Multiobjective optimization of trusses using Backtracking Search Algorithm

A. Tchvagha Zeine[1,2,*], N. El hami[2], S. Ouhimmou[4], R. Ellaia[3], A. El hami[1]

[1]LOFIMS, Institut National de Sciences Appliques Rouen-France
BP. 08, Avenue de l'Universite, 76800 Saint Etienne de Rouvray, France, *nouneahmed@gmail.com
[2]LERMA, Mohammadia School of Engineers Rabat-Morocco
BP. 765, Ibn Sina avenue, Agdal, Rabat, Morocco
[3]ENSA-Kenitra National School of Applied Science of Kenitra-Morocco
[4]Laboratory of Mechanics, University Hassan II Casablanca, Morocco

ABSTRACT. In engineering, under complex nonlinear constraints, the majority of design problems are generally multi-objective. For multi-objective problems, the computing effort can often rise significantly through the number of objectives and constraints evaluations. Metaheuristics are nowadays seen as powerful algorithms to deal with multiobjective optimization problems. In this paper, we using Backtracking Search Algorithm (BSA) as a tool to solve multiobjective optimization problems in structures, named (BSAMO). A well known benchmark multi-objective problem has been chosen from the literature to demonstrate the validity of the proposed method, applicability of the method for structural problems has been tested through a truss problem and promising results were obtained. The numerical results show not only the effectiveness and the best performance of BSAMO compared to NSGA-II, but also its rapidity and efficiency.

KEYWORDS. Multiobjective optimization; Backtracking Search; Structural optimization; Truss structures; Design optimization

1. Introduction

In a single objective optimization engineering design problem, the optimal solution can be clearly defined for example in reliability engineering [LOP 11, LOP 13]. However, most of the engineering design problems involve multiple and often conflicting design objectives. In multi-objective optimization engineering design process is usually to optimize different objectives simultaneously, that are in conflict with each other in most cases. Multi-objective optimization (MO)can be formulated as a minimization problem as follows:

$$\begin{cases} \min\limits_{\mathbf{x}\in\Omega} \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}))^{\mathrm{T}} \\ \text{Subject to} \\ \qquad g_i(x) \le 0, \ (i = 1, \cdots, l) \\ \qquad h_j(x) = 0, \ (j = 1, \cdots, k), \end{cases} \tag{1}$$

where m is the number of objective functions, $\mathbf{x} = (x_1, \cdots, x_n) \in \Omega$ 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 $i = 1, \cdots, n_x$. $g_i(x)$ $(i = 1, \cdots, l)$ are l inequality constraints and $h_j(x)$ $(j = 1, ..., k)$ are k equality constraints.

That fact, the ideal solution is usually located outside the possible design space, so the aim is to get a set of solutions representing the best of objectives also called Pareto front. In mathematics, if a solution cannot be improved with respect to the goal without infecting at least the quality of the other objectives it is called Pareto solution. Also in the multi-objective optimization, Pareto front designates the space formed by Pareto solutions in objective space. To analyze tradeoffs in real life, policymakers are usually able to analyze a limited number of solutions. Therefore, they require a sufficiently uniformly distributed set of solutions which are representative of the entire Pareto front. In One hand, it is possible to meet the challenges that are generating a well spread all difficult solutions such as discontinuities in the Pareto frontier, the non-uniform density achievable solutions, non-convexity and nonlinear objective functions and constraints. In the scientific literature, Multiobjective Optimization Problems (MOPs) are typically very difficult to solve. Although with all these difficulties, multiobjective optimization has many effective algorithms with many successful applications [ZHO 11, MOH 06, TAL 09, ABD 14]. Recently, a significant amount of multi-objectives evolutionary algorithms (MOEAs) have been used to solve (MOPs) in science and engineering in a single runs. The best known state-of-the-art include (MOEAs) SPEA2 [ZIT 02], PAES [COR 2000] and NSGA-II [DEB 02]. On the other hand, many review articles adopted the MOEAs to solve engineering design problems, see for example [FAR 02, COE 05, MEH 07, GON 09, YAN 13].

In this paper, we used Backtracking Search Algorithm for solving the Multiobjective Design Optimization (BSAMO) which is a new way to extend Backtracking Search Algorithm (BSA) [CIV 13] to be suitable for solving engineering design problems. This paper is organized as follows: In section 2 we will introduce the (BSA). The section 3, is focused in the problem formulation of MOPs. In section 4 we will propose a BSAMO method which solves the MOPs introduced in section 3. The simulations of BSAMO is demonstrated in section 5 and will be compared with various multi-objective optimization algorithms. In the last section we will apply this algorithm to solve design optimization problems in engineering and we will finish by giving a short conclusion.

2. Backtracking Search Algorithm

Backtracking Search Algorithm (BSA) [CIV 13] is an evolutionary algorithm recently developed by Civicioglu. According to the author, BSA showed good convergence performance compared to other algorithms and needs only one control parameter. As described in [CIV 13], BSA includes five evolutionary mechanisms: initialization, selection-I, mutation [5], crossover [1], and selection-II. Hereafter, we give a detailed description of evolutionary mechanism used in the original version of BSA:

2.1. *Initialization*

To generate an initial population, BSA algorithm uses a uniform random distribution function. The individuals of the initial population are built based on Equation [2]:

$$\mathrm{P} \sim \mathrm{rand}(\mathrm{x}_{li}, \mathrm{x}_{ui}), \qquad [2]$$

2.2. *Selection I*

This first selection step aims to choose the historical population P_h for determining the search direction. In the same way as for the initial population, the individuals of the historical population are redefined beginning of each iteration and computed following the equation [3]:

$$\text{if } \mathrm{a} < \mathrm{b}|_{\mathrm{a,b}\sim\mathrm{rand}(0,1)} \text{ then } \mathrm{P}_h \leftarrow \mathrm{P} \qquad [3]$$

The order of the values of this historical population P_h is changed randomly according to the permuting i.e. a random shuffling function in the 'if/then' equation [4]:

$$\mathrm{P}_h \leftarrow \mathrm{permutting}(\mathrm{P}_h) \tag*{[4]}$$

2.3. *Mutation operator*

The initial from of the trial population is generated by the mutation operator of BSA using the following equation [5]:

$$\mathrm{P}_m = \mathrm{P} + \mathrm{F} \times (\mathrm{P}_h - \mathrm{P}) \tag*{[5]}$$

where F is a parameter controlling the amplitude of the search direction matrix computed as the difference between the historical and the current populations matrix $(\mathrm{P}_h - \mathrm{P})$. In this paper, we use the value $\mathrm{F} = \alpha * \mathrm{N}(0, 1)$, where $\alpha = 3$ and N is the normal distribution.

2.4. *Crossover Operator*

From the initial trial population built using the mutation process, the crossover mechanism of BSA creates the final form of the trial population P_c. Thus, the crossover operator works on the P_m population. Based on their fitness values, the best individuals in the trial population are selected to guide the search through the target population individuals. The crossover operator of BSA combines two main parts. First, a binary integer-valued $n_\mathrm{P} \times n_x$ matrix called '**map**' is computed. This matrix determines the individuals of the trial population P_c to be mixed with relevant individuals of P_m. To update the population of T with $\mathrm{P}_c = \mathrm{P}_m$ if $\mathbf{map}_{i,j} = 1, \{i, j\} \in \{1, \cdots, n_\mathrm{P}\} \times \{1, \cdots, n_x\}$. The second step of the BSA's crossover strategy consists in repairing some individuals of the trial population generated in the first step and which overflow the allowed search-sapce. Those individuals are thus regenerated so that they keep in the allowed search-space. Algorithm 1 presents the BSA's crossover process:

Algorithm 1 Crossover

```
Input : $\mathrm{P}_m$,$n_\mathrm{P}$ , N and $n_x$.
Output : Crossover population: $\mathrm{P}_m$
$m_{1:n_\mathrm{P},1:n_x} = 0$
if a $\leq$ b then
    $a, b \sim rand$
    for each particle $i = 1, \cdots, n_\mathrm{P}$ do
        $\mathbf{map}_{i,1:u(1:rnd*n_x)} = 1 \mid u = \mathrm{permeting}_{(1,\cdots,n_x)}$
    end for
else
    for each particle $i = 1, \cdots, n_\mathrm{P}$ do
        $\mathbf{map}_{i,randi(n_x)} = 1$
    end for
end if
$\mathrm{P}_c = \mathrm{P}_m$
$\mathrm{P}_c = \mathrm{P}_m * \mathbf{map} + P* \sim \mathbf{map}$
```

2.5. *Selection II*

The BSA algorithm performs second selection step from the trial population generated during the crossover step. Thus the individuals from the trial population having better fitness than some individuals in the old population are used to updated this last one. Similarly, if the so far obtained global minimum is has worst fitness than best individual, BSA uses this the best individual as new global minimum in the next iteration.

3. The proposed algorithm

In this approach we used , method of weighted sum for solving multi-objectives optimization problems. It has to combine all multiple objectives into a composite single objective

$$\min \mathbf{f} = \sum_{i=1}^{m} w_i f_i, \qquad \sum_{i=1}^{m} w_i = 1, \qquad [6]$$

where m the numbers of functions objectives and w_i, $(i = 1, \cdots, m)$ are non-negative weights. Then, the weights w_i can be calculated by random, which can be drawn from a uniform distribution for Pareto front.

4. Design optimization problems

In this section we will solve four design optimization problems taken from [GON 09, YAN 13] use our approach BSAMO algorithm. To test the efficiency a applicability of the algorithm for multiobjective design optimization, the selected this problem.

4.1. *I-Beam optimization*

This example has been proposed by [ERF 13] as is illustrated on the figure 1. We want to minimize the weight of the beam and to maximize its displacement.

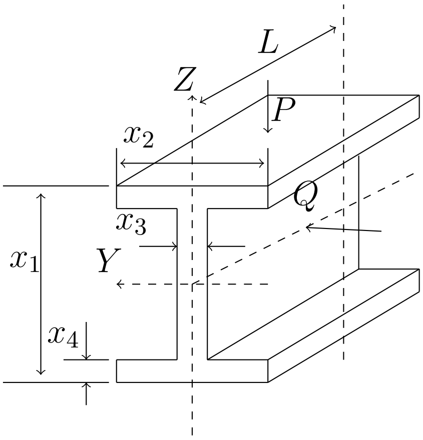


Figure 1. *I-Beam optimization*

The mathematical definition of the problem is:

$$[H]F : \begin{cases} \min\limits_{\mathbf{x}} \ f_1(\mathbf{x}) = 2\mathbf{x}_1\mathbf{x}_2 + \mathbf{x}_3(\mathbf{x}_1 - 2\mathbf{x}_4) \\ \max\limits_{\mathbf{x}} \ f_2(\mathbf{x}) = \frac{\mathbf{PL}^3}{48EI} \\ \text{Subject to} \quad g(\mathbf{x}) = \frac{\mathrm{M}_y}{\mathrm{W}_y} + \frac{\mathrm{M}_z}{\mathrm{W}_z} - \mathrm{k}_g \geq 0, \\ \qquad\qquad 10 \leq \mathbf{x}_1 \leq 80 \\ \qquad\qquad 10 \leq \mathbf{x}_2 \leq 50 \\ \qquad\qquad 0.9 \leq \mathbf{x}_3 \leq 5 \\ \qquad\qquad 0.9 \leq \mathbf{x}_4 \leq 5 \end{cases} \qquad [7]$$

$$\mathrm{W}_y = \left[\frac{2\mathbf{x}_2\mathbf{x}_4(4\mathbf{x}_4^2 + 3\mathbf{x}_1(\mathbf{x}_1 - 2\mathbf{x}_4)) + \mathbf{x}_3(\mathbf{x}_1 - \mathbf{x}_4)^3}{6\mathbf{x}_1}\right]$$

$$\mathrm{W}_z = \frac{2\mathbf{x}_4\mathbf{x}_2^3 + \mathbf{x}_3^3(\mathbf{x}_1 - \mathbf{x}_4)}{6\mathbf{x}_2}$$

$$\mathrm{I} = \left[\frac{2\mathbf{x}_2\mathbf{x}_4(4\mathbf{x}_4^2 + 3\mathbf{x}_1(\mathbf{x}_1 - 2\mathbf{x}_4)) + \mathbf{x}_3(\mathbf{x}_1 - 2\mathbf{x}_4)^3}{12}\right]$$

In equation 7, the function f_1 represents the weight of the beam, and the function f_2 represents the maximum displacement of bream. The values of the deterministic parameters are given in Table 1. In this table, k_g and E designate the stress yield point and the Young's modulus of the material, respectively, and M_y and M_z designate maximum bending moments of a beam around the x and y axes respectively.

Table 1. *The parameter values*

Var	P(KN)	Q(KN)	E(KN/CM^2)	k_g(KN/CM^2)	L(CM)	M_y(KN.CM)	M_z(KN.CM)
Valeur	600	50	2×10^4	16	200	30000	2500

The results shown in Figure 2, shows the Pareto-optimal front obtained by the approaches that we compare with BSAMO and NSGA-II algorithms, after 100 iteration, we use a population size of 100 points respectively. The Pareto front obtained by this algorithms is shown in right Figure 2, the non-dominated solutions obtained by BSAMO with a good distribution original Pareto front (PF) and good spread compared with results obtained by algorithm NSGA-II shown in right) Figure 2.

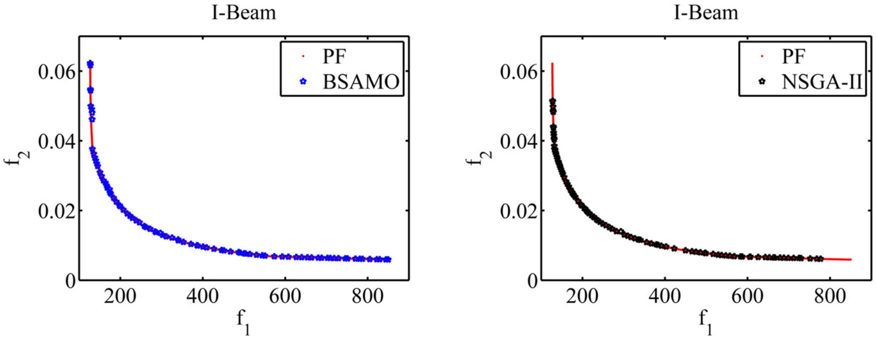


Figure 2. *Pareto front for BSAMO (left) and NSGA-II (right) the I-Beam*

4.2. *Two bars truss design*

This case study, using a two-bar truss design problem, which has been well studied by many researchers [COL 05] and [GON 09]. The problem is shown in Figure 5.

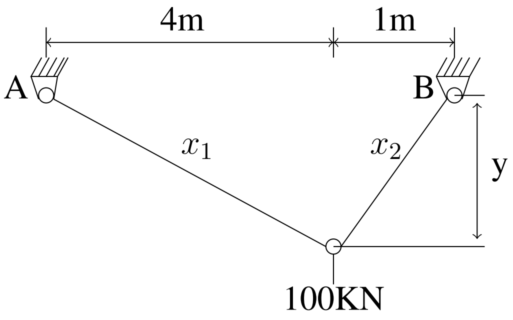


Figure 3. *The two bars truss design*

The mathematical description of the problem is as follows:

$$\begin{cases} \min\limits_{x} \left(f_1(x), f_2(x)\right) \\ \text{such that} \\ f_1(x) = x_1\sqrt{16+x_3^2} + x_2\sqrt{1+x_3^2} \\ f_2(x) = 20\sqrt{16+x_3^2}/(x_1 y) \\ \text{Subject to} \\ f_1 \leq 0.1,\ \ f_1 \leq 10^5 \\ f_{\text{stress,BC}} = 80\sqrt{1+x_3^2}/(x_2 y) \leq 10^5 \\ \text{where } 0 \leq x_1, x_2, and\ 1 \leq y \leq 3 \end{cases} \qquad [8]$$

The resultant Pareto optimal front curve is shown on figure 4 obtained after 100 iterations and the population size is set as 100 points. Figure 4 shows also the convergence efficiency of BSAMO and the uniformly distributed solutions on the Pareto front after 100 point non-dominated. However, we note that the optimization method that we developed offers better convergence to PF compared to results obtained by NSGA-II.

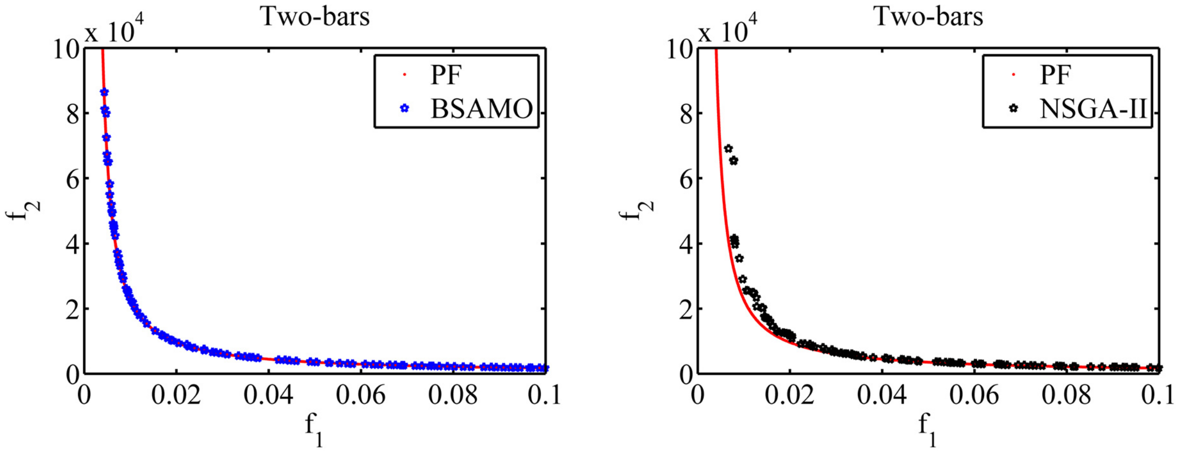


Figure 4. *Pareto front for BSAMO (left) and NSGA-II (right) the two bars truss design*

4.3. *Four bars*

As an example, the multi-objective optimization problem of four bars taken from [COE 05] and [ALE 12] shown in Figure 5. The objective is to find an optimal structure of the four bars simultaneously minimizing the total mass and displacement of node C. We consider the following functions:

- $min(f_1)$:The total volume
- $min(f_2)$:The displacement

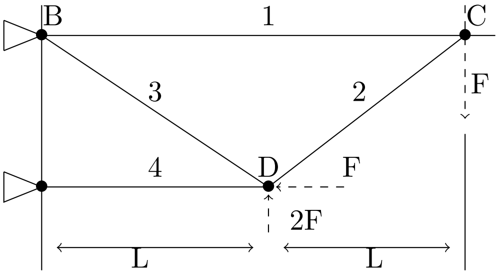


Figure 5. *Four bars*

The mathematic formulations in this functions followings:

$$\begin{cases} \min\limits_{x} \left(f_1(x), f_2(x)\right) \\ \text{such as} \\ f_1(x) = \mathrm{L}(2x_1 + \sqrt{2}x_2 + \sqrt{x_3} + x_4) \\ f_2(x) = \frac{\mathrm{FL}}{\mathrm{E}}(\frac{2}{x_1} - \frac{2\sqrt{2}}{x_2} + \frac{2}{x_4}) \\ \text{Subject to} \\ \frac{\mathrm{F}}{\sigma} \leq x_1, x_4 \leq 3\frac{F}{\sigma} \\ \sqrt{2}\frac{\mathrm{F}}{\sigma} \leq x_2, x_3 \leq 3\frac{F}{\sigma} \end{cases} \qquad [9]$$

The resultant Pareto optimal front curve in figure 6 obtained after 250 iterations and the population size is set as 50 points. In figure 6, demonstrate the convergence efficiency of BSAMO, we note that the uniformly distributed solutions on the Pareto front after 50 and 100 point non-dominated. However, we note that the optimization method that we developed offers better convergence to PF.

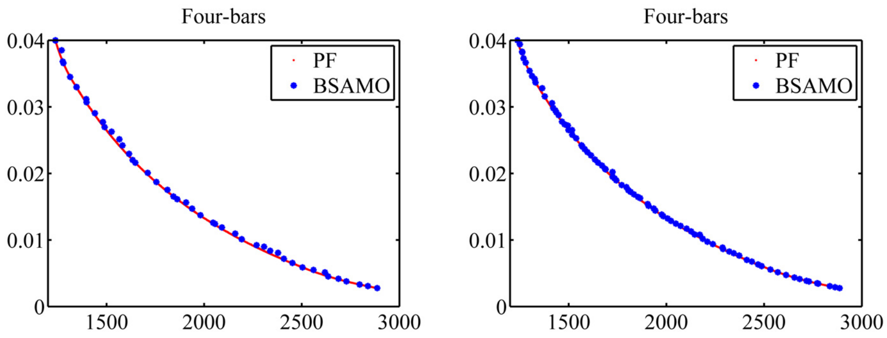


Figure 6. *Pareto front of the Four Bars*

4.4. *Application*

We are going to analyze the Pylon of a line of electricity transportation that one assimilates to a truss plan. Two identical loads F of 1.8 KN are applied to the two superior extremities of the following pylon with angle equals to $\theta = 15$. The bars forming the Pylon are in steel of which Young's modulus $\mathrm{E} = 210e^9$ Pa and the Poisson coefficient $\gamma = 0.29$. The section of every bar is worth $\mathrm{A} \in [20\ \mathrm{cm}^2, 30\ \mathrm{cm}^2]$. The hypothesis for this problem is that the weight of each bar of the Pylon is negligible in front of the applied efforts (see figure 7).

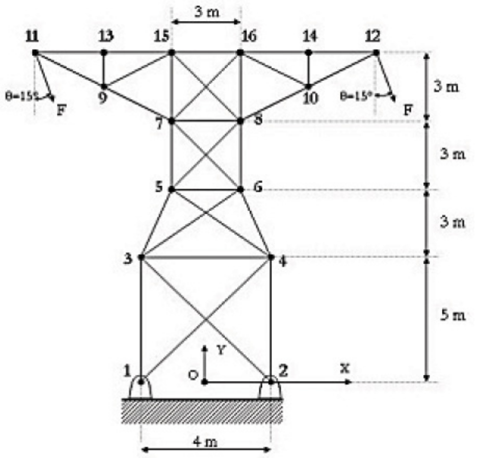
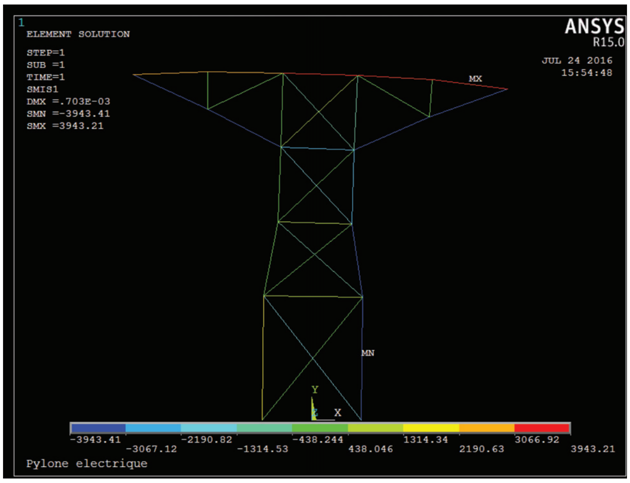


Figure 7. *The Pylon*

For the structural modelling, and static analysis in which the FEA software Ansys is used to calculate the structural responses. Theses responses are the displacement of node number 12 (maximum displacement see Figure 8) and the volume of the Pylon. In this study we are interesting to optimize both the maximum displacement and the volume of Pylon under reliability constraint (the probability of failure of the Pylon).



```
1
ELEMENT SOLUTION
STEP=1
SUB =1
TIME=1
SMIS1
DMX =.703E-03
SMN =-3943.41
SMX =3943.21
```

Figure 8. *The maximum displacement for the Pylon*

In this study, two objective functions have to be minimized:

- Volume of the Pylon.

- The maximum displacement.

Subject to the probability of failure of this Pylon is less than 0.406. This probability of failure is calculated by Monte Carlo method in Ansys.

$$\begin{cases} \min\left(\text{Volume}, \text{Displecement}\right) \\ \text{Subject to} \quad \mathrm{P}_f \leq 0.406 \end{cases} \qquad [10]$$

The Figure 9 shows the Pareto front obtained by BSAMO algorithm for the problem of the Pylon. The population size is 7 and the number of iterations is 10. It can be seen from Figure 9, that the good distribution on the Pareto front.

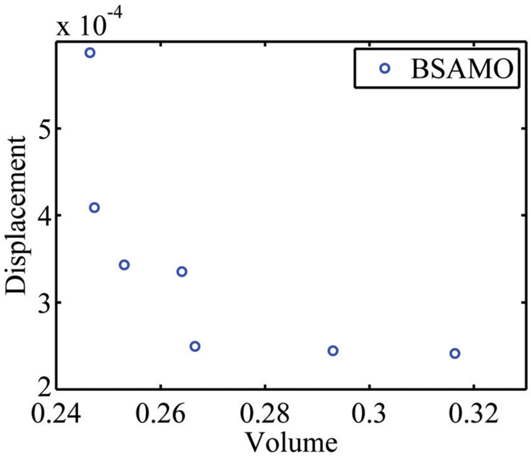


Figure 9. *The Pareto front for the Pylon*

5. Conclusions

In this paper, we propose a new algorithm based on Backtracking Search Algorithm (BSA) for solving multiobjective optimization problems in structures. The performance of this approach is evaluated for four benchmark functions of multiobjective problems, experimental results show our algorithm finds optimal solutions better than other algorithms. Although all tested examples are bi-objective problems, it is demonstrated that the proposed method is general and efficient to obtain uniformly distributed Pareto optimum points, and which show the robustness of the our algorithm. According to the results, BSAMO involves smaller populations, less generations and attains better efficiency and good convergence near to the Pareto-optimal front.

Bibliography

[LOP 11] Lopeza R. H., Lemossea D., de Cursia J. E. S., Rojasb J., El-Hami A., An approach for the reliability based design optimization of laminated composites, Engineering Optimization, 43, 1079–1094, 2011.

[LOP 13] Lopeza R. H., Lemossea D., de Cursi J. E. S., Rojasb J., El-Hami A., Iterative projection on critical states for reliability-based design optimization, Engineering Optimization, 45, 577–590, 2013.

[ZHO 11] Zhoua A. , Qub H. B., Zhao S., Suganthan P., Zhangd Q., Multiobjective evolutionary algorithms: A survey of the state of the art, Swarm and Evolutionary Computation, 1, 32–49, 2011.

[MOH 06] MOHSINE A., KHARMANDA G., EL-HAMI A., Improved hybrid method as a robust tool for reliability- based design optimization, Structural and Multidisciplinary Optimization, 32, 203–213, 2006.

[TAL 09] TALBI E., Metaheuristics from design to implentation, John Wiley & Sons, 2009.

[ABD 14] ABDESSALEM A. B., EL-HAMI A., Global sensitivity analysis and multi-objective optimisation of loading path in tube hydroforming process based on metamodelling techniques, International Journal of Advanced Manufacturing Technology, 71, 753–773, 2014.

[ZIT 02] ZITZLER E., LAUMANNS M., THIELE L., Spea2: improving the strength pareto evolutionary algo- rithm,a in evolutionary methods for design, Optimization and Control with Applications to Industrial Problems, Athens, Greece, 95–100, 2002.

[COR 2000] CORNE D. W., KNOWLES J. D., Paes: Approximating the nondominated front using the pareto archived evolution strategy, Evolutionary Computation. 8(2), 149172, 2000.

[DEB 02] DEB K., PRATAP A., AGARWAL S., MEYARIVAN T., A fast and elitist multiobjective genetic algorithm: Nsga-ii, IEEE Transactions on Evolutionary Computation, 6(2), 182âĂŞ-197, 2002.

[FAR 02] FARHANG-MEHR A., AZARM S., Entropy-based multi-objective genetic algorithm for design opti- mization, Structural and Multidisciplinary Optimization, 24, 351–361, 2002.

[COE 05] COELLO C., PULIDO G., Multiobjective structural optimization using a microgenetic algorithm, Structural and Multidisciplinary Optimization, 30, 388–403, 2005.

[MEH 07] MEHR1 F., AZARM A., Hybridization of genetic algorithm with immune system for optimization problems in structural engineering, Structural and Multidisciplinary Optimization, 24, 415–429, 2007.

[GON 09] GONG W., CAI Z., ZHU L., An effective multiobjective differential evolution algorithm for engineering design, Structural and Multidisciplinary Optimization, 38, 137–157, 2009.

[YAN 13] YANG X., DEB S., Multiobjective cuckoo search for design optimization, Computers and Operations Research, 40, 1616–1624, 2013.

[CIV 13] CIVICIOGLU P., Backtracking search optimization algorithm for numerical optimization problems, Applied Mathematics and Computation 219(15), 8121–8144, April 2013.

[ERF 13] ERFANI T., UTYUZHNIKOV S. V., KOLO B., A modified directed search domain algorithm for mul- tiobjective engineering and design optimization, Structural and Multidisciplinary Optimization, 48(6), 1129–1141, 2013.

[ALE 12] ALEM W. E., HAMI A. E., ELLAIA R., Pareto-optimal solutions for a truss problem, advanced materials research, 423, 53–64, 2012.