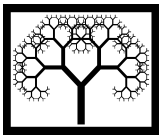
Paper 66



Proceedings of the Eleventh International Conference
on Computational Structures Technology,
B.H.V. Topping, (Editor),
Civil-Comp Press, Stirlingshire, Scotland

Multi-Objective Ant Colony Approaches for Structural Optimization Problems

J.S. Angelo[1], H.S. Bernardino[1] and H.J.C. Barbosa[1,2]
[1]National Laboratory for Scientific Computing
LNCC / MCTI, Petropolis RJ, Brazil
[2]Federal University of Juiz de Fora, UFJF, MG, Brazil

Abstract

In this paper two ant colony optimization (ACO) techniques are proposed, based on the ant system (AS) and the ant colony system (ACS), to solve multi-objective structural optimization problems. The algorithms are equipped with an adaptive penalty method (APM) in order to handle the constraints of the model, corresponding to the stress of the bars. The results obtained in the computational experiments are used to analyse the proposed methods as well as to compare with values found in the literature. The five structural optimization test problems in which design variables assume discrete values and the objective is to minimize both the structure's weight and maximum displacement of its nodes. Finally, two different performance metrics, namely the hypervolume and the empirical attainment function, are used to indicate the best performing algorithm.

Keywords: ant colony, multi-objective, structural optimization, constraint.

1 Introduction

Optimization techniques are fundamental tools in structural design, indicating promising ways in which a designer or a decision maker can extract good performance from the available resources. Due to the increasing competitiveness in industry and consequently, of the quality of its products (according to a set of measures), the design and analysis of novel optimization techniques in general, and nature inspired techniques in particular, are being widely explored.

Nature inspired metaheuristics can help overcome the challenges of the structural optimization problems, such as multiple objectives, discrete valued design variables, low regularity of the objective function(s), the large number of non-linear implicit constraints, and expensive and/or unreliable gradients. In fact, besides the constraint

in the stress of the bars, the problem of optimization of a framed structure can be formulated considering multiple and conflicting objective functions, for example the weight of the structure and the displacement of its nodes. Also, the design variables are often to be chosen from a list of commercially available sizes leading to a discrete search space. Thus, the use of nature inspired techniques becomes attractive.

Most classical methods to solve multi-objective problems are based on reducing the multiple objectives to a single one by aggregating them, usually by a linear combination. These classical techniques have serious drawbacks, as they require *a priori* information about the problem (weights or thresholds). Moreover, many runs are needed in order to obtain different solutions, since just one is obtained in each run of the classical methods.

In a typical multi-objective optimization situation, there is a set of alternatives that are superior to the remainder when all the objectives are considered. This set of the so called non-dominated solutions is the Pareto optimum set, and provides many options to the decision-maker.

Among the metaheuristics, ant colony algorithms are probabilistic techniques inspired by the collective behavior of ants which are used for solving a wide range of optimization problems, specially those which can be reduced to finding good paths through graphs. In the case of multi-objective problems, the ant colony optimization (ACO) techniques considered here deal with the objectives using one pheromone matrix per objective, and combining them to build the solutions.

Using nature inspired metaheuristics, such as ACO, one way to handle constraints is penalizing the objective function(s) according to the level of constraint violation. However, to generate an unique performance measure combining these values is not an easy task. Here the penalty coefficients are automatically calculated by using the adaptive penalty technique (APM) proposed in [1].

Combining all those ideas, we propose here a multi-objective ant colony (MOACO) approach, equipped with APM, to solve discrete and multi-objective structural optimization problems.

The computational experiments include five structural optimization problems, namely 10-, 25-, 60-, 72-, and 942-bar trusses, in which the structure's weight and the maximum displacement of its nodes (both to be minimized) are the objective functions. The results are discussed and the proposed techniques are compared using two multi-objective performance metrics, namely the hypervolume and the empirical attainment function.

2 Multi-objective Optimization Problems

In single-objective optimization one has to find a unique solution that minimizes or maximizes an objective function subject to given constraints. On the other hand, in multi-objective optimization (MOO) several objectives have to be simultaneously op-

timized. Thus, without loss of generality, this type of problem can be formulated as

$$\begin{aligned} &\text{minimize } \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_k(\mathbf{x})] \\ &\text{subject to } \mathbf{x} = (x_1, x_2, ..., x_n) \in \mathcal{S} \end{aligned} \tag{1}$$

where $\mathcal{S}$ is the *feasible set*, $\mathbf{f}(\mathbf{x})$ is an objective vector with $k(\geq 2)$ objective functions to be minimized and $\mathbf{x}$ is the vector of decision variables, which is a feasible solution if $\mathbf{x} \in \mathcal{S}$. The image of the feasible set, denoted by $\mathcal{Z}(= \mathbf{f}(\mathcal{S}))$, is known as the *feasible objective set*. The elements of $\mathcal{Z}$ are the objective functions with $\mathbf{z} = (z_1, z_2, ..., z_k)^T$, where $z_i = f_i(\mathbf{x})$ for all $i = 1, ..., k$ are objective values. Frequently these objectives are conflicting and possibly non-commensurable. In such cases, usually there is no single optimal solution, but a set of alternatives that outperform the remainders when all objectives are considered.

Assuming a problem in which all objectives should be minimized, a solution $\mathbf{x} \in \mathcal{S}$ dominates another solution $\mathbf{x}' \in \mathcal{S}$ (written as $\mathbf{x} \prec \mathbf{x}'$), when

$$f_i(\mathbf{x}) \leq f_i(\mathbf{x}') \quad \forall i \in \{1, ..., k\} \quad \text{and} \quad \exists j \in \{1, ..., k\} : f_j(\mathbf{x}) < f_j(\mathbf{x}') \tag{2}$$

i.e., the solution $\mathbf{x}$ is no worse than $\mathbf{x}'$ in all objectives and better in at least one. Thus, all possible pairwise solutions can be compared in order to find which solutions are non-dominated. Finally, the set of non-dominated solutions is called *Pareto-optimal set* and the corresponding image in the objective space defines the *Pareto front*, giving the best collection of solutions found for the problem.

3 Multi-objective Structural Optimization Problem

The multi-objective structural optimization problem solved here consists in finding a set of discrete cross-sectional areas $\mathbf{A} = \{A_1, A_2, ..., A_N\}$ which minimizes both the total weight $w(\mathbf{A})$ of a given structure and the maximum displacement $d(\mathbf{A})$ of its nodes, subject to stress constraints. The problem can be expressed as minimizing

$$w(\mathbf{A}) = \sum_{k=1}^{N} \gamma A_k L_k \tag{3}$$

$$d(\mathbf{A}) = \max(|u_{i,l}|), \quad \text{with } 1 \leq i \leq M, \quad 1 \leq l \leq N_L \tag{4}$$

subject to the (normalized) stress constraints

$$\frac{|s_{j,l}|}{s_{adm}} - 1 \leq 0, \quad 1 \leq j \leq N, \quad 1 \leq l \leq N_L, \tag{5}$$

where N is the total number of bars in the truss structure, M is the number of translational degrees of freedom, γ is the specific weight of the material, L_k is the length of the k-th bar, N_L is the number of load cases applied to the structure, u_i is the nodal displacement of the i-th translational degree of freedom, s_j is the stress of the j-th bar, and s_{adm} is the allowable stress for the material.

Although the function $w(\mathbf{A})$ from Eq. (3) is simple, $d(\mathbf{A})$ (Eq. 4) and the constraints are complex implicit functions of the design variables $\mathbf{A}$. They require the solution of the equilibrium equations of the discrete model

$$\mathbf{K}(\mathbf{A})\mathbf{u}_l = \mathbf{f}_l, \quad 1 \leq l \leq N_L, \tag{6}$$

where $\mathbf{K}$ is the symmetric and positive definite stiffness matrix of the structure, and $\mathbf{u}_l$ and $\mathbf{f}_l$ are, respectively, the vector of nodal displacements and the vector of nodal forces for the l-th load condition. For each load condition, the system (Eq. 6) is solved for the displacement field and the stress in the j-th bar is then calculated according to Hooke's Law as

$$s_{j,l} = E\delta(\mathbf{u}_l), \tag{7}$$

where E is the Young's modulus and δ is the unit change in length of the bar.

4 Ant Colony Optimization

Ant Colony Optimization is a metaheuristic inspired by the behavior of real ant colonies [2, 3]. In this technique artificial ants collaborate while exploring the search space in order to find good solution for the problem. ACO algorithms are essentially constructive, as ants build solutions by adding solution components to an initially empty solution until the solution is complete. While moving, artificial ants deposit pheromone on the "edges crossed" marking a path that may be followed by other members of the colony, which then reinforce the pheromone on that path. This self-organizing behavior results in a self-reinforcing process that leads to the formation of a path marked by high pheromone concentration. Less used paths tend to have a diminishing pheromone level due to evaporation.

The movement of the ants is guided by (i) a heuristic information (η), that represents *a priori* information about the problem instance; and (ii) a pheromone trail (τ), that encodes a memory about the ant colony search process which is continuously updated by the ants. These values are used by the ant's heuristic rule to decide (probabilistically) the next node to be visited. When all ants have generated their solutions, the pheromone trail is updated according to the quality of the corresponding candidate solutions: the components of good solutions are reinforced (positive feedback) and a certain level of pheromone evaporation is applied to all edges.

A pseudo-code for a standard ACO metaheuristic is given by Algorithm 1. In *ConstructAntSolution()*, ants create a solution path according to the decision policy, which is composed by the pheromone trail and the heuristic information associated with the problem. After constructing a complete path, it is possible to apply a local search procedure in order to improve the solution obtained. Then, the pheromone trails are updated in *UpdateGlobalPheromone()*, considering the quality of the candidate solutions generated as well as a certain level of pheromone evaporation. The cycle is repeated until the termination criteria are met, when the algorithm returns the best solution found.

Algorithm 1: Pseudo-code for a standard ACO metaheuristic.

```
1 Initialize pheromone trails τ and heuristic information η ;
2 while termination criteria not met do
3     ConstructAntSolution();
4     ApplyLocalSearch(); // (optional)
5     UpdateGlobalPheromone();
6 Return the best solution;
```

5 The Proposed Ant Colony Algorithms

In the last years, ACO algorithms have been proposed for solving combinatorial and multi-objective optimization problems. A review of multi-objective ACO (MOACO) algorithms can be found in [4].

A pseudo-code for a standard MOACO algorithm is given by Algorithm 2, where K is the number of objectives. The algorithm starts generating the pheromone trails, heuristic information, and an empty Pareto set. Besides the steps described for the single objective case, MOACO includes *UpdateParetoSet()*, which updates $\mathcal{P}$ with the newly created solutions, keeping only the non-dominated ones. Also, it is important to highlight that here multiple pheromones, corresponding to each objective, are updated and are combined to guide the construction of the solutions.

Algorithm 2: Pseudo-code of a standard MOACO algorithm.

```
1 Initialize pheromone trails τ (τ^k) and heuristic information η (η^k) ;
2 Initialize an empty Pareto set P ;
3 while termination criteria not met do
4     ConstructAntSolution();
5     ApplyLocalSearch(); // (optional)
6     UpdateParetoSet();
7     UpdateGlobalPheromone();
8 Return the Pareto set P;
```

In this work, two ant colony algorithms are proposed to solve the discrete and multiple objective structural optimization problems. The first one, named MOAS, was elaborated based on the Ant System (AS) algorithm [5, 6, 7], while the second one, named MOACS, based on the Ant Colony System (ACS) algorithm [8, 9].

5.1 The MOAS algorithm

As mentioned earlier, the MOAS is based on the AS algorithm, which has to be adapted to solve multi-objective optimization problems. For the bi-objective problems dealt with in this work, two pheromone matrices τ_{ij}^1 and τ_{ij}^2 are used, each one asso-

ciated with the objectives: the structure's weight, and the maximum displacement of its nodes, respectively. In this way, τ_{ij}^k is the k-th pheromone matrix entry associated with the i-th design variable and the j-th cross-section area. The heuristic information is associated with the cross-sections areas of the structure and is given by $\eta_j = \frac{A_{max}}{A_j}$, where A_{max} is the maximum cross-section area available and A_j is the j-th candidate cross-section area.

In MOAS, the ant h chooses the cross-section area j for the design variable i according to the following probabilistic rule:

$$p_{ij}^h = \frac{[\tau_{ij}^1]^{\lambda\alpha_1}[\tau_{ij}^2]^{(1-\lambda)\alpha_2}[\eta_j]^{\lambda\beta}}{\sum_j [\tau_{ij}^1]^{\lambda\alpha_1}[\tau_{ij}^2]^{(1-\lambda)\alpha_2}[\eta_j]^{\lambda\beta}}, \quad \forall j, \tag{8}$$

where α_1, α_2 and β are parameters that balance the influence of the pheromone and the heuristic information, respectively. In order to emphasize differently each objective when different solutions are constructed, λ is calculated for each ant $h \in \{1, ..., m\}$ as $\lambda = (h-1)/(m-1)$. Thus, the first ant ($\lambda = 0$) considers only the second objective, while only the first objective contributes for the decision of the m-th ant ($\lambda = 1$). When all solutions (paths) are created, the Pareto set is updated.

The next step is to update the pheromone matrices, using the solutions stored in the Pareto set. As in AS, the pheromone update step starts by evaporating all pheromone trails on both matrices, as

$$\tau_{ij}^1 \leftarrow (1-\rho_1)\tau_{ij}^1, \quad \tau_{ij}^2 \leftarrow (1-\rho_2)\tau_{ij}^2, \tag{9}$$

where ρ_1 and ρ_2 are the pheromone evaporation factors. In the sequence, the solutions in the Pareto set are allowed to update the pheromone matrices, as follows

$$\tau_{ij}^1 \leftarrow \tau_{ij}^1 + \frac{w_{min}}{f_1(s)}, \quad \tau_{ij}^2 \leftarrow \tau_{ij}^2 + \frac{d_{min}}{f_2(s)}, \tag{10}$$

where w_{min} and d_{min} are, respectively, the minimum possible weight for the structure and the minimum displacement. The objective functions $f_1(s)$ and $f_2(s)$, corresponding to the solution s, are calculated via the APM method.

Finally, when the termination criteria are met, the algorithm returns the Pareto set.

5.2 The MOACS algorithm

The MOACS algorithm differs from MOAS in two main points: (i) it explores the search space using a more aggressive action choice rule; and (ii) it updates the pheromone trails each time an ant crosses an arc (i, j), by removing some pheromone from that arc, in order to increase the exploration of alternative choices. In MOACS, the ant can choose the next cross-section area j according to the so called *pseudo-random proportional* rule, given by

$$j = \begin{cases} \arg\max_{\forall j}\{[\tau_{ij}^1]^{\lambda\alpha_1}[\tau_{ij}^2]^{(1-\lambda)\alpha_2}[\eta_j]^{\lambda\beta}\}, & \text{if } q \leq q_0 \\ \text{Eq. (8)} & \text{otherwise,} \end{cases} \tag{11}$$

where q is a random value uniformly distributed in $[0, 1]$ and $q_0 \in [0, 1]$ is a user defined parameter. Notice that an ant chooses with probability q_0 the best value as indicated in the pheromone matrices and heuristic information, intensifying the search, and performs a diversification in the search space with probability $(1 - q_0)$.

The update of the local pheromone, performed every time an ant chooses a new cross-section area, is given by

$$\tau^1_{ij} \leftarrow (1-\xi)\tau^1_{ij} + \xi\tau_0, \quad \tau^2_{ij} \leftarrow (1-\xi)\tau^2_{ij} + \xi\tau_0, \tag{12}$$

where $\xi \in (0, 1)$ is the local pheromone evaporation rate and τ_0 is the initial pheromone value. After all solutions are created, the Pareto set is updated and its elements are used to update the global pheromone as

$$\tau^1_{ij} \leftarrow (1-\rho_1)\tau^1_{ij} + \rho_1\frac{w_{min}}{f_1(s)}, \quad \tau^2_{ij} \leftarrow (1-\rho_2)\tau^2_{ij} + \rho_2\frac{d_{min}}{f_2(s)} \tag{13}$$

Finally, the algorithm returns the Pareto set.

5.3 Adaptive Penalty Method

Introducing a penalty function is a common way to handle the constraints when using nature inspired techniques for optimization. However, the penalty coefficients are highly problem dependent and need to be tuned for each application. The adaptive penalty method (APM) proposed in [1] aims at alleviating the user from the task of defining those values. The technique automatically sets those values using feedback from the search process. The idea is to observe how each constraint is being violated and set a higher penalty coefficient to those constraints which seem to be more difficult to satisfy. The quantities included in the definition of the penalty values are the average value of the objective function of the solutions and the average violation of the j-th constraint, both for a given set of solutions.

Considering each objective separately and assuming their minimization subject to m constraints, the quality of a given solution can be written as [10]

$$F(x) = \begin{cases} f(x), & \text{if } x \text{ is feasible} \\ \overline{f}(x) + \sum_{j=1}^{m} k_j v_j(x), & \text{otherwise} \end{cases} \tag{14}$$

$$\overline{f}(x) = \begin{cases} f(x), & \text{if} \quad f(x) > \langle f(x)\rangle \\ \langle f(x)\rangle, & \text{otherwise} \end{cases} \tag{15}$$

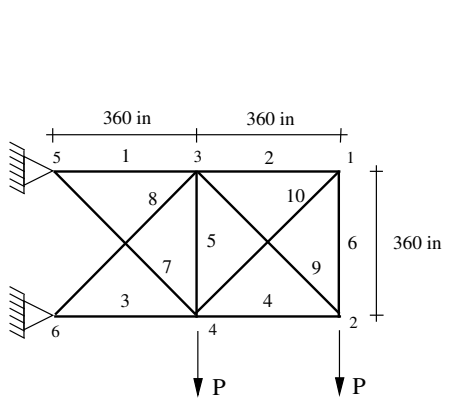
The penalty coefficient k_j corresponding to the $j-$th constraint is defined at every iteration of the MOAS and MOACS algorithms by

$$k_j = |\langle f(x)\rangle| \frac{\langle v_j(x)\rangle}{\sum_{l=1}^{m} [\langle v_l(x)\rangle]^2}, \tag{16}$$

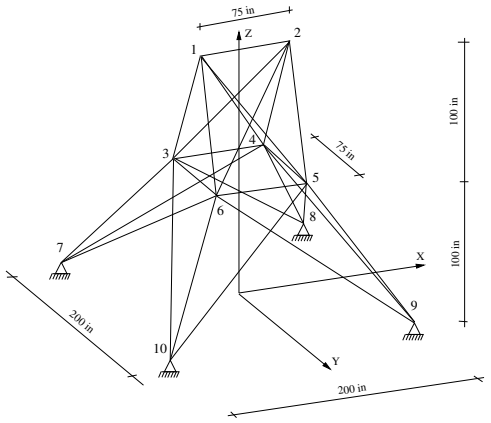
where $\langle f(x)\rangle$ is the average fitness in the current population and $\langle v_l(x)\rangle$ is the violation of the l-th constraint averaged over the current population.

6 Computational Experiments

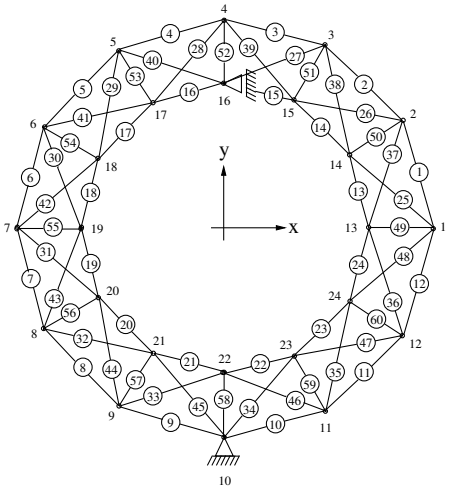
Five multi-objective structural engineering problems from the literature, namely the 10-, 25-, 60-, 72-, and 942-bar trusses, are used to assess the performance of the proposed techniques. These structures are illustrated (not to scale) by Figures 1 and 2.



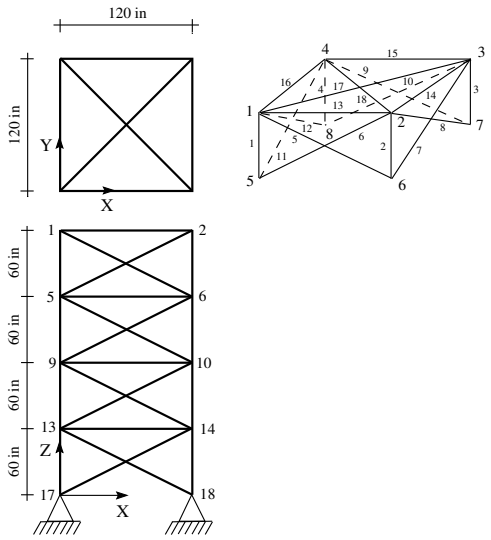
(a) The 10-bar truss



(b) The 25-bar truss



(c) The 60-bar truss



(d) The 72-bar truss

Figure 1: Illustration of some truss structures of the computational experiments.

Each algorithm was executed 100 times, with $m = 50$ ants and 1000 iterations, for all test-problems. The results obtained by the proposed techniques are displayed in figures, containing the Pareto front, the (normalized) hypervolume, and the attainment surfaces. Although no work was found solving exactly the same test-problems, we collected reference solutions from the literature, for each one. These results are indicated by a point-shaped "x" and correspond to (i) the weight of the structures as presented in [11], for the 10-, 25-, 60-, and 72-bar trusses, and in [12], for the 942-

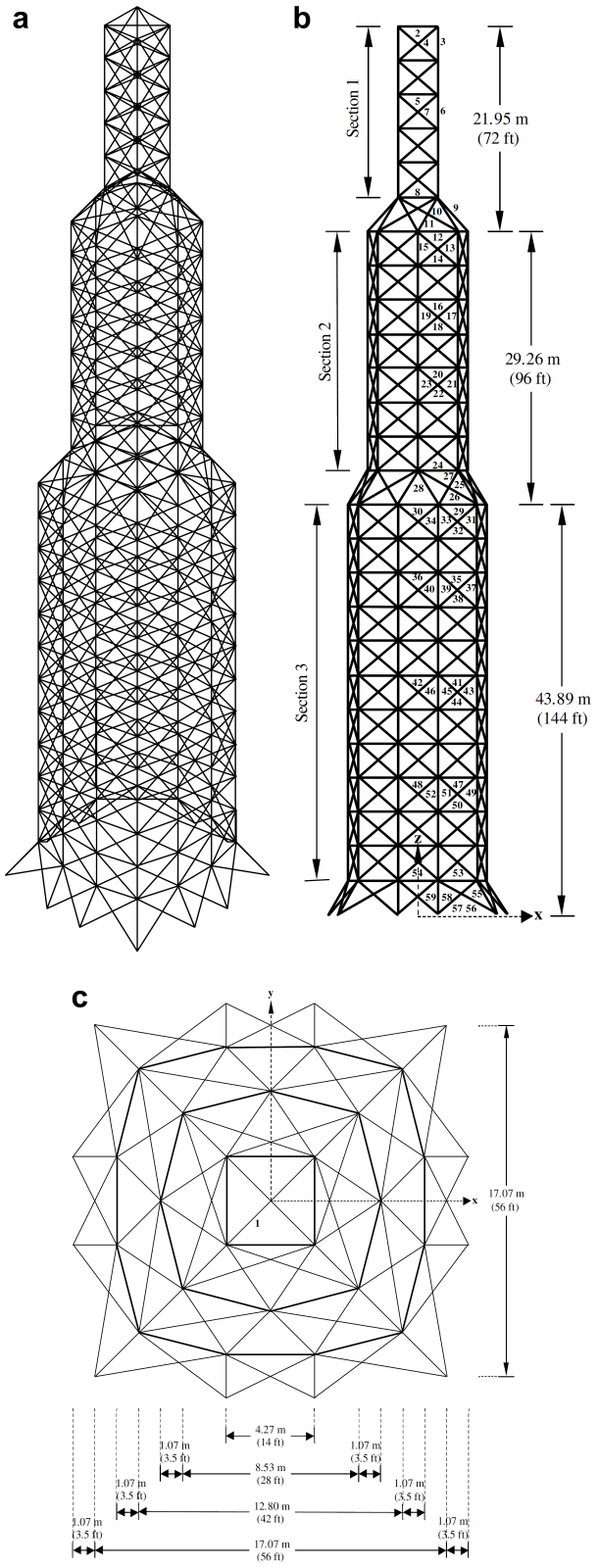


Figure 2: Illustration of the 942-bar truss structure.

bar truss; and (ii) the maximum displacements allowed in the single-objective cases, where the displacement of the nodes are constraints of the problem.

Some preliminary experiments were performed to find reasonable values for the user defined parameters. The parameters chosen for MOAS were: $\alpha_1 = 1$, $\alpha_2 = 1$, $\beta = 2$, $\rho_1 = 0.2$, and $\rho_2 = 0.1$. For the MOACS, the parameters settings are: $\alpha_1 = 2$, $\alpha_2 = 1$, $\beta = 1$, $\rho_1 = 0.1$, $\rho_2 = 0.2$, $q_0 = 0.2$, and $\xi = 0.1$. The initial values of the pheromone matrices, indicated by the literature, were $\tau_0 = m$ for the MOAS and $\tau_0 = 1.0/m$ for the MOACS.

6.1 Performance Assessment

Performance analysis in multi-objective optimization is a difficult task, because different aspects of the set of solutions should be considered. Zitzler *et al* [13] describes three distinct goals that should be taken into account when assessing the quality of the non-dominated solutions set: (i) the solutions should be as close as possible to the Pareto-optimal set (when it is known); (ii) a wide range of distinct solutions should be presented; and (iii) the solutions should be well distributed along the obtained non-dominated front. The inclusion of these desirable aspects makes the assessment itself a multi-objective problem.

Many quality measures or performance metrics, are used for comparing solutions sets, in order to assist the search for the best one. A list of performance metrics can be found in [14, 15]. In [15] the metrics are classified as: (i) metrics that evaluate closeness to the Pareto-optimal front; (ii) metrics evaluating diversity among non-dominated solutions; and (iii) metrics that evaluate both closeness and diversity. Here, the Hypervolume metric [16] and the Empirical Attainment Function (EAF) [17] are used to analyze the results of the computational experiments. They were chosen because of their capacity in evaluating closeness and diversity in a combined sense.

6.1.1 Hypervolume

The hypervolume metric (HV) calculates the hypervolume of a multi-dimensional region (in the objective space) formed by a set of solutions A and a reference point, in order to calculate the "size" of the region dominated by the set A. A convenient reference point needs to be chosen so as not to induce misleading results. Notice that higher values are preferable.

6.1.2 Empirical Attainment Function

The empirical attainment function gives the probability distribution of an outcome set generated by a stochastic algorithm using the notion of goal-attainment. In this sense, the attainment function indicates the chances that an arbitrary point in the objective space is attained by (dominated by or equal to) the outcome of a single run of a particular algorithm.

EAF express the multiple runs of an algorithm in terms of the $k\%$-attainment set, by means of the line separating the objective space attained by $k\%$ of the runs. For instance, the *median* attainment surface indicates the region attained by 50% of the runs, while the *worst* attainment surface delimits the region attained by all runs (100%-attainment surface). It is easy to see that the *best* attainment surface indicates the region covered by the non-dominated solutions including the solutions obtained in all independent runs. Finally, notice that the *best* attainment surface is bounded by the region attained by at least one run and the solution vectors attained by any run.

6.2 The 10-bar Truss Design

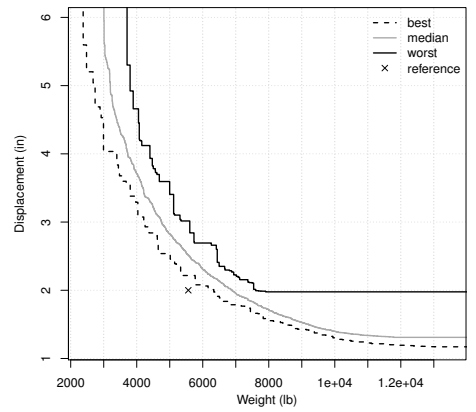
The problem presented in this section is the 10-bar truss (see Figure 1a). For this test-problem, the material has $\gamma = 0.1$ lb/in^3, and $E = 10^4$ ksi. Vertical downward loads of 100 kips are applied at nodes 2 and 4, and the stress in each member is limited to $\pm$ 25 ksi. The values of the cross-sectional areas (in^2) are chosen from the set: 1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.50, 13.50, 13.90, 14.20, 15.50, 16.00, 16.90, 18.80, 19.90, 22.00, 22.90, 26.50, 30.00, and 33.50; resulting in 42 options.

The results obtained by the proposed techniques for this test-problem, as well as a reference point, are presented in Figure 3. The reference point considered is $(5562.355, 2)$, which are, respectively, the weight obtained by the DUVDE+APM algorithm in [11], and the maximum allowed displacement in the mono-objective case.

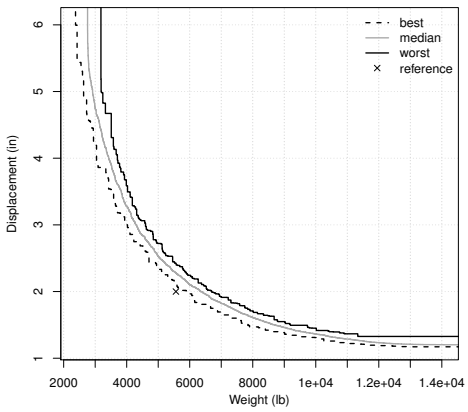
By checking the Pareto front of both algorithms, one can conclude that MOACS is slightly better than MOAS. The EAFs plots, shown in Figures 3a and 3b, show that MOACS seems to be more stable, since the distance between the best and the worst surface is shorter than in MOAS. This indicates that the MOACS was able to generate solutions close to each other considering all runs, while the solutions found by MOAS are more spread out towards the best front. The EAFs associated with the outcomes of the algorithms (Figure 3d) demonstrated that MOACS generates solutions near the best attainment surface, considering the results obtained by both algorithms, indicating the best performance of MOACS. Finally, the value of the hypervolume achieved by the MOACS results is larger than that calculated for MOAS, reinforcing the previous conclusions.

6.3 The 25-bar Truss Design

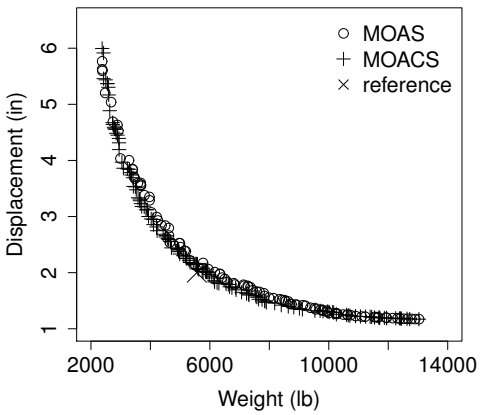
Here the 25-bar truss structure (Figure 1b) is used as test-problem. In this case, the material has $\gamma = 0.1$ lb/in^3, $E = 10^4$ ksi, and the constraints require that the maximum stresses in the members remain in the interval $[-40, 40]$ ksi. The design variables are to be chosen from the set (in^2): 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.8, 3.0, 3.2, 3.4.



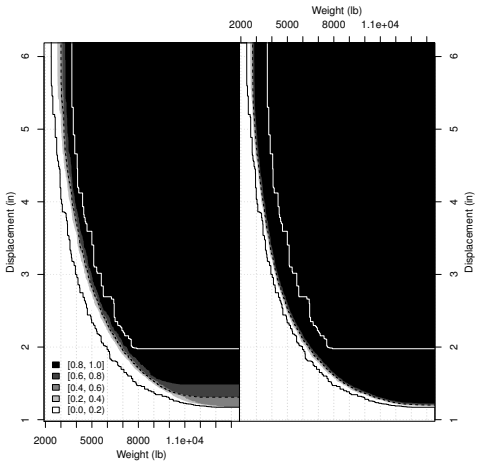
(a) EAF from MOAS results.



(b) EAF from MOACS results.



(c) Final Pareto front.



(d) EAFs plots for MOAS and MOACS.

Figure 3: Results for the 10-bar truss. The final Pareto fronts lead to $HV = 0.9727$ for MOAS and $HV = 1$ for MOACS.

Finally, the loading data for the current structure and the design variables, linked in eight groups, are detailed in the Table 1 and Table 2, respectively.

Figure 4 presents the results obtained by the proposed techniques when applied to the 25-bar truss problem. The reference point is $(485.90, 0.35)$, which are, respectively, the weight obtained by DUVDE+APM technique in [11], and the maximum displacement allowed in the mono-objective view of the problem.

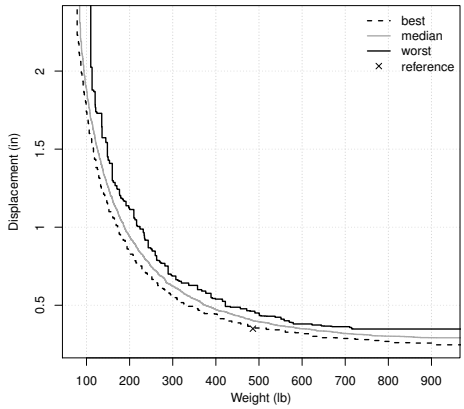
It can be observed that the Pareto front of both algorithms are very close, making it hard to distinguish the performance of the contenders. The same can be said, by examining the EAF plots (Figures 4a and 4b) where it is possible to perceive stability in both algorithms. Although both algorithms seems to perform similarly to this test-problem, the hypervolume metric points out that MOACS performs better than MOAS.

Node	F_x	F_y	F_z
1	1	-10.0	-10.0
2	0	-10.0	-10.0
3	0.5	0	0
6	0.6	0	0

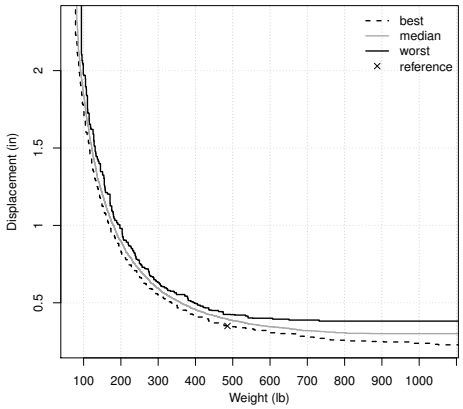
Table 1: Loading data (kips) - 25-bar truss.

Group	Connectivities
A_1	1-2
A_2	1-4, 2-3, 1-5, 2-6
A_3	2-5, 2-4, 1-3, 1-6
A_4	3-6, 4-5
A_5	3-4, 5-6
A_6	3-10, 6-7, 4-9, 5-8
A_7	3-8, 4-7, 6-9, 5-10
A_8	3-7, 4-8, 5-9, 6-10

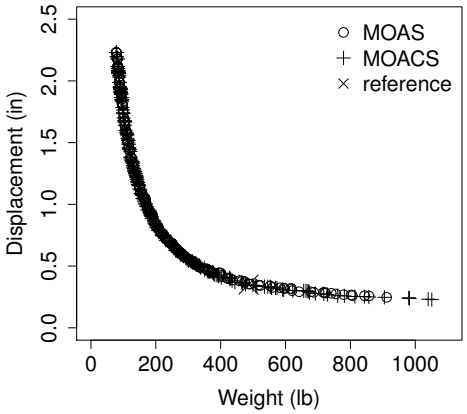
Table 2: Member grouping - 25-bar truss.



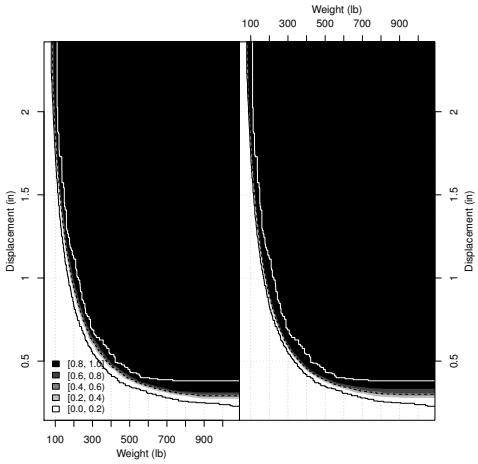
(a) EAF from MOAS results.



(b) EAF from MOACS results.



(c) Final Pareto front.



(d) EAFs plots for MOAS and MOACS.

Figure 4: Results for the 25-bar truss. The final Pareto fronts lead to $HV = 0.9741$ for MOAS and $HV = 1$ for MOACS.

6.4 The 60-bar Trussed Ring

This test-problem involves the 60-bar trussed ring structure, depicted in Figure 1c (not to scale), under three load cases as given in Table 3. The outer radius of the ring is 100 in. and the inner radius is 90 in. The material has $E = 10^4$ ksi and $\gamma = 0.1$ lb/in^3. There are 180 constrains which refer to allowable stress (σ_i = 60 ksi, i = 1 to 60). The grouping of the cross-sectional areas in 25 design variables, to be chosen from the set $\{0.5, 0.6, \ldots, 4.8, 4.9\}$, is listed in Table 4.

Node	F_x	F_y
Load case 1		
1	-10.0	0
7	9.0	0
Load case 2		
15	-8.0	3.0
18	-8.0	3.0
Load case 3		
22	-20.0	10.0

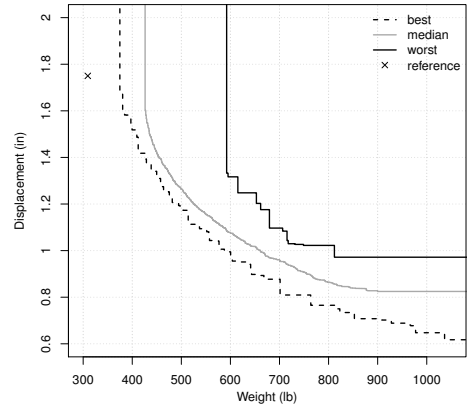
Table 3: Loading data (kips) - 60-bar truss.

Group	Bars	Group	Bars
A_1	49 to 60	A_{14}	25, 37
A_2	1, 13	A_{15}	26, 38
A_3	2, 14	A_{16}	27, 39
A_4	3, 15	A_{17}	28, 40
A_5	4, 16	A_{18}	29, 41
A_6	5, 17	A_{19}	30, 42
A_7	6, 18	A_{20}	31, 43
A_8	7, 19	A_{21}	32, 44
A_9	8, 20	A_{22}	33, 45
A_{10}	9, 21	A_{23}	34, 46
A_{11}	10, 22	A_{24}	35, 47
A_{12}	11, 23	A_{25}	36, 48
A_{13}	12, 24		

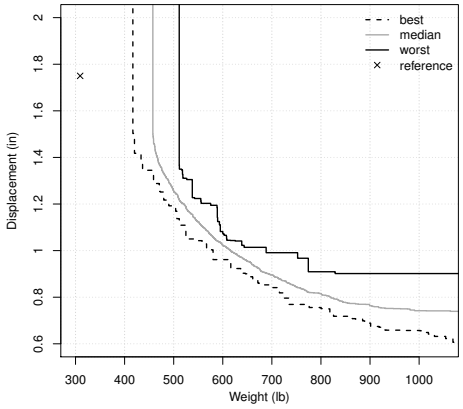
Table 4: Member grouping - 60-bar truss.

The results found by the proposed techniques for the test-problem involving the 60-bar truss structure is presented in Figure 5. Here the reference point is $(309.44, 1.75)$, formed by the weigh obtained by DUVDE+APM algorithm in [11] and the minimum displacement allowed in the mono-objective version of the problem, for all load cases. It is important to highlight that the design variables were chosen in a continuous search space in [11], while only some discrete possibilities are allowed here.

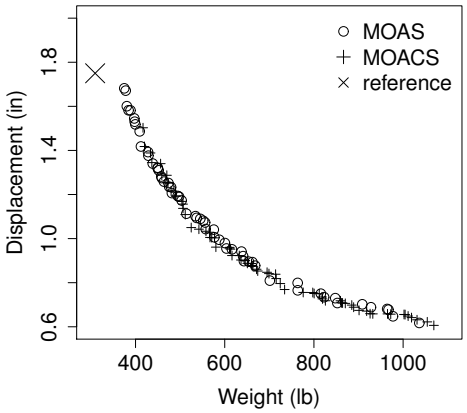
The Pareto front of both algorithms are somewhat similar, making it hard to define the best performing one for this test-problem. The MOAS technique performed better toward the weight objective, while MOACS performs better toward the maximum displacement objective. One can have the same conclusion by checking the EAFs of the outcomes (Figure 5d). The EAF plots, shown in Figures 5a and 5b, indicate that MOACS is more stable than MOAS, since the median surface is closer to the best and worst surface than in MOAS. For this test-problem, the hypervolume indicates that MOAS performs better than MOACS.



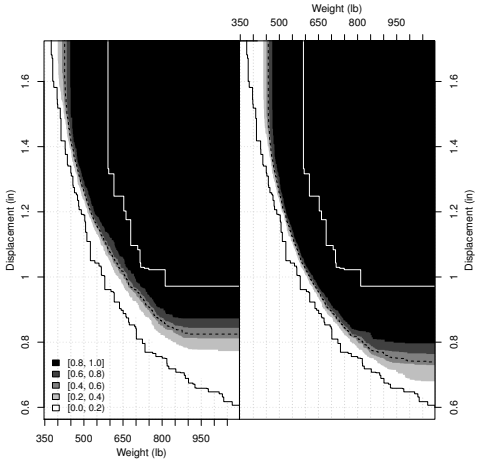
(a) EAF from MOAS results.



(b) EAF from MOACS results.



(c) Final Pareto front.



(d) EAFs plots for MOAS and MOACS.

Figure 5: Results for the 60-bar truss. The final Pareto fronts lead to $HV = 1$ for MOAS and $HV = 0.9616$ for MOACS.

6.5 The 72-bar Truss

The 72-bar truss structure, illustrated in Figure 1d, composes the current test-problem, in which the design variables are linked in sixteen groups detailed in the Table 6, the material has $\gamma = 0.1$ lb/in^3, $E = 10^4$ ksi, and is submitted to two load cases (see Table 5). The stress in each bar is restricted to the range $[-25, 25]$ ksi while the the cross-sectional areas are chosen from the set $\{0.1, 0.2, \ldots, 2.4, 2.5\}$.

The reference point for the 72-bar truss structure test-problem was set to $(379.667, 0.25)$, which are the weight found by DUVDE+APM algorithm in [11] and the maximum allowed displacement, verifying all load cases. Notice that, similarly to the 60-bar truss structure, the reference solves a continuous variation of this test-problem

Node	F_x	F_y	F_z
Load case 1			
1	5	5	-5
Load case 2			
1	0	0	-5
2	0	0	-5
3	0	0	-5
4	0	0	-5

Table 5: Loading data (kips) - 72-bar truss.

Group	Members
A_1	1, 2, 3, 4
A_2	5, 6, 7, 8, 9, 10, 11, 12
A_3	13, 14, 15, 16
A_4	17,18
A_5	19, 20, 21, 22
A_6	23, 24, 25, 26, 27, 28, 29, 30
A_7	31, 32, 33, 34
A_8	35,36
A_9	37, 38, 39, 40
A_{10}	41, 42, 43, 44, 45, 46, 47, 48
A_{11}	49, 50, 51, 52
A_{12}	53,54
A_{13}	55, 56, 57, 58
A_{14}	59, 60, 61, 62, 63, 64, 65, 66
A_{15}	67, 68, 69, 70
A_{16}	71,72

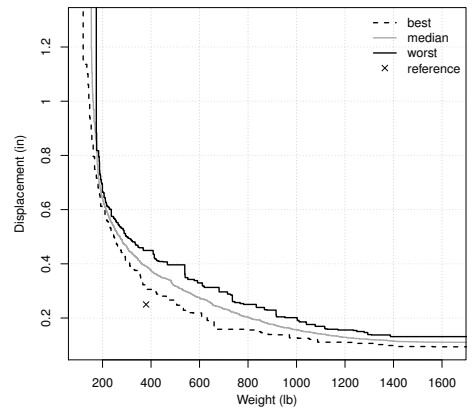
Table 6: Member grouping - 72-bar truss.

while here a small set of values are allowed. The results found for the 72-bar truss problem in the experiments are shown in Figure 6.

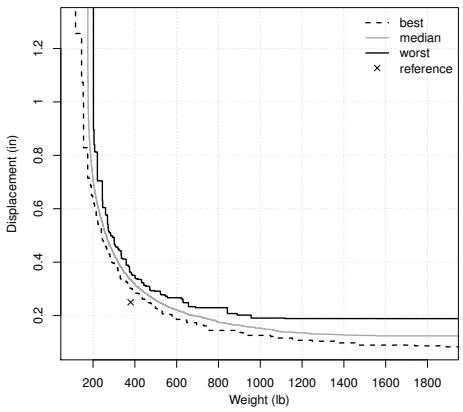
The Pareto fronts indicate that both have the same performance toward the reference point. However, the solutions found by MOACS are preferable because of the (larger) number of non-dominated results. Also, examining the EAF plots, presented in Figure 6d, it is possible to verify that MOACS attains more regions closer to the best surface against MOAS. Finally, the results obtained by MOAS generate only 95.08% of the hypervolume calculated for the values found by MOACS.

6.6 The 942-bar Truss Design

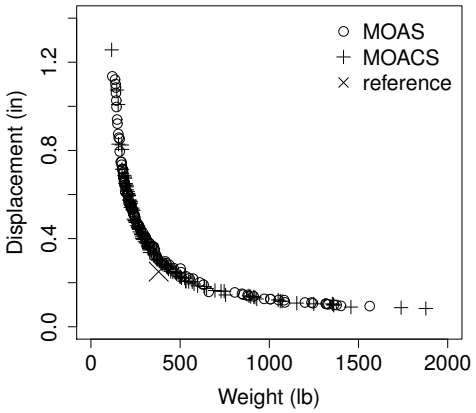
The symmetry of the tower around the x and y-axes is employed to group the 942 truss members into 59 independent size variables, as illustrated in Figure 2. The tower, described in [18], is subject to a single loading condition consisting of both horizontal and vertical loads, as follows: (i) the vertical loads in the z direction are -3.0 kips, -6.0 kips and -9.0 kips at each node in the first, second and third sections, respectively; (ii) the lateral loads in the y direction are 1.0 kips at all nodes of the tower; and (iii) the lateral loads in the x direction are 1.5 kips and 1.0 kips at each node on the left and right sides of the tower, respectively. Here, the design variables are integer values in the interval with lower and upper bounds equal to 1.0 in^2 and 200 in^2, respectively. The constraints of this problem include a stress limitation of 25.0 ksi, both in tension and compression, for all members. The modulus of elasticity is $E = 10^4$ksi and the density is $\gamma = 0.1$ lb/in^3. This problem has been previously studied in [18, 19, 20],



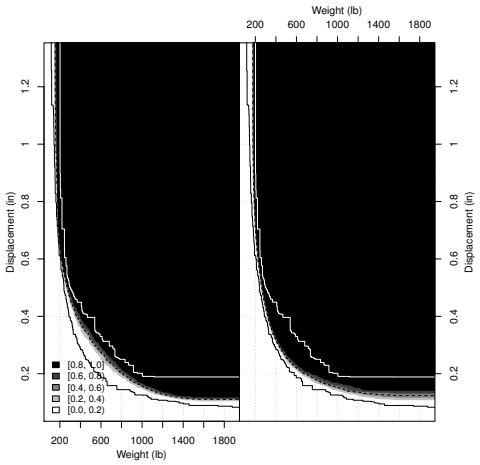
(a) EAF from MOAS results.



(b) EAF from MOACS results.



(c) Final Pareto front.



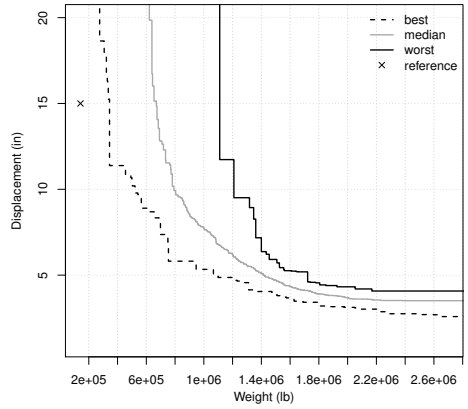
(d) EAFs plots for MOAS and MOACS.

Figure 6: Results for the 72-bar truss. The final Pareto fronts lead to $HV = 0.9508$ for MOAS and $HV = 1$ for MOACS.

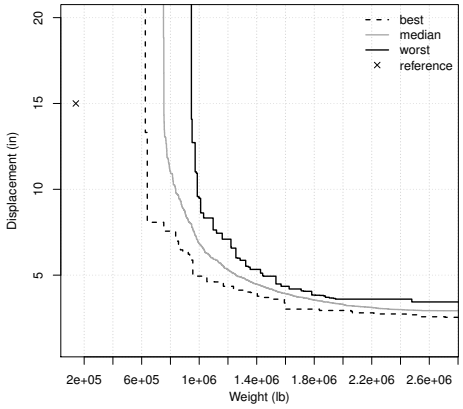
where more detailed information is allowed.

The results obtained for the 942-bar truss problem are presented in Figure 7. For this test-problem, the reference point was set to $(142295.75, 15)$, which are the best weight found by a GA algorithm in [12] and allowed displacement for the mono-objective case of this test-problem.

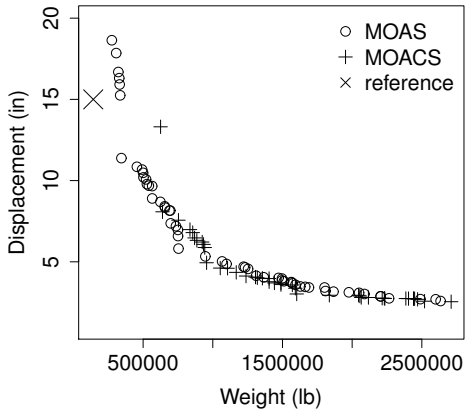
The generation of better solutions among the non-dominated solutions and closer to the reference point indicates that MOAS performed much better than MOACS for this test-problem. Also, the solutions found by MOAS attain more regions closer to the best attainment surface, as presented in Figure 7d. However, MOACS shows to be more stable in the independent runs, as one can see in Figures 7a and 7b. Finally, it is possible to highlight the difference between the hypervolumes for this test-problem,



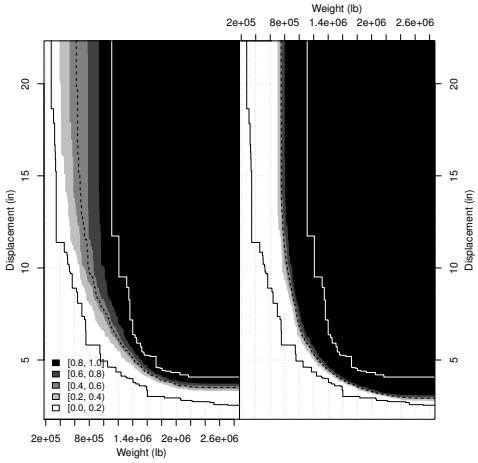
(a) EAF from MOAS results.



(b) EAF from MOACS results.



(c) Final Pareto front.



(d) EAFs plots for MOAS and MOACS.

Figure 7: Results for the 942-bar truss. The final Pareto fronts lead to $HV = 1$ for MOAS and $HV = 0.7189$ for MOACS.

where MOACS found solutions generating a HV which is only 71.89% of the obtained by the results found by MOAS.

7 Concluding Remarks

In this paper two variants of an ant colony approach were proposed to solve discrete and multi-objective structural optimization problems. The techniques proposed were combined with an adaptive penalty method used for dealing with the constraints of the model. From the experiments performed one can see that the MOACS presents a more stable behaviour in the independent runs over the MOAS. In all test problems,

specially in the 10-, 20- and 72-bar problems, the algorithms were capable of generating a large surface (diversity among the non-dominated solution) very close to the reference point (proximity to the ideal solution), suggesting their good performance in view of the qualities expected for non-dominated solution sets.

Acknowledgments

The authors acknowledge the support from CNPq (grants 308317/2009-2, 300192/2012-6 and 141519/2010-0).

References

[1] H.J.C. Barbosa, A.C.C. Lemonge, "An New Adaptive Penalty Scheme for Genetic Algorithms", *Information Sciences*, 156: 215–251, 2003.

[2] M. Dorigo, G.D. Caro, "Ant Colony Optimization: A New Meta-Heuristic", in P.J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, A. Zalzala (Editors), *IEEE Congress on Evolutionary Computation*, pages 1470–1477. Piscataway, NJ, 1999.

[3] M. Dorigo, T. Stützle, *Ant Colony Optimization*, The MIT Press, Cambridge, Massachusetts, 2004, ISBN 0-262-04219-3.

[4] J.S. Angelo, H.J.C. Barbosa, *Ant Colony Optimization: methods and applications*, Chapter Ant Colony Optimization Algorithms for Multiobjective Problems, InTech, 2011.

[5] M. Dorigo, V. Maniezzo, A. Colorni, "Positive feedback as a search strategy", Technical report, Dipartimento di Elettronica e Informatica, Politecnico de Milano, 1991.

[6] M. Dorigo, *Optimization, Learning and Natural Algorithms*, PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy, 1992.

[7] M. Dorigo, V. Maniezzo, A. Colorni, "The Ant System: Optimization by a colony of cooperating agents", in *IEEE Trans. on Systems, Man, and Cybernetics - Part B*, Volume 26, pages 29–41, 1996.

[8] M. Dorigo, L.M. Gambardella, "Ant Colonies for The Traveling Salesman Problem", *Bio Systems*, 43(2): 73–81, 1997.

[9] M. Dorigo, L.M. Gambardella, "Ant Colony System: A cooperative learning approach to the traveling salesman problem", *IEEE Trans. on Evolutionary Computation*, 1(1): 53–66, 1997.

[10] H.J.C. Barbosa, A.C.C. Lemonge, "An Adaptive Penalty Scheme for Genetic Algorithms in Structural Optimization", *International Journal for Numerical Methods in Engineering*, 59: 703–736, 2004.

[11] E.K. Silva, H.J.C. Barbosa, A.C.C. Lemonge, "An adaptive constraint handling techinique for differential evolution with dynamic use of variants in engineering optimization", *Optimization and Engineering*, 12: 31–54, 2011.

[12] F. Erbatur, O. Hasançebi, I. Tütüncü, H. Kiliç, "Optimal design of planar and space structures with genetic algorithms", *Computers & Structures*, 75(2): 209 – 224, 2000.

[13] E. Zitzler, K. Deb, L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results", *Evolutionary Computation*, 8(2): 173–195, 2000.

[14] J.D. Knowles, L. Thiele, E. Zitzler, "A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers", Technical Report 214, Computer Engeneering and Network Laboratory, ETH Zurich, 2005.

[15] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Chichester, 2001.

[16] E. Zitzler, L. Thiele, "Multiobjective optimization using evolutionary algorithms: a comparative case study", in *Fifth International Conference on Parallel Problem Solving from Nature*, pages 292–301. Berlin, Germany, 1998.

[17] C.M. Fonseca, P.J. Fleming, "On the performance assessment and comparison of stochastic multiobjective optimizers", in *Proceedings of Parallel Problem Solving from Nature IV (PPSN-IV)*, pages 584–593, 1996.

[18] O. Hasançebi, "Adaptive evolution strategies in structural optimization: Enhancing their computational performance with applications to large-scale structures", *Computers & Structures*, 86(1-2): 119 – 132, 2008, ISSN 0045-7949.

[19] H. Adeli, N.T. Cheng, "Concurrent Genetic Algorithms for Optimization of Large Structures", *Journal of Aerospace Engineering*, 7(3): 276–296, 1994.

[20] O. Hasançebi, F. Erbatur, "On efficient use of simulated annealing in complex structural optimization problems", *Acta Mechanica*, 157: 27–50, 2002.