Discrete sizing/layout/topology optimization of truss structures with an advanced Jaya algorithm

Discrete sizing/layout/topology optimization of truss structures with an advanced Jaya algorithm

S. O. Degertekin*[1], L. Lamberti[2], I. B. Ugur[1]

[1] *Department of Civil Engineering, Dicle University, 21280, Diyarbakır, Turkey*
[2] *Dipartimento di Meccanica, Matematica e Management, Politecnico di Bari, 70126, Bari, Italy*

Abstract. Discrete optimization of truss structures is a hard computing problem with many local minima. Metaheuristic algorithms are naturally suited for discrete optimization problems as they do not require gradient information. A recently developed method called Jaya algorithm (JA) has proven itself very efficient in continuous engineering problems.Remarkably, JA has a very simple formulation and does not utilize algorithm-specific parameters. This study presents a novel JA formulationfor discrete optimization of truss structures under stress and displacement constraints. The new algorithm, denoted as discrete advanced JA (DAJA), implements efficient search mechanisms for generating new trial designs including discrete sizing, layout and topology optimization variables. Besides the JA's basic concept of moving towards the best design of the population and moving away from the worst design, DAJA tries to form a set of descent directions in the neighborhood of each candidate designs thus generating high quality trial designs that are very likely to improve current population. Results collected in seven benchmark problems clearly demonstrate the superiority of DAJAover other state-of-the-art metaheuristic algorithms and multi-stage continuous-discrete optimization formulations.

Keywords: Metaheuristic algorithms; Discrete optimization of truss structures; Sizing/layout/topology variables; Descent directions and approximate line search.

1. Introduction

In discrete structural optimization, trial designs are generated by selecting values of design variables from available sets of discrete values. However, if there are many optimization variables and many available discrete values for each variable, the design problem becomes a hard computing problem with many local minima. Metaheuristic algorithms [1] are naturally suited for discrete optimization in view of their inherent ability to perform global search without needing gradient information.

Metaheuristic algorithms likegenetic algorithms (GA) [2,3], simulated annealing (SA)[4,5], particle swarm optimization (PSO) [6], artificial bee colony(ABC) [7], differential evolution (DE) [8,9], ant colony optimization (ACO) [10,11], harmony search (HS) [12], adaptive step-size search (SASS)[13], big bang-big crunch optimization(BBBC) [14], firefly algorithm (FA) [15], teaching-learning-based optimization (TLBO) [16], mine blast algorithm (MBA)[17], swallow swarm optimization(SSO) [18], backtracking search optimization algorithm (BSA) [19], grey wolf optimizer (GWO) [20], symbiotic organisms search (SOS) [21],colliding bodies optimization (CBO)[22] and water evaporation optimization (WEO) [23] have been successfully applied to many engineering optimization problems.

Truss design is the most classical benchmark in structural optimization[24,25]. Implementation is definitely more challenging in the case of discrete problems that usually entail a very complex design space with multiple local optima. Just to limit the overview to the last 10 years, we can mention the study of Lee et al. [26] that used HS for discrete sizing optimization of four trusses with 25, 47, 52 and 72 bars: HS was found to be more efficient than GA variants. Li et al. [27] developed a heuristic particle swarm optimization (HPSO) method for discrete sizing optimization, which outperformed several PSO variants. A hybrid particle swarm ant colony optimization (DHPSACO) for discrete sizing optimization was developed by Kaveh and Talatahari [28] and successfully tested on truss structures with up to 582 elements: DHPSACO obtained better designs than GA, HS and PSO. A discrete BBBC algorithm for optimal design of skeletal structures was proposed by Kaveh and Talatahari [29] and the same authors later developed a charged system

search (CSS) with a fly to boundary method for discrete optimum design of truss structures [30].Sonmez [31] developed an artificial bee colony algorithm (ABC), very competitive with GA, HS and PSO in terms of optimized weight but slower than other methods.

Sadollah et al. [17] proposed the mine blast algorithm (MBA), which was very robust and obtained better designs than other optimization methods. Kaveh and his collaborators[32,33] utilized colliding bodies optimization (CBO) and enhanced colliding bodies optimization (ECBO) algorithms, finding them competitive with GA, HS, HPSO, DHPSACO, ABC, BBBC, ICA (imperialist competitive algorithm, [34]) and IRO (improved ray optimization, [35]). The self-adaptive step-size search (SASS) algorithm originally proposed by Nolle [13] was improved by Azad and Hasançebi [36] by introducing an elitist self-adaptive step-size search (ESASS) strategy: ESASS obtained better results than other metaheuristic methods such as GA and ABC. Dede [37] implemented an efficient teaching-learning-based-optimization (TLBO) algorithm for discrete optimization of truss structures with up to 72 bars. Baghlani et al. [38] developed an accelerated firefly algorithm (AFA) which converged to optimum within less structural analyses than standard FA. Sadollah et al. [39] compared water cycle (WCA), mine blast (MBA) and improved mine blast (IMBA) algorithms: all of these algorithms were very competitive with the most advanced metaheuristic optimization techniques.

The hybrid harmony search algorithm (HHS) developed by Cheng et al. [40] successfully combined harmony memory and pitch adjustment of HS, the global-best particle PSO and neighborhood search. Huu et al. [41] developed an adaptive elitist differential evolution (aeDE) for discrete optimization, which outperformed standard DE and other methods in many test cases.KazemzadehAzad [42] demonstrated that seeding the initial population with feasible solutions can improve the computational efficiency of metaheuristic algorithms for structural optimization.KazemzadehAzad [43] also developed hybrid algorithms by integrating a design oriented strategy into the stochastic search properties of adaptive dimensional search, modified BBBC and exponential BBBC.

Finding global optimum at low computational cost regardless of problem type and with limited sensitivity to initial population and internal parameters are critical issues in metaheuristic optimization. Computational cost can be minimized only if the exploration and exploitation phases are properly balanced. However, this is a very complicated task, which strongly depends on the algorithm formulation as well as on the problem at hand.The Jaya algorithm (JA) proposed by Rao [44] is an interesting metaheuristic method with a simple formulation and a powerful search engine that tries to move away from low quality solutions and direct search towards high quality solutions. The efficiency of JA was demonstrated in mathematical optimization problems [44], mechanical design problems [45], multi-objective optimization [46], and continuous optimization of truss structures [47]. In general, JA shows a very satisfactory performance compared to other metaheuristic algorithms and multi-objective evolutionary methods. The continuous optimization engine developed in [47] was then extendedto discrete problems by rounding continuous optimal solutionsto discrete values available from problem statement.

The present study presents a fully discrete advanced JA formulation (DAJA) for truss design problems. The optimization problem is now solved in a single loop without the need of carrying out the preliminary continuous optimization done in [47]. Explicit gradient or pseudo-gradient information are utilized together with approximate line search in order to speed up the generation of discrete designs and reduce the number of structural analyses required in the optimization process. Basically,a set of descent directions is generated in the neighborhood of each candidate design to increase the probability of improving population in each design cycle. This simplifies by a great extent the optimization process and eliminates biases towards local minima that may be originated from rounding continuous optimum solutions improperly.

Seven design examples including 10, 25, 47, 52, 72, 200 and 942-bar trusses are solved in order to verify the efficiency and robustness of the proposed DAJA formulation. The results obtained

byDAJA are compared with those reported in the literaturefor other state-of-art optimization methods.Thevalidity of the search engine developed in this study is fully demonstrated. The proposed approach is clearly superior over the multi-stage optimization strategy[47] and very competitive with state-of-the-art metaheuristic formulations.

The paperis structured as follows. Section 2 recallsthe general formulation of trussdesign problems. Section 3 describes in detail the DAJA algorithm. Section 4 presentstest problems and optimization results. Section 5 summarizes the main findings of this study.

2. Discrete design optimization of truss structures

Truss design optimization aims to minimize structural weight under design constraints on element stresses and nodal displacements. The optimization problem for a trussincluding*nm* elements (grouped in *ng* groups) and *nn* nodes can be stated as:

$$\text{Minimize} \quad W(\boldsymbol{A},\boldsymbol{X_{conf}},\boldsymbol{\Lambda}) = \gamma \sum_{i=1}^{nm} \lambda_i A_i \sqrt{(x_{i1}-x_{i2})^2+(y_{i1}-y_{i2})^2+(z_{i1}-z_{i2})^2} \quad (1)$$

Subjected to

$$\begin{aligned} &\sigma_i^c \le \sigma_i \le \sigma_i^t, && i=1,2,\ldots nm \\ &\delta_{\min} \le \delta_j \le \delta_{\max}, && j=1,2,\ldots..,nn \quad (2) \\ &A_{\min} \le A_k \le A_{\max}, && k=1,2,\ldots.,ng \\ &(x,y,z)_{\min} \le (x_{i1},x_{i2},y_{i1},y_{i2},z_{i1},z_{i2}) \le (x,y,z)_{\max} && i=1,2,\ldots,nm \end{aligned}$$

In Eq. (1), W is the weight of the truss structure and γ_i is the material density. In Eq. (2), $\sigma_i$is the stress developed in the ith element with the corresponding limits σ_i^c and σ_i^t in compression (including buckling strength) and tension; $\delta_j$represent the displacement components of the jth nodewith the corresponding limits $\delta_{\min}$ and $\delta_{\max}$.

The $\boldsymbol{A}$vector contains the sizing variables (i.e. cross-sectional areas of bars), selected from a list of discrete values. The $\mathbf{X_{conf}}$ vector contains the layout variables, which are the coordinates $x_{i1,2}$, $y_{i1,2}$, $z_{i1,2}$ of the nodes limiting the ith element of the structure. While sizing variables are always discrete, layout variables may be discrete or not. The $\boldsymbol{\Lambda}$ vector includes the topology variables λ_i that can take value 1 or 0, respectively, if the ith element (or element group) is keptin the design or removed from the truss. The optimization problem (1) has*NDV* design variables, including*ng* element cross-sectional areas taken as sizing variables and*NLAY* nodal coordinates taken as layout variables. In topology optimization, elements are removed from the truss if their cross-sectional area becomes very small (i.e., in [47], the area limit was set equal to10^{-7} in^2).

Here, stress and displacement constraints were handled by penalty functions. The penalized objective function $F_p$is defined as:

$$F_p = W(\boldsymbol{A},\boldsymbol{X_{conf}},\boldsymbol{\Lambda}) \times (1+\phi)^{\varepsilon} \quad (3)$$

The sum of stress and displacement penalties, ϕ, is defined as:

$$\phi = \sum_{i=1}^{nm} \phi_\sigma^i + \sum_{j=1}^{nn} \phi_\delta^j \quad (4)$$

Stress constraint penalty ϕ_σ^i for the *i-th*member and displacement constraint penalty ϕ_δ^j for the*j-th*node are respectively defined as:

$$\begin{cases} \phi_\sigma^{\mathrm{i}} = 0 & \text{if} \quad \sigma_{\mathrm{i}}^{\mathrm{c}} \le \sigma_{\mathrm{i}} \le \sigma_{\mathrm{i}}^{\mathrm{t}} \\ \phi_\sigma^{\mathrm{i}} = \dfrac{\left|\sigma_{\mathrm{i}} - \sigma^{\mathrm{t,c}}\right|}{\left|\sigma^{\mathrm{t,c}}\right|} & \text{if} \quad \sigma_{\mathrm{i}} < \sigma_{\mathrm{i}}^{\mathrm{c}} \text{ or } \sigma_{\mathrm{i}} > \sigma_{\mathrm{i}}^{\mathrm{t}} \end{cases} \tag{5}$$

$$\begin{cases} \phi_\delta^{\mathrm{j}} = 0 & \text{if} \quad \delta_{\min} \le \delta_{\mathrm{j}} \le \delta_{\max} \\ \phi_\delta^{\mathrm{j}} = \dfrac{\left|\delta_{\mathrm{j}} - \delta^{\max,\min}\right|}{\left|\delta^{\max,\min}\right|} & \text{if} \quad \delta_{\mathrm{j}} < \delta_{\min} \text{ or } \delta_{\mathrm{j}} > \delta_{\max} \end{cases} \tag{6}$$

The penalty function exponent ε varies with the number of iterations as $\varepsilon_{iter}=\varepsilon_o(1+it/it_{max})$.The initial value ε_o is chosen between 1.001 and 10000;*it* denotes the current iteration and it_{max} is the user-specified limit number of optimization iterations. This strategy allows to amplify the effect of penalty as the search process approaches constraint domain boundaries. Remarkably, JA convergence behavior was found to be insensitive to such a refinement (see, for example, [47]).

3. The discrete advanced Jaya (DAJA) algorithm for truss design optimization

3.1. *Classical JA formulation*

The JA algorithm is a population-based metaheuristic optimization method based on the idea that search process should always move towards the best design and move away from the worst design of the population. Unlike the vast majority of metaheuristic methods used for structural optimization, JA does not use algorithm-specific internal parameters but itneedsonly two standard control parameters such as population size and maximum number of iterations. This allows complicated tuning processes to be bypassed thus increasing robustness of optimization search.

Classical implementation of JA is very simple and includesjust one equation for generating new trial designs. Let us assume to have a population including *np* designs with*NDV* optimization variables. Let $X_{k,l,it}$be the value assigned to the k^{th} design variable of the l^{th} design stored in the populationat the it^{th} iteration. JA modifies the $X_{k,l,it}$ value as follows:

$$X_{\mathrm{k,l,it}}^{\mathrm{new}} = X_{\mathrm{k,l,it}} + r_{1,\mathrm{k,it}}\left(X_{\mathrm{k,best,it}} - \left|X_{\mathrm{k,l,it}}\right|\right) - r_{2,\mathrm{k,it}}\left(X_{\mathrm{k,worst,it}} - \left|X_{\mathrm{k,l,it}}\right|\right) \begin{cases} k = 1,\ldots,NDV \\ l = 1,\ldots,np \end{cases} \tag{7}$$

In Eq. (7), $X_{\mathrm{k,l,it}}^{\mathrm{new}}$ is the new value generated for the optimization variable $X_{\mathrm{k,l,it}}$; $r_{1,k,it}$ and $r_{2,k,it}$ are two randomly generated real numbers in the range [0,1] for the k^{th}optimization variable at the it^{th} iteration; $X_{\mathrm{k,best,it}}$ is the value assigned to the k^{th}optimization variable of the current best design$\boldsymbol{X}_{\mathrm{best,it}}$; $X_{\mathrm{k,worst,it}}$ is the value assigned to the k$^{\text{th}}$optimization variable of the worst design$\boldsymbol{X}_{\mathrm{worst,it}}$.

The term $r_{1,\mathrm{k,it}}\left(X_{\mathrm{k,best,it}} - \left|X_{\mathrm{k,l,it}}\right|\right)$ indicates the tendency ofsearch process to approach the best design, while the term $-r_{2,\mathrm{k,it}}\left(X_{\mathrm{k,worst,it}} - \left|X_{\mathrm{k,l,it}}\right|\right)$ indicates the tendency ofsearch process to move away from the worst design. Random factorsr_1 and r_2 ensure a good exploration of search space. Taking the absolute value of candidate solution ($|X_{\mathrm{k,l,it}}$) in Eq. (7) further enhances JA's exploration ability [47].

Once all variables of the l^{th} design stored in the population are updated with Eq. (7), the new trial design $\boldsymbol{X}_l^{new}$ thus generated is compared with the corresponding design $\boldsymbol{X}_l^{pre}$ stored in the previous iteration. If $\boldsymbol{X}_l^{new}$ is better than $\boldsymbol{X}_l^{pre}$, population is updated by replacing $\boldsymbol{X}_l^{pre}$ with $\boldsymbol{X}_l^{new}$. This process is repeated until some convergence criterion is satisfied.

3.2. *Improved JA formulation for discrete optimization*

Unlike other metaheuristic algorithms naturally suited for discrete optimization such as, for example, GA, HS and ACO (just to mention a few), the variable updating equation (7) of classical JA can provide real values for design variables. However, the optimization problem includes discrete values. The most common approach adopted in metaheuristic optimizationis to round real values to the nearest discrete values available from problem statement or to introduce additional terms in the variable updating equations. The latter strategy has the purpose of reducing not too much the number of available combinations of optimization variables. However, the abovementioned approaches do not necessarily improve population each time a new trial design is generated. For this reason, the following strategy has been implemented in the discrete advanced JA (DAJA) algorithm developed in the present study. Let$\boldsymbol{X}_{\mathrm{l,it}}$ be the candidate design currently perturbed by DAJA.Also, let $X_{\mathrm{k,NDSC}}$ and$X_{\mathrm{k,2nd\text{-}NDSC}}$bethe two nearest available discrete values to $X_{\mathrm{k,l,it}}^{\mathrm{new}}$ such that$X_{\mathrm{k,NDSC}} \le X_{\mathrm{k,l,it}}^{\mathrm{new}} \le X_{\mathrm{k,2nd\text{-}NDSC}}$.

The new discrete value of the k^{th} variable of the l^{th} candidate design is selected as follows:

$$\begin{aligned}
&(X_{\mathrm{k,NDSC}} - X_{\mathrm{k,l,it}})\frac{\partial W(\boldsymbol{X})}{\partial x_k}\Big|_{\boldsymbol{X}=\boldsymbol{X}_{\mathrm{l,it}}} < 0 \quad \& (X_{\mathrm{k,2nd\text{-}NDSC}} - X_{\mathrm{k,l,it}})\frac{\partial W(\boldsymbol{X})}{\partial x_k}\Big|_{\boldsymbol{X}=\boldsymbol{X}_{\mathrm{l.it}}} > 0 \Rightarrow X_{\mathrm{k,l,it}}^{\mathrm{new}} = X_{\mathrm{k,NDSC}} \\
&(X_{\mathrm{k,2nd\text{-}NDSC}} - X_{\mathrm{k,l,it}})\frac{\partial W(\boldsymbol{X})}{\partial x_k}\Big|_{\boldsymbol{X}=\boldsymbol{X}_{\mathrm{l.it}}} < 0 \ \& \quad (X_{\mathrm{k,NDSC}} - X_{\mathrm{k,l,it}})\frac{\partial W(\boldsymbol{X})}{\partial x_k}\Big|_{\boldsymbol{X}=\boldsymbol{X}_{\mathrm{l,it}}} > 0 \quad \Rightarrow X_{\mathrm{k,l,it}}^{\mathrm{new}} = X_{\mathrm{k,2nd\text{-}NDSC}}
\end{aligned} \tag{8}$$

where $\partial W/\partial x_k$ is the sensitivity of cost function with respect to the k^{th} design variable, evaluated at $\boldsymbol{X}_{\mathrm{l,it}}$. Basically,DAJA is forced to perturb optimization variables by moving along descent directions with respect to the candidate design $\boldsymbol{X}_l$ currently perturbed. Since for a descent direction $\boldsymbol{S}$ originating from a generic point $\boldsymbol{X}$ of design space it holds $\Delta W = \boldsymbol{S}^T \nabla W(\boldsymbol{X}) < 0$ and the total variation of cost function $\Delta W$is$\Sigma_{\mathrm{k=1,NDV}}\ \Delta X_{\mathrm{i}}(\partial \mathrm{W}/\partial \mathrm{x_k})$, cost function reductionswill be more significant if variations relative to all design variablesare forced to be negative.

If both inequalities $(X_{\mathrm{k,NDSC}} - X_{\mathrm{k,l,it}})\frac{\partial W(\boldsymbol{X})}{\partial x_k}\Big|_{\boldsymbol{X}=\boldsymbol{X}_{\mathrm{l,it}}} < 0$ and $(X_{\mathrm{k,2nd\text{-}NDSC}} - X_{\mathrm{k,l,it}})\frac{\partial W(\boldsymbol{X})}{\partial x_k}\Big|_{\boldsymbol{X}=\boldsymbol{X}_{\mathrm{l.it}}} < 0$ are satisfied, both discrete values $X_{\mathrm{k,NDSC}}$ and $X_{\mathrm{k,2nd\text{-}NDSC}}$ available for rounding the k^{th} design variable x_k may potentially improve the candidate design $\boldsymbol{X}_l$. In this case, DAJA tries to maximize the improvement in cost by taking the largest perturbation of design variable. Therefore, it sets:

$$\begin{cases}
\text{If } |X_{\mathrm{k,NDSC}} - X_{\mathrm{k,l,it}}| > |X_{\mathrm{k,2nd\text{-}NDSC}} - X_{\mathrm{k,l,it}}| \Rightarrow X_{\mathrm{k,l,it}}^{\mathrm{new}} = X_{\mathrm{k,NDSC,}} \\
\text{If } |X_{\mathrm{k,2nd\text{-}NDSC}} - X_{\mathrm{k,l,it}}| > |X_{\mathrm{k,NDSC}} - X_{\mathrm{k,l,it}}| \Rightarrow X_{\mathrm{k,l,it}}^{\mathrm{new}} = X_{\mathrm{k,2nd\text{-}NDSC.}}
\end{cases} \tag{9}$$

Strictly speaking, gradients are not defined for discrete optimization problems. However, in truss design optimization, cost function gradients with respect to sizing variables are constant over the whole design space whilethey are explicitly available for layout variables. Hence, Eq. (8) can always be used and it is computationally cheap. The same argument holds true for any optimization problemincludingan explicit cost function.

In the general case where gradients were not available,the goal of each optimization cyclehowever remains to improve the current population. In this regard, it should be noted thatthe

vector $(\boldsymbol{X}_{best,it}-\boldsymbol{X}_{l,it})$ defines a descent direction with respect to the candidate design $\boldsymbol{X}_{l,it}$ because $W(\boldsymbol{X}_{l,it})>W(\boldsymbol{X}_{best,it})$. Hence, considering the cost function gradient $\nabla W(\boldsymbol{X}_{l,it})$ evaluated at $\boldsymbol{X}_{l,it}$, it holds $(\boldsymbol{X}_{best,it}-\boldsymbol{X}_{l,it})^T \nabla W(\boldsymbol{X}_{l,it})<0$. In order to increase the probability of improving $\boldsymbol{X}_{l,it}$, other descent directions originating from $\mathbf{X}_{l,it}$ should be generated: for example, the $(\boldsymbol{X}_{l,it}^{new}-\boldsymbol{X}_{l,it})$ direction yielding $(\boldsymbol{X}_{l,it}^{new}-\boldsymbol{X}_{l,it})^T \nabla W(\boldsymbol{X}_{l,it})<0$. By combining the two conditions $(\boldsymbol{X}_{best,it}-\boldsymbol{X}_{l,it})^T \nabla W(\boldsymbol{X}_{l,it})<0$ and $(\boldsymbol{X}_{l,it}^{new}-\boldsymbol{X}_{l,it})^T \nabla W(\boldsymbol{X}_{l,it})<0$, it appears logical to accept a new trial design $\boldsymbol{X}_{k,l,it}^{new}$ if it holds $(\boldsymbol{X}_{best,it}-\boldsymbol{X}_{l,it})^T(\boldsymbol{X}_{l,it}^{new}-\boldsymbol{X}_{l,it})>0$. In view of this, Eq. (8) can be rewritten as:

$$\begin{cases}(\boldsymbol{X}_{best,it}-\boldsymbol{X}_{l,it})^T(\boldsymbol{X}_{l,it}^{k,NDSC}-\boldsymbol{X}_{l,it})>0 \Rightarrow X_{k,l,it}^{new}=X_{k,NDSC}\\ (\boldsymbol{X}_{best,it}-\boldsymbol{X}_{l,it})^T(\boldsymbol{X}_{l,it}^{k,2nd\text{-}NDSC}-\boldsymbol{X}_{l,it})>0 \Rightarrow X_{k,l,it}^{new}=X_{k,2nd\text{-}NDSC}\end{cases} \quad (10)$$

where the two design vectors $\boldsymbol{X}_{l,it}^{k,2nd-NDSC}(x_{1,l,it},x_{2,l,it},\ldots,X_{k,2nd\text{-}NDSC},\ldots,x_{NDV-1,l,it},x_{NDV,l,it})$ and $\boldsymbol{X}_{l,it}^{k,NDSC}(x_{1,l,it},x_{2,l,it},\ldots,X_{k,NDSC},\ldots,x_{NDV-1,l,it},x_{NDV,l,it})$ include the two nearest discrete values to $X_{k,l,it}^{new}$.

If both inequalities in Eq. (10) are satisfied, both discrete values $X_{k,NDSC}$ and $X_{k,2nd\text{-}NDSC}$ may potentially improve the candidate design $\boldsymbol{X}_{l,it}$. Hence, $X_{k,l,it}^{new}$ is rounded so as to take the largest perturbation steps along descent directions thus maximizing improvements in cost function. That is,

$$\begin{cases}\text{If } \|\boldsymbol{X}_{l,it}^{k,NDSC}-\boldsymbol{X}_{l,it}\|>\|\boldsymbol{X}_{l,it}^{k,2nd\text{-}NDSC}-\boldsymbol{X}_{l,it}\| \Rightarrow X_{k,l,it}^{new}=X_{k,NDSC}\\ \text{If } \|\boldsymbol{X}_{l,it}^{k,2nd\text{-}NDSC}-\boldsymbol{X}_{l,it}\|>\|\boldsymbol{X}_{l,it}^{k,NDSC}-\boldsymbol{X}_{l,it}\| \Rightarrow X_{k,l,it}^{new}=X_{k,2nd\text{-}NDSC}.\end{cases} \quad (11)$$

Equation (8) is simplified if the currently perturbed k^{th} design variable is continuous. In particular, it follows:

$$\begin{aligned}&(X_{k,l,it}^{new}-X_{k,l,it})\frac{\partial W(\boldsymbol{X})}{\partial x_k}\Big|_{\boldsymbol{X}=\boldsymbol{X}_{l,it}}<0 \Rightarrow \text{Accept } X_{k,l,it}^{new} \quad (12)\\ &(X_{k,l,it}^{new}-X_{k,l,it})\frac{\partial W(\boldsymbol{X})}{\partial x_k}\Big|_{\boldsymbol{X}=\boldsymbol{X}_{l,it}}>0 \Rightarrow \text{Reset } X_{k,l,it}^{new} \text{ as } (X_{k,l,it}^{new})' = X_{k,l,it}^{new}+\eta_k(X_{k,l,it}^{new}-X_{k,l,it})\end{aligned}$$

where η_k is a random number in the [0,1] interval. Basically, if $X_{k,l,it}^{new}$ and $X_{k,l,it}$ limit a descent direction with respect to $\boldsymbol{X}_{l,it}$, the trial point $\boldsymbol{X}_{l,it}^{k,new}(x_{1,l,it},x_{2,l,it},\ldots,X_{k,l,it}^{new},x_{NDV-1,l,it},x_{NDV,l,it})$ may potentially improve the design. Conversely, DAJA attempts to transform the non-descent direction $(X_{k,l,it}^{new}-X_{k,l,it})$ in the descent direction $-(X_{k,l,it}^{new}-X_{k,l,it})$ by "mirroring" the trial point $\boldsymbol{X}_{l,it}^{k,new}$ about $\boldsymbol{X}_{l,it}$. The random number η_k limits movements to make search remain in the feasible design space.

Once a candidate design is updated with Eqs. (7-11), cost function W^{new} is preliminary evaluated for the new design $\boldsymbol{X}_l^{new}$. Structural analysis and penalty evaluation are performed only if $W^{new}<W^{pre}$, that is if $\boldsymbol{X}_l^{new}$ can potentially improve the previous design $\boldsymbol{X}_l^{pre}$. The case $W^{new}>W^{pre}$ entails a new constraint evaluation only if $\boldsymbol{X}_l^{pre}$ is infeasible. Clearly, if $\boldsymbol{X}_l^{pre}$ is feasible and $\boldsymbol{X}_l^{new}$ is heavier than $\boldsymbol{X}_l^{pre}$ regardless of penalty terms, it is not logical to perturb design on the non-descent direction $(\boldsymbol{X}_l^{new}-\boldsymbol{X}_l^{pre})$. This strategy resulted very efficient in the case of continuous optimization [40] and allowed computational cost to be reduced on average by 30%. It should be noted that since DAJA tries to take the largest perturbations as possible along the descent directions generated in the neighborhood of candidate designs by properly selecting the available discrete values of optimization variables, it is increased the probability of generating trial designs for which it holds $W^{new}<W^{pre}$.

In summary, the standard JA formulation has been modified for discrete optimization and selection of discrete variables has been enhanced by including gradient information (if explicitly available) or pseudo-gradient information and approximate line search (general case). In this way, the two-step discrete optimization process described in [47] is no longer necessary and discrete optimization becomes affordable also if the discrete optimum design is formed by a large number of different discrete values of cross-sectional areas rather than by a few values. This turns very useful when the design space includes different types of variables that may have very different scales (e.g. sizing vs. layout variables). The risk of biasing search process towards designs including sub-optimally rounded values is greatly reduced. The superiority of the present approach in terms of computational speed will be demonstrated in the rest of the article.

3.3. *DAJA formulation for topology optimization*

An interesting issue is how DAJA deals with topology variables. Degertekin et al. [47]utilized a recursive approach based on the progressive reduction of the number of bars. A new continuous optimization run was performed each time somebarswere removed or some sizing/layout variables were rounded. Discrete values of layout variables were selected by evaluating cost function and constraints for both nearest discrete values to the correspondingoptimum value of the continuous solution. Elements were removed as soon as their cross-sectional areas $A_k$became smaller than 10^{-7} in^2. In the subsequent optimization cycles, A_k was made to vary between 0.999999999×10^{-7} and 1.000000001×10^{-7}, thus leavinglocal topology unvaried until the end of optimization process.Since structural weight will decrease as cross-sectional areas become smaller and the largest weight reductions can be achieved by removing bars, a metaheuristic formulation including gradient informationmay entail the risk of reducing structural weight too rapidly. Hence, constraint violation would increase to such an extent that new trial designs never improve population. This effect may become even more significantwhenmaterial is unnecessarily/prematurely removed from some region of the structure, thus biasing search process towards suboptimal designs.

In order to overcome the above mentioned limitations, in the present study, the discrete set of cross-sections was artificially expanded by adding the following values to the available set: $[10^{-10}A_{min}, 10^{-9}A_{min}, 10^{-8}A_{min}, 10^{-7}A_{min}, 10^{-6}A_{min}, 10^{-5}A_{min}, 10^{-4}A_{min}, 10^{-3}A_{min}, 10^{-2}A_{min}, 10^{-1}A_{min}]$. The topology variable $\lambda_i$defined for the i^{th} element (or element group) is set equal to 0 (element removal)only if DAJA selects the value $10^{-10}A_{min}$ for an element cross-sectional area. The transition from $A_{min}$to $10^{-10}A_{min}$ is rather smooth because DAJA always considers the two nearest discrete values to the current value assigned to a design variable. In this way, DAJA can automatically update sizing variables and eventually remove bars without shifting too much from feasible to infeasible regions in the neighborhood of promising solutions.

Interestingly, the smooth transition from A_{min} to $10^{-10}A_{min}$ adopted by DAJA for topology optimization may help avoidingthatW^{new}becomes smaller than W^{pre}just because of the removal of some bars regardless of the values taken by all other optimization variables.Sincea cross-sectional areais scaled by a factor 10 each time DAJA operates on smaller values than A_{min}, stress constraints arevery sensitive to design perturbations whenDAJA changes the topology of the structure. The ability of DAJA to improve design in spite of such a large variation in structural response confirms the validity of the proposed formulation.

3.4. *Flow chart and implementation steps of DAJA*

The flow chart of the discrete advanced JA formulation developed in this study to include gradient/pseudo-gradient information, approximate line search and reduced number of structural analyses is shown in Figure 1. The implementation steps of DAJA are summarized below.

<u>*Step 1*</u> Generate the initial population including *np* candidate designs$\boldsymbol{X}^{pre}$. For each discrete

variable, generate a random integer between 1 and the number of available discrete values for that variable. For continuous variables, use the following relationship:

$$X^{o}_{k,l} = X^{k}_{min} + rand\,(0,1)\,(X^{k}_{max} - X^{k}_{min}) \qquad k=1,2,\ldots,NDV; \qquad l=1,2,\ldots,np \qquad (13)$$

where $rand(0,1)$ is a random number uniformly generated in the [0,1] interval.
Set all topological variables equal to 1.
Compute cost function values $W^{pre}$and penalized objective function values $F_p{}^{pre}$for all designs ofthe population using Eqs. (1-6).
Initialize iteration counter as it=0 and setthe maximum iteration counter (it_{max}).

Step 2 Increase the iteration counter, $it=it+1$.

Step 3 Identify best design $\boldsymbol{X}_{best}$and worst design $\boldsymbol{X}_{worst}$of the population.

Step 4 Modify discreteoptimization variables with Eqs. (7-11); use Eqs. (7,12) for continuous variables. Try to maximize perturbation steps in order to make cost improvements the largest as possible. Generate np new trial designs$\boldsymbol{X}^{new}$. The generation of discrete trial designs is completely renewed respect to classical approaches followed in literature including multi-stage algorithms, which alternate continuous and discrete optimizations until all design variables become equal to the available discrete values.

Step 5 Compute objective function values$W^{new}$for all new trial designs$\boldsymbol{X}^{new}$generated in Step 4. Compare W^{new} with the corresponding valuesW^{pre}computed at the previous iteration.

Step 6 If $W_l{}^{new} < W_l{}^{pre}$, compute value of penalized cost function $F_{p,l}{}^{new}$ and compare it with its counterpart$F_{p,l}{}^{pre}$ computed for the l^{th} candidate design in the previous iteration.

Step 7 If $F_{p,l}{}^{new} < F_{p,l}{}^{pre}$, replace the l^{th} candidate design with the new design generated in Step 4.

Step 8 If $W_l{}^{new} > W_l{}^{pre}orF_{p,l}{}^{new} > F_{p,l}{}^{pre}$, leave the l^{th}candidate design unchanged.

Repeat Steps 6 through 8 for all candidate designs.

Step 9 If $it<it_{max}$, go to Step 2.
If $it>it_{max}$,store current best design $\boldsymbol{X}_{best}$ as optimum solution of the problem at hand.
Besides the maximum number of iterations set so as to have a computational budget of 20,000 structural analyses, DAJA checks the variation of cost function$W(\boldsymbol{X}_{best})$ throughout the optimization process. In all design examples, DAJAconverged asymptotically to the optimum solution and the standard deviation of design vectors included in the population rapidly dropped below 10^{-7}only 80-100 structural analyses after DAJA found theoptimum.

3.5. *Critical comparison of DAJA and other metaheuristic algorithms*

The description of DAJA formulation made in this section highlightssome important differences and clear advantages of the present approach over other metaheuristic algorithms.

The first important advantage of DAJA isits very simple formulation based on just one perturbation equation (Eq. (7));Eqs. (8-12) simply serve to transform continuous values into available discrete values. Second, DAJA requires only two standard parameters: population size and limit number of iterations. While the former parameter is common to all population-based algorithms, the latter parameter may significantly affect convergence behavior. However, high quality trial designs generated by DAJA throughout search process always allow the present algorithm to find global optimum much before than the limit number of iterations are completed.

Third, DAJA includes information on cost function gradients in the process of generating new trial designs while metaheuristic optimizers usually do not perform gradient evaluations.However, disposing of gradient information allows descent directions to be always selected at any stage of the optimization process thus increasing the probability of generating high quality designsin all iterations. This approach is very suited for optimization of skeletal structures where cost function gradients are explicitly available for sizing variables orthey can be evaluated at low computational cost for layout variables.

Fourth, metaheuristic algorithms usuallydefine a new trial design using a pseudo-random approach (i.e. reproducing some natural phenomenon) and then simply check if this design improves current population. This is usually done without taking care if the current best record is improved or not. Conversely, DAJA attempts to form high quality trial designs that always lie on descent directions. This may allow to immediately improve the current best record.

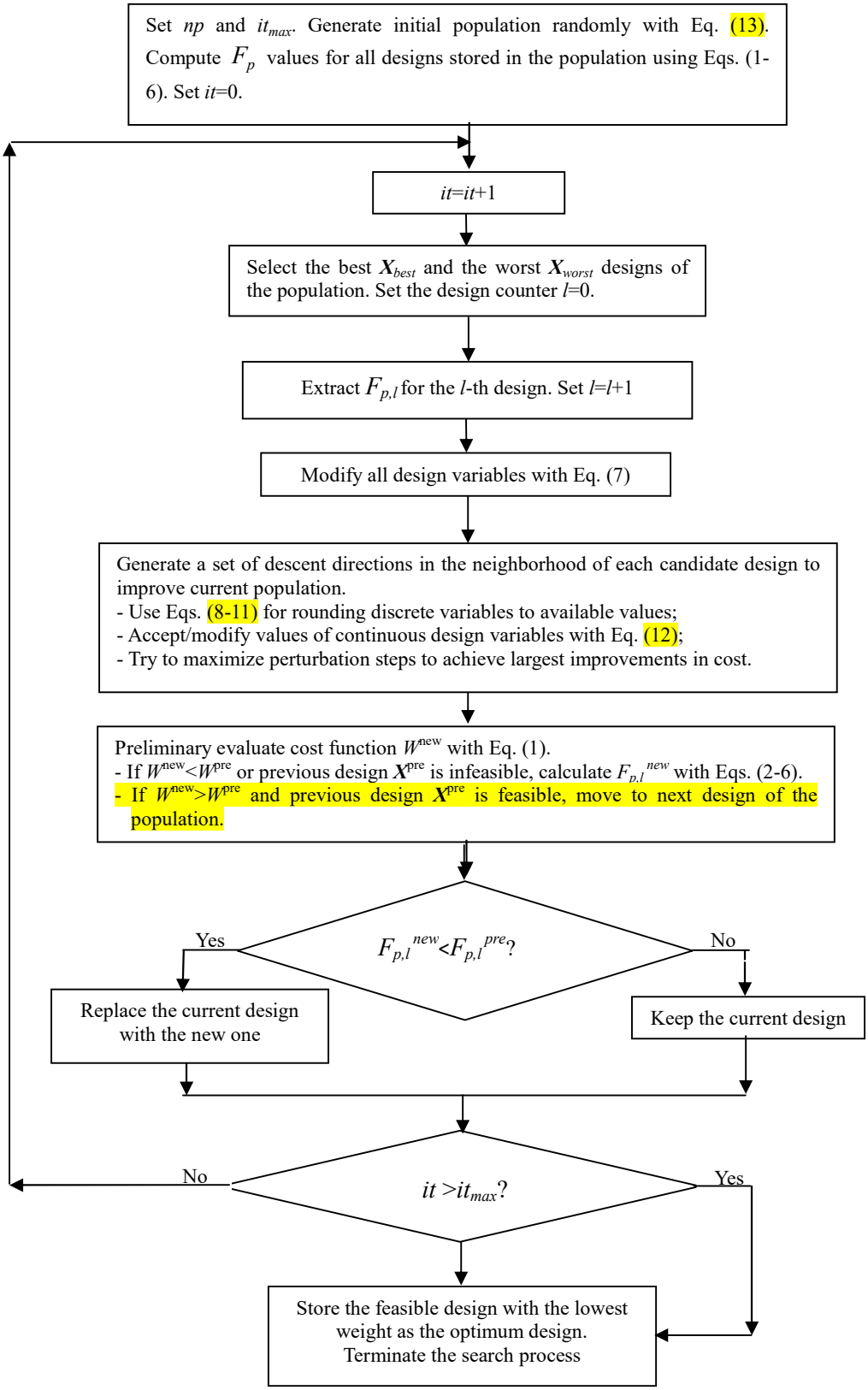


Fig. 1. Flowchart of the DAJA search process for discrete optimization of truss structures.

Fifth, some metaheuristic algorithms adopt an elitist strategy to survive the best individuals of the current iteration also in the next iteration. However, this entails additional structural analyses. In this regard, the upper bound strategy (UBS) implemented in [48] is very efficient because optimization constraints are not evaluated (thus saving structural analyses) if the structural weight computed for the trial design X_l^{new} is larger than the worst penalized weight included in the population. Basically, UBS rejects all trial designs that are certainly worse than the worst element currently included in the population of candidate designs. DAJA implements an even more severe criterion than UBS as the trial design X_l^{new} is immediately rejected if it is heavier than the corresponding design X_l^{pre} regardless that X_l^{pre} is the worst design of the population or not.

Sixth, metaheuristic algorithms usually include different mathematical models for updating optimization variables in the exploration and exploitation phases. Conversely, DAJA utilizes just a single equation – Eq. (7) – to perturb design variables and always uses gradient information to accept/reject/round updated values. Exploration and exploitation phases naturally alternate in DAJA based on the size of perturbation steps given to optimization variables. Movements ($X_{\mathrm{k,l,it}}^{\mathrm{new}} - X_{\mathrm{k,l,it}}$) become smaller in size as the distance between the best design $\boldsymbol{X}_{\mathrm{best,it}}$ and the worst design $\boldsymbol{X}_{\mathrm{worst,it}}$ included in the population decreases.

Last, DAJA adaptively handles search process based on the quality of available discrete valuesin the neighborhood of the current solution. Other metaheuristic algorithms may adaptively change their internal parameters to balance exploration and exploitation but some particular adaptation criterion may increase the amount of heuristics entailed by optimization process.

Table 1 summarizes the arguments made above comparing main characteristics of DAJA and other metaheuristic optimizers. It appears that DAJA's formulation is inherently superiorover other metaheuristic algorithms and does not have significant weaknesses.

Table 1. Critical comparison between DAJA and other metaheuristic algorithms.

Characteristics	DAJA	Other metaheuristic algorithms
Simplicity of formulation	Very simple	Sometimes very complicated formulations especially for hybrid algorithms.
Setting of control parameters	Not required	Often required. Convergence behavior highly sensitive to setting of internal parameters.
Gradient information	Used for generating high quality trial designs. Works on explicit gradients.	Not utilized
Trial design improves population?	High probability of generating high quality designs. Current best record can be improved in each iteration.	Best design ofpopulation not necessarily improved by new trial designs.
Elitism	Intrinsically elitist with no additional structural analyses.	Sometimes elitist. Additional structural analyses usually needed.
Exploration/exploitation	Naturally balanced. Just one mathematical model.	Not necessarily balanced. Two mathematical models.

Is the algorithm adaptive?	Adaptive search strategy. Movements given to variables depend on distance between $\boldsymbol{X}_{\text{best,it}}$ and $\boldsymbol{X}_{\text{worst,it}}$as well as on quality of available discrete values in the neighborhood of current solution.	Not necessarily adaptive

4. Test problems and optimization results

The DAJA algorithm developed in this research was testedinseven classical truss weight minimization problems including discrete sizing, layout and topology optimization variables. Following the classical approach adopted by structural optimization experts, DAJA results were compared with the best solutions available in literature in terms of optimized weight and number of structural analyses. Because of the random nature of DAJA, each design example was run twenty times starting from randomly generated initial populations. Result tables report the best design obtained in these independent runs along with the best weight, the average weight, the worst weight and the standard deviation (SD) on optimized weight. The number of structural analyses(NSA) and percent constraint violation (CV%) for the best design also are specified. The rate of success achieved by DAJA over the twenty independent optimization runs also is specifiedfor each design example, thus providing further information on the robustness of the proposed algorithm. Interestingly, this information is not very often available in structural optimization literature.

The test problems considered in this study were divided in three groups: (i) “small” scale problems with up to 16 sizing variables, 5 layout variables and 8 topology variables; (ii) “average” scale problems with up to 29 sizing variables, 17 layout variables and 27 topology variables; (iii) “large” scale problems with up to 59 sizing variables. The global optimum was available for the small scale problems (such a design could be obtained by many other metaheuristic algorithms documented in literature) and the main goal of DAJA’s search was hence to converge to the target solution within much less structural analyses than referenced algorithms. For the average/large scale problems, since no previously developed metaheuristic algorithm could find the global optimum because of the fairly large number of design variables, problem formulation or input data characteristics, the DAJA’s objective was twofold: to obtain a lower structural weight and reduce computational cost with respect to the best solutions available in the literature. If all of the aforementioned goals are accomplished for all test problems, DAJA should be regarded as a powerful method for discrete optimization of truss structures. This approach is commonly adopted in metaheuristic optimization to test the efficiency of a new algorithm.

DAJA was coded in the MATLAB programming language under the Windows environment. Optimization runs were performed on a standard personal computer with an Intel Core i7 processor and 4 GB RAM. The population size *np* was set equal to 20 for all test problems. For that purpose, a sensitivity analysis was performed by changing *np* from 10 to 1000.Sensitivity analysis results for two representative design examples, the 10-bar and 200-bar truss problems, are reported in Table 2. Similar results were obtained for the other test problems and are omitted for the sake of brevity. Table 3 presents the details ofstructural weight and number of structural analyses obtained from each independent run of the 10-bar and 72-bar truss problems (small scale examples with, respectively, 10 and 16 sizing variables), 200-bar truss problem (average scale example with 29 sizing variables), and 942-bar tower problem (large scale example with 59 sizing variables). Remarkably, DAJA resulted insensitive to initial population and population size: in fact, the ratio between standard deviation on optimized weight and average optimized weight ranged between 0% and 1.55% while the largest deviation on the number of structural analyses was about 30%.These results facilitated comparisons between DAJA and other metaheuristic algorithms that achieved their best performance by setting population sizes different from *np*=20.

Table 2 demonstrates clearly the superiority of DAJA over classical JA. In fact, since classical JA performs *np* analyses for each optimization iteration (see Section 3.1), its computational cost increases linearly with population size at a fixed number of iterations. Conversely, computational cost of DAJA increased at most by a factor 3.6 even though the largest population considered in sensitivity analysis was 100 times larger than the smallest one (i.e. 1000 vs. 10).

Table 2. Results of sensitivity analysis on the effect of population size.

Population size (ps)	10-bar truss Case-1		10-bar truss Case-2		200-bar truss	
	Weight (lb)	NSA	Weight (lb)	NSA	Weight (lb)	NSA
10	5498.37	1270	5093.30	1187	28,587.39	6000
20	**5490.74**	**687**	**5067.33**	**1022**	**27,282.54**	**4693**
30	5490.74	1293	5081.48	1022	28,131.93	7421
40	5491.72	1299	5092.02	1019	27,584.34	6057
50	5490.74	1324	5074.79	1352	27,794.01	7318
60	5498.20	1075	5082.24	1168	28,062.63	7899
70	5499.35	1272	5070.42	1409	28,435.44	8516
80	5504.16	1543	5085.33	1087	27,816.01	7545
90	5497.22	1420	5096.38	1486	27,650.36	6302
100	5504.16	1930	5095.87	1606	27,580.40	7481
250	5499.35	2438	5082.24	1758	28,510.57	7049
500	5502.52	2037	5093.30	2511	28,348.64	7496
1000	5504.16	2048	5079.16	2118	28,529.11	7865

Another important issue in comparing algorithms is the limit number of structural analyses that can be performed in the optimization process. This parameter may change significantly for population-based methods because it is commonly defined as the product between population size and limit number of iterations. Since DAJA found very soon the optimum design regardless of the size of the problem at hand, the effective computational budget exploited by DAJA was much smaller than for the referenced algorithms that continued to generate trial designs not able to improve the current best record until completing the limit number of structural analyses while DAJA rapidly converged 80-100 structural analyses after having found the optimum. This made the comparison with literature more challenging as DAJA was actually forced to search the optimum design over a small number of structural analyses.

Table 3. Detailed results of twenty independent optimization runs for some design examples.

Run No.	10-bar truss Case-1		10-bar truss Case-2		72-bar truss Case-1		72-bar truss Case-2		200-bar truss		942-bar tower	
	Weight (lb)	NSA	Weight (lb)	NSA	Weight (lb)	NSA	Weight (lb)	NSA	Weight (lb)	NSA	Weight (lb)	NSA
1	5490.74	980	5081.48	2087	**385.54**	**1873**	389.684	3619	28,096.93	6904	377,485.0	52,116
2	5490.74	964	5081.48	2401	386.95	1809	389.458	4247	27,875.64	8851	379,090.9	51,145
3	5491.72	1018	**5067.33**	**1022**	385.54	2166	389.458	3997	28,108.61	10,641	385,454.5	42,005
4	5490.74	887	5067.33	2081	385.54	1905	389.334	5164	27,846.24	8768	384,545.5	46,772
5	5490.74	688	5067.33	2158	387.94	2172	389.458	5149	27,491.48	9641	383,636.4	44,117

6	5490.74	908	5067.33	1245	385.54	2447	389.828	5232	**27,282.57**	**4693**	382,727.3	52,115
7	**5490.74**	**687**	5067.33	3739	387.94	1643	389.334	3475	28,101.21	10,783	380,909.1	50,023
8	5491.72	1383	5081.48	2156	385.54	1957	389.601	3971	27,786.14	6810	377,485.0	52,117
9	5490.74	1354	5081.48	1633	385.54	2087	389.458	4248	28,096.93	7248	382,727.3	45,967
10	5490.74	1578	5070.42	2893	385.54	2891	**389.334**	**3376**	28,096.93	4941	379,090.9	48,771
11	5491.72	1614	5067.33	1447	387.94	1817	389.334	3569	28,101.21	9049	378,181.8	44,781
12	5490.74	1491	5067.33	1693	385.54	1903	389.458	3721	28,108.61	10,117	377,485.0	54,253
13	5490.74	1446	5067.33	2105	385.54	2058	389.458	4019	28,096.93	7243	380,909.1	54,771
14	5490.74	998	5070.42	2986	385.54	2179	389.828	5437	27,786.14	5973	378,181.8	53,778
15	5491.72	1559	5067.33	1967	386.95	1795	389.458	4763	27,875.64	7985	**377,485.0**	**49,097**
16	5490.74	1219	5081.48	2429	385.54	1958	389.334	3412	28,096.93	5418	382,727.3	50,017
17	5490.74	1005	5067.33	1098	385.54	2346	389.334	4033	27,846.24	9009	379,090.9	55,411
18	5490.74	1314	5067.33	2388	387.94	2013	389.601	3599	28,096.93	6428	384,545.5	51,123
19	5491.72	1063	5067.33	2171	385.54	1971	389.458	4107	27,282.57	5102	380,909.1	53,771
20	5491.72	1177	5081.48	1552	385.54	2034	389.684	3383	27,491.48	10,089	380,909.1	54,781

4.1.*Planar 10-bar truss structure*

The first design example regards the 10-bar planar truss structure with 10 elements and 6 nodes shown in Figure 2. The material density and the modulus of elasticity are 0.1 lb/in^3 and 10 Msi, respectively. The allowable stress for all members is ±25,000psi for tension/compression. Displacementsof all freenodes are restricted to ±2.0 in. The loads applied to the truss structure are equal to P_1=100 kips and P_2=0.Each member is defined as a design variable, and hence there are ten sizing variables in the structure. This is a very classical test problem and was included in this study because the large amount of data available in the literature makes comparison very challenging.

Two optimization cases are considered. For Case 1, discrete variables are selected from the set D=[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, 33.50] (in^2). For Case 2, the following set is used D=[0.1, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5, 10.0, 10.5, 11.0, 11.5, 12.0, 12.5, 13.0, 13.5, 14.0, 14.5, 15.0, 15.5, 16.0, 16.5, 17.0, 17.5, 18.0, 18.5, 19.0, 19.5, 20.0, 20.5, 21.0, 21.5, 22.0, 22.5, 23.0, 23.5, 24.0, 24.5, 25.0, 25.5, 26.0, 26.5, 27.0, 27.5, 28.0, 28.5, 29.0, 29.5, 30.0, 30.5, 31.0, 31.5] (in^2).

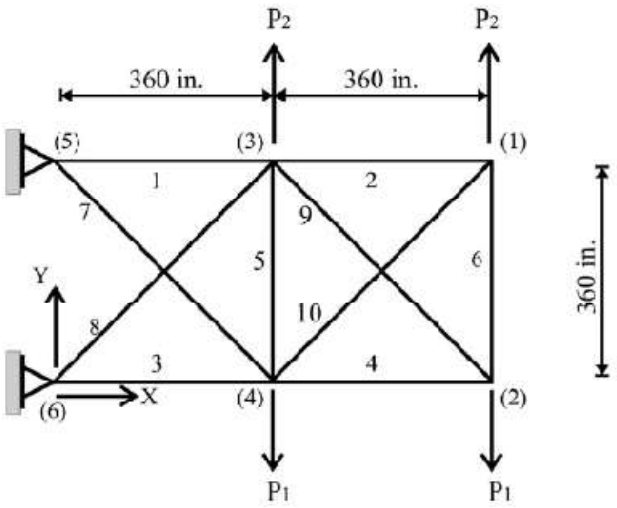


Fig. 2. Schematic of the planar 10-bar truss structure

The best solution quoted in the literature for Case 1 (see, for example, Refs. [31,37,40,41]) is {33.50; 1.62; 22.90; 14.20; 1.62; 1.62; 7.97; 22.90; 22.00; 1.62} in^2 yielding an optimum weight of 5490.74 lb. For Case 2, the available best solution (see, for example, Refs. [17,40]) is {29.5; 0.1; 24.0; 15.0; 0.1; 0.5; 7.5; 21.5; 21.5; 0.1} in^2 with anoptimum weight of 5067.33 lb. While the rather small number of design variables included in the 10-bar truss problem and the huge number of combinations available from the D sets (respectively, 10^{42} and 10^{64}, for Case 1 and Case 2) allowed many metaheuristic algorithms to converge to the global optimum, data on computational cost of the optimization were highly dispersed, ranging from 1000 to about 26,000 structural analyses for Case 1 and from about 2300 to 3000 structural analyses for Case 2. However, computational cost appears to be very large for a truss design problem including only 10 discrete sizing variables. It is hence very important to develop a new metaheuristic algorithm able to find the global optimum also minimizing the computational cost of the optimization process.

Tables 4 and 5 compare the results obtained byDAJA and other state-of-art optimization methods forCases 1 and 2, respectively. The present algorithm and most methods converged to feasible designs corresponding to the target weights of 5490.74 lb for Case 1 and 5067.33 lb for Case 2. The AFA [38] and HPSO [27]algorithms were the worst optimizers overall. In particular, AFA designed a lighter structure but the solution violated optimization constraints in both optimization cases. HPSO converged instead to the largest weight. The most important result is that theDAJA algorithm required much less structural analyses than other methods in both optimization cases: for Case 1, only 687 vs. 1000 to 50,000 analyses; for Case 2, only 1022 analyses vs. 2291 to 50,000 analysesrequired by the other optimizers. Hence, the main objective of the optimization was fully achieved for this problem.

Remarkably, DAJA was very robust as its standard deviation on optimized weight is at least one order of magnitude smallerthan for the other methods. The metaheuristic nature of the optimization search obviously resultedin nonzero standard deviation on structural weight. In particular, the rate of success of DAJA (i.e. the number of times DAJA reached the global optima out of 20 independent optimization runs conducted from different initial populations) was 70% for Case 1 and60% for Case 2. However, this rate increases to 100% by considering nearly optimal designs that are up to 0.28% heavier than the target global optimum.

The two-stage JA optimization process developed in [47] was applied to this test problem to check the improvement in computational efficiency brought by DAJA. The continuous optimum solutions found by two-stage JA [47] were consistent with the target weightsquoted in literature for

both Cases 1 and 2 (about 5060.9 lb) but two-stage JA required about 12,000 structural analyses to complete just the continuous optimization, i.e. between 12 and 17 times the total computational cost of DAJA to find the discrete optimum solution. For Case 1, the continuous optimum found by two-stage JA[47] is {30.4973; 0.1; 23.1693; 15.2573; 0.1; 0.569; 7.452; 20.9837; 21.5897; 0.1} in^2.Thisleads to the discrete optimum design {33.50; 1.62; 22.90; 15.50; 1.62; 1.62; 7.22; 22.90; 22.00;1.62} in^2, whichstill satisfies displacement and stress constraints but is slightly heavier than DAJA's optimum design (i.e. 5499.36 lb vs 5490.74 lb, see Table 3). For Case 2, the continuous optimum of two-stage JA is rounded to{30.5; 0.1; 23.5; 15.5; 0.1; 0.5; 8; 21.5; 22.0; 0.1} in^2, a feasible design yet slightly heavier than the best discrete design of DAJA (i.e. 5145.25 lb vs 5067.33 lb, see Table5). This is a further proof of the efficiency of DAJA, which can avoid to bias discrete search space should an optimization variable be assigned a discrete value not corresponding to the global optimum.

Table 4.Optimization results obtained for Case 1 of the 10-bar truss problem.

Design variables A_i (in^2)	HPSO [27]	ABC [31]	MBA [17]	TLBO [37]	AFA [38]	HHS [40]	aeDE [41]	**Present study DAJA**
A_1	30.00	33.50	30.00	33.50	33.50	33.50	33.50	**33.50**
A_2	1.62	1.62	1.62	1.62	1.62	1.62	1.62	**1.62**
A_3	22.90	22.90	22.90	22.90	22.90	22.90	22.90	**22.90**
A_4	13.50	14.20	16.90	14.20	13.90	14.20	14.20	**14.20**
A_5	1.62	1.62	1.62	1.62	1.62	1.62	1.62	**1.62**
A_6	1.62	1.62	1.62	1.62	1.62	1.62	1.62	**1.62**
A_7	7.97	7.97	7.97	7.97	7.97	7.97	7.97	**7.97**
A_8	26.50	22.90	22.90	22.90	22.90	22.90	22.90	**22.90**
A_9	22.00	22.00	22.90	22.00	22.00	22.00	22.00	**22.00**
A_{10}	1.80	1.62	1.62	1.62	1.62	1.62	1.62	**1.62**
Weight (lb)	5531.98	5490.74	5507.758	5490.74	5479.94	5490.74	5490.74	**5490.74**
CV (%)	None	None	None	None	0.1945	None	None	**None**
NSA	50000	25800	3600	1000	4250	3533	2380	**687**
Worst weight (lb)	N/A	5536.97	5536.965	N/A	N/A	N/A	5549.20	**5491.72**
Mean weight (lb)	N/A	5510.35	5527.296	N/A	N/A	5493.49	5502.62	**5491.03**
SD (lb)	N/A	N/A	11.38	N/A	N/A	10.46	20.78	**0.461**

Table 5.Optimization results obtained for Case 2 of the 10-bar truss problem.

Design variables A_i (in^2)	HPSO [27]	MBA [17]	AFA [38]	HHS [40]	**Present study DAJA**
A_1	31.50	29.5	31.00	29.50	**29.50**
A_2	0.10	0.1	0.10	0.10	**0.10**

A_3	24.50	24	23.00	24.00	**24.00**
A_4	15.50	15	15.00	15.00	**15.00**
A_5	0.10	0.1	0.10	0.10	**0.10**
A_6	0.50	0.5	0.50	0.50	**0.50**
A_7	7.50	7.5	7.50	7.50	**7.50**
A_8	20.50	21.5	21.00	21.50	**21.50**
A_9	20.50	21.5	21.50	21.50	**21.50**
A_{10}	0.10	0.1	0.10	0.10	**0.10**
Weight (lb)	5073.51	5067.33	5059.87	5067.33	**5067.33**
CV (%)	None	None	0.037	None	**None**
NSA	50000	3000	4750	2291	**1022**
Worst weight (lb)	N/A	N/A	N/A	N/A	**5081.48**
Mean weight (lb)	N/A	N/A	N/A	5068.36	**5071.88**
SD (lb)	N/A	N/A	N/A	N/A	**6.512**

The convergence curves recorded for the best optimization runs relative to Cases 1 and 2 are compared in Fig. 3. In Case 1 (see Fig. 3a), HHS [40] and aeDE [41] started from better initial populationsthan DAJA (i.e. their best individuals weighed between 640 and 900 lb less than the 8048 lb of the DAJA's best design), but the present algorithm soon generated much better intermediate solutions.This happened in spite of the hybrid/elitist strategies implemented by HHSand aeDE, thus confirming the advantages of including gradient/pseudo-gradient information and approximate line search in the generation of trial designs. MBA [17] and AFA [38] started their optimization runs from populations including more conservative designs than DAJA, HHS and aeDE and converged more slowly to the optimum. The same observationsmade for Case 1 hold true also for Case 2 (see Fig. 3b) even though DAJA required about twice more structural analyses than in the former case (i.e. 1022 vs. 687).

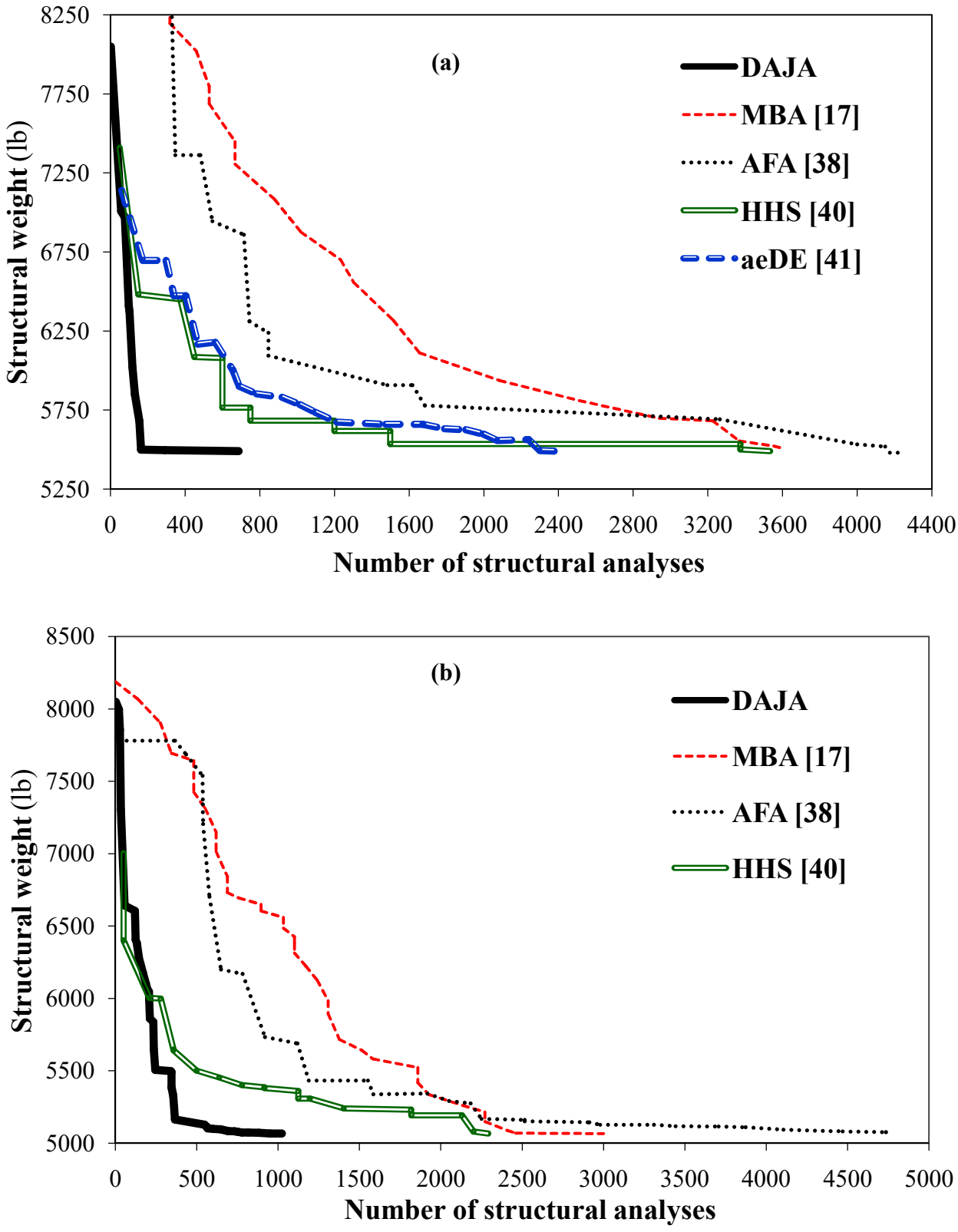


Fig. 3. Comparison of convergence curves for the 10-bar truss problem: (a) Case 1; (b) Case 2.

4.2.*Spatial 25-bar tower*

The second truss structure optimized in this study is the spatial 25-bar towershown in Fig. 4. Material properties are the same as in the first example. Elements are divided into eight independentgroups: (1) A_1, (2) A_2–A_5, (3) A_6–A_9, (4) A_{10}–A_{11}, (5) A_{12}–A_{13}, (6) A_{14}–A_{17}, (7) A_{18}–A_{21} and (8) A_{22}–A_{25}. The allowable stress is ±40,000 psi for tension/compression while nodal displacements must be less than ±0.35 in. This is another very classical design example widely analyzed in the optimization literature.

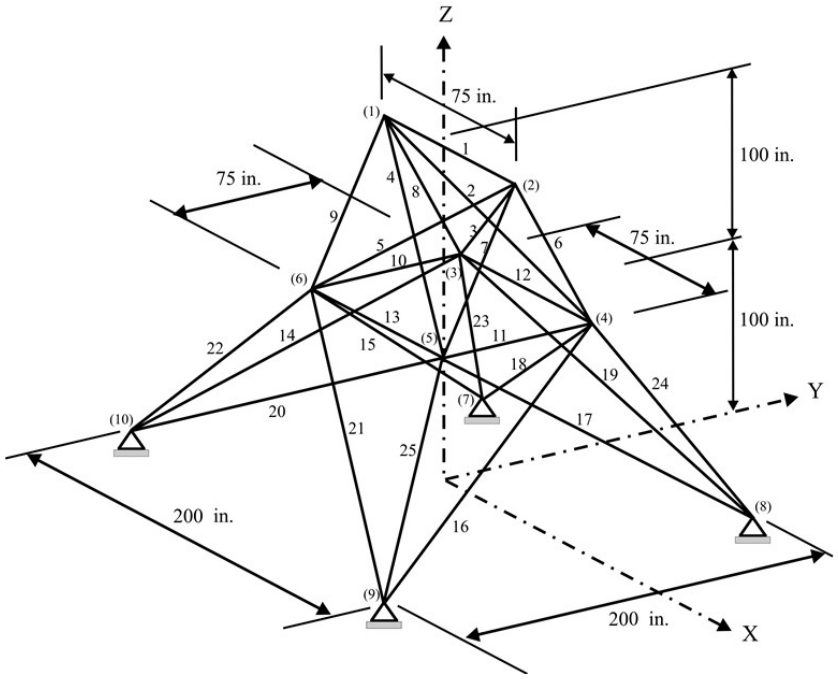


Fig. 4. Schematic of the spatial 25-bar truss tower

Three optimization cases including different available discrete sets of sizing variables and loading conditions (see Table 6) were considered. In Case 1, sizing variables are selected from the discrete set D=[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, 2.6, 2.8, 3.0, 3.2, 3.4] (in^2) and the truss is subject to the first loading condition listed in Table 5. In Case 2, discrete sizing variables are selected from the set D=[0.01, 0.4, 0.8, 1.2, 1.6, 2.0, 2.4, 2.8, 3.2, 3.6, 4.0, 4.4, 4.8, 5.2, 5.6, 6.0] (in^2). In Case 3, discrete sizing variables are selected from American Institute of Steel Construction specifications [49] using Table 7. Loading conditions 2 and 3 listed in Table 6 were used for optimization Cases 2 and 3.

The best solution quoted in the literature for Case 1 (see, for example, Refs. [17,27,31,33,35,37,38,40,41,50,51]) is {0.1; 0.3; 3.4; 0.1; 2.1; 1.0; 0.5; 3.4} in^2 yielding a structural weight of 484.85 lb. For Case 2, the available best solution (see, for example, Refs. [17,27,40]) is {0.01; 2.0; 3.6; 0.01; 0.01; 0.8; 1.6; 2.4} in^2 with a structural weight of 560.59 lb. For Case 3, the available best solution (see, for example, Refs. [17,27,28,38]) is {0.111; 2.130; 2.880; 0.111; 0.111; 0.766; 1.620; 2.620} in^2 with a structural weight of 551.14 lb.

Similar to the 10-bar truss problem, many algorithms were able to converge to the global optima of the three problem variants listed above. This happened becauseof the huge number of available combinations of discrete sizing variables from the D sets: respectively, $1.547 \cdot 10^{26}$ (i.e. 8^{29}), $2.815 \cdot 10^{14}$ (i.e. 8^{16}) and $6.277 \cdot 10^{57}$ (i.e. 8^{64}), for Case 1, Case 2 and Case 3. However, computational cost entailed by metaheuristic optimization was very high also for this problem, which includes only 8 sizing variables: between 2450 and 25,000 structural analyses for Case 1; between 950 and 25,000 structural analyses for Case 2; between 2400 and 25,000 structural

analyses for Case 3.

Table 6.Loading conditions acting on the spatial 25-bar tower.

Optimization cases	Loading conditions	Nodes	Forces (kips)		
			P_x	P_y	P_z
1	1	1	1.0	-10.0	-10.0
		2	0.0	-10.0	-10.0
		3	0.5	0.0	0.0
		6	0.6	0.0	0.0
2 and 3	2	1	0.0	20.0	-5.0
		2	0.0	-20.0	-5.0
	3	1	0.0	10.0	-5.0
		2	1.0	10.0	-5.0
		3	0.5	0.0	0.0
		6	0.5	0.0	0.0

Table 7.Available values of cross-sectional areas of bars taken from AISC specifications[49].

No.	Area (in^2)	No.	Area (in^2)	No.	Area (in^2)	No.	Area (in^2)
1	0.111	17	1.563	33	3.840	49	11.500
2	0.141	18	1.620	34	3.870	50	13.500
3	0.196	19	1.800	35	3.880	51	13.900
4	0.250	20	1.990	36	4.180	52	14.200
5	0.307	21	2.130	37	4.220	53	15.500
6	0.391	22	2.380	38	4.490	54	16.000
7	0.442	23	2.620	39	4.590	55	16.900
8	0.563	24	2.630	40	4.800	56	18.800
9	0.602	25	2.880	41	4.970	57	19.900
10	0.766	26	2.930	42	5.120	58	22.000
11	0.785	27	3.090	43	5.740	59	22.900
12	0.994	28	3.130	44	7.220	60	24.500
13	1.000	29	3.380	45	7.970	61	26.500
14	1.228	30	3.470	46	8.530	62	28.000
15	1.266	31	3.550	47	9.300	63	30.000
16	1.457	32	3.630	48	10.850	64	33.500

Tables 8through10compareDAJA optimization results with the literature. All methodsconverged to the same target optimum designs weighing 484.85 lb and 551.14 lb for Cases 1 and 3, respectively. In Case 2, JA found the same optimum design as the other methods except DHPSACO [28] and AFA [38]. However, the design found byDAJA satisfied constraints while

DHPSACO and AFA converged to the same infeasible solution.

It should be noted that the present algorithm always was the fastestoptimizer and required about half of the structural analyses required by the 2nd fastest algorithm:in Case 1, only 511 analyses vs. 1440 analyses of aeDE [41];in Case 2, only 530 analyses vs. 950 analyses of MBA [17]; in Case 3, only 946 analyses vs. 2400 analyses of MBA. The main objective of the optimization, to significantly reduce the computational cost of the metaheuristic search, was hence fully accomplished by DAJA also for this test problem.

Remarkably, DAJA always converged to the Case 1's target weight in all independent runs thus achieving a 100% rate of success in spite of the metaheuristic nature of the optimization search. However, standard deviation on DAJA's optimized weight was slightly larger than for MBA [17] (this algorithm also achieved 100% rate of success) and HHS [40] for Case 2,and MBA [17]forCase 3.The corresponding rate of success achieved by DAJA in these two cases decreased to about 80%. This may be due to the much higher convergence rate (i.e. about 3 times higher than for HHS in Case 2; about 2.5 times higher than for MBA in Case 3) exhibited by the proposed algorithm. A similar relationship between fast convergence and robustness was observed for the CBO and ECBO algorithms [28]: in fact, the latter was 3.5 times slower but achieved a better average solution.

The two-stage JA algorithm of Ref. [47] was applied to Case 2, the classical problem variant presented in the literature for the continuous optimization of the 25-bar truss. The discrete solution, {0.01; 2.0; 3.2; 0.01; 0.01; 0.8; 1.6; 2.8} in^2, obtained by rounding the continuous solution {0.01; 1.982567; 3.00; 0.010004; 0.010002; 0.684767; 1.677633; 2.659208} in^2 found by two-stage JA [47] is feasible but corresponds to a slightly heavier weight than DAJA (i.e. 564.86 lb vs 560.59 lb, see Table 9). Furthermore, the target weight of the two-stage JA continuous solution was foundafter about 8,000 structural analyses, i.e. about 16 times the total computational cost required by DAJA for finding the discrete optimum. This confirms the validity of the proposed approach.

Table 8. Optimization results obtained for Case 1 of the 25-bar truss problem (sizing variables only).

Design variables A_i (in^2)	HPSO [27]	ABC [31]	MBA [17]	CBO [33]	ECBO [33]	TLBO [37]	AFA [38]	HHS [40]	aeDE [41]	DBB-BC [52]	**Present study DAJA**
A_1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	**0.1**
A_2-A_5	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	**0.3**
A_6-A_9	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	**3.4**
A_{10}-A_{11}	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	**0.1**
A_{12}-A_{13}	2.1	2.1	2.1	2.1	2.1	2.1	2.1	2.1	2.1	2.1	**2.1**
A_{14}-A_{17}	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	**1.0**
A_{18}-A_{21}	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	**0.5**
A_{22}-A_{25}	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	3.4	**3.4**
Weight (lb)	484.85	484.85	484.85	484.85	484.85	484.85	484.85	484.85	484.85	484.85	**484.85**
CV (%)	None	None	None	None	None	None	None	None	None	None	**None**
NSA	25000	24250	2150	2040	7050	4000	7100	1739	1440	20000	**511**
Worst weight	N/A	485.05	485.048	N/A	N/A	N/A	N/A	N/A	486.1	N/A	**484.85**

(lb)											
Mean weight (lb)	N/A	484.94	484.885	486.87	485.89	N/A	N/A	484.95	485.01	N/A	**484.85**
SD (lb)	N/A	N/A	0.072	N/A	N/A	N/A	N/A	0.365	0.273	N/A	**0**

Table 9.Optimization results obtained for Case 2 of the 25-bar truss problem (sizing variables only).

Design variables A_i (in^2)	HPSO [27]	DHPSACO [28]	MBA [17]	AFA [38]	HHS [40]	**Present study DAJA**
A_1	0.01	0.01	0.01	0.01	0.01	**0.01**
A_2-A_5	2.00	1.60	2.00	1.60	2.00	**2.00**
A_6-A_9	3.60	3.20	3.60	3.20	3.60	**3.60**
A_{10}-A_{11}	0.01	0.01	0.01	0.01	0.01	**0.01**
A_{12}-A_{13}	0.01	0.01	0.01	0.01	0.01	**0.01**
A_{14}-A_{17}	0.80	0.80	0.80	0.80	0.80	**0.80**
A_{18}-A_{21}	1.60	2.00	1.60	2.00	1.60	**1.60**
A_{22}-A_{25}	2.40	2.40	2.40	2.40	2.40	**2.40**
Weight (lb)	560.59	551.61	560.59	551.61	560.59	**560.59**
CV (%)	None	0.1	None	0.1	None	**None**
NSA	25000	4550	950	6200	1400	**530**
Worst weight (lb)	N/A	N/A	560.59	N/A	N/A	**564.86**
Mean weight (lb)	N/A	N/A	560.59	N/A	560.785	**561.44**
SD (lb)	N/A	N/A	0	N/A	0.743	**1.88**

Table 10.Optimization results obtained for Case 3 of the 25-bar truss problem(sizing variables only).

Design variables A_i (in^2)	HPSO [27]	DHPSACO [28]	MBA [17]	AFA [38]	**Present study DAJA**
A_1	0.111	0.111	0.111	0.111	**0.111**
A_2-A_5	2.130	2.130	2.130	2.130	**2.130**
A_6-A_9	2.880	2.880	2.880	2.880	**2.880**
A_{10}-A_{11}	0.111	0.111	0.111	0.111	**0.111**
A_{12}-A_{13}	0.111	0.111	0.111	0.111	**0.111**
A_{14}-A_{17}	0.766	0.766	0.766	0.766	**0.766**

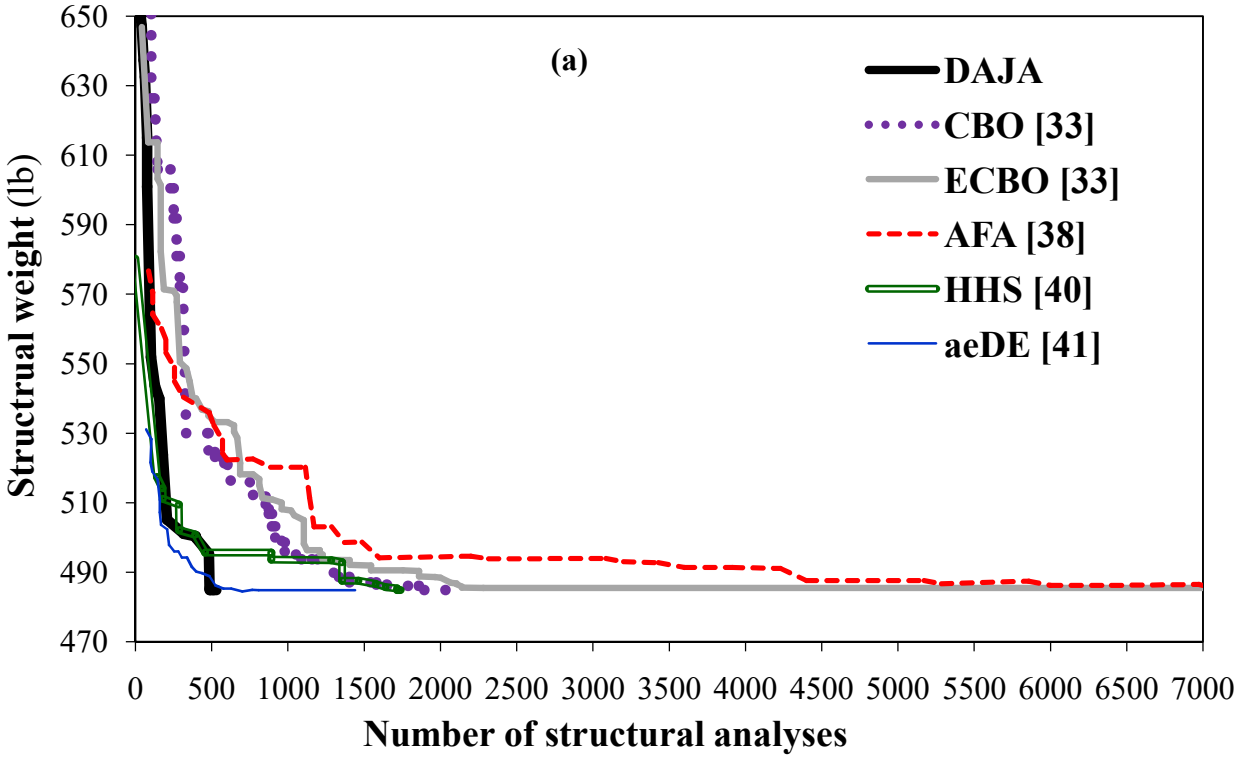
A_{18}-A_{21}	1.620	1.620	1.620	1.620	**1.620**
A_{22}-A_{25}	2.620	2.620	2.620	2.620	**2.620**
Weight (lb)	551.14	551.14	551.14	551.14	**551.14**
CV (%)	None	None	None	None	**None**
NSA	25000	4450	2400	9100	**946**
Worst weight (lb)	N/A	N/A	554.067	N/A	**554.51**
Mean weight (lb)	N/A	N/A	551.54	N/A	**551.81**
SD (lb)	N/A	N/A	0.987	N/A	**1.62**

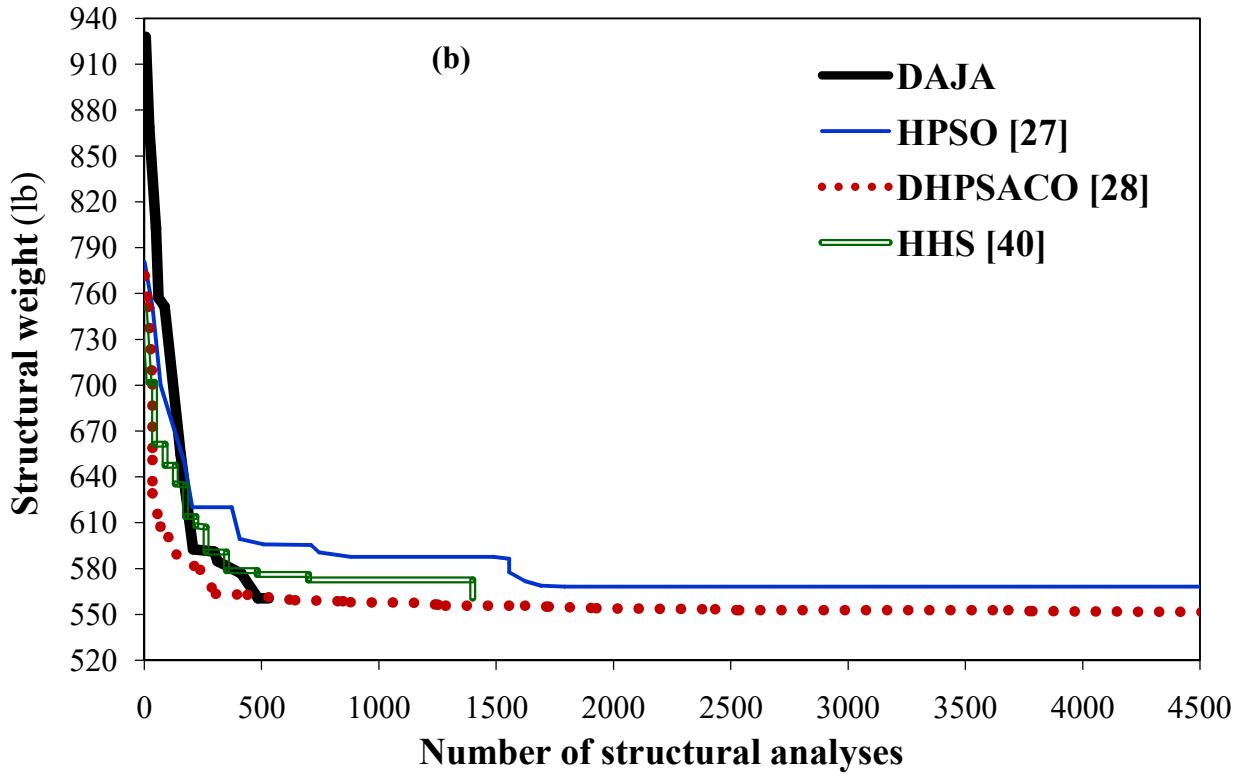
Figure 5 compares the convergence curves recorded in the three sizing optimization problemssolvedfor the 25-bar tower. Each curve shown in the figure refers to the best optimization run performed for the corresponding algorithm.For the sake of clarity, some plots include less structural analyses than the slowest optimizer.

In Case 1 (see Fig. 5a), DAJA started from a population including a best design which wasabout 130 lb heavier than its aeDE's counterpart [41]. Similar to the 10-bar truss example, the elitist strategy used by aeDE was less efficient than the approximate line searches utilized by DAJA. In fact, the present algorithm recovered the weight gap within about 450 structural analyses and immediately directed search towards the best region of design space while aaDE improved trial solutions by a smaller extent and required some 1000 additional structural analyses to complete the optimization process. AFA [38] and HHS [40] started their optimization runs from populations including about 80 lb lighter best individuals than DAJA. Harmony search is inherently more efficient than firefly algorithm and this explains why convergence curves of HHS and DAJA were very close until 450 structural analyses while AFA generated considerably heavier intermediate solutions than DAJA after only 150 structural analyses. CBO and ECBO [33] initial populations were similar to DAJA's one but their convergence curves soon diverged from DAJA's optimization history and their intermediate designs approached those of AFA [38]. Such a behavior was expected because ECBO and CBO include at most a weak elitist criterion, intrinsically less efficient than the search for descent directions carried out by DAJA.

In Case 2 (see Fig. 5b), DAJA started its search from a larger initial structural weight than HPSO [27], DHPSACO [28] and HHS [40]. While DHPSACO generated infeasible intermediate designs and finally converged to an infeasible solution, DAJA could find lighter intermediate designs than DHPSACO after about 480 analyses never violating constraints.Furthermore, the present algorithm required only 200 structural analyses to recover the initial gap from HPSO and HHS's best designs: HHS remained competitive with DAJA until 400 structural analyses while HPSO soon generated considerably higher intermediate designs than DAJA.

In Case 3 (see Fig. 5c), DAJA and MBA [17] started optimization process from the largest (about 900 lb) and smallest (700 lb) initial weight, respectively. Again, DAJA required only 450 structural analyses to recover the gap and generate better intermediate designs than MBA. HPSO [27], DHPSACO [28] and AFA [38] generated heavier intermediate designs than DAJA since the very beginning of the optimization process even though they started from better populations than the present algorithm.


(a)
DAJA
CBO [33]
ECBO [33]
AFA [38]
HHS [40]
aeDE [41]
Structrual weight (lb)
650
630
610
590
570
550
530
510
490
470
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
5500
6000
6500
7000
Number of structural analyses


(b)
DAJA
HPSO [27]
DHPSACO [28]
HHS [40]
Structural weight (lb)
940
910
880
850
820
790
760
730
700
670
640
610
580
550
520
0
500
1000
1500
2000
2500
3000
3500
4000
4500
Number of structural analyses

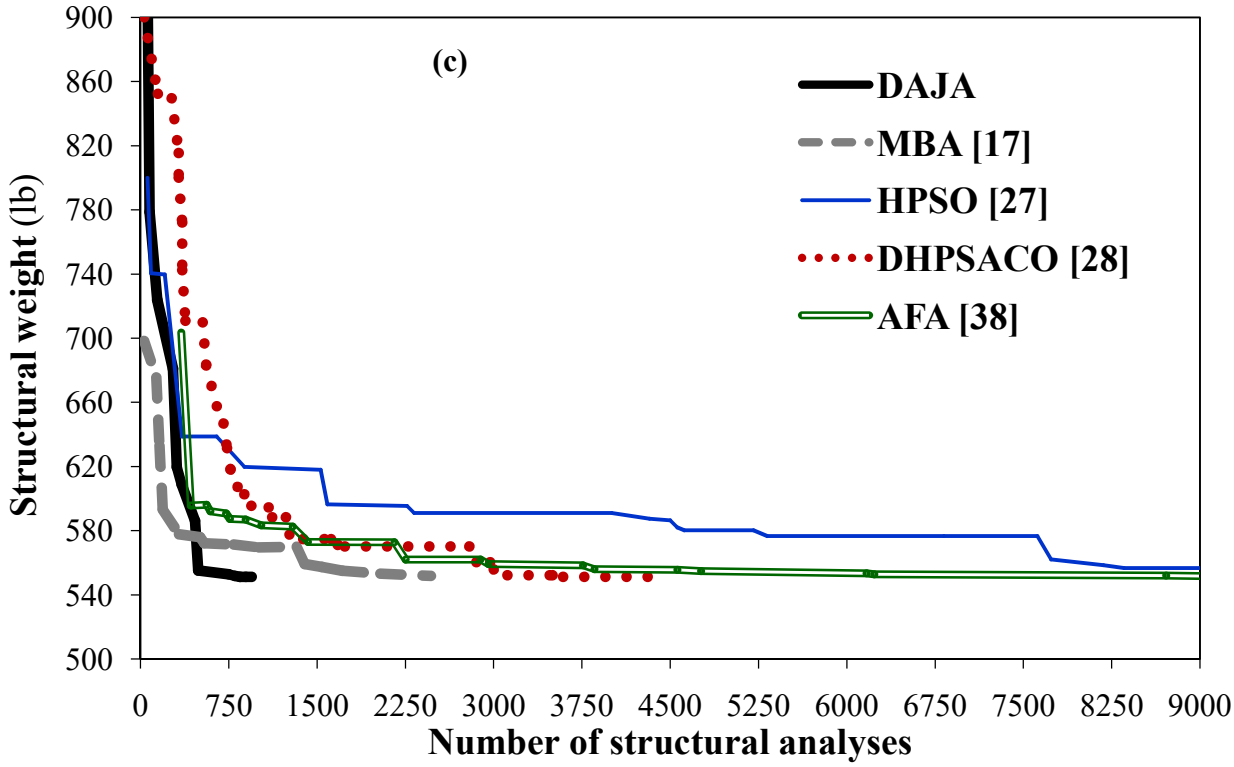


Fig. 5. Comparison of convergence curves for the 25-bar truss problem: (a) Case 1; (b) Case 2; (c) Case 3.

The 25-bar tower optimized in Case 1was re-optimized including layout and topology variables. There arefivecontinuous layout variables,the coordinates X_4, Y_4, Z_4, X_8 and Y_8 of nodes 4 and 8. Hence, this problem variant is a mixed discrete-continuous optimization problem including 13 sizing/layout variables and 8 topology variables. The side constraints for geometry variables are: $20 \leq X_4 = X_5 = -X_3 = -X_6 \leq 60$ in, $40 \leq Y_3 = Y_4 = -Y_5 = -Y_6 \leq 80$ in, $90 \leq Z_3 = Z_4 = Z_5 = Z_6 \leq 130$ in, $40 \leq X_8 = X_9 = -X_7 = -X_{10} \leq 80$ in, $100 \leq Y_7 = Y_8 = -Y_9 = -Y_{10} \leq 140$ in.

The optimization results are presented in Table 11. It can be seen that DAJA removed the same elements indicated for the other literature solutions(based on genetic algorithms, firefly algorithm and evolutionary search) and found the best design overall both in terms of structural weight and number of required structural analyses. The optimum design is kinematically stable and does not lead to the presence of local mechanisms. Interestingly, DAJA converged to the same optimum layout as the multi-stage JA algorithm developed in [47] for topology optimization. This is becausethe 25-bar tower problemincluded only 8 sizing variables which took only 3 values at the optimum solution. Hence, it is rather easy to identify the best region of sizing design space. However, the present formulation could reduce the number of required structural analyses from 4877 to 3404thanks to the advanced search strategy described in Section 3. The standard deviation on optimized weight resulting from independent runs was only 0.003 kg, relatedto small fluctuations in the optimum values of layout variables. This is consistent with the fact that design search is driven by a rather small number of discrete values of cross-sectional areas.

Table 11. Optimization results of the 25-bar tower topology optimization problem.

Design Variables	GA [53]	GA [54]	FFA [55]	FSD-ES [56]	Multi-stage JA[47]	**Present study DAJA**
A_1 (in^2)	Removed	Removed	Removed	Removed	Removed	**Removed**
A_2	0.1	0.1	0.1	0.1	0.1	**0.1**
A_3	0.9	0.9	1.1	0.9	1.0	**1.0**
A_4	Removed	Removed	Removed	Removed	Removed	**Removed**
A_5	Removed	Removed	Removed	Removed	Removed	**Removed**
A_6	0.1	0.1	0.1	0.1	0.1	**0.1**
A_7	0.1	0.1	0.1	0.1	0.1	**0.1**
A_8	1.0	1.0	0.9	1.0	0.9	**0.9**
X_4 (in)	39.91	38.7913	38.50	38.8713	38.909	**38.909**
Y_4	61.99	66.1110	64.35	61.5207	59.087	**59.087**
Z_4	118.23	112.9787	112.87	119.1785	123.247	**123.247**
X_8	53.13	48.7924	49.13	49.4146	51.227	**51.227**
Y_8	138.49	138.8910	134.94	137.9423	140.104	**140.104**
Weight (kg)	52.045	51.877	52.880	51.899	51.388	**51.388**
CVP (%)	None	None	None	None	None	**None**
NSA	6000	10000	6000	8660	4877	**3404**

5.3.*47-bar power line*

The third design example regards the planar 47-bar power line tower shown in Fig. 6, including 47 elements connected by 22 nodes. The material density is 0.3 lb/in^3 while the modulus of elasticity is 30Msi. Bars are divided in 27 groups: (1) A_1=A_3, (2) A_2=A_4, (3) A_5=A_6, (4) A_7, (5) A_8=A_9, (6)A_{10}, (7) A_{11}=A_{12}, (8) A_{13}=A_{14}, (9) A_{15}=A_{16}, (10) A_{17}=A_{18}, (11) A_{19}=A_{20}, (12) A_{21}=A_{22}, (13) A_{23}=A_{24}, (14) A_{25}=A_{26}, (15) A_{27}, (16) A_{28}, (17) A_{29}=A_{30}, (18) A_{31}=A_{32}, (19) A_{33}, (20)A_{34}=A_{35}, (21) A_{36}=A_{37}, (22) A_{38}, (23) A_{39}=A_{40}, (24) A_{41}=A_{42}, (25) A_{43}, (26) A_{44}= A_{45}, and (27) A_{46}=A_{47}.

The structure is subject to three independent loading conditions: (i) 6 kips acting in the positive *X*-direction and 14 kips acting in the negative *Y*-direction at nodes 17 and 22; (ii) 6 kips acting in the positive *X*- direction and 14 kips acting in the negative *Y*-direction at node 17; (iii) 6 kips acting in the positive *X*-direction and 14 kips acting in the negative *Y*-direction at node 22.

The truss must be designed against stress and buckling constraints. In particular, stress limits are 15,000 psi in compression and 20,000 psi in tension. The Euler buckling limit is:

$$\sigma_i^{cr} = \frac{-KEA_i}{L_i^2} \qquad i=1,2,\ldots,47 \qquad (12)$$

In Eq. (12),the value of the *K*constant depends on cross-sectional geometry (here,*K*=3.96);*E* is the Young's modulus of the material;$A_i$and L_i, respectively, are the cross-sectional area and length of the i^{th}truss element.

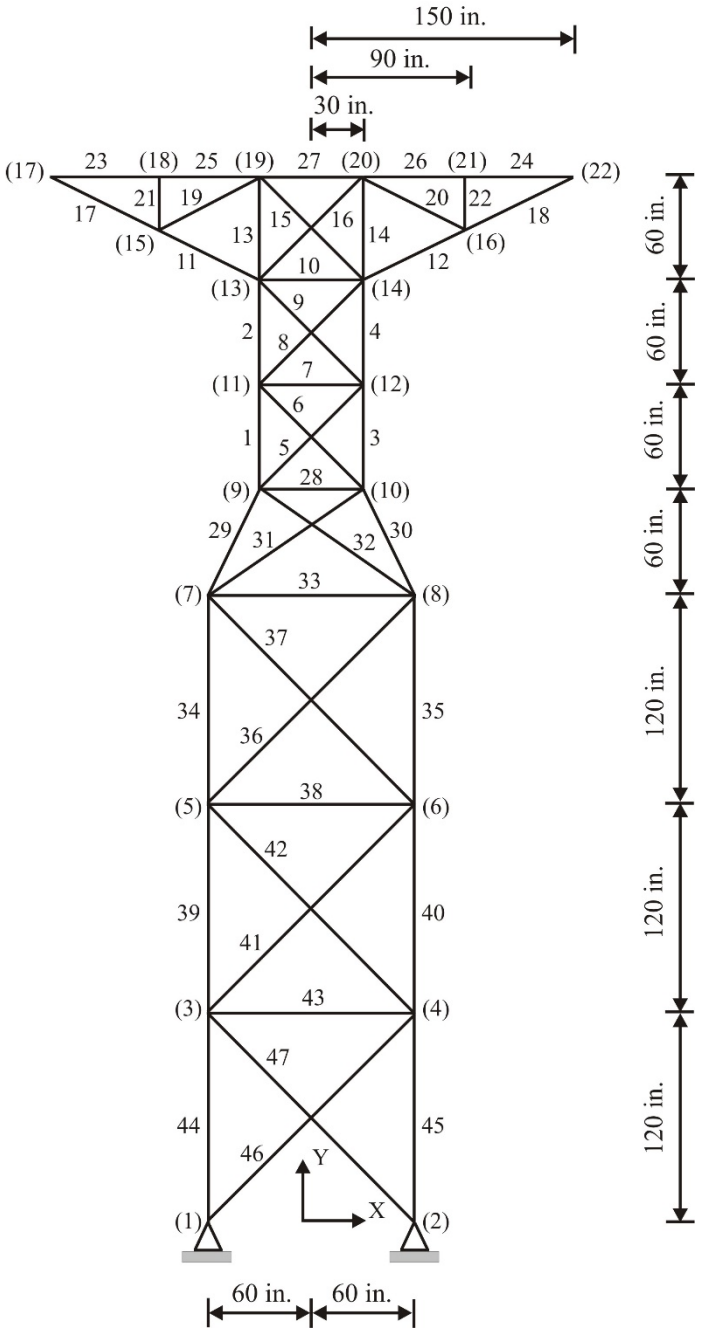


Fig. 6. Schematic of the planar 47-bar power line tower.

The simplest variant of this problem includes only 27 sizing variables corresponding to the cross-sectional areas of the 27 groups of elements. Under this assumption, the tower was optimized by Lee et al. [26] using harmony search (HS) and Kaveh and Mahdavi [32] using colliding bodies optimization (CBO). Discrete values of cross-sectional areas were selected from the AISC set reported in Table 6. The presence of buckling constraints contributed to the difficulty of metaheuristic algorithms to find a global optimum solution for this design example in spite of the very large number of available discrete combinations of sizing variables, $4.048 \cdot 10^{91}$ (i.e. 27^{64}).

The results obtained by DAJA are compared with the literature in Table 12.It can be seen that the present algorithm not only designed a lighter structure than HS and CBO (i.e. between 10 and 20 kg weight reduction) but also required significantly less structural analyses than the referenced

algorithms. In particular,DAJA obtained the optimum design after only 8046 structural analyses while HS and CBOfound their best solutions after 45,557 and 25,000 structural analyses, respectively. Once again, DAJA was able to fulfill the basic requirement of metaheuristic optimization: to explore large fractions of designs space at low computational cost. Table 12 shows also that DAJA is more robust than CBO. The rate of success of DAJA for this test problem was about 77%.

Table 12.Optimization resultsofthe 47-bar truss problem including onlydiscrete sizing variables.

Designvariables A_i (in^2)	HS [26]	CBO [32]	**Present study DAJA**
A_1	3.840	3.840	**3.840**
A_2	3.380	3.380	**3.380**
A_3	0.766	0.785	**0.766**
A_4	0.141	0.196	**0.111**
A_5	0.785	0.994	**0.785**
A_6	1.990	1.800	**1.990**
A_7	2.130	2.130	**2.130**
A_8	1.228	1.228	**1.228**
A_9	1.563	1.563	**1.563**
A_{10}	2.130	2.130	**2.130**
A_{11}	0.111	0.111	**0.111**
A_{12}	0.111	0.111	**0.111**
A_{13}	1.800	1.800	**1.800**
A_{14}	1.800	1.800	**1.800**
A_{15}	1.457	1.563	**1.457**
A_{16}	0.442	0.442	**0.563**
A_{17}	3.630	3.630	**3.630**
A_{18}	1.457	1.457	**1.457**
A_{19}	0.442	0.307	**0.250**
A_{20}	3.630	3.090	**3.090**
A_{21}	1.457	1.266	**1.266**
A_{22}	0.196	0.307	**0.307**
A_{23}	3.840	3.840	**3.840**
A_{24}	1.563	1.563	**1.563**
A_{25}	0.196	0.111	**0.141**
A_{26}	4.590	4.590	**4.590**
A_{27}	1.457	1.457	**1.457**
Weight (lb)	2396.8	2386.0	**2376.019**
CV (%)	None	None	**None**
NSA	45557	25000	**8046**

Worst weight (lb)	N/A	2467.73	**2418.19**
Mean weight (lb)	N/A	2405.91	**2399.92**
SD (lb)	N/A	19.61	**13.15**

The convergence curves relative to the best optimization runs of DAJA, HS [26] and CBO [32]for the 47-bar truss problem including only sizing variables are shown in Fig. 7. It can be seen that DAJA and HS started their best run from very similar populations but the present algorithm rapidly improved design while HS spent about 18,000 structural analyses before reducing significantly structural weight. This happened because the HS formulation of [26] generates new trial designs by changing one optimization variable at time while the present DAJA algorithm updates all population in a more dynamic way. The CBO algorithm of Ref. [27] started the optimization process from a more conservative design than DAJA and showed a higher rate of reduction in cost function than DAJA for the first 1000 analyses. However, this led CBO to carry out search near the boundaries of feasible design space and reduce step sizes in order not to generate infeasible trial designs.

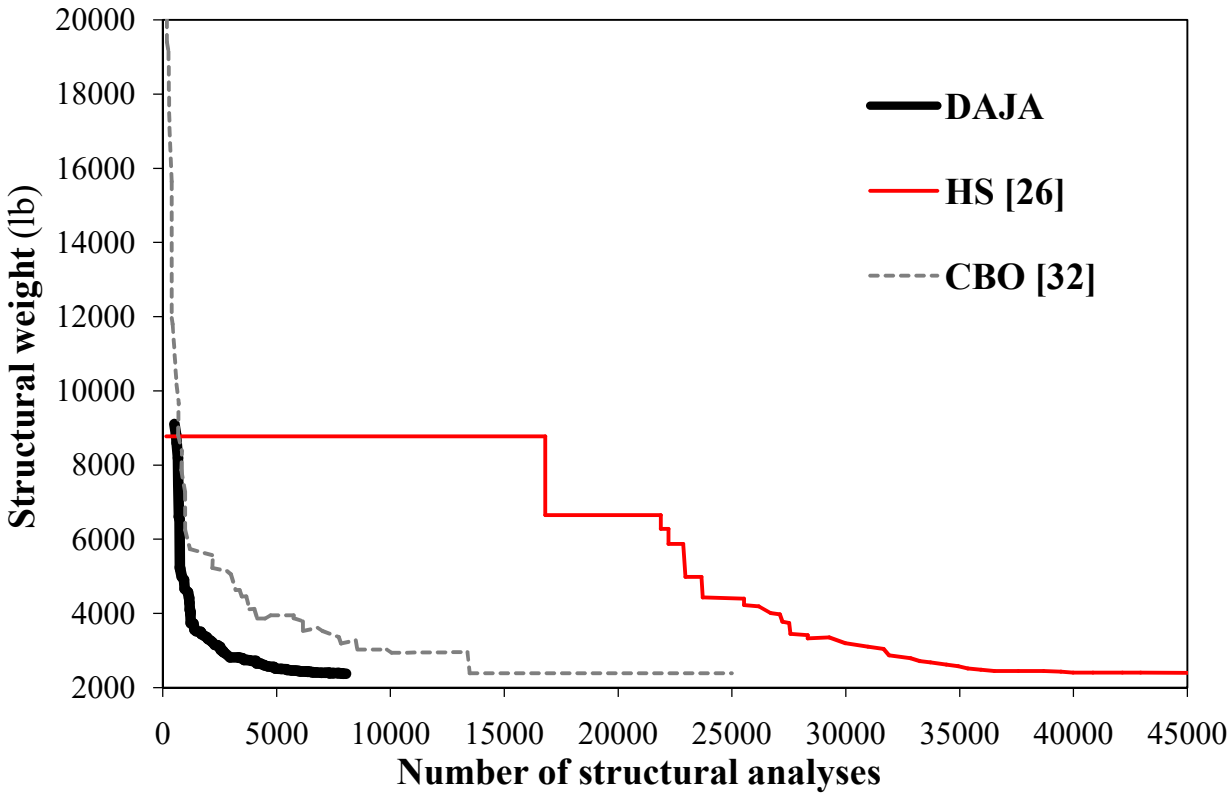


Fig. 7. Comparison of convergence curves for the 47-bar truss problem including only sizing variables.

A much more challenging variant of the 47-bar tower problem includes also discrete layout and topology optimization:the total number of design variables increases to 44 sizing/layout variables and 27 topology variables. The 17 layout variables are the nodal co-ordinates X_2, X_4, Y_4, X_6, Y_6, X_8, Y_8, X_{10}, Y_{10}, X_{12}, Y_{12}, X_{14}, Y_{14}, X_{20}, Y_{20}, X_{21} and Y_{21}, which can take discrete values with a

resolution of 1 in between two consecutive values.Structural symmetry about the Y-axis must be preserved in each design cycle.

Table 13 presents the results obtained for this design example. Discrete JA was compared with the multi-stage JA formulation of Ref. [47], genetic algorithm [57] and simulated annealing [58]. The best and worst designs obtained by DAJA in the independent optimization runs are listed in the table. All optimized designs are kinematically stable. It appears that the best design of DAJA ranked overall third in terms of optimized weight.However, DAJA's solutionshould be considered the best overall becauseit violatesdesign constraints less than the optimum solutions found by the other algorithms. For example, if we multiply cross-sectional areas by a factor (1+CV/100) to recover maximum constraint violation, DAJA's optimized weights would range between 841.1 and 843.7 kg vs. 844.8 kg of multi-stage JA [47] and 860.3 kg of SA [58], yet considerably lighter than the 855.1 kg weight of the GA's feasible design [57].

Unlike multi-stage JA, which rounded continuous optimum solutions weighing from 823.805 to 823.980 kg (practically includingthe same distribution of cross-sectional areas)andhence obtained always the same discrete optimum design regardless of initial population, DAJA explored a larger fraction of design space as it perturbed the whole set of sizing, layout and topology variables until the end of optimization process. The gradient/pseudo-gradient information and approximate line search strategy implemented by DAJA allowed to generate high quality trial designs throughout optimization process without biasing search after some variables are roundedand/or removed from the design process. This explains why the best and worst designs found by DAJA were very competitive and anyhow better than the optimum design of multi-stage JA.

Table 13. Results of fully discrete optimization of the 47-bar truss structurewith sizing, layout and topology variables.

Design variables	GA [57]	SA [58]	Multi-stage JA [47]	**Present study DAJA – Best**	**Present study DAJA - Worst**
A_1 (in^2)	2.6	2.9	2.6	**2.7**	**2.6**
A_2	2.4	2.4	2.5	**2.5**	**2.5**
A_5	0.8	0.5	0.8	**0.8**	**0.8**
A_7	Removed	Removed	Removed	**Removed**	**Removed**
A_8	1.1	1.7	1.0	**1.0**	**1.0**
A_{10}	1.3	Removed	1.0	**1.2**	**1.0**
A_{11}	1.7	1.6	1.7	**1.9**	**1.8**
A_{13}	0.6	0.5	0.8	**0.6**	**0.8**
A_{15}	1.0	0.9	0.9	**0.8**	**0.9**
A_{17}	1.4	1.3	1.2	**1.3**	**1.2**
A_{19}	0.5	0.8	0.3	**0.4**	**0.3**
A_{21}	1.1	1.1	1.1	**1.2**	**1.1**
A_{23}	1.0	1.0	0.9	**1.0**	**0.9**
A_{25}	1.0	0.8	0.9	**0.9**	**0.9**
A_{27}	0.8	0.5	0.8	**0.8**	**0.8**
A_{28}	Removed	Removed	Removed	**Removed**	**Removed**
A_{29}	2.7	2.6	2.6	**2.7**	**2.6**
A_{31}	1.0	1.1	0.8	**0.8**	**0.8**
A_{33}	Removed	Removed	Removed	**Removed**	**Removed**
A_{34}	2.9	3.1	2.8	**2.9**	**2.8**
A_{36}	0.9	0.5	0.9	**0.9**	**0.9**
A_{38}	Removed	Removed	Removed	**Removed**	**Removed**
A_{39}	3.1	2.9	3.0	**3.1**	**3.0**
A_{41}	1.1	1.4	1.1	**1.0**	**1.1**
A_{43}	Removed	Removed	Removed	**Removed**	**Removed**
A_{44}	3.2	3.3	3.3	**3.2**	**3.3**
A_{46}	1.0	0.3	1.1	**1.1**	**1.1**
X_2 (in)	104.0	112.0	100.0	**105.0**	**100.0**
X_4	93.0	88.0	87.0	**86.0**	**87.0**
Y_4	116.0	140.0	134.0	**135.0**	**134.0**

X_6	74.0	68.0	70.0	**68.0**	**70.0**
Y_6	223.0	241.0	261.0	**248.0**	**261.0**
X_8	64.0	61.0	61.0	**58.0**	**61.0**
Y_8	302.0	326.0	342.0	**330.0**	**342.0**
X_{10}	54.0	47.0	54.0	**53.0**	**54.0**
Y_{10}	391.0	410.0	411.0	**396.0**	**413.0**
X_{12}	46.0	44.0	43.0	**44.0**	**43.0**
Y_{12}	458.0	450.0	476.0	**459.0**	**476.0**
X_{14}	51.0	65.0	44.0	**46.0**	**44.0**
Y_{14}	507.0	502.0	514.0	**506.0**	**514.0**
X_{20}	19.0	1.0	2.0	**1.0**	**1.0**
Y_{20}	595.0	598.0	594.0	**580.0**	**594.0**
X_{21}	90.0	58.0	94.0	**81.0**	**94.0**
Y_{21}	626.0	635.0	634.0	**632.0**	**634.0**
Weight (kg)	855.053	811.603	834.118	**834.606**	**836.157**
CV (%)	None	3.2 (Tens. stress) 6 (Compr. stress)	0.89 (Compr. stress) 1.28 (Buckling)	**0.779 (Compr. stress)** **0.636 (Buckling)**	**0.907 (Compr. stress)** **0.475 (Buckling)**
NSA	100000*	13000^{+}	6945	**5174**	**4576**

***** Estimated as the product between number of optimization cycles and population size

$^{+}$ Estimated as the product between number of optimization cycles and number of optimized variables

The present algorithm was much faster than GA and SA and required 25% less structural analyses than multi-stage JA. Again, this is due to the inherent ability of DAJA of generating high quality trial designs because the search engine always tries to improve design by operating on the whole set of optimization variables. As a further proof to this statement, Fig. 8 compares the optimum shapes found by different algorithms. The worst design of DAJA corresponds to a similar shape to multi-stage JA but the present design DAJA violated buckling by less than 0.5%, which is about 1/3 of the violation reported in [47] for multi-stage JA. The best design of DAJA is characterized by the presence of shorter elements in the truss regions most critical to buckling, thus making it easier to satisfy this type of constraint.

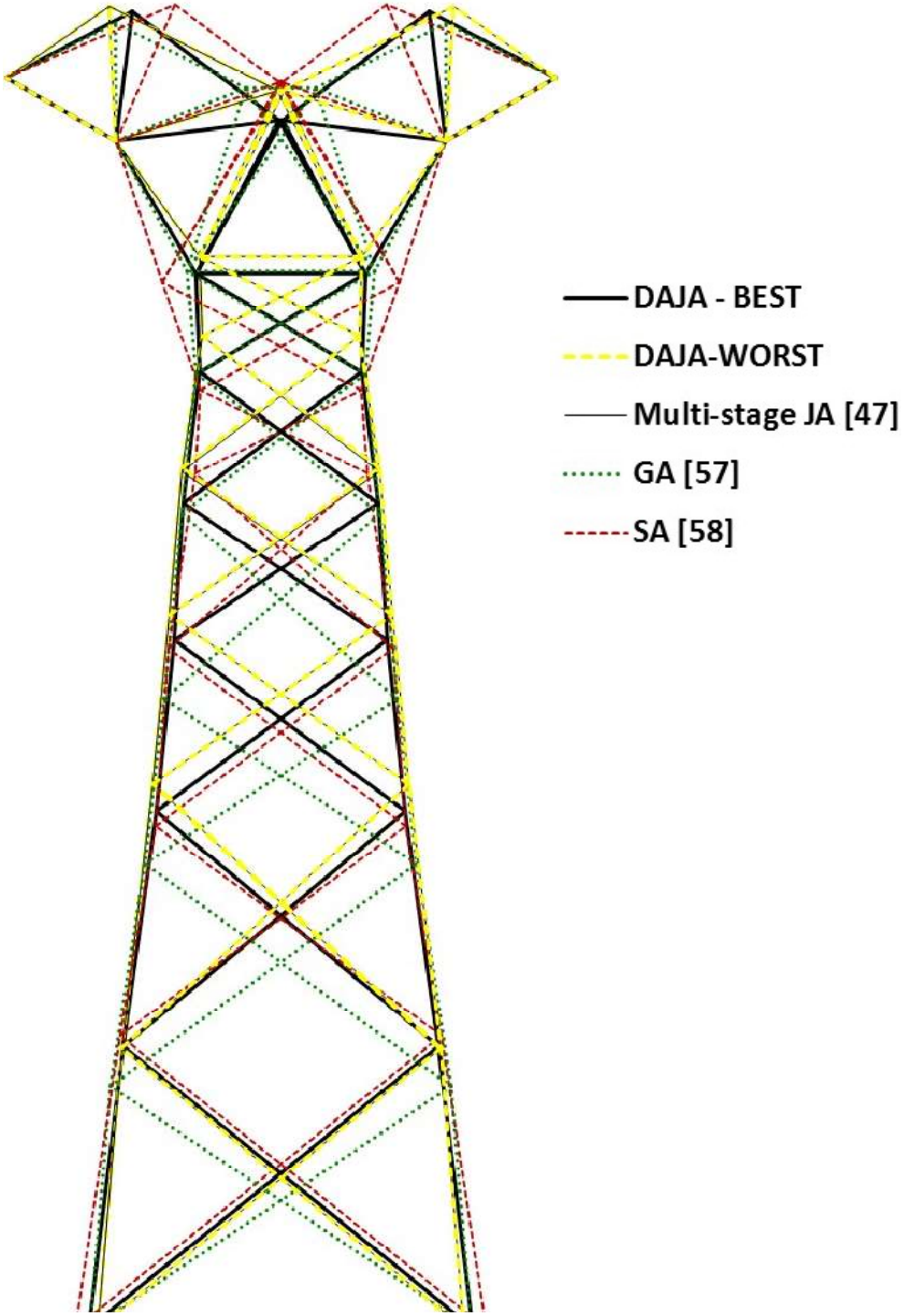


Fig. 8. Comparison of optimized topologies found for the fully discrete 47-bar tower problem.

Finally, DAJA is robust enough in terms of structural weight and number of structural analyses required in the optimization process. In fact, the standard deviation on optimized weight was only 1.359 kg, which is about 0.325% of the average optimized weight. Furthermore, the standard deviation on the number of structural analyses was about 15% of the average number of analyses. The robustness of DAJA is quite remarkable if we consider that this design example included a rather large number of optimization variables.

4.4. *Planar 52-bar truss*

The forth structure optimized in this study was the planar 52-bar truss shown in Fig. 9.The structure includes 52 elements connected by 20 nodes. Material properties are the same as in the 47-bar tower problem. The allowable stress of bars is ±26,010psi in tension/compression. A single loading condition acts in the structure:concentrated forcesP_x=22.48 kips and P_y=44.96 kips applied to nodes 17-20. Bars are divided in 12 groups of elements with the same cross-sectional areas: (1) A_1-A_4, (2) A_5-A_{10}, (3) A_{11}-A_{13}, (4) A_{14}-A_{17}, (5) A_{18}-A_{23}, (6) A_{24}-A_{26}, (7) A_{27}-A_{30}, (8) A_{31}-A_{36}, (9) A_{37}-A_{39}, (10) A_{40}-A_{43}, (11) A_{44}-A_{49}, and (12) A_{50}-A_{52}. Hence, this design example included 12 discrete sizing variables selected from the discrete values listed in Table 6.

The best solution quoted in the literature for this test problem (see, for example, Refs. [21,37-41]) is {4658.055; 1161.288; 494.1930; 3303.219; 940.0000; 494.1930; 2238.705; 1008.385; 494.1930; 1283.868; 1161.288; 494.1930} mm^2 yielding a structural weight of 1902.605 kg. The huge number of available combinations of discrete sizing variables from the D set (i.e. $1.168 \cdot 10^{69}$, that is12^{64}) and the rather small number of discrete sizing variables (only 12) allowed many metaheuristic algorithms to find the aforementioned optimum design. However, computational cost entailed by metaheuristic optimization (i.e. between about 3700 and 7100 structural analyses) appears to be very high also for this design example.

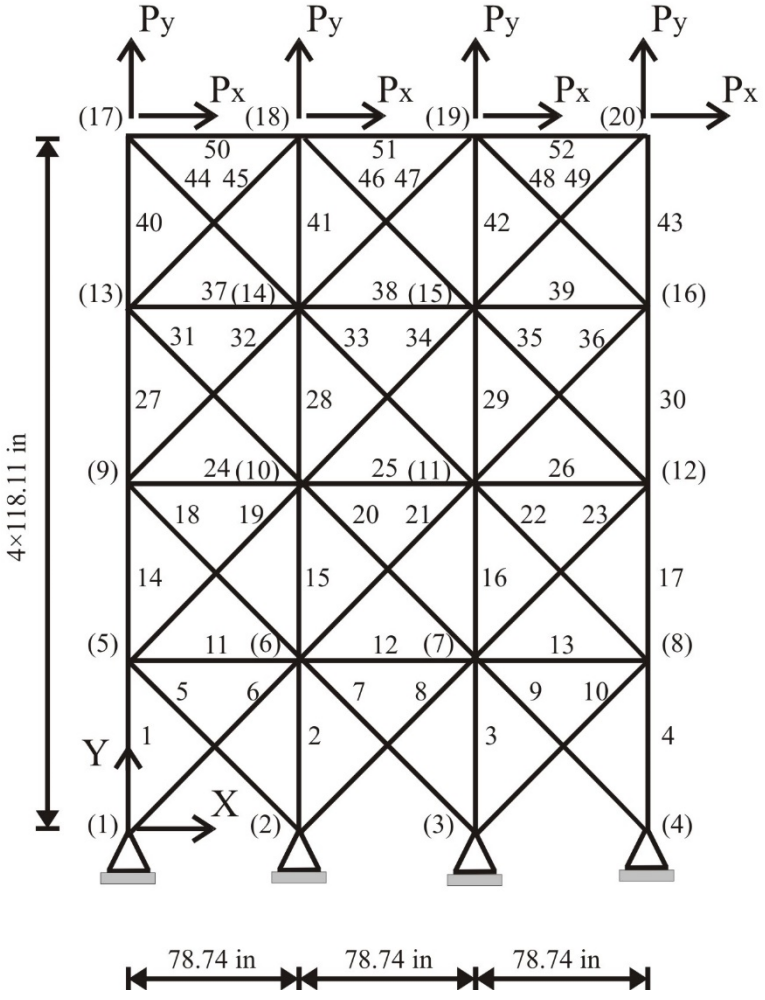


Fig. 9. Schematic of the planar 52-bar truss.

The optimization results of DAJA are comparedin Table 14with the many solutions published in the literature. Interestingly, DAJA and most of the referencedalgorithms convergedto a feasible design corresponding to the target optimum weight of 1902.605 kg.The HPSO [27],AFA[38] and DHPSACO [28]algorithms designed slightly heavier trusses than DAJA. Furthermore, the optimum design of DHPSACO also violated constraints. CSS[30] found the lowest structural weight overall (1897.62kg) but the corresponding optimum design violatedstressconstraints by 0.114%.

Once again, DAJA was the fastest and most robust optimization algorithm. In particular, DAJA found its best solutionafter 3321 structural analyses whereas the other methods required between 3720 and 150,000 analyses. It should be noted that the SOS algorithm described in [21] obtained the same structural weight as DAJA within 47 iterations but the actual number of structural analyses required in the optimization process was not specified in Ref. [21]. Interestingly, DAJA required 46 iterations (hence, one less iteration than SOS) to converge to the optimum weight of 1902.605 kg although it started from a considerably heavier design than SOS: 7117.6 kg vs. only 3950 kg of SOS. This result was achieved by DAJA in spite of having used a population of only 20 designs vs. the 50 designs included in the SOS population. Hence, the present algorithm should actually be considered faster than SOS.

As far as it concerns standard deviation on optimized weight, the 2nd most robust algorithm, IMBA, showed a 2.5 times larger deviation than DAJA. In summary, DAJA reduced computational cost for this design example by 11% with respect to the fastest referenced algorithm (aaDE of Ref. [41]) but was 15 times more robust than that algorithm. The rate of success achieved by DAJA for this design example was about 88%.

Table 14. Optimization results obtained for the 52-bar truss problem

Design variables A_i (mm^2)	HPSO [27]	DHPSACO [28]	CSS [30]	CBO [32]	TLBO [37]	AFA [38]	WCA [39]	IMBA [39]	HHS [40]	aeDE [41]	**Present DAJA**
A_1-A_4	4658.055	4658.055	4658.055	4658.055	4658.055	4658.055	4658.055	4658.055	4658.055	4658.055	**4658.055**
A_5-A_{10}	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288	**1161.288**
A_{11}-A_{13}	363.2250	494.1930	388.3860	388.3860	494.1930	363.2250	494.1930	494.1930	494.1930	494.1930	**494.1930**
A_{14}-A_{17}	3303.219	3303.219	3303.219	3303.219	3303.219	3303.219	3303.219	3303.219	3303.219	3303.219	**3303.219**
A_{18}-A_{23}	940.0000	1008.385	940.0000	939.9980	940.0000	939.9880	940.0000	940.0000	940.0000	940.0000	**940.0000**
A_{24}-A_{26}	494.1930	285.1610	494.1930	506.4510	494.1930	494.1930	494.1930	494.1930	494.1930	494.1930	**494.1930**
A_{27}-A_{30}	2238.705	2290.318	2238.705	2238.705	2238.705	2238.705	2238.705	2238.705	2238.705	2238.705	**2238.705**
A_{31}-A_{36}	1008.385	1008.385	1008.385	1008.385	1008.385	1008.385	1008.385	1008.385	1008.385	1008.385	**1008.385**
A_{37}-A_{39}	388.3860	388.3860	494.1930	506.4510	494.1930	641.2890	494.1930	494.1930	494.1930	494.1930	**494.1930**
A_{40}-A_{43}	1283.868	1283.868	1283.868	1283.868	1283.868	1283.868	1283.868	1283.868	1283.868	1283.868	**1283.868**

A_{44}-A_{49}	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288	1161.288	**1161.288**
A_{50}-A_{52}	792.2560	506.4510	494.1930	506.4510	494.1930	494.1930	494.1930	494.1930	494.1930	494.1930	**494.1930**
Weight (kg)	1905.50	1904.83	1897.62	1899.35	1902.605	1903.37	1902.605	1902.605	1902.605	1902.605	**1902.605**
CV (%)	None	0.2726	0.114	0.0484	None	None	None	None	None	None	**None**
NSA	150000	5300	5000	3840	6000	52600	7100	4750	4253	3720	**3321**
Worst weight (kg)	N/A	N/A	N/A	2262.8	N/A	N/A	1912.646	1904.83	N/A	1925.714	**1903.944**
Mean weight (kg)	N/A	N/A	N/A	1963.12	N/A	N/A	1909.856	1903.076	1904.587	1906.735	**1902.74**
SD (kg)	N/A	N/A	N/A	106.01	N/A	N/A	7.09	1.13	1.309	6.679	**0.446**

Figure 10 compares the convergence curves obtained for the best optimization runs of DAJA, TLBO [37], IMBA [39], HHS [40] and aeDE [41]. DAJA and aeDE started their best runs from almost the same initial weight but the present algorithmwas able to reduce structural weight since the very beginning of optimization process. This is because differential evolution (even DE schemes including elitist strategies) does not necessarily direct all trial designs towards the best design of the current population. HHS, IMBA and TLBO started from considerably better initial designs than DAJA but either generated infeasible intermediate designs (for example, HHS approached constraint domain boundaries too quickly) or biased search towards suboptimal designs. The present algorithm could soon recover the initial gap in structural weight with respect to HHS, IMBA and TLBO because it is easier to define a large number of descent directions and assignvery large perturbations to sizing variables when the optimization process starts from conservative designs. The considerably smaller amount of heuristics included in the DAJA formulation, which, unlike all other metaheuristic algorithms, does not require setting of internal parameters, certainly concurs to speed up search towards the optimum solution.

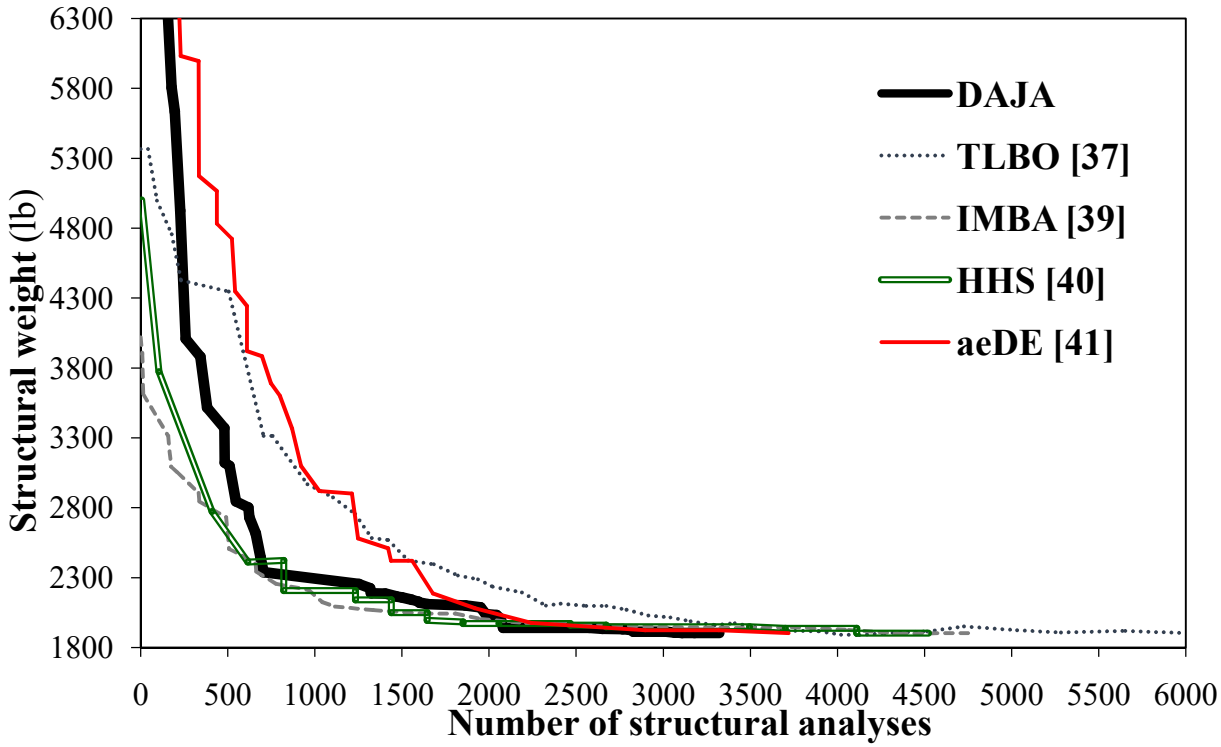


Fig. 10. Comparison of convergence curves obtained for the 52-bar truss problem.

4.5. *Spatial 72-bar truss*

The fifth structure optimized in this study is thespatial 72-bar truss shown in Fig. 11. Thistruss includes 72 bars connected by 20 nodes. Material properties and stress limits are the same as for the 10-bar truss design example whilethe allowable displacements of the top nodes of trussmust be lessthan ±0.25 in in both *X* and *Y*-directions. The structure is subject to the two independent loading conditions listed in Table 15. This test problem included 16 discrete sizing variables corresponding to the cross-sectional areas of the groups of elements that form the truss: (1) A_1–A_4, (2) A_5–A_{12}, (3) A_{13}–A_{16}, (4) A_{17}–A_{18}, (5) A_{19}–A_{22}, (6) A_{23}–A_{30}, (7) A_{31}–A_{34}, (8) A_{35}–A_{36}, (9) A_{37}–A_{40}, (10) A_{41}–A_{48}, (11) A_{49}–A_{52}, (12) A_{53}–A_{54}, (13) A_{55}–A_{58}, (14) A_{59}–A_{66}, (15) A_{67}–A_{70}, (16) A_{71}–A_{72}.

Two optimization cases were considered. InCase 1, discrete sizing variables were selected from the set D=[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, 2.7, 2.8, 2.9, 3.0, 3.1, 3.2] (in^2). In Case 2, discrete sizing variables were selected from the available values listed in Table 6.

The best solution quoted in the literature for Case 1 (see, for example, Refs. [28,32,37-40]) is {1.9; 0.5; 0.1; 0.1; 1.4; 0.5; 0.1; 0.1; 0.5; 0.5; 0.1; 0.1; 0.2; 0.6; 0.4; 0.6} in^2 yielding a structural weight of 385.54 lb. For Case 2, the best design quoted in the literature (see, for example, Refs. [28,30,33,34]) is {1.990; 0.563; 0.111; 0.111; 1.228; 0.442; 0.111; 0.111; 0.563; 0.563; 0.111; 0.111; 0.196; 0.563; 0.391; 0.563} in^2 yielding a structural weight of 389.334 lb. Similar to the 10, 25 and 52-bar design examples, the 72-bar truss problem includes a very large number of available combinations from the discrete sets D (respectively, $3.403 \cdot 10^{38}$– 16^{32}– and $1.158 \cdot 10^{76}$ – 16^{64}– for Case 1 and Case 2) and a rather small number of discrete sizing variables (only 16). While this allowed several metaheuristic algorithms to find the global optima for the two problem variants, computational cost remains an important issue also for this test problem because the reported number of structural analyses ranged from 3200 to 12,200 for Case 1 and from 4600 to about 17,000 for Case 2.

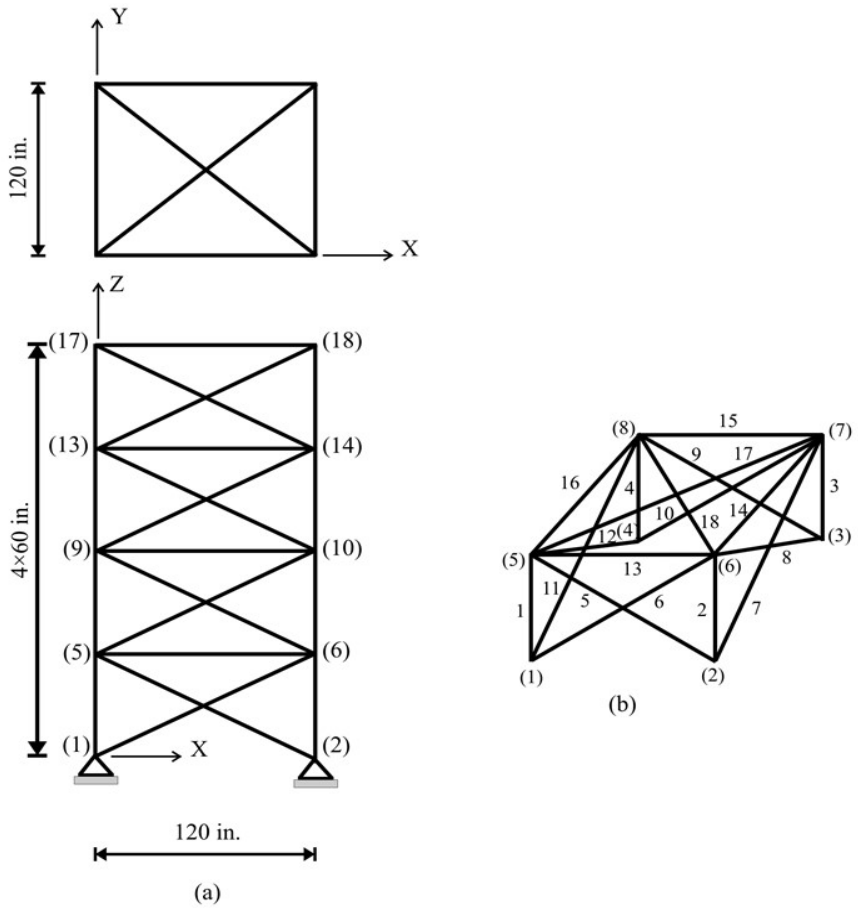


Fig. 11. Schematic of the spatial 72-bar truss structure: (a) top and side view (b) member and node numbering pattern for the first story.

Table 15. Loadingconditions acting on the spatial 72-bar truss structure.

Node	Loading condition 1			Loading condition 2		
	F_x (kips)	F_y (kips)	F_z (kips)	F_x (kips)	F_y (kips)	F_z (kips)
17	5.0	5.0	-5.0	0.0	0.0	-5.0
18	0.0	0.0	0.0	0.0	0.0	-5.0
19	0.0	0.0	0.0	0.0	0.0	-5.0
20	0.0	0.0	0.0	0.0	0.0	-5.0

Tables 16 and 17 compare the optimization results of DAJA with the many solutions available in the literaturefor Case 1 and Case 2, respectively. For optimization Case 1, all methods converged to the target feasible optimum design weighing 385.54 lb except HPSO [24] that found a weight of 388.94 lb. For optimization Case 2, DAJA,ECBO [33], AFA [38], WCA [39] and IMBA [39]converged to the target feasible optimumdesign while HPSO [27], DHPSACO [28], CSS [30] and CBO [32,33] obtained heavier designs that sometimes even violate optimization constraints. JA was definitely the fastest optimizer in both optimization cases, always followed by WCA. In particular, in optimization Case 1, DAJA required only 1873 structural analyses vs. 3200 analyses (thus achieving about 42% reduction in computation cost) required by the 2nd fastest optimizer WCA. In optimization Case 2, DAJA completed the design process within 3376 structural analyses, about 27% less than the 2nd fastest optimizer WCA.The

main goal of substantially reducing the computational cost of the optimization was hence very well accomplished by DAJA also for this test problem.

DAJA showed once again a very small standard deviation on optimized weight, ranking 3rd after IMBA and WCA in Case 1 (this may be due to the fact that the present algorithm found the optimum design after a very small number of structural analyses compared to the other two algorithms) and 1st overall in Case 2.This confirms the validity of the proposed approach, which achieved a success rate of about 80% for Case 1 and about 90% for Case 2, if we consider optimized designs up to 0.35% heavier than the target global optimum.

The continuous optimum design found by two-stage JA [40] in Case 1, {1.883750; 0.513814; 0.100001; 0.100000; 1.263169; 0.511173; 0.100000; 0.100002; 0.524452; 0.516183; 0.100000; 0.100089; 0.156420; 0.545762; 0.412597; 0.570632} in^2, can be rounded to the same discrete optimum design quoted in Table 16 for DAJA. However, the continuous optimization of two-stage JA required by itself about 11,000 structural analyses, i.e. about 6 times the total computational cost of DAJA. The two-stage continuous-discrete optimization process, although appropriate for this test problem, confirms to be computationally very expensive.

Table 16.Optimization results obtained for Case 1 of the 72-bar truss problem.

Design variables	HPSO [27]	DHPSACO [28]	CBO [32]	TLBO[37]	AFA [38]	WCA [39]	IMBA [39]	HHS [40]	**Present study DAJA**
A_1-A_4	2.1	1.9	1.9	1.9	2.0	1.9	1.9	1.9	**1.9**
A_5-A_{12}	0.6	0.5	0.5	0.5	0.5	0.5	0.5	0.5	**0.5**
A_{13}-A_{16}	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	**0.1**
A_{17}-A_{18}	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	**0.1**
A_{19}-A_{22}	1.4	1.3	1.4	1.4	1.3	1.4	1.4	1.3	**1.4**
A_{23}-A_{30}	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	**0.5**
A_{31}-A_{34}	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	**0.1**
A_{35}-A_{36}	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	**0.1**
A_{37}-A_{40}	0.5	0.6	0.5	0.5	0.5	0.5	0.5	0.6	**0.5**
A_{41}-A_{48}	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	**0.5**
A_{49}-A_{52}	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	**0.1**
A_{53}-A_{54}	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	**0.1**
A_{55}-A_{58}	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	**0.2**
A_{59}-A_{66}	0.5	0.6	0.6	0.6	0.6	0.6	0.6	0.6	**0.6**
A_{67}-A_{70}	0.3	0.4	0.4	0.4	0.4	0.4	0.4	0.4	**0.4**
A_{71}-A_{72}	0.7	0.6	0.6	0.6	0.6	0.6	0.6	0.6	**0.6**
Weight (lb)	388.94	385.54	385.54	385.54	385.54	385.54	385.54	385.54	**385.54**
CV (%)	None	0.0164	None	None	None	None	None	None	**None**
NSA	50000	5330	4500	4000	12200	3200	5750	3294	**1873**
Worst weight(lb)	N/A	N/A	460.98	N/A	N/A	386.80	387.942	N/A	**387.943**
Mean weight(lb)	N/A	N/A	401	N/A	N/A	385.842	385.765	386.04	**386.161**
SD (lb)	N/A	N/A	16.99	N/A	N/A	0.55	0.41	1.155	**1.008**

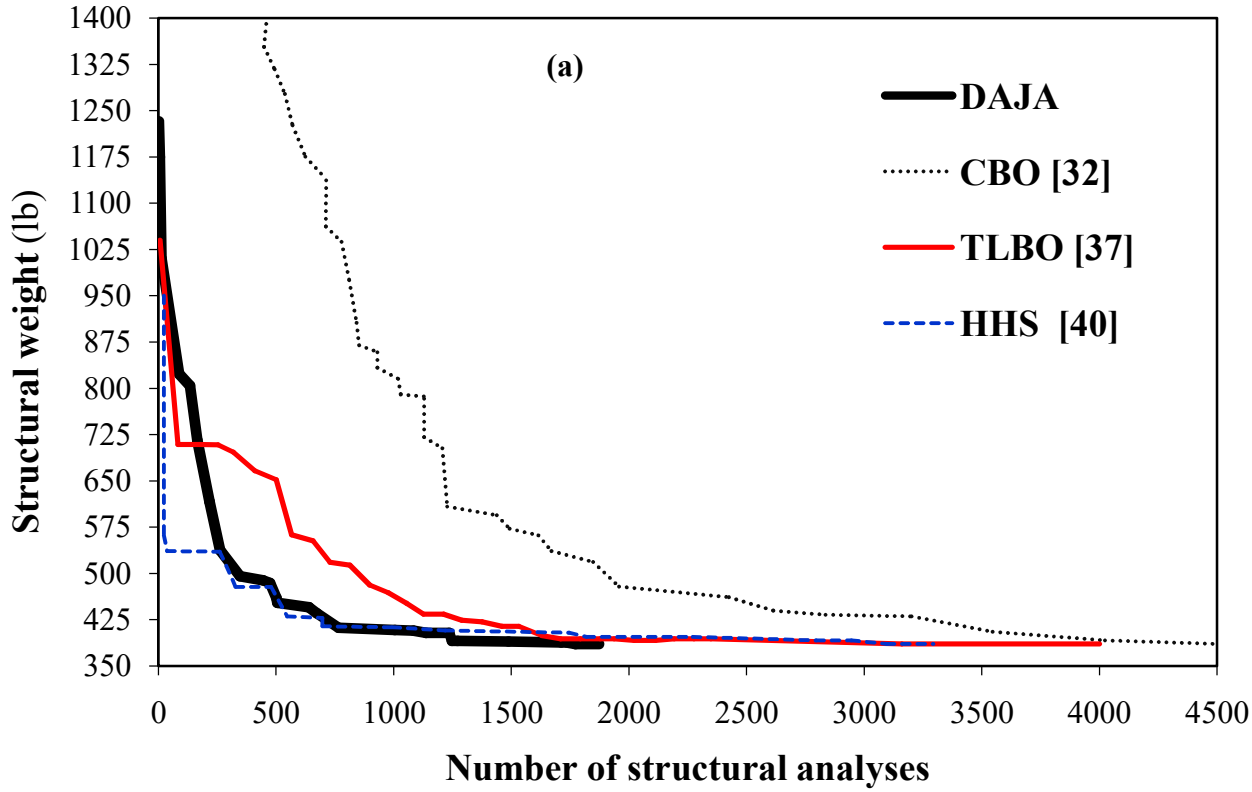
Table 17. Optimization results obtained for Case 2 of the 72-bar truss problem.

Design variables A_i (in^2)	HPSO [27]	DHPSACO [28]	CSS [30]	CBO(1) [32]	CBO(2) [33]	ECBO [33]	AFA [38]	WCA [39]	IMBA [39]	**Present study DAJA**
A_1-A_4	4.970	1.800	1.990	1.620	2.130	1.990	1.990	1.990	1.990	**1.990**
A_5-A_{12}	1.228	0.442	0.442	0.563	0.563	0.563	0.563	0.442	0.442	**0.563**
A_{13}-A_{16}	0.111	0.141	0.111	0.111	0.111	0.111	0.111	0.111	0.111	**0.111**
A_{17}-A_{18}	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	**0.111**
A_{19}-A_{22}	2.880	1.228	0.994	1.457	1.228	1.228	1.228	1.228	1.228	**1.228**
A_{23}-A_{30}	1.457	0.563	0.563	0.442	0.442	0.442	0.442	0.563	0.563	**0.442**
A_{31}-A_{34}	0.141	0.111	0.111	0.111	0.141	0.111	0.111	0.111	0.111	**0.111**
A_{35}-A_{36}	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	**0.111**
A_{37}-A_{40}	1.563	0.563	0.563	0.602	0.442	0.563	0.563	0.563	0.563	**0.563**
A_{41}-A_{48}	1.228	0.563	0.563	0.563	0.563	0.563	0.563	0.563	0.563	**0.563**
A_{49}-A_{52}	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	0.111	**0.111**
A_{53}-A_{54}	0.196	0.250	0.111	0.111	0.111	0.111	0.111	0.111	0.111	**0.111**
A_{55}-A_{58}	0.391	0.196	0.196	0.196	0.196	0.196	0.196	0.196	0.196	**0.196**
A_{59}-A_{66}	1.457	0.563	0.563	0.602	0.563	0.563	0.563	0.563	0.563	**0.563**
A_{67}-A_{70}	0.766	0.442	0.442	0.391	0.391	0.391	0.391	0.391	0.391	**0.391**
A_{71}-A_{72}	1.563	0.563	0.766	0.563	0.563	0.563	0.563	0.563	0.563	**0.563**
Weight (lb)	393.094	393.380	393.05	391.07	391.23	389.334	389.334	389.334	389.334	**389.334**
CV (%)	None	0.0424	None	0.0076	None	None	None	None	None	**None**
NSA	50000	5330	5370	4500	4620	17010	13200	4600	6250	**3376**
Worst weight (lb)	N/A	N/A	N/A	495.97	N/A	N/A	N/A	393.778	389.457	**389.828**
Mean weight (lb)	N/A	N/A	N/A	403.71	456.69	391.59	N/A	389.941	389.823	**389.495**
SD (lb)	N/A	N/A	N/A	24.8	N/A	N/A	N/A	1.43	0.84	**0.159**

Figure 12 compares the convergence curves obtained for the best optimization runs of DAJA and other metaheuristic methods carried out in Case 1 and Case 2. In the first case (see Fig. 12a), CBO [32] started from more conservative designs than the other algorithms and hence converged more slowly to the optimum solutions. DAJA, TLBO [37] and HHS [40] started from similar populations including almost the same best design. The global best PSO strategy used by HHS made search very fast in the initial design cycles but soon pushed search process near constraint domain boundaries. DAJA and HHS optimization histories practically coincided from 500 to about 1250 structural analyses but DAJA could later exploit its inherent ability to always generate a set of descent directions in the neighborhood of any trial designs. TLBO progressively recovered the gap and behaved like HHS after about 2000 structural analyses.

The advantage deriving from the use of gradient/pseudo-gradient information and approximate line search in discrete sizing optimization of truss structures (as is done by DAJA) appears evident from Fig. 12b which presents convergence curves for Case 2. In fact, DAJA generated better intermediate designs than TLBO [37] throughout the optimization process and required only 750 structural analyses to outperform designs of IMBA[39]although the best design included in the DAJA's initial population was about two times heavier than those ofTLBO and IMBA. CBO [32,33] and ECBO [33] always generated lower quality intermediate designs than DAJA over the whole optimization process either when those referenced algorithms started

optimization search from lighter or heavier designs than the present algorithm. WCA [39] was muchfaster than CBO [33] and ECBO [33] and its convergence history practically coincided with that of DAJA after about 2500 structural analyses.Conversely, CBO and ECBO yet generated almost 15% heavier designs than DAJA after about 3400 analyses, that is when the present algorithm already completed the optimization process.



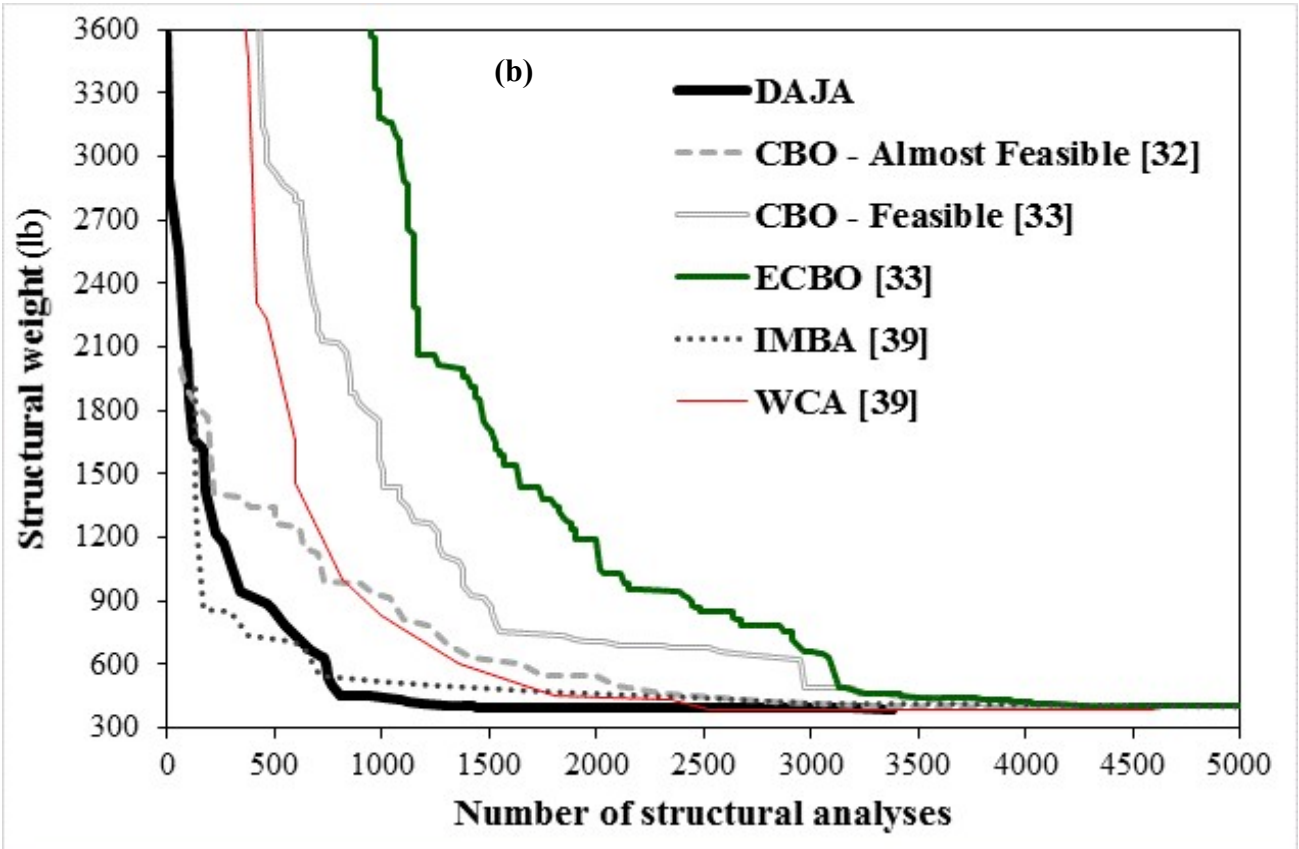


Fig. 12. Comparison of convergence curves for the 72-bar problem: (a) Case 1; (b) Case 2.

4.6. Planar 200-bar truss structure

The sixth structure optimized in this study is the planar 200-bar truss structure shown in Fig. 13. The structure includes 200 elements connected by 77 nodes. The Young's modulus and mass density are 30Msi and 0.283 lb/in^3, respectively. The allowable stress is ±10,000psi in tension/compression andthere are no displacement constraints. Elementsare divided into the 29 groups listed in Table 18. The structure is subject to three independent loading conditions: (1) 1.0 kip acting in the positive *x*-direction at nodes 1, 6, 15, 20, 29, 34, 43, 48, 57, 62 and 71; (2) 10.0 kips acting in the negative *y*-direction at nodes 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 15, 16, 17, 18, 19, 20, 22, 24, 26, 28,29 30, 31, 32, 33, 34,36, 38, 40, 42, 43, 44, 45, 46, 47, 48, 50, 52, 54, 56, 57, 58, 59, 60, 61, 62, 64, 66, 68, 70, 71, 72, 73, 74 and 75; (3) load cases (1) and (2) acting together.

This classical average-scale design example includes 29 discrete sizing variables (corresponding to cross-sectional areas of element groups), which must be selected from the following set: D = [0.1, 0.347, 0.44, 0.539, 0.954, 1.081, 1.174, 1.333, 1.488, 1.764, 2.142, 2.697, 2.8, 3.131, 3.565, 3.813, 4.805, 5.952, 6.572, 7.192, 8.525, 9.3, 10.85, 13.33, 14.29, 17.17, 19.18, 23.68, 28.08, 33.7] (in^2). The total number of available discrete combinations of sizing variables is $7.446 \cdot 10^{43}$ (i.e. 29^{30}).

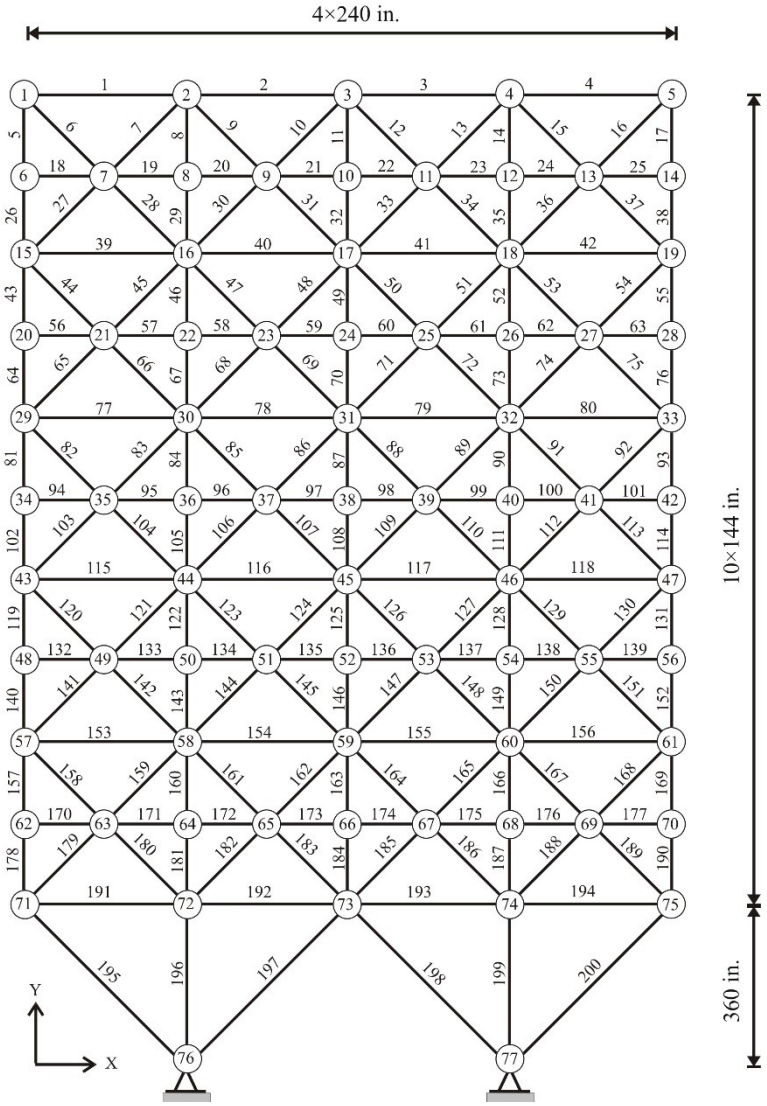


Fig. 14. Schematic of the planar 200-bar truss structure.

Table 18.Variable linking used for the 200-bar truss problem.

Design variables	Member number	Design variables	Member number
1	1,2,3,4	16	82,83,85,86,88,89,91,92, 103,104,106,107,109,110,112,113
2	5,8,11,14,17	17	115,116,117,118
3	19,20,21,22,23,24	18	119,122,125,128,131
4	18,25,56,63,94,101,132, 139,170,177	19	133,134,135,136,137,138
5	26,29,32,35,38	20	140,143,146,149,152
6	6,7,9,10,12,13,15,16,27, 28,30,31,33,34,36,37	21	120,121,123,124,126,127,129, 130,141,142,144,145,147,148, 150,151
7	39,40,41,42	22	153,154,155,156
8	43,46,49,52,55	23	157,160,163,166,169
9	57,58,59,60,61,62	24	171,172,173174,175,176
10	64,67,70,73,76	25	178,181,184,187,190
11	44,45,47,48,50,51,53,54, 65,66,68,69,71,72,74,75	26	158,159,161,162,164,165,167, 168,179,180,182,183,185,186, 188,189
12	77,78,79,80	27	191,192,193,194
13	81,84,87,90,93	28	195,197,198,200
14	95,96,97,98,99,100	29	196,199
15	102,105,108,111,114		

Similar to the 47-bar tower problem, no global optimum solution is available in the literature for this design example. Besides the fairly large number of sizing variables included in these two test cases (respectively, 29 and 27 variables vs. up to 16 variables defined for the 10, 25, 52 and 72-bar trusses), the difficulty for metaheuristic algorithms to find a global optimum for the 200-bar truss problem lies in the fact that the size of the discrete set D is almost the same as the number of sizing variables (i.e. 30 vs. 29). This limits by a great extent the number of candidate designs formed by selecting all different cross-sectional areas (i.e. when a design vector cannot include two identical cross-sectional areas) and consequently affects the exploration phase of the metaheuristic search process if the optimal design does not include many similar cross-sectional areas.

Table 19 compares the optimization results obtained byDAJA, ESASS [36], HHS [40] and GA[59].The present algorithmwas the most efficient optimizer and obtained a feasible design weighing 27,282.57 lb, respectively 2.9% and 4.5% lighter than the feasible designs obtained by GA and ESASS. The lighter design obtained by HHS violatedstress constraints by 12.53% and would be increased to about 30,600 lb by rescaling sizing variables by a factor equal to (1+CV/100). Furthermore, JA required the smallestnumber of structural analyses to complete the

optimization process and was the most robust optimization algorithm. Remarkably, the computational cost of the optimization process was reduced by DAJA by almost 58% with respect to the literature. The success rate achieved by DAJA was rather high also for thisaverage scale discreteoptimization problem. In fact, about 80%of independent optimization runs converged to solutions up to 3% heavier than DAJA's best weight of 27,282.57 lb. Such a dispersion is smaller than the 4.1% ratio between standard deviation and average weight obtained by HHS at the cost of violating stress constraints. It can be concluded that DAJA represents a noticeable advancement with respect to other metaheuristic algorithms also for this design example.

Table 19. Optimization results obtained for the 200-bar truss problem.

Design variables A_i (in^2)	GA [59]	ESASS [36]	HHS [40]	**Present study DAJA**
1	0.347	0.1	0.1	**0.1**
2	1.081	0.954	0.954	**0.954**
3	0.1	0.1	0.1	**0.347**
4	0.1	0.1	0.1	**0.1**
5	2.142	2.142	2.142	**2.142**
6	0.347	0.347	0.347	**0.347**
7	0.1	0.1	0.1	**0.1**
8	3.565	3.131	3.131	**3.131**
9	0.347	0.1	0.1	**0.1**
10	4.805	4.805	4.805	**4.805**
11	0.44	0.347	0.44	**0.44**
12	0.44	0.1	0.347	**0.347**
13	5.952	5.952	5.952	**5.952**
14	0.347	0.1	0.347	**0.1**
15	6.572	6.572	6.572	**6.572**
16	0.954	0.44	0.954	**0.954**
17	0.347	0.539	0.347	**0.1**
18	8.525	7.192	8.525	**8.525**
19	0.1	0.44	0.1	**0.539**
20	9.3	8.525	9.3	**9.3**
21	0.954	0.954	1.081	**0.954**
22	1.764	1.174	0.347	**0.1**
23	13.3	10.85	13.33	**13.33**
24	0.347	0.44	0.954	**0.1**
25	13.3	10.85	13.33	**13.33**
26	2.142	1.764	1.764	**0.954**
27	4.805	8.525	3.813	**5.952**
28	9.3	13.33	8.525	**10.85**
29	17.17	13.33	17.17	**14.29**
Weight (lb)	28,544.014	28,075.488	27,163.59	**27,282.57**
CV (%)	None	None	12.53	**None**
NSA	51,360	11,156	5000	**4693**
Worst weight (lb)	N/A	N/A	N/A	**28,108.61**
Mean weight (lb)	N/A	N/A	28,159.59	**27,878.27**
SD (lb)	N/A	N/A	1149.91	**282.88**

The two-stage JA algorithm started the rounding process from the continuous optimum solution {0.147258; 0.940434; 0.100109; 0.100098; 1.941704; 0.296783; 0.100096; 3.106749; 0.100095; 4.108109; 0.403975; 0.193079; 5.434236; 0.100095; 6.434203; 0.575306; 0.135485; 7.980200; 0.100157; 8.980345; 0.709002; 0.437247; 10.89123; 0.100150; 11.89141; 1.049144; 6.610648; 10.77913; 13.87830} in^2 reported in [40].The discrete solution thus obtained is {0.347; 1.081; 0.1; 0.1; 2.142; 0.347; 0.347; 3.131; 0.1; 4.805; 0.539; 0.1; 5.952; 0.1; 6.572; 0.539; 0.539; 8.525; 0.1; 8.525; 0.954; 0.44; 10.85; 0.1; 13.33; 0.954; 6.572; 10.85; 13.33} in^2. The corresponding structural weight is 26,488.83 lb, but the above reported discrete design violates stress constraints by 4.071%. By relaxing sizing variables by a factor (1+CV) to recover constraint violation, the structural weight would increase to 27,567.19 lb, which is 1.043% larger than the optimum weight of 27,282.57 lb obtained by DAJA. Furthermore, the present algorithm completed the whole optimization process within only 4693 structural analyses while two-stage JA [40] required 31,580 analyses just to perform the continuous optimization. Once again, the present discrete formulation was about one order of magnitude faster than the two-stage optimization JA formulation of Ref. [47]. This confirms the utility of using gradient information in the generation of trial designs.

Figure14compares the convergence curves for the best optimization runs of DAJA, ESASS[36] and HHS [40]. All optimizations started from similar populations including best designs with approximately the same weight. It can be seen that DAJA and ESASS had the same convergence rate over the first 3000 structural analyses but later became considerably less efficient than DAJA. Such a behavior can be explained as follows.ESASS updates population by taking a fraction of the distance between the currently perturbed design and another design randomly selected;an elitist strategy can enhance search by sampling new designs in the neighborhood of the current best record. As optimization cyclesprogress, population becomes more and more condensed about the current optimum and hence distances between pairs of candidate designs become smaller. This limits the search ability of ESASS. Conversely, DAJA can generate feasible directions with no limitation on the available step size to be given to design variables.

Again, the global PSO search strategy implemented by HHS resulted in a high rate of reduction of structural weight in the first optimization cycles but this made intermediate designs turn infeasible while DAJA was able to generate descent directions in the neighborhood of candidate designs always remaining in the feasible design space.

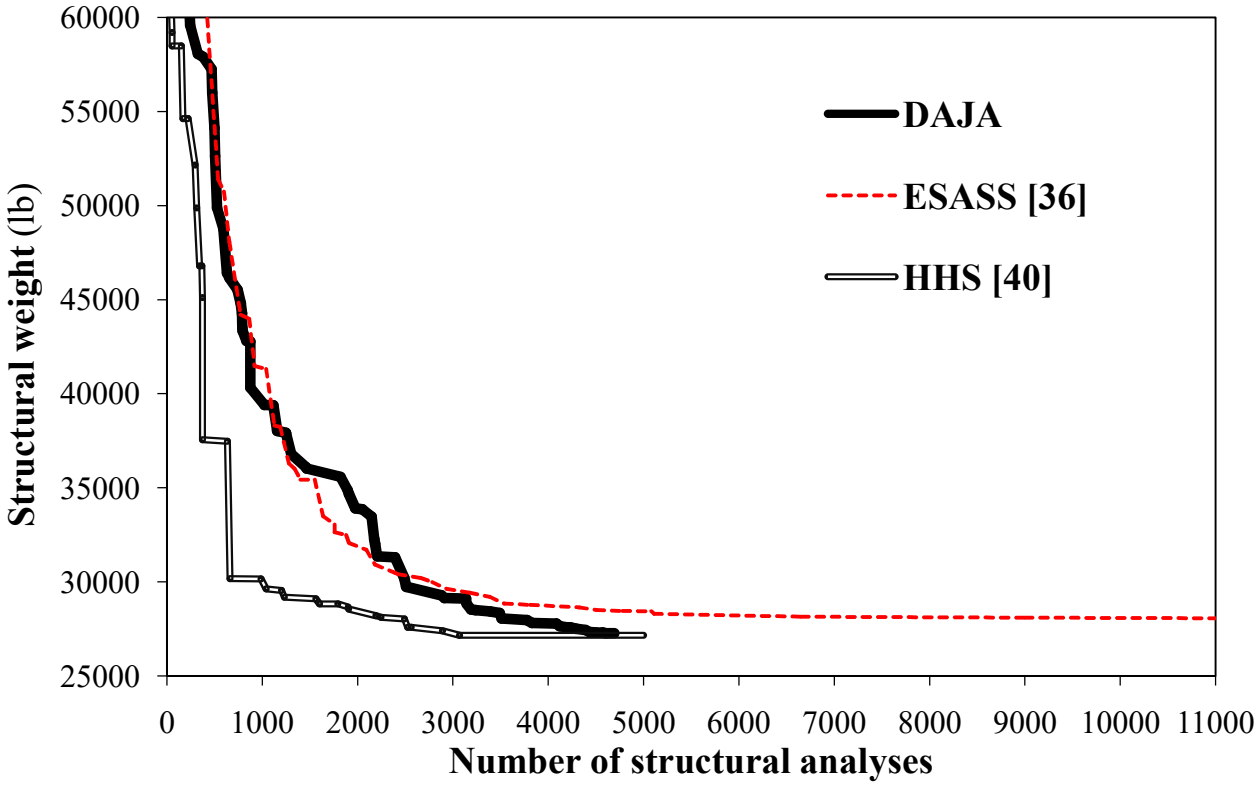


Fig.14. Comparison of convergence curves obtained for the 200-bar truss problem.

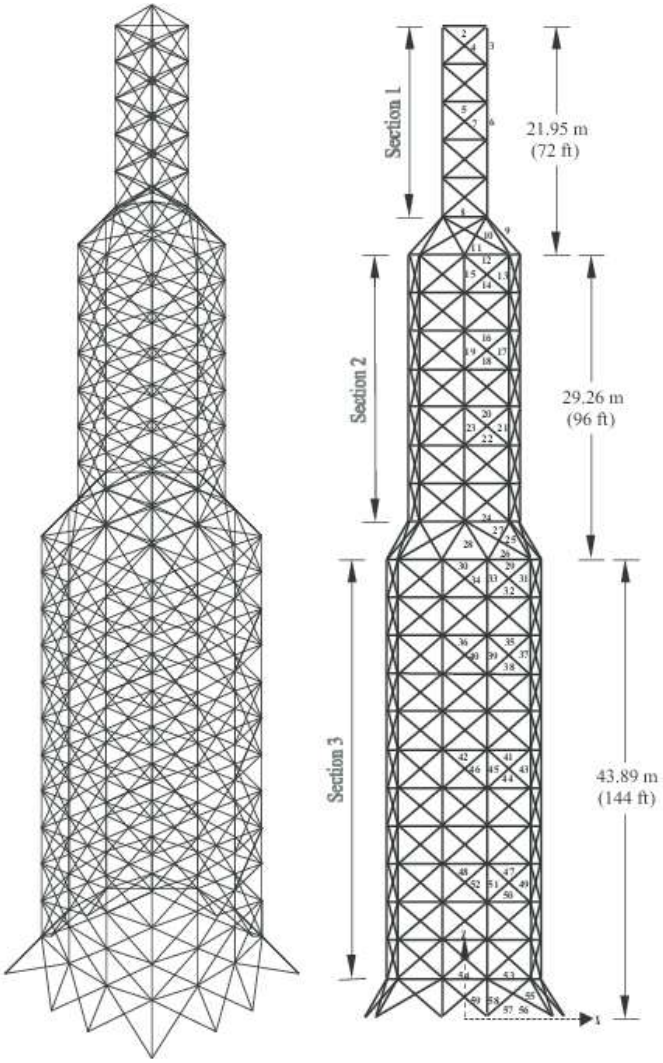
It should be noted that the 200-bar problem is a rather complicated test problem for metaheuristic algorithms also in the case of continuous optimization. This statement is proven by the fact that none of the currently available metaheuristic methods could converge to the target optimum of 25446.7 lb. Discrete optimization makes the problem even more complex and this is confirmed by the presence of several steps also in the convergence curve of DAJA (see Fig. 14).

4.7. Spatial942-bar tower

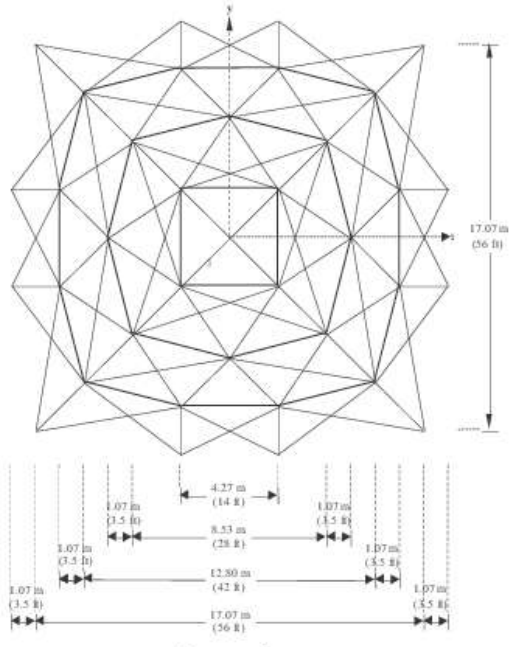
The last design example is the sizing optimization of the spatial 942-bar tower shown in Fig. 15. The structure is made of steel with Young's modulusequal to 29 Msi. Variable linking is used to exploit the symmetry of thestructure about X and Y axes(see group numbering in Fig. 15).Hence, the optimization problem includes 59 discrete sizing variables, which can be selected from a discrete list of295 ready W-sections. The following single loading condition is applied to the structure: (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; (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. The stress and stability constraints are taken from AISC-ASD specifications [49], setting the yield stress equal to 36,000 psi. Displacements of free nodes in allcoordinate directionsmust be less than 15 in, about 1/250 of the total height of the tower (312 ft).

This large scale problemwas previously solvedby other authors using SA [58], a simple genetic algorithm (SGA) [60], evolution strategies (ESs) [60] and a bat inspired algorithm (BI) [61]. The very large number of available discrete combinations of sizing variables for this design example, $2.520 \cdot 10^{522}$ (i.e. 59^{295}), explains the significant dispersion observed in terms of optimized weight and number of structural analyses for the documented solutions. Hence, there is not a target

global optimum quoted in the literature. Furthermore, computational cost of optimization was reported to range between about 41500 and 200,000 structural analyses.



(a) 3-D view (b) side view



(c) top view

Fig. 15. Schematic of the spatial 942-bar tower.

The optimum results of DAJA are compared with those of SA, SGA, ESs and BI in Table 20.Denominations and area sizes of optimized cross-sections are listed in the table. The DAJA algorithm again was the best method in terms of optimized weight and computational cost. In particular, DAJA obtained a structural weight of 377,485 lb while the 2nd best design, obtained by BI, weighs 377,567 lb. This was achieved within one half structural analyses with respect to BI (i.e. 49,097 vs. 95,700). Furthermore, DAJA generated an intermediate feasible design lighter than the optimum design of SA (i.e. 379,630 vs 379,660 lb) after only 37,521 structural analyses vs. the 41,462 analyses required by SA for completing the optimization process.

The standard deviation on optimized weight achieved by DAJA(see also details of independent runs given in Table 1) is less than 1% of mean weight, thus confirming the robustness of DAJA also for this large-scale problem. Interestingly, the rate of success of DAJA was about 80% by considering optimized designs up to 1.4% heavier than the best weight of 377,485 lb. Such a dispersion is similar to the 1.1% ratio between standard deviation and average weight achieved by the 2nd best optimizer, the BI algorithm.

Table 20. Optimization results obtained for the 942-bar tower problem.

Design variables A_i (in^2)	SA [58]	SGA [60]	ESs [60]	BI [61]	Present study DAJA
1	W6×9 (2.68)	W10×22 (6.49)	W6×9 (2.68)	W6×9 (2.68)	**W6×9 (2.68)**
2	W6×9 (2.68)	W6×9 (2.68)	W8×10 (2.96)	W6×9 (2.68)	**W6×9 (2.68)**
3	W6×9 (2.68)	W6×9(2.68)	W6×9 (2.68)	W6×9 (2.68)	**W6×9 (2.68)**
4	W6×15 (4.43)	W6×15 (4.43)	W6×15 (4.43)	W6×15 (4.43)	**W6×15 (4.43)**
5	W6×9 (2.68)	W6×9 (2.68)	W6×9 (2.68)	W6×9 (2.68)	**W6×9 (2.68)**
6	W6×15 (4.43)	W5×19 (5.54)	W6×15 (4.43)	W6×15 (4.43)	**W6×15 (4.43)**
7	W6×15 (4.43)	W5×16 (4.68)	W6×15 (4.43)	W6×15 (4.43)	**W6×15 (4.43)**
8	W6×9 (2.68)	W14×22 (6.49)	W6×9 (2.68)	W6×9 (2.68)	**W6×9 (2.68)**
9	W6×20 (5.87)	W18×50 (14.70)	W6×20 (5.87)	W6×20 (5.87)	**W6×20 (5.87)**
10	W8×24 (7.08)	W8×24 (7.08)	W6×25 (7.34)	W8×24 (7.08)	**W8×24 (7.08)**
11	W6×15 (4.43)	W6×15 (4.43)	W6×15 (4.43)	W6×15 (4.43)	**W6×15 (4.43)**
12	W6×9 (2.68)	W6×9 (2.68)	W6×9 (2.68)	W6×9 (2.68)	**W6×9 (2.68)**
13	W6×20 (5.87)	W10×22 (6.49)	W6×20 (5.87)	W6×20 (5.87)	**W6×20 (5.87)**
14	W6×15 (4.43)	W6×15 (4.43)	W6×15 (4.43)	W6×15 (4.43)	**W6×15 (4.43)**
15	W4×13 (3.83)	W5×16 (4.68)	W4×13 (3.83)	W4×13 (3.83)	**W4×13 (3.83)**
16	W6×9 (2.68)	W6×9 (2.68)	W6×9 (2.68)	W6×9 (2.68)	**W6×9 (2.68)**
17	W8×28 (8.25)	W8×28 (8.25)	W8×28 (8.25)	W8×28 (8.25)	**W8×28 (8.25)**
18	W6×15 (4.43)	W6×15 (4.43)	W6×15 (4.43)	W6×15 (4.43)	**W6×15 (4.43)**
19	W6×15 (4.43)	W6×15 (4.43)	W6×15 (4.43)	W5×16 (4.68)	**W6×15 (4.43)**
20	W6×9 (2.68)	W6×9 (2.68)	W6×9 (2.68)	W6×9 (2.68)	**W6×9 (2.68)**
21	W8×35 (10.30)	W8×35 (10.30)	W8×35 (10.30)	W8×35 (10.30)	**W8×35 (10.30)**
22	W6×20(5.87)	W6×20 (5.87)	W6×20 (5.87)	W6×20 (5.87)	**W6×20 (5.87)**
23	W6×25 (7.34)	W8×31 (9.13)	W8×24 (7.08)	W8×24 (7.08)	**W8×24 (7.08)**
24	W8×35 (10.30)	W12×40 (11.80)	W10×45 (13.30)	W8×35 (10.30)	**W8×35 (10.30)**
25	W10×49 (14.40)	W8×58 (17.10)	W8×58 (17.10)	W10×49 (14.40)	**W10×49 (14.40)**
26	W8×31 (9.13)	W10×33 (9.71)	W8×31 (9.13)	W8×31 (9.13)	**W8×31 (9.13)**
27	W6×15 (4.43)	W6×15 (4.43)	W6×15 (4.43)	W6×15 (4.43)	**W6×15 (4.43)**
28	W8×24 (7.08)	W12×26 (7.65)	W8×24 (7.08)	W8×24 (7.08)	**W8×24 (7.08)**
29	W14×26 (7.69)	W8×24 (7.08)	W6×25 (7.34)	W8×24 (7.08)	**W8×24 (7.08)**
30	W8×21 (6.16)	W14×22 (6.49)	W10×22 (6.49)	W8×21 (6.16)	**W8×21 (6.16)**
31	W12×87 (25.60)	W10×68 (20.00)	W14×90 (26.50)	W27×84 (24.80)	**W27×84 (24.80)**
32	W6×20 (5.87)	W8×24 (7.08)	W6×20 (5.87)	W6×20 (5.87)	**W6×20 (5.87)**
33	W6×20 (5.87)	W6×15 (4.43)	W6×15 (4.43)	W5×19 (5.54)	**W5×19 (5.54)**

34	W6×15 (4.43)	W6×15 (4.43)	W6×15 (4.43)	W6×15 (4.43)	**W6×15 (4.43)**
35	W6×9 (2.68)	W6×9 (2.68)	W6×9 (2.68)	W6×9 (2.68)	**W6×9 (2.68)**
36	W6×9 (2.68)	W6×9 (2.68)	W6×9 (2.68)	W6×9 (2.68)	**W6×9 (2.68)**
37	W14×99 (29.10)	W24×104 (30.60)	W14×99 (29.10)	W14×99 (29.10)	**W14×99 (29.10)**
38	W8×24 (7.08)	W8×24 (7.08)	W8×24 (7.08)	W8×24 (7.08)	**W8×24 (7.08)**
39	W6×15 (4.43)	W6×15 (4.43)	W6×15 (4.43)	W6×15 (4.43)	**W6×15 (4.43)**
40	W6×20 (5.87)	W6×20 (5.87)	W6×20 (5.87)	W6×20 (5.87)	**W6×20 (5.87)**
41	W6×9 (2.68)	W6×9 (2.68)	W6×9 (2.68)	W6×9 (2.68)	**W6×9 (2.68)**
42	W6×9 (2.68)	W4×13 (3.83)	W8×10 (2.96)	W6×9 (2.68)	**W6×9 (2.68)**
43	W24×131 (38.50)	W12×136 (39.90)	W24×131 (38.50)	W24×131 (38.50)	**W24×131 (38.50)**
44	W8×31 (9.13)	W8×31 (9.13)	W8×31 (9.13)	W8×31 (9.13)	**W8×31 (9.13)**
45	W6×15 (4.43)	W6×15 (4.43)	W6×15 (4.43)	W6×15 (4.43)	**W6×15 (4.43)**
46	W8×24 (7.08)	W8×24 (7.08)	W8×24 (7.08)	W8×24 (7.08)	**W8×24 (7.08)**
47	W4×13 (3.83)	W8×18 (5.26)	W4×13 (3.83)	W4×13 (3.83)	**W4×13 (3.83)**
48	W6×9 (2.68)	W6×20 (5.87)	W6×9 (2.68)	W6×9 (2.68)	**W6×9 (2.68)**
49	W14×145 (42.70)	W14×145 (42.70)	W14×145 (42.70)	W14×145 (42.70)	**W14×145 (42.70)**
50	W8×31 (9.13)	W8×31 (9.13)	W8×31 (9.13)	W8×31 (9.13)	**W8×31 (9.13)**
51	W8×28 (8.25)	W6×20 (5.87)	W12×30 (8.79)	W8×28 (8.25)	**W8×28 (8.25)**
52	W8×24 (7.08)	W8×31 (9.13)	W8×24 (7.08)	W8×24 (7.08)	**W8×24 (7.08)**
53	W10×60 (17.60)	W14×61 (17.90)	W12×65 (19.10)	W12×65 (19.10)	**W12×65 (19.10)**
54	W24×68 (20.10)	W8×48 (14.10)	W21×73 (21.5)	W21×73 (21.5)	**W21×73 (21.5)**
55	W14×132 (38.80)	W14×120 (35.30)	W14×132 (38.80)	W14×132 (38.80)	**W14×132 (38.80)**
56	W8×35 (10.30)	W8×31 (9.13)	W8×31 (9.13)	W8×31 (9.13)	**W8×31 (9.13)**
57	W12×79 (23.20)	W10×100 (29.40)	W12×72 (21.10)	W12×72 (21.10)	**W12×72 (21.10)**
58	W8×24 (7.08)	W10×33 (9.71)	W8×28 (8.25)	W8×28 (8.25)	**W8×28 (8.25)**
59	W8×35 (10.30)	W10×33 (9.71)	W8×31 (9.13)	W8×31 (9.13)	**W8×31 (9.13)**
Weight (lb)	379,660	394,321	377,947	377,567	**377,485** [a]
CV (%)	None	None	0.07	None	**None**
NSA	41,462	200,000	150,000	95,700	**49,097**
Worst Weight (lb)	N/A	N/A	N/A	N/A	**385,455**
Mean Weight (lb)	N/A	N/A	N/A	382,017	**380,679**
SD (lb)	N/A	N/A	N/A	3970	**2666.5**

[a] JA found an intermediate design weighing 379,660 lb (less than the optimum weight of SA) after only 37521 structural analyses.

The convergence curves of the best optimization runs of DAJA, SA [58] and ESs [60] are compared in Fig. 16. The number of structural analyses and weight of intermediate designs shown in the plot axes are respectively limited to 80,000 and 900,000 lb for the sake of clarity. SA started the optimization process from a very conservative design, weighing almost 3.5 times as the best designs included in the initial populations of DAJA and ESs (i.e. about $2.5 \cdot 10^6$ vs. about $7\text{-}8 \cdot 10^5$ lb). SA recovered this gap within about 22,000 structural analyses but it then improved marginally weight from 30,000 structural analyses to the end of optimization process. This happened because the SA formulation used in [58] operated on a single design at a time also perturbingonly one sizing variable at a time. ESs was the slowest optimizer overall and always generated heavier intermediate designs than DAJA and SA: after 50,000 structural analyses, ESs found an intermediate design still weighing about 388,600 lb while DAJA and SA already converged to their optimum designs weighing less than 378,000 lb. This could be due to the presence of many search operators that introduce a considerable amount of heuristics in ESs. Such a limitation affected at a smaller (but still significant) extent also the convergence rate of the BI algorithm [61]. Interestingly, both ESs and BI operated with a much larger population than DAJA: respectively, 150 offspring and 50 individuals.However, DAJA was able to better explore design space even operating on only 20

candidate designs. In summary, DAJA was definitely the fastest optimizer also in this large-scale problem.

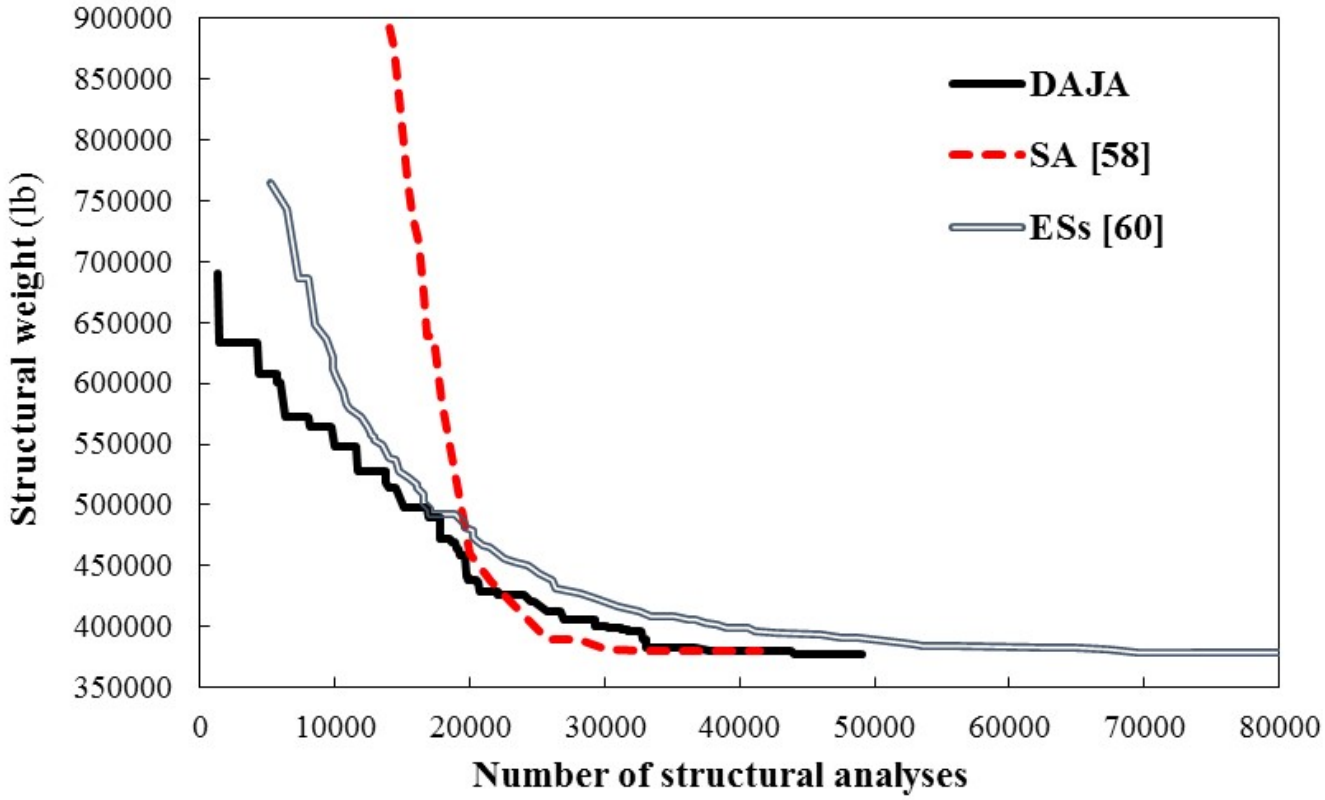


Fig. 16. Comparison of convergence curves obtained for the 942-bar truss problem.

5. Conclusions

This paper presented an advanced formulation ofthe Jaya algorithm for discrete optimization of truss structures including sizing, layout and topology design variables. The new algorithm (discrete advanced JA or DAJA) attempts to improve population by generating a set of descent directions in the neighborhood of each candidate design. For that purpose, explicit gradient information (if available) or pseudo-gradient information (in the general case) and approximate line search strategies are utilized to perturb design variables, rounding the values given by the JA search engine.

DAJA was tested on several classical weight minimization problems formulated forseven truss structures. The largest test case included 59 discrete sizing variables while the second largest design example included 27 discrete sizing variables, 17 discrete layout variables and 27 topology variables (to determine if elements should be maintained or removed). The present algorithmalways converged to the target optimum designs quoted in the literature and, in the most complicated test problems, found better solutions than other methods. Remarkably,DAJA always required much less structural analyses than other optimization methods and was definitely more robust than its competitors in most design examples. The direct discrete optimization search performed by DAJA allowed computational cost to be reduced on average by one order of magnitude with respect to multi-stage JA continuous-discrete optimization[47].

DAJA's formulation can easily be adapted to other engineering problems. For example, we successfully tested DAJA in the weight minimization of a frame structure and a gravity dam. Remarkably, DAJA resulted highly competitive with other hybrid algorithms that use SA, HS and

BBBC as metaheuristic search engines [62].Further studies are currently conducted for using DAJA in discrete optimization of structures with dynamic constraints.

References

[1] Glover F, Kochenberger GA.*Handbook of Metaheuristics*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2003.

[2] Holland JH. *Adaptation in Natural and Artificial Systems*. Ann Arbor: The University of Michigan Press; 1975.

[3] Goldberg DE. *Genetic Algorithms in Search, Operation and Machine Learning*, Addison-Wesley, Reading (MA), USA, 1989.

[4] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing, Science 1983; 220: 671-680.

[5] Van Laarhoven PJM, Aarts EHL. *Simulated Annealing: Theory and Applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1987.

[6] Eberhart RC, Kennedy J. A new optimizer using particle swarm theory. In: Proceedings of The Sixth International Symposium on Micro Machine and Human Science, Nagoya (Japan) 1995, Vol. 4, pp. 1942-1948.

[7] Karaboga D. *An Idea Based on Honey Bee Swarm for Numerical Optimization*. Technical report-TR06 Erciyes University Engineering Faculty Computer Engineering Department, Kayseri (Turkey), 1995.

[8] Storn R, Price K. *Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces*, Technical Report No. TR-95-012, International Computer Science Institute, Berkley (CA), USA, 1995.

[9] Storn R, Price K. Differential evolution- a simple and efficient heuristic for global optimization over continuous spaces.J Glob Optim 1997; 11:341-359.

[10] Dorigo M, Maniezzo V, Colorni A. The ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern B 1996; 26: 29–41.

[11] Dorigo M, Stutzle T. *Ant Colony Optimization*, MIT Press, Cambridge (MA), USA, 2004.

[12] Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: harmony search. Simulation 2001; 76: 60-68.

[13] Nolle L. On a hill-climbing algorithm with adaptive step size: towards a control parameter-less black-box optimisation algorithm.In: B. Reusch (Ed.) *Computational Intelligence Theory and Applications,* pp. 587-595, 2006.

[14] Erol OK, Eksin I. A new optimization method: big bang-big crunch. Adv Eng Softw 2006; 37: 106-111.

[15] Yang XS.*Engineering Optimization: An Introduction with Metaheuristic Applications*, John Wiley and Sons, 2010.

[16] Rao RV, Savsani VJ, Vakharia DP. Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems.Comput Aided Des 2011; 43: 303-315.

[17] Sadollah A, Bahreininejad A, Eskandar H, Hamdi M. Mine blast algorithm for optimization of truss structures with discrete variables.Comput Struct 2012; 102-103: 49-63.

[18] Neshat M, Sepidnam G, Sargolzaei M. Swallow swarm optimization algorithm: a new method to optimization.Neural Comput Appl 2013; 23: 429-454.

[19] Civicioglu P. Backtracking search optimization algorithm for numerical optimization problems. Appl Math Comput 2013; 219: 8121-8144.

[20] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Adv Eng Softw 2014; 69, 46-61.

[21] Cheng MY, Prayogo D. Symbiotic organisms search: a new metaheuristic optimization algorithm. Comput Struct 2014; 139: 98-112.

[22] Kaveh A, Mahdavi VR. Colliding bodies optimization: a novel meta-heuristic method. Comput

Struct 2014; 139: 18-27.
[23] Kaveh A, Bakhshpoori T. Water Evaporation Optimization: A novel physically inspired optimization algorithm. Comput Struct 2016; 167: 69-85.
[24] Lamberti L, Pappalettere C. Metaheuristic design optimization of skeletal structures: a review. Computational Technology Reviews 2011; 4: 1-32.
[25] Kaveh A. *Advances in Metaheuristic Algorithms for Optimal Design of Structures*, Springer International Publishing, Switzerland, 2014.
[26] Lee KS, Geem ZW, Lee SH, Bae KW. The harmony search algorithm for discrete structural optimization.Eng Optimiz 2005;37: 663-684.
[27] Li LJ, Huang ZB, Liu F. A heuristic particle swarm optimization method for truss structures with discrete variables.Comput Struct 2009; 87: 435-443.
[28] Kaveh A, Talatahari S. A particle swarm ant colony optimization for truss structures with discrete variables.J Constr Steel Res 2009; 65: 1558-1568.
[29] Kaveh A, Takatahari S. A discrete big bang – big crunch algorithm for optimal design of skeletal structures. Asian J Civil Eng 2010; 11: 103-122.
[30] Kaveh A, Takatahari S. A charged system search with a fly to boundary method for discrete optimum design of truss structures. Asian J Civil Eng 2010; 11: 277-293.
[31] Sonmez M. Discrete optimum design of truss structures using artificial bee colony algorithm.Struct Multidisc Opt 2011; 43, 85-97.
[32] Kaveh A, Mahdavi VR. Colliding Bodies Optimization method for optimum discrete design of truss structures.Comput Struct 2014; 139: 43-53.
[33] Kaveh A, Ilchi Ghazaan M. A comparative study of CBO and ECBO for optimal design of skeletal structures. Comput Struct 2015; 153: 137-147.
[34] Kaveh A, Talatahari S. Optimum design of skeletal structure using imperialist competitive algorithm. Comput Struct 2010; 88: 1220-1229.
[35] Kaveh A, Ilchi Ghazaan M, Bakhshpoori T. An improved ray optimization algorithm for design of truss structures. Period Polytech 2013; 57: 1-15.
[36] Azad SK, Hasançebi O. An elitist self-adaptive step-size search for structural design optimization, Appl Soft Comput 2014; 19: 226-235.
[37] Dede T. Application of teaching-learning-based-optimization algorithm for the discrete optimization of truss structures.KSCE J Civil Eng 2014; 18: 1759-1767.
[38] Baghlani A, Makiabadi MH, Sarcheshmehpour M. Discrete optimum design of truss structures by an improved firefly algorithm.Adv Struct Eng 2014; 17: 1517-1530.
[39] Sadollah A, Eskandar H, Bahreininejad A, Kim JH. Water cycle, mine blast and improved mine blast algorithms for discrete sizing optimization of truss structures.Comput Struct 2015; 149: 1-16.
[40] Cheng MY, Prayogo D, Wu YW, Lukito MM. A Hybrid Harmony Search algorithm for discrete sizing optimization of truss structure. Auto Constr 2016; 69: 21-33.
[41] Ho-Huu V, Nguyen-Thoi T, Vo-Duy T, Nguyen-Trang T. An adaptive elitist differential evolution for truss optimization with discrete variables. Comput Struct 2016, 165: 59-75.
[42] Kazemzadeh Azad S. Seeding the initial population with feasible solutions in metaheuristic optimization of steel trusses. Eng Optimiz 2018; 55: 89-105.
[43] Kazemzadeh Azad S. Enhanced hybrid metaheuristic algorithms for optimal sizing of steel truss structures with numerous discrete variables. Struct Multidiscip Optim 2017; 55: 2159-2180.
[44] Rao RV. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int J Ind Eng Comp 2016; 7: 19–34.
[45] Rao RV, Waghmare GG. A new optimization algorithm for solving complex constrained design optimization problems. Eng Optimiz 2017; 49: 60-83.
[46] Rao RV, More KC Taler J, Oclon P. Dimensional optimization of a micro-channel heat sink using Jaya algorithm. Appl Therm Eng 2016; 103: 572-582.

[47] Degertekin SO, Lamberti L, Ugur IB. Sizing, layout and topology design optimization of truss structures using the Jaya algorithm. Appl Soft Comp 2018; 70: 903-928.
[48] Kazemzadeh Azad S, Hasancebi O, Kazemzadeh Azad S, Erol OK. Upper bound strategy in optimum design of truss structures: a big bang-big crunch algorithm based application.Adv Struct Eng 2013; 16: 1035-1046.
[49] AISC-ASD.*Manual of Steel Construction Allowable Stress Design*, 9th Ed. American Institute of Steel Construction, Chicago (IL), USA, 1989.
[50] Camp CV, Bichon BJ. Design of space trusses using ant colony optimization. ASCE J Struct Eng 2004; 130:741-751.
[51] Camp CV. Design of space trusses using big bang–big crunch optimization. ASCE J Struct Eng 2007;133:999-1008.
[52] Prayogo D, Cheng MY, Wu YW, Herdany AA, Prayogo H. Differential Big Bang - Big Crunch algorithm for construction-engineering design optimization. Auto Constr 2018; 85: 290-304.
[53] Tang W, Tong L, Gu, Y. Improved genetic algorithm for design optimization of truss structures with sizing, shape and topology variables. Int J Numer Methods Eng 2005; 62:1737-1762.
[54] Rahami H, Kaveh A, Gholipour Y. Sizing, geometry and topology optimization of trusses via force method and genetic algorithm. Eng Struct 2008; 30: 2360-2369.
[55] Miguel Leandro FF, Lopez RH, Miguel Leticia FF. Multimodal size, shape, and topology optimization of truss structures using the Firefly algorithm. Adv Eng Soft 2013; 56: 23-37.
[56] Ahrari A, Atai AA, Deb K. Simultaneous topology, shape and size optimization of truss structures by fully stressed design based on evolution strategy. Eng Optimiz 2015; 47: 1063-1084.
[57] Hasançebi O, Erbatur F. Layout optimization of trusses using improved GA methodologies. Acta Mech 2001; 146: 87-107.
[58] Hasançebi O, Erbatur F. On efficient use of simulated annealing in complex structural optimization problems. Acta Mech 2002; 157: 27-50.
[59] Togan V, Daloglu AT. An improved genetic algorithm with initial population strategy and self-adaptive member grouping.Comput Struct 2008; 86: 1204–1218.
[60] Hasançebi O. Adaptive evolution strategies in structural optimization: enhancing their computational performance with applications to large-scale structures. Comput Struct 2008;86: 119-132.
[61] Hasancebi, O, Teke, T, Pekcan O. A bat-inspired algorithm for structural optimization. Comput Struct 2013;128: 77-90.
[62] Ficarella E, Lamberti L, Degertekin SO. Comparison of three novel hybrid metaheuristic algorithms for non-smooth engineering optimization problems. To be submitted to Comput. Methods Appl. Mech. Engrg.