to increase the structural frequency, the effective area of elements that are nearer from the embeded boundary conditions is increased, while it is decreased far from this region, leading to intermediate designs from the minimum weight one and the minimum displacements one.
The results of the optimum parameters of several runs for the best solutions obtained on the Pareto front for the individual minima are shown in Table 2 and in Table 3 (with different initial population). These are meaningful results for the minimal weight since all non zero sized elements are fully stressed. Note that two equivalent solutions are proposed for the 14 bar truss when considering only the mass objective. This is a comprehensive result since the element 13 is in parallel with the elements 1 and 2, leading to a non uniqueness in sizing them. Note also that a 4-th column is added in Table 3 that involves only four bars. This result is frequently proposed in the literature, but we can see that it is a suboptimal topology, although it is also fully stressed. Figures 4 and 6 show all these topologies as well as mechanical stresses on the element members. As we can see, optimal topologies are significantly affected when traveling from one objective to another. Hence, if the minimum weight design is requested, it does not lead to the same topology than the minimum displacement design or the maximum first natural frequency design.

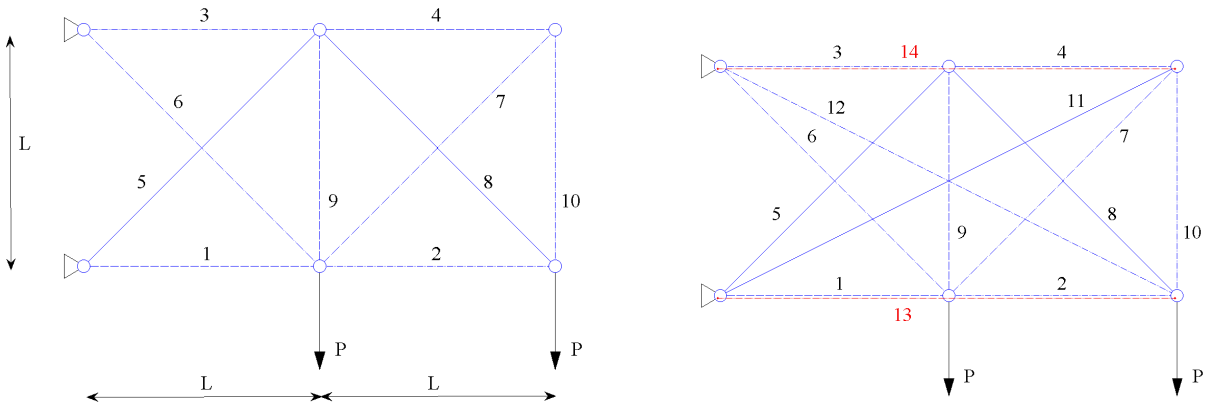


Figure 2: Ten-bar (left) and 14-bar (right) trusses ground structures

Table 1: Algorithms main parameters and performance for the ten-bar and the 14-bar trusses

10 bars truss	population size	# of generation	# of function called	# of nondominated
NSGA-II	100	3.10^4	$\sim 3\,10^6$	35
MOPSO	5120	3.10^2	$\sim 1.5\,10^6$	140-2.10^5
MISA	20	10^3	$\sim 10^5$	160-235
14 bars truss	**population size**	**# of generation**	**# of function called**	**# of nondominated**
NSGA-II	100	3.10^4	$\sim 3\,10^6$	35
MOPSO	16384	10^2	$\sim 1.5\,10^6$	159-1267
MISA	20	10^3	$\sim 10^5$	100-200

Table 2: 10-bar truss results

10 bars truss	$\min(f_1 = w)$	$\min(f_2 = u)$	$\min(f_3 = -f)$
Weight w [t]	0.724	3.82	1.97
Displacement u [mm]	183	45.5	105
Frequency f [Hz]	21.1	14.5	28.2
x_1	0.366	1	1
x_2	0.183	1	0.183
x_3	0.366	1	1
x_4	0	1	0
x_5	0.259	1	1
x_6	0.259	1	1
x_7	0	1	0
x_8	0.259	1	0.259
x_9	0	0	0.12
x_{10}	0	1	0

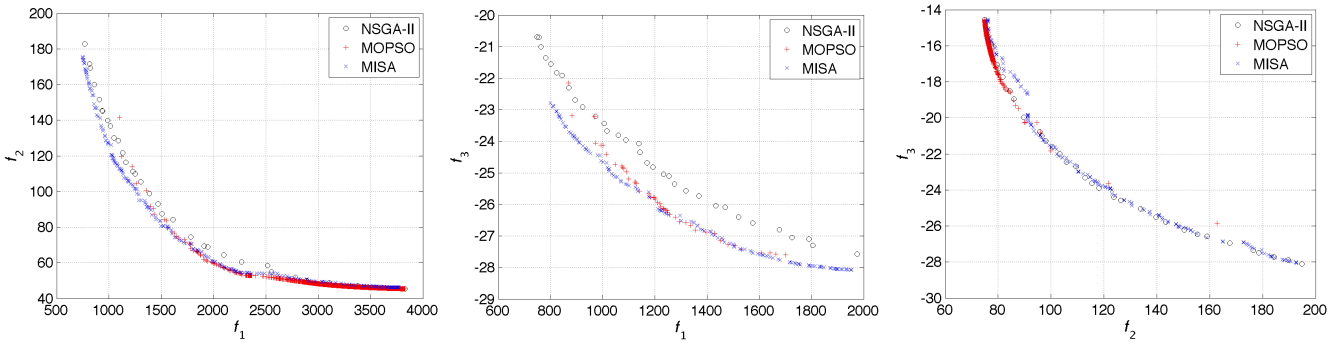


Figure 3: Pareto fronts of the ten-bar truss for the 3 objectives functions considering them two by two

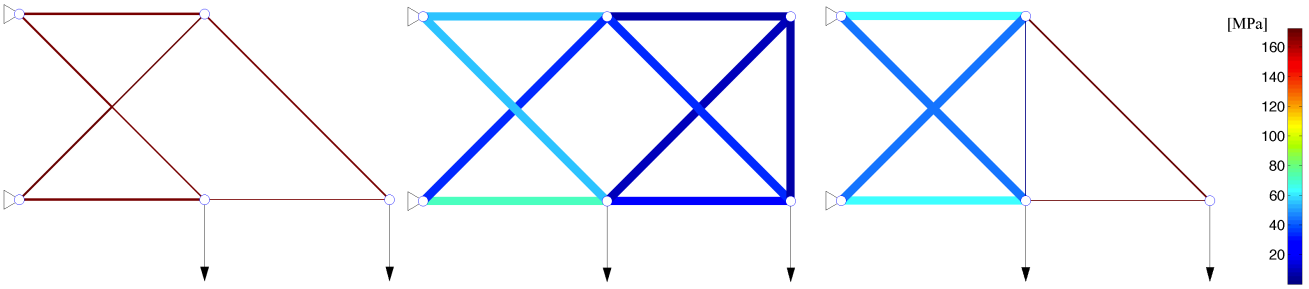


Figure 4: Topologies of the ten-bar truss for the 3 individual objectives functions (stresses are colorized on the bar element; $\min(f_1 = w)$ left, $\min(f_2 = u)$ middle, $\min(f_3 = -f)$ right)

Table 3: 14 bars truss results

14 bars truss	$\min(f_1 = w)$	$\min(f_1 = w)$	-	$\min(f_2 = u)$	$\min(f_3 = -f)$
Weight w [t]	0.724	0.724	0.789	6.36	1.975
Displacement u [mm]	183	183	206	24.4	105
Frequency f [Hz]	18.8	21.1	15.6	13.6	28.2
x_1	0.183	0.366	0.549	1	1
x_2	0	0.183	0.366	1	0.183
x_3	0.366	0.366	0	1	1
x_4	0	0	0	1	0
x_5	0.259	0.259	0	1	1
x_6	0.259	0.259	0.259	1	1
x_7	0	0	0	0	0
x_8	0.259	0.259	0	1	0.259
x_9	0	0	0	0	0.12
x_{10}	0	0	0	1	0
x_{11}	0	0	0	1	0
x_{12}	0	0	0.409	1	0
x_{13}	0.183	0	0	1	0
x_{14}	0	0	0	1	0

Conclusion The working principle of AMOPSO algorithm is presented in this paper. It is applied to multi-objective topology optimization of truss structures. While no adaptations of the algorithm are developed specifically for this particular application, it is observed that the general method AMOPSO provides adequate results and convergence of its strategies are comparable to other algorithms, like NSGA II and MISA. However, we found cases where diversity of solutions may be relatively poor.

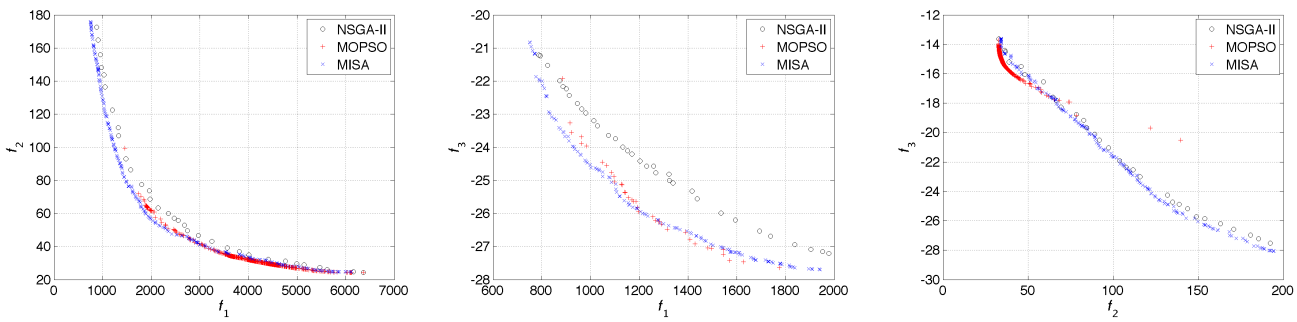


Figure 5: Pareto fronts of the 14-bar truss considering the 3 objectives functions two by two

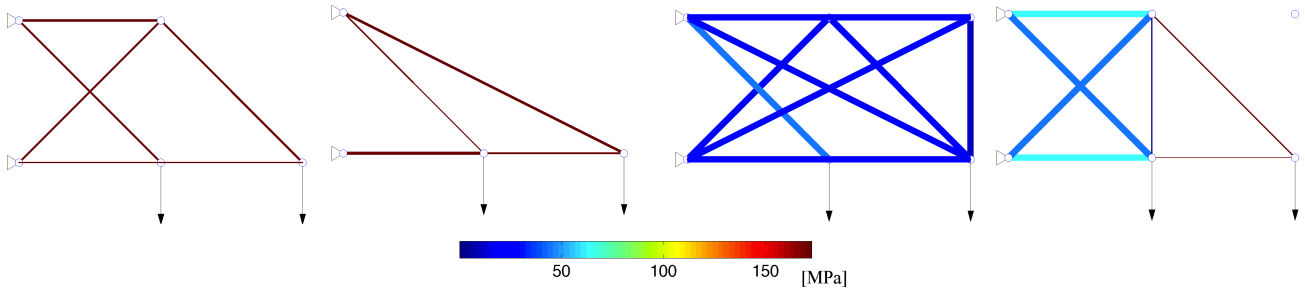


Figure 6: Topologies of the 14-bar truss obtained for the 3 individual objectives functions (stresses are colorized on the bar element, $\min(f_1 = w)$ left, $\min(f_2 = u)$ middle right, $\min(f_3 = -f)$ right). Subfigure at the middle left is the representation of the suboptimal case presented in the 4th column of the Table 3

References

[1] J.E. Alvarez-Benitez, R.M. Everson, and J.E. Fieldsend, A MOPSO algorithm based exclusively on Pareto dominance concepts. Lecture notes in computer science Vol. 3410, 459-473. Springer-Verlag, 2005.

[2] M. Arkadiusz, Geometrical aspects of optimum truss like structures for three-force problem, Structural and Multidisciplinary Optimization, 2012.

[3] M.P. Bendsøe, O. Sigmund, Topology Optimization , Springer, Berlin, 2003.

[4] H. Bilil, R. Ellaia & M. Maaroufi, A New Optimal Multi-objective Reactive Power Dispatch Approach Using Particle Swarm Optimization Based on Matrix Computation. International Conference on Multimedia Computing and Systems (ICMCS), 1119 - 1124, 2012.

[5] B. Bochenek, P. Foryś, Structural optimization for post buckling behavior using particle swarms, Struct Multidisc Optim, 521 531, 2006.

[6] L.,Cagnina , S. Esquivel and A. Coello Coello Carlos, A Particle Swarm Optimizer for Multi-Objective Optimization, JCS & T vol. 5 No. 4, 204-210, 2005.

[7] Carlos A. Coello Coello, Handling Multiple Objectives With Particle Swarm Optimization, IEEE Transactions on Evolutionary Computation, Vol. 8, No. 3, 256-279, 2004.

[8] C.A. Coello Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimization, IEEE Transactions on Evolutionary Computation Vol 8, No 3, 256-279, 2004.

[9] Y. Collette and P. Siarry Multiobjective optimization: principles and case studies. Berlin: Springer, 2003.

[10] K. Deb, Multi-objective Optimization using Evolutionary Algorithms, John Wiley & Sons, Chichester, England, 2001.

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

[12] R.C. Eberhart , Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the IEEE congress evolutionary computation, San Diego, CA, 84-88, 2000.

[13] J.E. Fieldsend, & S.A. Singh, Multi-objective algorithm based upon particle swarm optimization, an efficient data structure and turbulence. In Proceedings of the 2002 UK workshop on computational intelligence, Birmingham, UK, 37-44, 2002.

[14] F.M. Hemez, E. Pagnacco, Statics and inverse dynamics solvers based on strain-mode disassembly, Revue Européenne des Eléments Finis, Vol 9, No 5, 511-560, 2000.

[15] V. L Huang, P.N. Suganthan, J.J. Liang, Comprehensive learning particle swarm optimizer for solving multiobjective optimization problems, International Journal of Intelligent Systems 21 (2), 209-226, 2006.

[16] X. Hu, & R. Eberhart, Multi objective optimization using dynamic neighborhood particle swarm optimization. In Congress on evolutionary computation (CEC?2002), Piscataway, New Jersey, IEEE Service Center. Vol. 2, 1677-1681, 2002.

[17] J. Kennedy, & R. C. Eberhart, Particle swarm optimization. In Proceeding of the IEEE international conference on neural networks, Piscataway: IEEE, 1942-1948, 1995.

[18] J. Kennedy, R.C. Eberhart and Y. Shi, Swarm intelligence, San Francisco: Morgan Kaufmann Publishers, 2001.

[19] A.G.M. Michell, The limits of economy in frame structures. Philosophical Magazine Sect. 6 , 8(47), 589-597, 1904.

[20] S. Mostaghim, and J. Teich, Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In Proceedings of the IEEE swarm intelligence symposium, 26-33, 2003.

[21] N.A. Moubayed, A. Petrovski and J. McCall, A novel smart multiobjective particle swarm optimization using decomposition, in: Parallel Problem Solving from Nature, PPSN XI, in: LNCS, vol. 6239, 1-10, 2010.

[22] K.E. Parsopoulos and M.N. Vrahatis, Particle Swarm Optimization Method in Multi-objective Problems, in Proceedings of the 2002 ACM Symposium on Applied Computing (SAC'2002), 603-607, 2002.

[23] P. Pedersen, On the optimal layout of multi-purpose trusses, Computers & Structures, 2, 695-712, 1972.

[24] W. Peng and Q.A. Zhang, decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization problems. In IEEE International Conference on Granular Computing, 2008. GrC, 534 -537, 2008.

[25] K.T. Praveen , B. Sanghamitra and Sankar Kumar Pal, Multi-Objective Particle Swarm Optimization with time variant inertia and acceleration coefficients, Information Sciences 177, 5033-5049, 2007.

[26] T. Ray, K.M. Liew, A Swarm Metaphor for Multiobjective Desing Optimization., Engineering Optimization, vol. 34, no. 2, 141- 153, 2002.

[27] U. Ringertz, On topology optimization of trusses. Engng Optirn. 9, 209-217, 1985.

[28] M.R. Sierra and C.A. Coello Coello, Multi-objective particle swarm optimizers: A survey of the state-of-the-art. International Journal of Computational Intelligence Research, 2(3), 287-308, 2006.

[29] H.R. Tizhoosh, Opposition-based reinforcement learning, Journal of Advanced Computational Intelligence and Intelligent Informatics 10 (3), 578-585, 2006.

[30] E. Zitzler, K. Deb and L. Thiele, Comparison of multi-objective evolutionary algorithms: Empirical results. Evolutionary Computation, 8(2), 173-195, 2000.

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

[32] L. Wen-Fung and Gary G. Yen, Dynamic Population Size in PSO-based Multiobjective Optimization, Proceedings of IEEE Congress on Evolutionary, Computation Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada, 6182-6189, 2006.