INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2011; 2:233-256

THE CMA EVOLUTION STRATEGY BASED SIZE OPTIMIZATION OF TRUSS STRUCTURES

A. Kaveh[*, †], M. Kalateh-Ahani and M.S. Masoudi
Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Narmak, Tehran-16, Iran

ABSTRACT

Evolution Strategies (ES) are a class of Evolutionary Algorithms based on Gaussian mutation and deterministic selection. Gaussian mutation captures pair-wise dependencies between the variables through a covariance matrix. Covariance Matrix Adaptation (CMA) is a method to update this covariance matrix. In this paper, the CMA-ES, which has found many applications in solving continuous optimization problems, is employed for size optimization of steel space trusses. Design examples reveal competitive performance of the algorithm compared to the other advanced metaheuristics.

Received: 15 March 2011; Accepted: 7 October 2011

KEY WORDS: The covariance matrix adaptation evolution strategy; metaheuristics; space truss structures; size optimization

1. INTRODUCTION

In the modern era, where resources are often severely limited, there is no company not involved in solving cost optimization problems. This optimization involves designing a project doing the most with consuming the least materials, energy and time. Since real-life optimization problems are often complex and difficult to solve, developing new solution strategies is always a challenging topic in engineering.

As the material cost is one of the major factors in final construction cost of a building, engineers are often asked to seek for a design that efficiently fulfills the requirements of the building codes with utilizing the least volume of materials. In other words, owners and

[*]Corresponding author: A. Kaveh, School of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran-16, Iran
[†]E-mail address: alikaveh@iust.ac.ir

designers always prefer to have optimal structures. Over the last two decades, various metaheuristics have been developed for structural optimization including Evolutionary Algorithms [1-14], Ant Colony Optimization [15-22], Particle Swarm Optimizer [23-25], Harmony Search [26-28], Big Bang-Big Crunch Optimization [29] and Charged System Search algorithm [30].

Truss optimization is one of the most active fields in structural mechanics. Size optimization of truss structures looks for optimum values of member cross-sectional areas that minimize the structural weight. This optimal solution should also satisfy the inequality constraints that limit design variable sizes (member cross-sectional areas) and structural responses (member stresses and nodal displacements).

The metaheuristic developed in this paper, belongs to a class of Evolution Strategies. In the nineteenth century, Mendel was the first to state the preliminary concepts of heredity from parents to offsprings [31]. Then in 1859, Darwin presented the theory of evolution in his famous book *On the Origin of Species* [32]. In the 1980s, these theories of creation of new species and their evolution have inspired computer scientists in designing Evolutionary Algorithms (EA). Different types of EAs have evolved independently during the past 40 years: Genetic Algorithms [33], Evolution Strategies [34], Evolutionary Programming [35], and Genetic Programming [36]. Each of these constitutes a different approach; however, they are inspired by the same principles of natural evolution. EAs are the most studied population-based metaheuristics and this has promoted the field known as Evolutionary Computation [31].

The CMA-ES is a stochastic method for continuous optimization of non-linear, non-convex problems, which was first introduced by Hansen et al. in 1995 [37]. In an ES, new candidate solutions are sampled according to a multivariate normal distribution. Pair-wise dependencies between the variables in this distribution are described by a covariance matrix. The CMA is a method to update the covariance matrix of this distribution. The CMA-ES is a second-order optimization approach, where only the ranking between candidate solutions is exploited for learning the sample distribution, and neither derivatives nor even the objective function values are required by the method. This makes the method feasible on ill-conditioned and non-continuous problems, as well as on multimodal or noisy problems [31].

After this section, the paper is organized as follows. Section 2 introduces the CMA-ES. In section 3, objective function and constraints of the problem are formulated. Section 4 studies various design examples to verify the efficiency of the algorithm. A discussion is provided in section 5 and finally the paper is concluded with section 6.

2. THE CMA-ES

In what follows, a brief description of the CMA-ES is presented. For more information about terminology and details, interested readers may refer to [38]. A summary of the algorithm together with a table of the default strategy parameters and their values are provided in Appendix A. First, a few required fundamentals are explained.

2.1. Preliminaries

2.1.1. Eigendecomposition of a positive definite matrix

A symmetric positive definite matrix, $\boldsymbol{C} \in R^{n\times n}$, is characterized by; for each $x \in R^n \backslash \{0\}$ holds $x^{\mathrm{T}} \boldsymbol{C}\, x > 0$. The matrix $\boldsymbol{C}$ has an orthonormal basis of eigenvectors, $\boldsymbol{B} = [b_1,\ldots,b_n]$, with corresponding eigenvalues, $d_1^2,\ldots,d_n^2 > 0$. The eigendecomposition of $\boldsymbol{C}$ is defined as

$$\boldsymbol{C} = \boldsymbol{B}\, \boldsymbol{D}^2\, \boldsymbol{B}^{\mathrm{T}} \tag{1}$$

where

$\boldsymbol{B}$ is the eigenvector matrix of $\boldsymbol{C}$;

$\boldsymbol{D}$ = diag $(d_1,\ldots, d_n)$, a diagonal matrix with square roots of eigenvalues of $\boldsymbol{C}$ as diagonal entries.

The matrix decomposition (1) is unique, apart from signs of columns of $\boldsymbol{B}$ and permutations of columns in $\boldsymbol{B}$ and $\boldsymbol{D}^2$, provided all the eigenvalues are distinct.

2.1.2. Multivariate Normal Distribution

A multivariate normal distribution, $\mathcal{N}(\boldsymbol{m}, \boldsymbol{C})$, has a unimodal bell-shaped density, where the top of the bell (the modal value) corresponds to the distribution mean, $\boldsymbol{m}$. The distribution $\mathcal{N}(\boldsymbol{m}, \boldsymbol{C})$ is uniquely determined by its mean $\boldsymbol{m} \in R^n$ and its symmetric and positive definite covariance matrix $\boldsymbol{C} \in R^{n\times n}$ Covariance matrices have an appealing geometrical interpretation; they can be uniquely identified with an (hyper-) ellipsoid $\{\boldsymbol{x} \in R^n \mid \boldsymbol{x}^{\mathrm{T}} \boldsymbol{C}^{-1} \boldsymbol{x} = 1\}$, as shown in Figure 1.

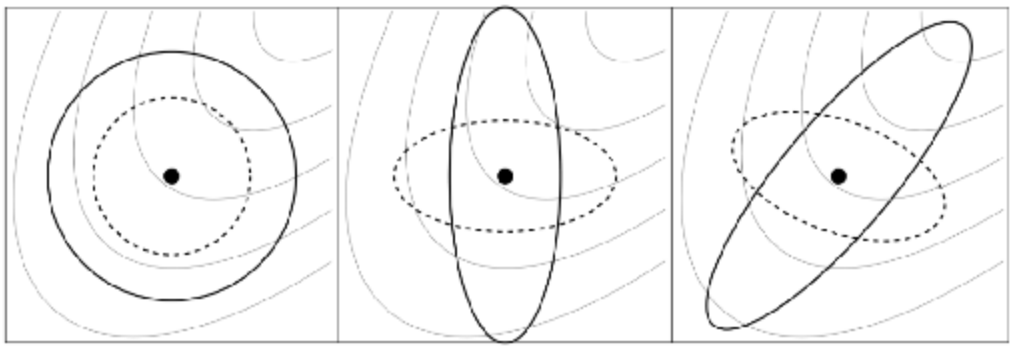

Figure 1. Ellipsoids depicting one–σ lines of equal density of six different normal distributions, where $\sigma \in R_+$, $\boldsymbol{D}$ is a diagonal matrix, and $\boldsymbol{C}$ is a positive definite covariance matrix. Gray lines depict possible objective function contour lines

The ellipsoid is a surface of equal density of the distribution. The principal axes of the ellipsoid correspond to the eigenvectors of $\boldsymbol{C}$, the squared axes lengths correspond to the eigenvalues. As mentioned, the eigendecomposition is denoted by $\boldsymbol{C} = \boldsymbol{B}\, \boldsymbol{D}^2\, \boldsymbol{B}^{\mathrm{T}}$. If $\boldsymbol{D} = \sigma\, \mathbf{I}$, where $\sigma \in R_+$ and $\mathbf{I}$ denotes the identity matrix, $\boldsymbol{C} = \sigma^2\, \mathbf{I}$ and the ellipsoid is isotropic (Figure

1, left). If $\boldsymbol{B} = \mathbf{I}$, then $\boldsymbol{C} = \boldsymbol{D}^2$ is a diagonal matrix and the ellipsoid is axis parallel oriented as is shown in Figure 1, middle. In the coordinate system given by the columns of $\boldsymbol{B}$, the distribution $\mathcal{N}(0, \boldsymbol{C})$ is always uncorrelated (Figure 1, right).

The normal distribution $\mathcal{N}(\boldsymbol{m}, \boldsymbol{C})$ can be written in different ways:

$$\begin{aligned} \mathcal{N}(\boldsymbol{m}, \boldsymbol{C}) &\sim \boldsymbol{m} + \mathcal{N}(0, \boldsymbol{C}) \\ &\sim \boldsymbol{m} + \boldsymbol{C}^{1/2}\, \mathcal{N}(0, \mathbf{I}) \\ &\sim \boldsymbol{m} + \boldsymbol{B}\,\boldsymbol{D}\,\boldsymbol{B}^{\mathrm{T}}\, \mathcal{N}(0, \mathbf{I}) \\ &\sim \boldsymbol{m} + \boldsymbol{B}\,\boldsymbol{D}\, \mathcal{N}(0, \mathbf{I}) \end{aligned} \tag{2}$$

where "~" denotes equality in distribution. The last row can be well interpreted, from right to left

$\mathcal{N}(0, \mathbf{I})$ produces an spherical (isotropic) distribution as in Figure 1, left.

$\boldsymbol{D}$ scales the spherical distribution within the coordinate axes as in Figure 1, middle.

$\boldsymbol{B}$ defines a new orientation for the ellipsoid, where the new principal axes of the ellipsoid correspond to the columns of $\boldsymbol{B}$ (Figure 1, right).

Equation (2) is useful to compute $\mathcal{N}(\boldsymbol{m}, \boldsymbol{C})$ distributed vectors, because $\mathcal{N}(0, \mathbf{I})$ is a vector of independent (0, 1)-normally distributed numbers that can easily be realized on a computer.

2.2. Basic equation: sampling

In the CMA-ES, a population of new search points (individuals, offspring) is generated by sampling a multivariate normal distribution. At each generation, the basic equation for sampling is

$$\boldsymbol{x}_k^{(g+1)} \sim \boldsymbol{m}^{(g)} + \sigma^{(g)}\, \mathcal{N}(0, \boldsymbol{C}^{(g)}) \qquad \text{for } k = 1, \ldots, \lambda \tag{3}$$

where

$\mathcal{N}(0, \boldsymbol{C}^{(g)})$ is a multivariate normal distribution with zero mean and covariance matrix $\boldsymbol{C}^{(g)}$.

$\boldsymbol{x}_k^{(g+1)} \in R^n$, k-th offspring from generation $g + 1$.

$\boldsymbol{m}^{(g)} \in R^n$, mean value of the search distribution at generation g.

$\sigma^{(g)} \in R_+$, overall standard deviation, step-size, at generation g.

$\boldsymbol{C}^{(g)} \in R^{n \times n}$, covariance matrix at generation g.

$\lambda \geq 2$, population size, number of offspring.

To define the complete iteration step, the remaining question is, how to calculate $\boldsymbol{m}^{(g+1)}$, $\boldsymbol{C}^{(g+1)}$ and $\sigma^{(g+1)}$ for the next generation $g + 1$. The next three sections will answer these questions, respectively.

2.3. Selection and recombination

The new mean $\boldsymbol{m}^{(g+1)}$ of the search distribution is a weighted average of μ selected points

from the sample:

$$\boldsymbol{m}^{(g+1)} = \sum_{i=1}^{m} w_i \boldsymbol{x}_{i:l}^{(g+1)} \tag{4}$$

where

$\mu \leq \lambda$ is the parent population size, i.e. the number of selected points.

$w_i \in R_+$, positive weight coefficients for recombination. $\sum_{i=1}^{m} w_i = 1$ and $w_1 \geq w_2 \geq \ldots \geq w_m \geq 0$.

$\boldsymbol{x}_{i:l}^{(g+1)}$, i-th best individual out of $\boldsymbol{x}_1^{(g+1)}, \ldots, \boldsymbol{x}_l^{(g+1)}$ from (3). The index $i:\lambda$ denotes the index of the i-th ranked individual and $f(\boldsymbol{x}_{1:l}^{(g+1)}) \leq f(\boldsymbol{x}_{2:l}^{(g+1)}) \leq \ldots \leq f(\boldsymbol{x}_{l:l}^{(g+1)})$, where f is the objective function to be minimized.

The measure $\mu_{eff} = (\sum_{i=1}^{m} w_i)^{-1}$ will be used in the following and can be paraphrased as variance effective selection mass. From the definition of w_i in (4), $1 \leq \mu_{eff} \leq \mu$ is derived.

2.4. Adapting the covariance matrix

The CMA-ES is based on two adaptation principles, which make it an efficient procedure for multimodal continuous problems. First, a maximum-likelihood principle, based on the idea to increase the probability of successful candidate solutions and search steps. For this purpose, the algorithm updates the covariance matrix of the distribution such that the likelihood of already applied successful steps is increased. Rank-μ-update performs this principle [31].

Second, an evolution path principle, based on memorizing the time evolution path of the distribution mean. These paths contain significant information about the correlation between consecutive steps. The evolution paths are exploited in two ways. One path is used for the covariance matrix adaptation procedure and facilitates a possibly much faster variance increase of favorable directions. Rank-one-update performs it. The other path is used to conduct an additional step-size control that effectively prevents premature convergence yet allowing faster convergence (see section 2.5) [31].

2.4.1. Rank-μ-update

Choosing $\boldsymbol{C}^{(0)}$ to be the unity matrix, then the new covariance matrix $\boldsymbol{C}^{(g+1)}$ is given by

$$\begin{aligned} \boldsymbol{C}^{(g+1)} &= (1-c_m)\boldsymbol{C}^{(g)} + c_m \frac{1}{s^{(g)^2}} \boldsymbol{C}_m^{(g+1)} \\ &= (1-c_m)\boldsymbol{C}^{(g)} + c_m \sum_{i=1}^{m} w_i \, \boldsymbol{y}_{i:l}^{(g+1)} \boldsymbol{y}_{i:l}^{(g+1)^{\mathrm{T}}} \end{aligned} \tag{5}$$

where

$0 \leq c_m \leq 1$ is learning rate for updating the covariance matrix. For $c_m = 1$, no prior

information is retained and For c_m = 1, no learning takes place.

$\boldsymbol{y}_{i:l}^{(g+1)} = (\boldsymbol{x}_{i:l}^{(g+1)} - \boldsymbol{m}^{(g)}) / s^{(g)}$, the selected steps.

This covariance matrix update is called rank-μ-update, because the sum of outer products in (5) is of rank μ. The number $1/c_m$ is the backward time horizon, which says approximately 37% of the information in $\boldsymbol{C}^{(g+1)}$ is older than the last $1/c_m$ generations.

2.4.2. Rank-one-update

A sequence of successive steps, the strategy takes over a number of generations, is called an evolution path. An evolution path can be expressed by a sum of consecutive steps. This summation is referred to as cumulation. The evolution path of the distribution mean is expressed by[1]

$$\boldsymbol{p}_c^{(g+1)} = (1-c_c)\boldsymbol{p}_c^{(g)} + \sqrt{\boldsymbol{c_c}(2-c_c)m_{eff}}\,\frac{\boldsymbol{m}^{(g+1)} - \boldsymbol{m}^{(g)}}{s^{(g)}} \tag{6}$$

where

$\boldsymbol{p}_c^{(g)} \in R^n$ is the evolution path at generation g and $\boldsymbol{p}_c^{(0)} = 0$.

c_c , learning rate for updating the evolution path.

The rank-one-update of the covariance matrix $\boldsymbol{C}^{(g)}$ via the evolution path $\boldsymbol{p}_c^{(g+1)}$ is defined as

$$\boldsymbol{C}^{(g+1)} = (1-c_1)\boldsymbol{C}^{(g)} + c_1 \boldsymbol{p}_c^{(g+1)} \boldsymbol{p}_c^{(g+1)^{\mathrm{T}}} \tag{7}$$

where

c_1 is the learning rate for the rank one update.

2.4.3. Combining Rank-μ-update and cumulation

The final CMA update of the covariance matrix combines (5) and (7):

$$\boldsymbol{C}^{(g+1)} = (1-c_1-c_m)\boldsymbol{C}^{(g)} + c_1 \boldsymbol{p}_c^{(g+1)} \boldsymbol{p}_c^{(g+1)^{\mathrm{T}}} + c_m \sum_{i=1}^{m} w_i \, \boldsymbol{y}_{i:l}^{(g+1)} \boldsymbol{y}_{i:l}^{(g+1)^{\mathrm{T}}} \tag{8}$$

Equation (8) combines the advantages of (5) and (7). On the one hand, the information within the population of one generation is used efficiently by the Rank-μ-update. On the other hand, information of correlations between generations is exploited by using the evolution path for the rank-one update. The former is important in large populations, the latter is in particular important in small populations.

Figure 2 demonstrates the concept behind the covariance matrix adaptation in the CMA-ES algorithm. As the generations progress, the algorithm approaches the global optimum while simultaneously the distribution shape adapts to an ellipsoidal landscape and the search is directed along an evolution path.

[1] In the final algorithm (6) is slightly modified, see Appendix A.

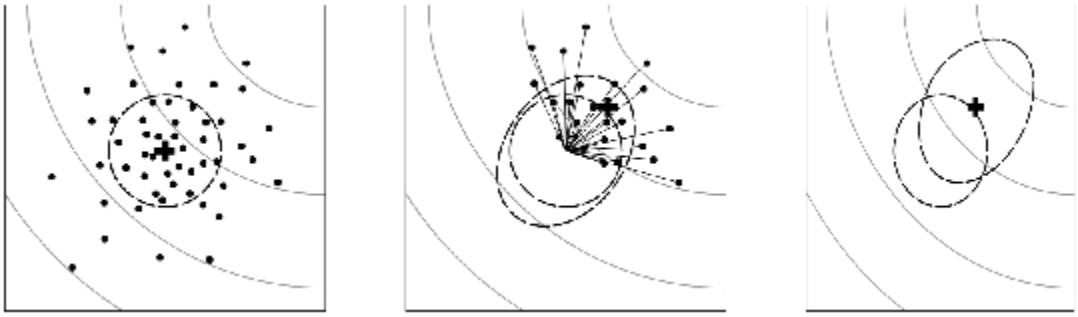

Figure 2. Estimation of the search distribution for the second generation. Left: producing the first population by sampling of $\mathcal{N}(0, \mathbf{I})$. Middle: selection of the new parents and updating the covariance matrix; solid lines determines the selected steps. Right: search distribution of the next generation (dashed ellipsoid). Contour lines (grayed) indicate that the strategy should move toward the upper right corner

2.5. Step-size control

The covariance matrix adaptation, introduced in the last section, does not explicitly control the "overall scale" of the distribution. Step-size control defines how much the distribution ellipsoid should be elongated or shortened, to achieve an optimal scale. The evolution path is utilized to control the step-size.

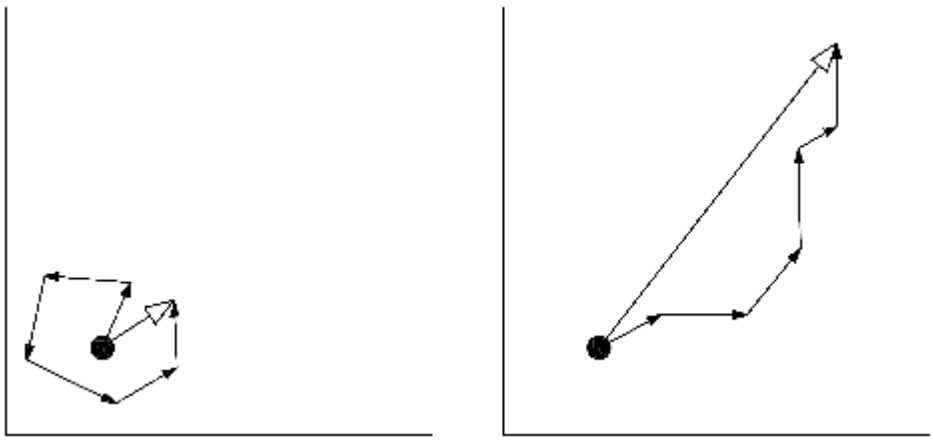

Figure 3. Two evolution paths of respectively six steps from various situations. The length of the evolution paths is remarkably different and is exploited for step-size control

The length of an evolution path is exploited, based on the following reasoning. Whenever the evolution path is short, single steps cancel each other out as is shown in Figure 3, left. Hence, they are called anti-correlated. If steps annihilate each other, the step-size should be decreased. Whenever the evolution path is long, the single steps are pointing to similar directions and they are called correlated (Figure 3, right). Because the steps are similar, the same distance can be covered by fewer but longer steps into the same directions. Consequently, the step-size should be increased.

To decide whether the evolution path is long or short, the length of the path is compared with its expected length under random selection, which is equal to the expectation of the Euclidean norm of a $\mathcal{N}(0, \mathbf{I})$ distributed random vector. If selection biases the evolution path to be longer than expected, σ is increased, and vice versa.

To calculate the step-size, a conjugate evolution path is constructed, because the expected length of the evolution path $\boldsymbol{p}_c$ from (6) depends on its direction. Initialized with $\boldsymbol{p}_s^{(0)} = 0$, the conjugate evolution path is given by

$$\boldsymbol{p}_s^{(g+1)} = (1-c_s)\boldsymbol{p}_s^{(g)} + \sqrt{\boldsymbol{c}_s(2-c_s)m_{eff}}\, \boldsymbol{C}^{(g)^{-\frac{1}{2}}} \frac{\boldsymbol{m}^{(g+1)} - \boldsymbol{m}^{(g)}}{s^{(g)}} \tag{9}$$

where

$\boldsymbol{p}_s^{(g)} \in R^n$ is the conjugate evolution path at generation g.

c_σ, the learning rate.

$\boldsymbol{C}^{(g)^{-\frac{1}{2}}} = \boldsymbol{B}^{(g)}\boldsymbol{D}^{(g)^{-1}}\boldsymbol{B}^{(g)^{\mathrm{T}}}$, this transformation makes the expected length of $\boldsymbol{p}_s^{(g+1)}$ independent of its direction.

The step-size update is formulated as

$$\sigma^{(g+1)} = \sigma^{(g)}\, exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\| p_\sigma^{(g+1)} \|}{\mathrm{E}\,\| \mathcal{N}(0, \mathbf{I}) \|} - 1\right)\right) \tag{10}$$

where

d_σ is damping parameter that scales the change magnitude of $\sigma^{(g)}$.

$\mathrm{E}\,\| \mathcal{N}(0, \mathbf{I}) \| = \sqrt{n}\left(1 - \frac{1}{4n} + \frac{1}{21n^2}\right)$, expected length under random selection where n is the search space dimension.

3. OPTIMIZATION PROBLEM

For optimal design of truss structures, the objective function can be expressed as

$$\text{Minimize} \qquad \mathrm{W}(\boldsymbol{x}) = \sum_{i=1}^{m} g_{\mathrm{i}}\, x_i\, l_i \tag{11}$$

where

$\mathrm{W}(\boldsymbol{x})$ is the weight of the structure.

m , number of members making up the structure.

g_{i} , material density of member i.

l_i, length of member i.

x_i , cross-sectional area of member i chosen between x_{min} and x_{max}.

The constraints are as follows

$$\begin{cases} \sigma_i \le \sigma_{allowable} & i = 1, 2, \dots, m \\ \delta_i \le \delta_{allowable} & i = 1, 2, \dots, n \end{cases} \tag{12}$$

where *m* and *n* are number of members and nodes, respectively. σ, δ and *allowable* represent member stress, nodal displacement and their upper bound, correspondingly.

According to AISC-ASD code [39], the stress limitation for tension members is $\sigma_{allowalbe} = 0.6F_y$, in which F_y is the yield stress of steel. For compression members this limitation is given by

$$\sigma_{allowalbe} = \begin{cases} [(1 - \frac{l_i^2}{2C_C})F_y]/(\frac{5}{3} + \frac{3l_i}{8C_C} - \frac{l_i^3}{8C_C^3}) & for\ l_i < C_c \\ \frac{12\pi^2 E}{23 l_i^2} & for\ l_i \ge C_c \end{cases} \tag{13}$$

where

E is the modules of elasticity.

$l_i = \frac{k\, l_i}{r_i}$, the slenderness ratio of member *i* where *k* and r_i are the effective length factor and the radius of gyration, respectively.

$C_C = \sqrt{2\pi^2 E / F_y}$, the slenderness ratio dividing the elastic and inelastic buckling regions.

4. DESIGN EXAMPLES

In this section, the proposed algorithm is implemented in MATLAB® and some test problems are optimized. All runs are performed with the default strategy parameter setting given in Appendix A. The structures are analyzed using the standard matrix stiffness method. The results are compared to those of the other advanced metaheuristics to demonstrate their comparative performance. The CPU-time consumption of the program is calculated for each case. All recorded times are obtained using an Intel® Core ™ 2 Duo T9300 @ 2.50 GHz processor equipped with 4 GBs of RAM.

4.1. A twenty five-bar space truss

The topology and nodal numbering of a 25-bar space truss structure are shown in Figure 4, known as a benchmark in the field of structural optimization. The material density is considered as 0.1 lb/in^3 (2768 kg/m^3) and the modulus of elasticity is taken as 10000 ksi (68950 MPa). Twenty-five members are categorized in eight groups as follows: (1) A1, (2) A2 –A5, (3) A6–A9, (4) A10–A11, (5) A12–A13, (6) A14–A17, (7) A18–A21, and (8) A22–A25.

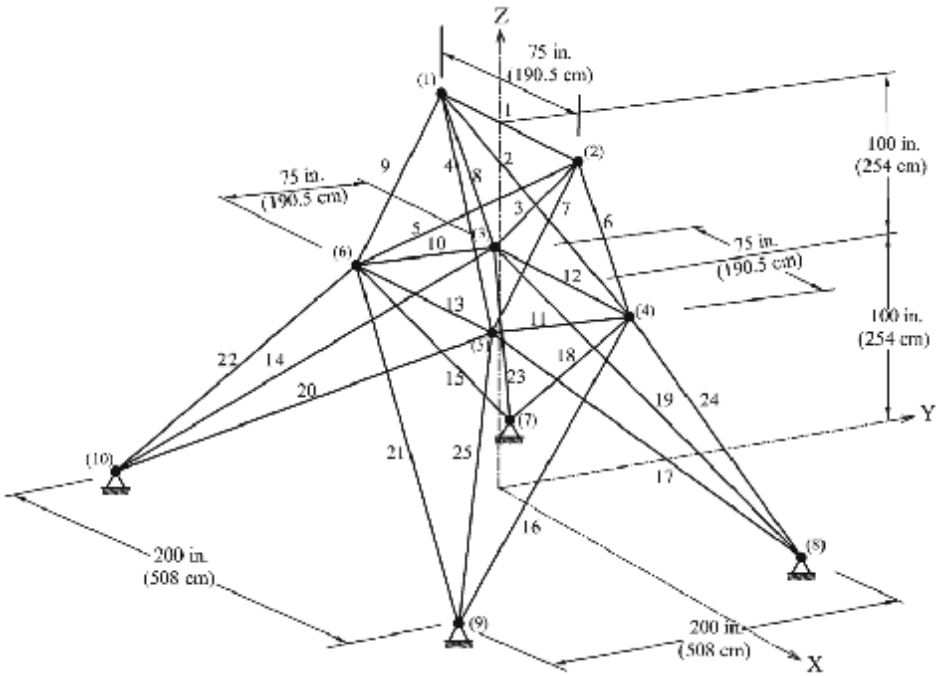

Figure 1. A 25-bar space truss

This truss is assumed to be subjected to two loading conditions, shown in Table 1. Maximum displacement limitations of ±0.35 in (±8.89 mm) are imposed on every node in every direction and the axial stress constraints are different for each group as shown in Table 2. Minimum cross-sectional area of all members is 0.01 in^2 (0.645 cm^2) and the maximum is taken as 3.4 in^2 (21.935 cm^2).

Table 1. Loading conditions for the 25-bar space truss

Node	Case 1			Case 2		
	P_x kips (kN)	P_y kips (kN)	P_z kips (kN)	P_x kips (kN)	P_y kips (kN)	P_z kips (kN)
1	0.0	20.0 (89)	-5.0 (22.5)	1.0 (4.45)	10.0 (44.5)	-5.0 (22.25)
2	0.0	20.0 (89)	-5.0 (22.5)	0.0	10.0 (44.5)	-5.0 (22.25)
3	0.0	0.0	0.0	0.5 (2.22)	0.0	0.0
6	0.0	0.0	0.0	0.5 (2.22)	0.0	0.0

Table 2. Member stress limitation for the 25-bar spatial truss

Element group	Compressive stress limitations ksi (MPa)	Tensile stress limitations ksi (MPa)
1 A_1	35.092 (241.96)	40.0 (275.80)
2 $A_2 \sim A_5$	11.590 (79.913)	40.0 (275.80)
3 $A_6 \sim A_9$	17.305 (119.31))	40.0 (275.80)
4 $A_{10} \sim A_{11}$	35.092 (241.96)	40.0 (275.80)
5 $A_{12} \sim A_{13}$	35.092 (241.96)	40.0 (275.80)
6 $A_{14} \sim A_{17}$	6.759 (46.603)	40.0 (275.80)
7 $A_{18} \sim A_{21}$	6.959 (47.982)	40.0 (275.80)
8 $A_{22} \sim A_{25}$	11.082 (76.410)	40.0 (275.80)

The best solution achived by the algorithm after fifty runs, as well as the results of some other methods are provided in Table 3. The CPU-time consumption was 35.7 seconds.

Table 3. Optimal design comparison for the 25-bar spatial truss

Element group	Optimal cross sectional areas (in^2)								
	Rajeev et al. [1]	Schutte et al. [23]	Lee et al.[26]	Kaveh et al.[21, 40, 29, 30]				Present work	
	GA	PSO	HS	IACS	HPSACO	HBB-BC	CSS	In^2	cm^2
1 A_1	0.100	0.010	0.047	0.010	0.010	0.010	0.010	0.010	0.0645
2 $A_2 \sim A_5$	1.800	0.010	2.022	2.042	2.054	1.993	2.003	1.981	12.7820
3 $A_6 \sim A_9$	2.300	2.893	2.950	3.001	3.008	3.056	3.007	3.002	19.3694
4 $A_{10} \sim A_{11}$	0.200	0.010	0.010	0.010	0.010	0.010	0.010	0.010	0.0645
5 $A_{12} \sim A_{13}$	0.100	0.010	0.014	0.010	0.010	0.010	0.010	0.010	0.0645
6 $A_{14} \sim A_{17}$	0.800	0.671	0.688	0.684	0.679	0.665	0.687	0.684	4.4140
7 $A_{18} \sim A_{21}$	1.800	1.611	1.657	1.625	1.611	1.642	1.655	1.678	10.8244
8 $A_{22} \sim A_{25}$	3.000	2.717	2.663	2.672	2.678	2.679	2.660	2.659	17.1569
Best weight (lb)	546.000	545.210	544.380	545.030	544.990	545.160	545.100	545.163	2424.848 N

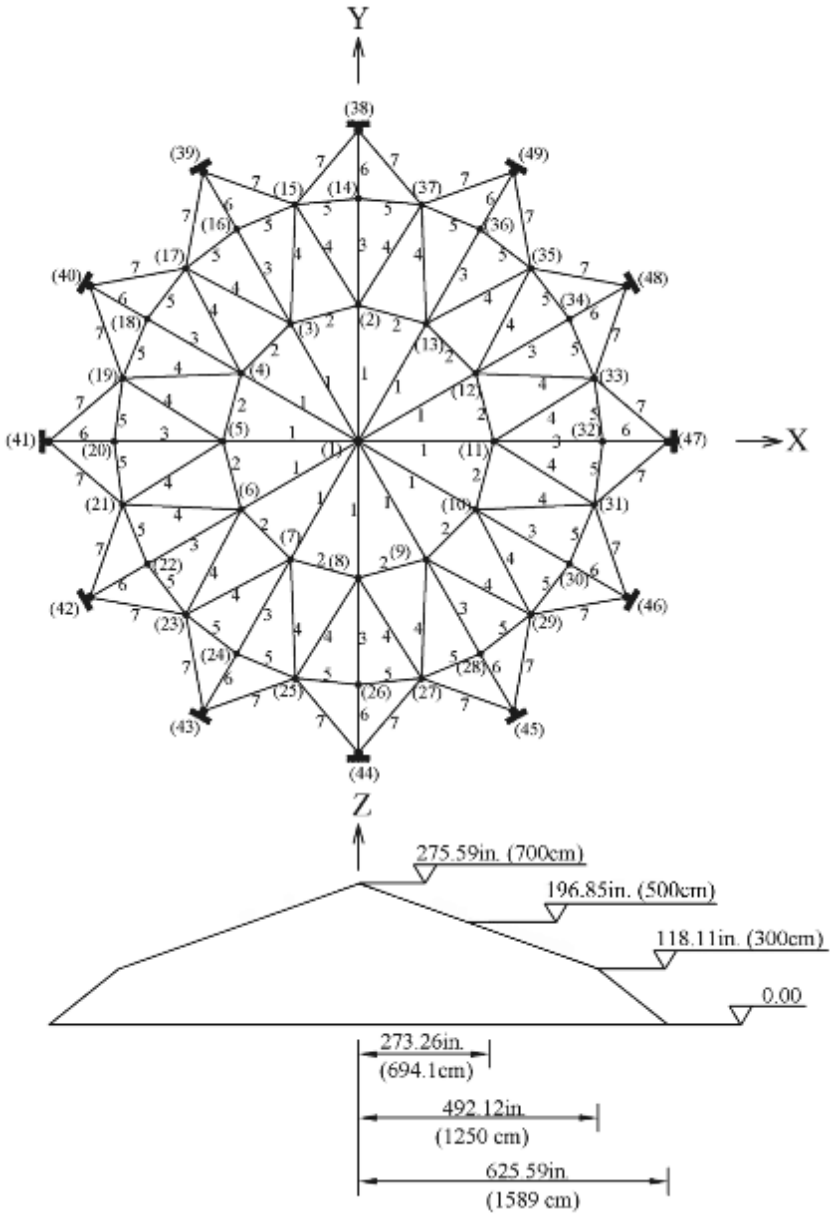

Figure 5. A dome truss

4.2. A dome truss

Figure 5 demonstrates the geometry and group numbers of a 120-bar dome truss. The modulus of elasticity is 30450 ksi (210000 MPa) and the material density is 0.288 lb/in^3 (7971.813 kg/m^3). The yield stress of steel is taken as 58.0 ksi (400 MPa). The dome is considered to be subjected to vertical loading at all the unsupported joints. These loads are taken as -13.49 kips (60 kN) at node 1, -6.744 kips (30 kN) at nodes 2 through 14, and -2.248 kips (10 kN) at the rest of the nodes.

The radius of gyration of each member (r_i) can be expressed in terms of its cross-sectional area, i.e., $r_i = aA_i^b$, which a and b are the constants depending on the types of sections adopted for the members. In this example, pipe sections (a = 0.4993 and b= 0.6777) are adopted for bars. The range of cross-sectional areas varies from 0.775 to 20.0 in^2 (5 to 129.032 cm^2). Displacement limitations of ±0.1969 in (±5 mm) are imposed on all nodes in x, y and z directions and stress constraints are assumed according to AISC-ASD code (13).

In Table 4, the best-found design over fifty runs, together with the results of six other heuristic algorithms are presented. The convergence history of the CMA-ES is illustrated in Figure 6. The CPU-time consumption was 198.5 seconds.

Table 4. Optimal design comparison for the dome truss

Element group	Optimal cross sectional areas (in^2)							
	Kaveh et al. [21,41,41,40 ,29,30]						Present work	
	IACS	PSOPC	PSACO	HPSACO	HBB-BC	CSS	in^2	cm^2
1 A_1	3.026	3.040	3.026	3.095	3.037	3.027	3.025	19.516
2 A_2	15.06	13.149	15.222	14.405	14.431	14.606	14.730	95.03
3 A_3	4.707	5.646	4.904	5.020	5.130	5.044	5.153	33.248
4 A_4	3.100	3.143	3.123	3.352	3.134	3.139	3.136	20.237
5 A_5	8.513	8.759	8.341	8.631	8.591	8.543	8.437	54.434
6 A_6	3.694	3.758	3.418	3.432	3.377	3.367	3.306	21.326
7 A_7	2.503	2.502	2.498	2.499	2.500	2.497	2.495	16.098
Best weight (lb)	33320.52	33481.2	33263.9	33248.9	33287.9	33251.9	33256.2	15084.7 N

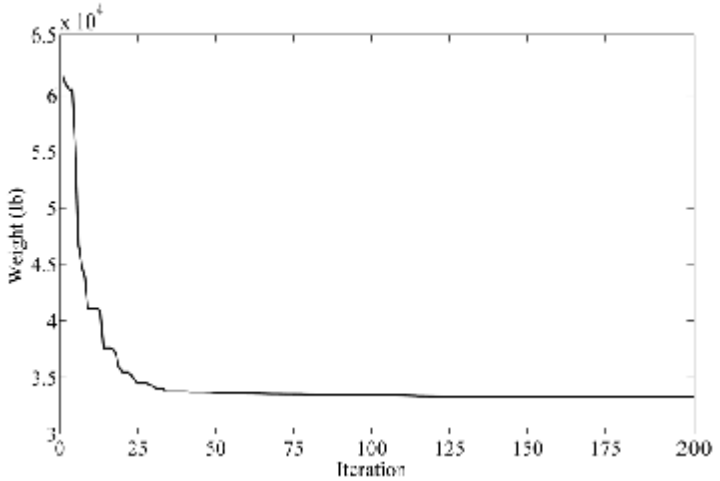

Figure 6. Convergence history of the CMA-ES for the dome truss

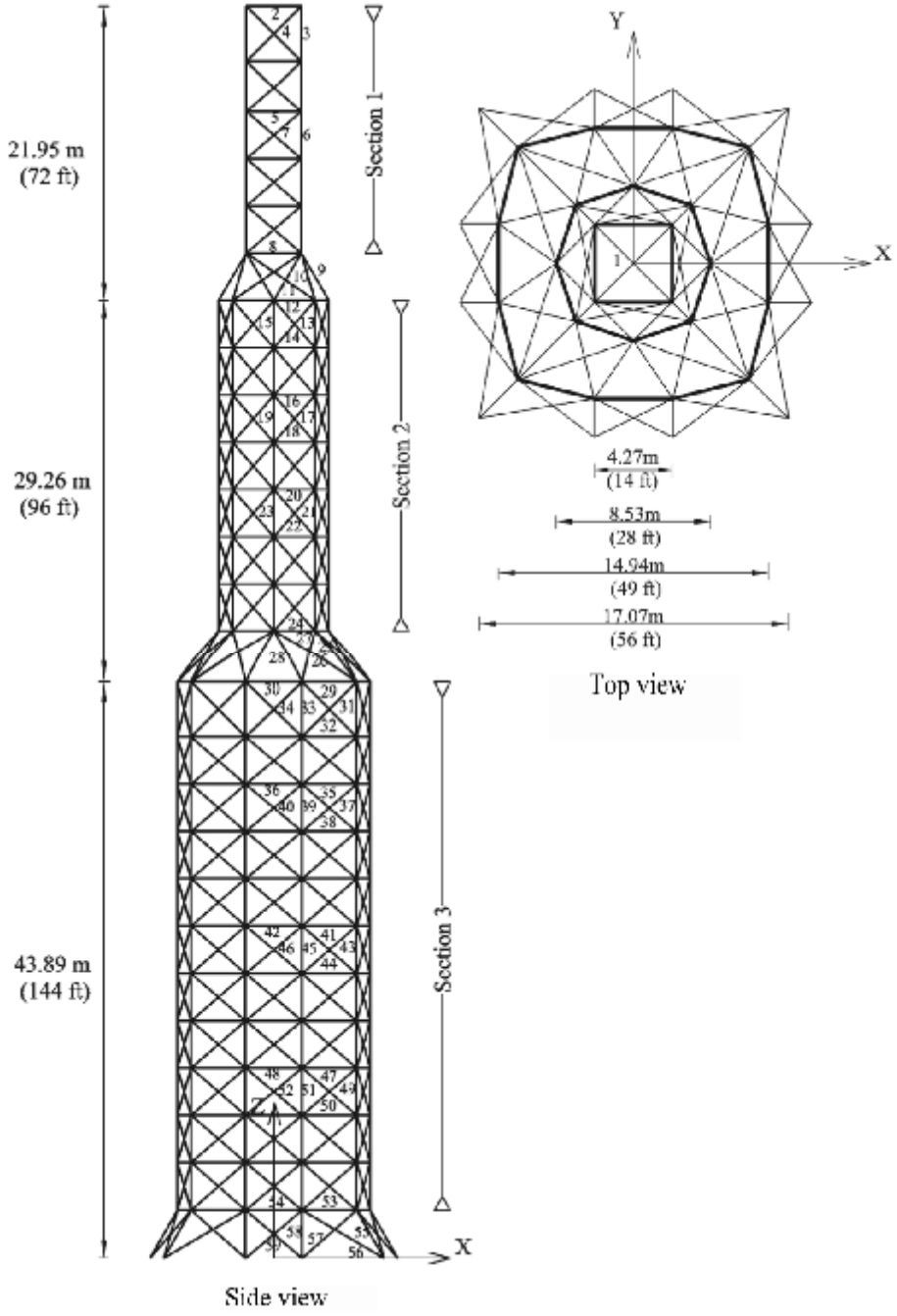

Figure 7. A 26-story-tower space truss

4.3. A 26-story-tower space truss

A 26-story-tower space truss containing 942 members and 244 nodes is examined as shown in Figure 7. Fifty-nine design variables are used to represent the cross-sectional areas of 59 member groups in this structure, employing the symmetry of the structure. The material

density is 0.1 lb/in^3 (2767.990 kg/m^3) and the modulus of elasticity is taken as 10000 ksi (68950 MPa). The members are subjected to the stress limits of ±25 ksi (172.375 MPa) and the four nodes of the top level in every directions are subjected to the displacement limits of ±15.0 in (38.10 cm). The allowable cross-sectional areas in this example are selected from 0.1 to 20.0 in^2 (from 0.6452 to 129.032 cm^2). The loading on the structure consists of:

1. The vertical load at each node in the first section is equal to -3 kips (13.344 kN).
2. The vertical load at each node in the second section is equal to -6 kips (26.688 kN).
3. The vertical load at each node in the third section is equal to -9 kips (40.032 kN).
4. The horizontal load at each node on the right side in the x direction is equal to -1 kips (4.448 kN).
5. The horizontal load at each node on the left side in the x direction is equal to 1.5 kips (6.672 kN).
6. The horizontal load at each node on the front side in the y direction is equal to -1 kips (4.448 kN).
7. The horizontal load at each node on the back side in the y direction is equal to 1 kips (4.448 kN).

Table 5. The optimal design achieved by the CMA-ES for the 26-story tower space truss

Optimal cross-sectional areas (in^2)											
Group No.	**Area**	**Group No.**	**Area**	**Group No.**	**Area**	**Group No.**	**Area**	**Group No.**	**Area**	**Group No.**	**Area**
1	13.596	11	0.107	21	3.851	31	13.374	41	0.107	51	4.536
2	8.200	12	0.713	22	0.703	32	0.916	42	2.408	52	0.529
3	5.654	13	2.576	23	5.759	33	2.602	43	11.840	53	11.132
4	0.413	14	0.504	24	8.770	34	0.622	44	0.723	54	14.911
5	0.128	15	7.752	25	8.490	35	0.100	45	4.482	55	16.117
6	4.157	16	0.225	26	4.001	36	1.191	46	0.572	56	2.700
7	0.547	17	7.770	27	2.452	37	7.519	47	0.250	57	2.188
8	12.506	18	0.566	28	3.725	38	0.625	48	0.952	58	3.824
9	8.950	19	6.007	29	5.435	39	4.293	49	15.948	59	1.706
10	6.207	20	0.250	30	9.336	40	0.612	50	1.402		

After 250 runs, the CMA-ES found an optimal design of weight 43055 lb (191505 N) as given in Table 5. The result comparison of different heuristic methods is provided in Table 6. Figure 8 shows the stress ratios, calculated by dividing the existing member stress by the allowable stress. In this figure, positive values belong to the compression members and negatives to the tension members. The maximum stress ratio is 99.99% and the maximum value of the displacement is equal to 15 in (38.10 cm). The convergence history is illustrated in Figure 9. The CPU-time consumption was 3557 seconds.

Table 6. Result comparison for the 26-story-tower space truss

	Kaveh et al. [29, 29, 29, 30]				Present work
	GA	PSO	HBB-BC	CSS	
Best weight (lb)	56343	60385	52401	47371	43055 (191505 N)

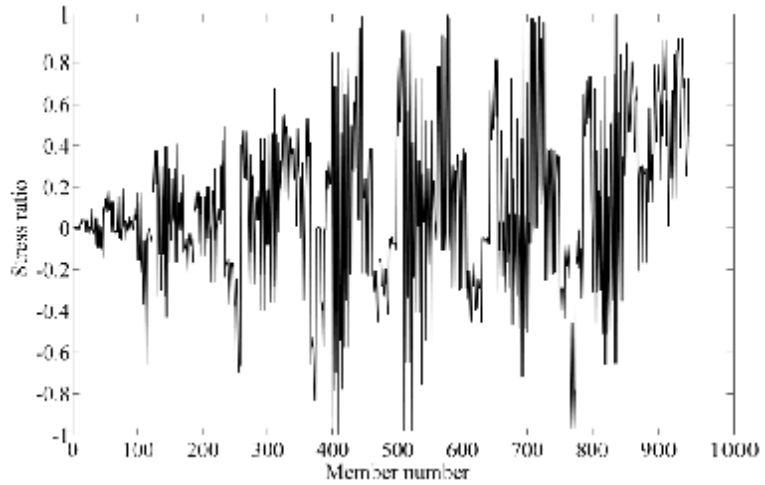

Figure 8. Stress ratios of the 26-story-tower space truss

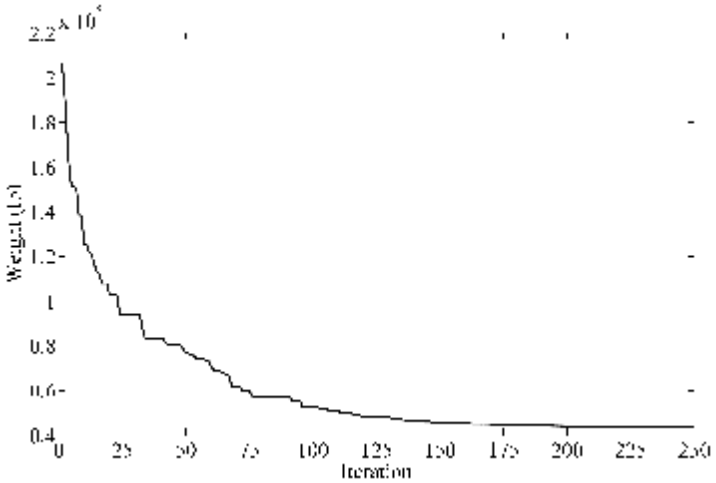

Figure 9. Convergence history of the CMA-ES for the 26-story-tower space truss

4.4. A cooling tower

A 17-story cooling tower structure containing 1700 members and 377 nodes is illustrated in Figure 10. In this example, the objective is to find an optimal design for the lateral load bearing system that consists of vertical and diagonal members. As shown in Figure 11, at each story, six member groups are considered for the member cross sectional areas, resulting in

102 member groups for the entire structure. In this figure, the members having a same number belong to a same group. All floors have identical plans as shown in Figure 10. In order to have rigid diaphragms, the modulus of elasticity for all horizontal members is taken 10^5 times larger than other members in the structure. The lateral loads are applied at the central node of each floor. The material density is taken as 0.2836 lb/in^3 (7850 kg/m^3) and the modulus of elasticity is 29869 ksi (205939 MPa). The loading includes:

1. The vertical load at each exterior node is equal to -11.023 kips (49.033 kN);
2. The horizontal load at each central node in both x and y directions is equal to 66.139i kips (294.200 kN), which i is the story number.

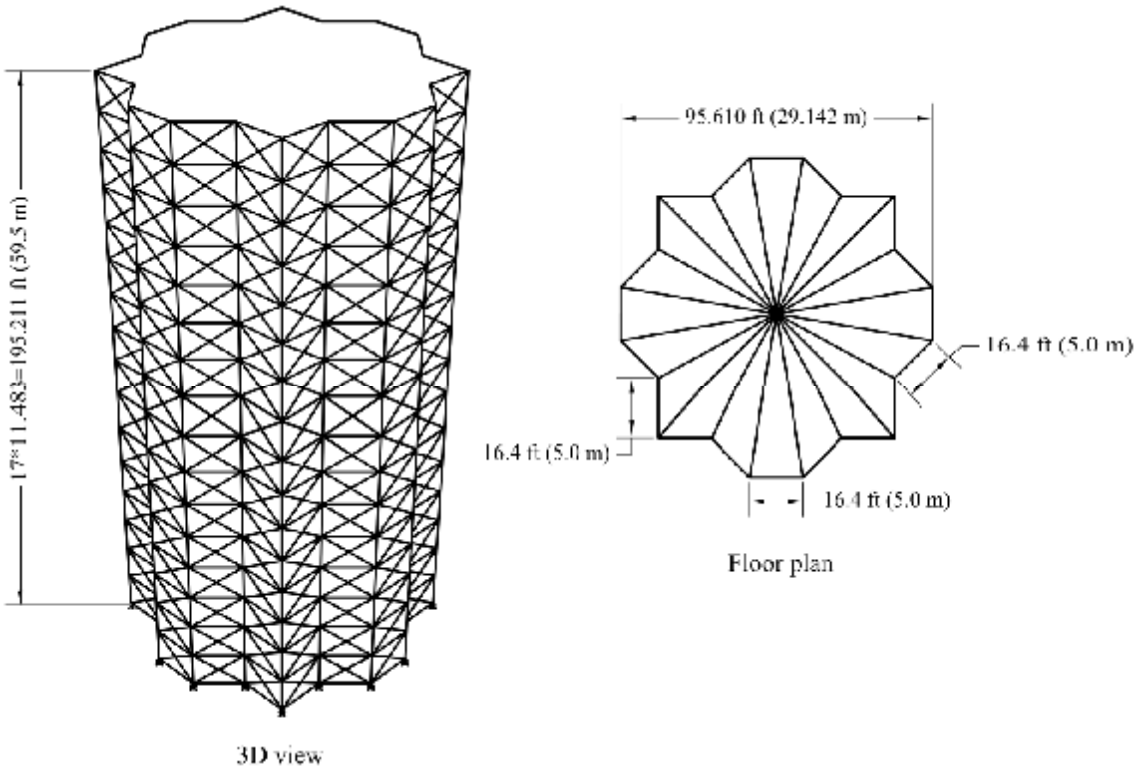

Figure 10. A cooling tower

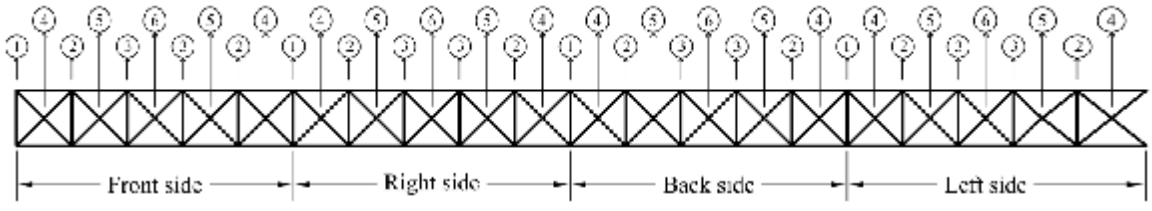

Figure 11. Member groups of the cooling tower at each story

Stress constraints are considered according to AISC-ASD code (13) and story drift limitations of 0.00143 is imposed. Story drift is the difference in horizontal deflection at the

top and bottom of a story divided by the story height. The cross-section of the members are square pipe sections of thickness 1.969 in (5 cm). Minimum width of the sections is 3.937 in (10 cm) and the maximum is taken as 59.055 (150 cm). Cross-sectional areas of all horizontal members are assumed to be equal to the maximum allowable area.

The weight of the best solution obtained over 300 runs is 4220836 lb (18774 kN), provided in Table 7. The stress ratio of the members and the drift ratio of the stories, computed by dividing the existing story drift by the allowable value, are demonstrated in Figure 12 and Figure13, respectively. As is indicated in these figures, the drift constraints are dominant in this example. The maximum stress ratio and drift ratio are 98.88% and 100%, respectively. The convergence history is shown in Figure 14. The CPU-time consumption was 8938 seconds.

Table 7. Optimal design achieved by the CMA-ES for the cooling tower

Optimal width of the square pipe sections (in)											
Group No.	**Width**	**Group No.**	**Width**	**Group No.**	**Width**	**Group No.**	**Width**	**Group No.**	**Width**	**Group No.**	**Width**
1	51.082	18	24.272	35	37.357	52	7.016	69	10.450	86	8.524
2	46.122	19	11.637	36	25.424	53	6.922	70	10.843	87	8.990
3	44.305	20	18.837	37	30.913	54	8.393	71	8.747	88	9.717
4	36.793	21	18.374	38	24.413	55	7.873	72	10.074	89	9.526
5	31.728	22	11.116	39	25.728	56	11.243	73	8.273	90	6.706
6	49.851	23	15.226	40	18.611	57	7.782	74	9.117	91	8.100
7	25.996	24	17.453	41	16.856	58	7.828	75	9.108	92	8.319
8	33.565	25	11.586	42	34.738	59	9.278	76	9.574	93	5.559
9	18.702	26	10.906	43	24.776	60	7.396	77	7.042	94	12.788
10	22.941	27	38.530	44	14.366	61	6.960	78	9.730	95	8.071
11	23.990	28	38.772	45	6.285	62	9.507	79	6.614	96	10.431
12	25.824	29	8.069	46	21.233	63	8.085	80	8.083	97	8.044
13	34.899	30	41.672	47	4.937	64	6.985	81	9.743	98	4.155
14	31.597	31	18.444	48	21.646	65	6.801	82	6.289	99	7.179
15	7.466	32	12.751	49	10.641	66	4.553	83	4.847	100	10.129
16	14.095	33	21.155	50	4.373	67	4.515	84	4.598	101	4.649
17	47.477	34	20.252	51	5.974	68	3.941	85	3.942	102	4.028

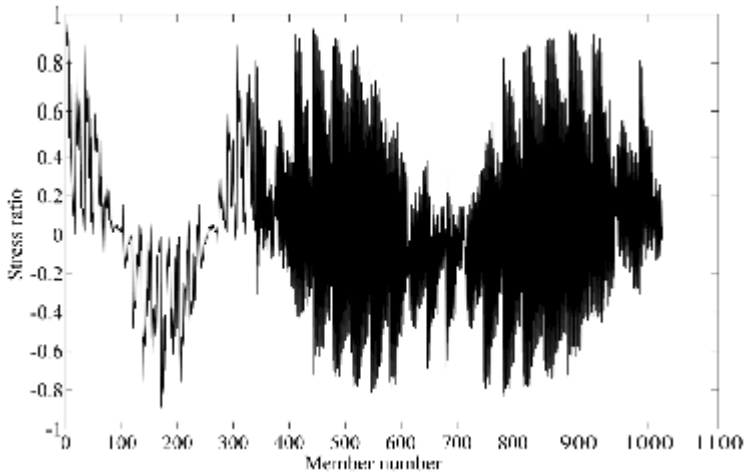

Figure 12. Stress ratios of the cooling tower

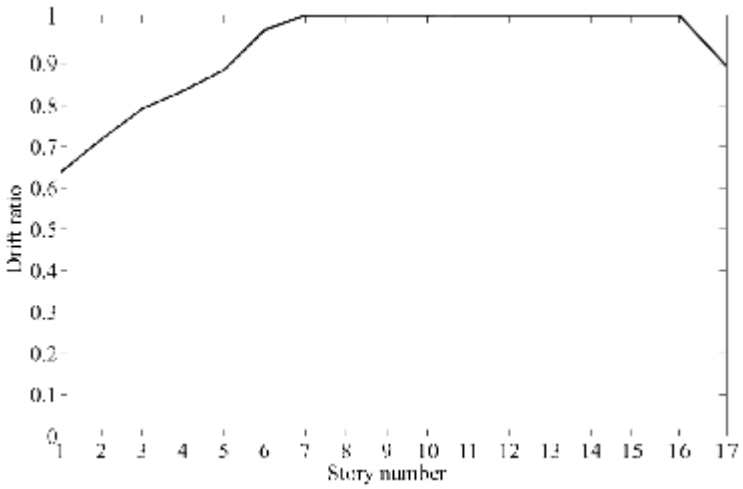

Figure 13. Drift ratios of the cooling tower

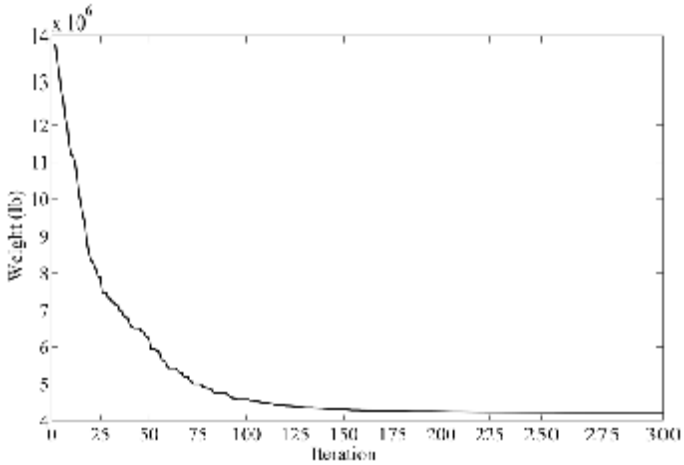

Figure 14. Convergence history of the CMA-ES for the cooling tower

5. DISCUSSION

The CMA-ES is a suitable alternative for continuous optimization, if classical search methods fail due to a non-convex or rugged search landscape. It overcomes typical problems that are often associated with evolutionary algorithms including

1. Poor performance on highly non-separable objective functions. A function f is called separable if the variables of f are independent in that the global optimum can be obtained by n one dimensional optimization procedures along the coordinate axes. Difficult real variable optimization problems exhibit essential dependencies between the variables. Learning these dependencies has been successfully addressed by the CMA, which learns all pair-wise dependencies by updating a covariance matrix for the sample distribution.
2. The inherent need to use large population sizes. A typical reason for the failure of population based search algorithms is the degeneration of the population into a subspace. This is usually prevented by a large population size (considerably larger than the problem dimension). In the CMA-ES, the population size can be freely chosen, because the learning rates c_1 and c_m prevent the degeneration even for small population sizes. Small population sizes generally lead to faster convergence, large population sizes help to avoid local optima.
3. Premature convergence of the population. Step-size control in the CMA-ES prevents the population to converge prematurely.

Invariance properties of a search algorithm denote identical behavior on a set, or a class of objective functions. The CMA-ES has several invariance properties. Two main of them are (1) invariance to order preserving transformations of the objective function value, since the algorithm only depends on the ranking of function values, And (2) invariance to angle preserving transformations of the search space (rotation, reflection and translation), if the initial search point is transformed accordingly.

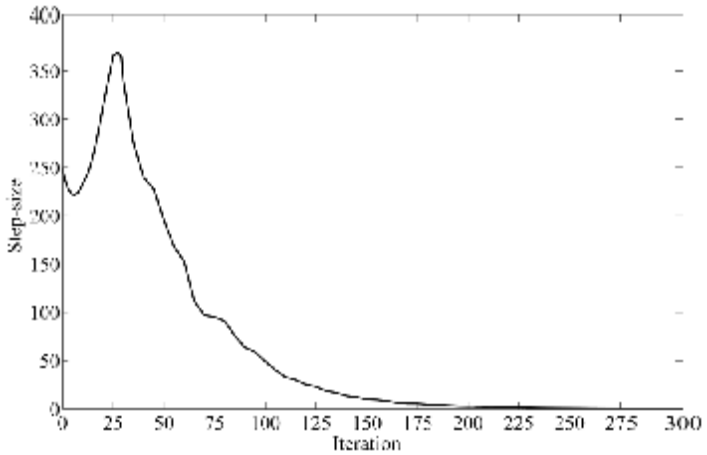

Figure 15. Adaptation history of the step-size for the cooling tower

The CMA-ES does not require a tedious parameter tuning for its application. In fact, the choice of strategy internal parameters is not left to the user. Finding good strategy parameters is considered as a part of the algorithm design, and not part of its application. For the

application of the CMA-ES, just an initial solution, an initial standard deviation (step-size) and, possibly, the termination criteria need to be set by the user.

Self-adaptation is an important feature of the CMA-ES. Self-adaptation in its purest meaning is a method to adjust setting of the strategy parameters. It is called self-adaptive because the algorithm controls the setting itself. Self-Adaptation aims at biasing the distribution towards promising regions of the search space while maintaining sufficient diversity of the search. Adaptation history of the step-size for the fourth design example is shown in Figure 15. When the step-size approaches zero, the population converge to an optimum solution.

6. CONCLUSIONS

In this paper, the CMA-ES, which rates among the most successful evolutionary algorithms for real variable optimization, is implemented to solve size optimization problems of truss structures. Various test problems are performed to evaluate the performance of the algorithm. The numerical results confirm that in comparison to state-of-the-art heuristic algorithms, the CMA-ES is a highly reliable and competitive optimization method, particularly for the large-scale problems.

REFERENCES

1. Rajeev S, Krishnamoorthy CS. Discrete optimization of structures using genetic algorithms. *J Struct Eng, ASCE,* 1992; **118**(5):1233–50.
2. Koumousis VK, Georgious PG. Genetic algorithms in discrete optimization of steel truss roofs. *J Comput Civil Eng, ASCE,* 1994;**8**(3):309–25.
3. Wu SJ, Chow PT. Integrated discrete and configuration optimization of trusses using genetic algorithms. *Comput Struct,* 1995;**55**(4):695–702.
4. Hajela P, Lee E. Genetic algorithms in truss topological optimization. *Int J Solids Struct,* 1995;**32**(22):3341–57.
5. Shrestha SM, Ghaboussi J. Evolution of optimization structural shapes using genetic algorithm. *J Struct Eng, ASCE,* 1998;**124**(11):1331–8.
6. Erbatur F, Hasancebi O, Tutuncil I, Kihc H. Optimal design of planar and space structures with genetic algorithms. *Comput Struct,* 2000;**75**:209–24.
7. Kameshki ES, Saka MP. Optimum design of nonlinear steel frames with semi rigid connections using a genetic algorithm. *Comput Struct,* 2001;**79**:1593–604.
8. Kaveh A, Kalatjari V. Genetic algorithm for discrete sizing optimal design of trusses using the force method. *Int J Numer Methods Eng,* 2002;**55**:55–72.
9. Saka MP. Optimum design of pitched roof steel frames with haunched rafters by genetic algorithm. *Comput Struct,* 2003;**81**:1967–78.
10. Kaveh A, Kalatjari V. Topology optimization of trusses using genetic algorithm, force method, and graph theory. *Int J Numer Methods Eng,* 2003;**58**(5):771–91.
11. Kaveh A, Abditehrani A. Design of frames using genetic algorithm, force method and graph theory. *Int J Numer Methods Eng,* 2004;**61**:2555–65.

12. Togˇan V, Dalogˇlu AT. Optimization of 3d trusses with adaptive approach in genetic algorithms. *Eng Struct*, 2006;**28**:1019–27.
13. Kaveh A, Rahami H. Analysis, design and optimization of structures using force method and genetic algorithm. *Int J Numer Methods Eng*, 2006;**65**(10):1570–84.
14. Hasancebi O. Adaptive evolution strategies in structural optimization: Enhancing their computational performance with applications to large-scale structures. *Comput Struct*, 2008;**86**:119–32.
15. Camp CV, Bichon J. Design of space trusses using ant colony optimization. *J Struct Eng, ASCE*, 2004;**130**(5):741–51.
16. Serra M, Venini P. On some applications of ant colony optimization metaheuristic to plane truss optimization. *Struct Multidisc Optim*, 2006;**32**(6):499–506.
17. Camp CV, Bichon J. Design of steel frames using ant colony optimization. *J Struct Eng, ASCE*, 2005;**131**:369–79.
18. Kaveh A, Shojaee S. Optimal design of skeletal structures using ant colony optimization. *Int J Numer Methods Eng*, 2007;**70**(5):563–81.
19. Kaveh A, Hassani B, Shojaee S, Tavakkoli SM. Structural topology optimization using ant colony methodology. *Eng Struct*, 2008;**30**(9):2559–65.
20. Kaveh A, Shahrouzi M. Dynamic selective pressure using hybrid evolutionary and ant system strategies for structural optimization. *Int J Numer Methods Eng*, 2008;**73**(4): 544–63.
21. Kaveh A, Farahmand Azar B, Talatahari S. Ant colony optimization for design of space trusses. *Int J Space Struct*, 2008;**23**(3):167–81.
22. Kaveh A, Talatahari S. An improved ant colony optimization for constrained engineering design problems, *Eng Computation*, 2010; **27**(1):155-82.
23. Schutte JJ, Groenwold AA. Sizing design of truss structures using particle swarms. *Struct Multidisc Optim*, 2003;**25**:261–9.
24. Li LJ, Huang ZB, Liu F, Wu QH. A heuristic particle swarm optimizer for optimization of pin connected structures. *Comput Struct*, 2007;**85**:340–9.
25. Perez RE, Behdinan K. Particle swarm approach for structural design optimization. *Comput Struct*, 2007;**85**:1579–88.
26. Lee KS, Geem ZW. A new structural optimization method based on the harmony search algorithm. *Comput Struct*, 2004;**82**:781–98.
27. Saka MP. Optimum geometry design of geodesic domes using harmony search algorithm. *Adv Struct Eng*, 2007;**10**(6):595–606.
28. Saka MP. Optimum design of steel sway frames to BS5950 using harmony search algorithm. *J Construct Steel Res*, 2009;**65**(1):36–43.
29. Kaveh A, Talatahari S. Size optimization of space trusses using Big Bang–Big Crunch algorithm. *Comput Struct*, 2009;**87**:1129–40.
30. Kaveh A, Talatahari S. Optimal design of skeletal structures via the charged system search algorithm. *Struct Multidisc Optim*, 2010;**41**:893–911.
31. Talbi E-G. Metaheuristics: from design to implementation. Wiley, New Jersey. 2009.
32. Darwin C. On the origin of species by means of natural selection. John Murray, London, 1859.
33. Holland J. H. Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor, MI, 1975.

34. Rechenberg I. Evolutionstrategie: Optimierung technischer systeme nach prinzipien derbiologischen evolution. Frommann-Holzboog, 1973.
35. Fogel LJ, Owens AJ, Walsh MJ. Artificial intelligence through simulated evolution. Wiley, 1966.
36. Koza J. R. Genetic Programming. MIT Press, Cambridge, MA, 1992.
37. Hansen N, Ostermeier A, Gawelczyk A. On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. In L. Eshelman (Ed.), Proceedings of the Sixth International Conference on Genetic Algorithms, Pittsburgh, Morgan Kaufmann, 1995.
38. Hansen N. The CMA evolution strategy: a tutorial. http://www.lri.fr/~hansen/ cmaesintro.html.
39. American Institute of Steel Construction (AISC). Manual of Steel Construction Allowable Stress Design. 9th ed. Chicago, IL; 1989.
40. Kaveh A, Talatahari S. Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. *Comput Struct*, 2009;**87**(5–6):267–83.
41. Kaveh A, Talatahari S. A hybrid particle swarm and ant colony optimization for design of truss structures. *Asian J Civil Eng*, 2008;**9**(4):329–48.

Appendix A. Algorithm summary

Table 8. Default strategy parameters

Selection and recombination

$$\lambda = 4 + \lfloor 3 \ln n \rfloor, \qquad \mu = \lfloor \mu' \rfloor, \qquad \mu' = \frac{\lambda}{2}$$

$$\omega_i = \frac{\omega_i'}{\sum_{i=1}^{\mu} \omega_i'}, \qquad \omega_i' = \ln(\mu' + 5) - \ln i \qquad \text{for } i = 1, \ldots, \mu$$

Step-size control

$$c_\sigma = \frac{\mu_{eff} + 2}{n + \mu_{eff} + 5}, \qquad d_\sigma = 1 + c_\sigma + 2 \max\left(0, \sqrt{\frac{\mu_{eff} - 1}{n+1}} - 1\right)$$

Covariance matrix adaptation

$$c_c = \frac{4 + \mu_{eff}/n}{n + 4 + 2\mu_{eff}/n}$$

$$c_1 = \frac{2}{(n+1.3)^2 + \mu_{eff}}$$

$$c_\mu = \min\left(1 - c_1, \frac{2(\mu_{eff} - 2 + 1/\mu_{eff})}{(n+2)^2 + \mu_{eff}}\right)$$

Set parameters

Set parameters λ, μ, $\omega_{i=1,\ldots,\mu}$, c_σ, c_1, c_c, c_σ and d_σ to their default values according to Table 8.

Set evolution paths $\boldsymbol{p}_\sigma^{(0)} = 0$, $\boldsymbol{p}_c^{(0)} = 0$, and covariance matrix $\boldsymbol{C}^{(0)} = \boldsymbol{I}$.

Choose distribution mean $\boldsymbol{m}^{(0)} \in \mathbb{R}^n$ and step-size $\sigma^{(0)} \in \mathbb{R}_+$, problem dependent.

Until termination criterion met

Sample new population of search points

$$\boldsymbol{x}_k^{(g+1)} \sim \boldsymbol{m}^{(g)} + \sigma^{(g)}\, \mathcal{N}(0,\, \boldsymbol{C}^{(g)}) \qquad \text{for } k = 1, \ldots, \lambda$$

Selection and recombination

$$\boldsymbol{m}^{(g+1)} = \sum_{i=1}^{\mu} \omega_i \boldsymbol{x}_{i:\lambda}^{(g+1)} \quad \text{where} \quad \sum_{i=1}^{\mu} \omega_i = 1\,,\ \omega_i > 0$$

Step-size control

$$\boldsymbol{p}_\sigma^{(g+1)} = (1-c_\sigma)\boldsymbol{p}_\sigma^{(g)} + \sqrt{c_\sigma(2-c_\sigma)\mu_{eff}}\ \boldsymbol{C}^{(g)^{-\frac{1}{2}}} \frac{\boldsymbol{m}^{(g+1)} - \boldsymbol{m}^{(g)}}{\sigma^{(g)}}$$

$$\sigma^{(g+1)} = \sigma^{(g)} \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\| \boldsymbol{p}_\sigma^{(g+1)} \|}{\mathrm{E}\,\| \mathcal{N}(0,\boldsymbol{I}) \|} - 1\right)\right)$$

Covariance matrix adaptation

$$h_\sigma^{(g+1)} = \begin{cases} 1 & \text{if } \dfrac{\| \boldsymbol{p}_\sigma^{(g+1)} \|}{\sqrt{1-(1-c_\sigma)^{2(g+1)}}} < \left(1.4 + \dfrac{2}{n+1}\right) \mathrm{E}\,\| \mathcal{N}(0,\boldsymbol{I}) \| \\ 0 & \text{Otherwise} \end{cases}$$

$$\boldsymbol{p}_c^{(g+1)} = (1-c_c)\boldsymbol{p}_c^{(g)} + h_\sigma^{(g+1)} \sqrt{c_c(2-c_c)\mu_{eff}}\ \frac{\boldsymbol{m}^{(g+1)} - \boldsymbol{m}^{(g)}}{\sigma^{(g)}}$$

$$\boldsymbol{C}^{(g+1)} = (1-c_1-c_\mu)\boldsymbol{C}^{(g)} + c_1 \boldsymbol{p}_c^{(g+1)} \boldsymbol{p}_c^{(g+1)^{\mathrm{T}}} + c_\mu \sum_{i=1}^{\mu} \omega_i\, \boldsymbol{y}_{i:\lambda}^{(g+1)} \boldsymbol{y}_{i:\lambda}^{(g+1)^{\mathrm{T}}}$$

Figure 16. Template of the CMA-ES