Iranian Journal of Science & Technology, Transaction B: Engineering, Vol. 34, No. B1, pp 15-34
Printed in The Islamic Republic of Iran, 2010
© Shiraz University

HYBRID GENETIC ALGORITHM AND PARTICLE SWARM OPTIMIZATION FOR THE FORCE METHOD-BASED SIMULTANEOUS ANALYSIS AND DESIGN*

A. KAVEH[1,2**] AND S. MALAKOUTI RAD[2]

[1]Institute for Mechanics of Materials and Structures, Vienna University of Technology, Karlsplatz 13, A-1040 Vienna, Austria, Email: alikaveh@iust.ac.ir
[2]Dept. of Civil Engineering, Iran University of Science and Technology, Tehran-16, I. R. of Iran

Abstract– The computational drawbacks of existing numerical methods have forced researchers to rely on heuristic algorithms. Heuristic methods are powerful in obtaining the solution of optimization problems. Although these methods are approximate methods (i.e. their solutions are good, but probably not optimal), they do not require the derivatives of the objective function and constraints. Also, the heuristics use probabilistic transition rules instead of deterministic rules. Here, an evolutionary algorithm based on the hybrid genetic algorithm (GA) and particle swarm optimization (PSO), denoted by HGAPSO, is developed in order to solve force method-based simultaneous analysis and design problems for frame structures. Suitability of the HGAPSO algorithm is compared to both GA and PSO for all the design examples, demonstrating its efficiency and superiority, especially for frames with a larger number of redundant forces.

Keywords– Simultaneous analysis and design, force method, trusses, frames, hybrid genetic algorithm and particle swarm optimization

1. INTRODUCTION

In the structural optimization literature, simultaneous analysis and design (SAND) formulation is a major class of alternative formulations, discussed since the 1960s. In this approach, the state and design variables are treated simultaneously as optimization variables. The analysis equations become an equality constraint in terms of the variables. SAND basically formulates the optimization problem in a mixed space of design and state variables to embed the analysis equations in a single optimization problem. Therefore, no explicit structural analysis or design sensitivity analysis is needed. Although theoretically very attractive, SAND formulations have one main difficulty: the problem becomes very large in terms of variables and constraints.

An early attempt to include state variables in the structural optimization problem was made by Schmit & Fox [1]. The basic idea was to transform an inequality-constrained minimization problem in the design variable space into an unconstrained problem in a space of mixed design and state variables. Detailed reviews of SAND formulations have been presented by Arora & Wang [2]. Most of these formulations are based on the displacement method of analysis, and equilibrium equations are used to form the equality constraints (Orozco & Ghattas [3, 4], Wu & Arora [5], Haftka & Kamat [6], Schmit & Fox [1], Ringertz [7], Kirsch & Rozvany [8], Larsson & Ronnqvist [9]). Force method is also employed for SAND by Kaveh & Rahami [10, 11] using GA.

*Received by the editors May 28, 2009; Accepted December 3, 2009.
**Corresponding author

In this paper, a force method-based SAND is presented and the compatibility equations are used to form the equality constraint based on the concept of minimum complementary strain energy. For optimization a hybrid form of GA and PSO is employed.

This paper is organized as follows. In section 2, the SAND problem based on the force method is formulated. In section 3, the PSO and GA algorithms are introduced briefly. In section 4, the HGAPSO algorithm and the optimization operators are described in detail. In section 5, HGAPSO is applied to different structures with continuous and discrete variables and the results are compared to those of standard PSO and standard GA. Finally, concluding remarks are presented in section 6.

2. FORCE METHOD BASED SAND FORMULATION

In the SAND approach, the formulation of the problem is modified by treating the state and design variables **q** (redundant forces) and **b** as independent optimization variables. In order to describe the approach, let us define a compound vector of optimization variables as $\mathbf{x} = [\mathbf{b} \quad \mathbf{q}]^{\mathrm{t}}$. The optimization problem is now expressed in terms of the vector **x** as follows:
Find **x** to minimize the weight function:

$$\boldsymbol{W}(\mathbf{b}) = \rho \sum_{s=1}^{nd} A(b_s) \sum_{i=1}^{nm} l_i \tag{1}$$

Subjected to the constraints:

$$h(\mathbf{x}) = 0 \tag{2.1}$$

$$g_i(\mathbf{x}) \le 0\,, \quad i = 1,\dots,n_g \tag{2.2}$$

where **x** is the vector containing the design variables and redundant forces; A(b_s) is the cross-sectional area for the *s*th group with *nm* members, related to the b_sth section available in the profile list *S*; *nd* is the number of design variables or the number of member groupings; n_g shows the number of inequality constraints.

The equality constraint: The formulation is based on the principle of minimum complementary strain energy ($\boldsymbol{U}^c$).

Suppose $\{\mathbf{p}\} = \{\mathrm{p}_1 \ \mathrm{p}_2 \ \cdots \ \mathrm{p}_N\}^{\mathrm{T}}$ is the vector of nodal forces, $\{\mathbf{q}\} = \{\mathrm{q}_1 \ \mathrm{q}_2 \ \cdots \ \mathrm{q}_R\}^{\mathrm{T}}$ contains the redundant forces, and $\{\mathbf{r}\} = \{\mathrm{r}_1 \ \mathrm{r}_2 \ \cdots \ \mathrm{r}_M\}^{\mathrm{T}}$ is comprised of the internal forces of the members. From equilibrium we have

$$\{\mathbf{r}\} = [\mathbf{B}_0]\{\mathbf{p}\} + [\mathbf{B}_1]\{\mathbf{q}\} = [\mathbf{B}_0 \ \ \mathbf{B}_1]\begin{bmatrix}\mathbf{p}\\ \mathbf{q}\end{bmatrix} \tag{3}$$

By removing the constraints or cutting the members corresponding to redundants, a statically determinate structure, known as the *basic (primary) structure*, will be obtained. $[\mathbf{B}_0]\{\mathbf{p}\}$ is known as a *particular solution* which satisfies equilibrium with the imposed load to the primary structure. $[\mathbf{B}_1]\{\mathbf{q}\}$ is a *complementary solution* formed from a maximal set of independent self-equilibrating stress systems, known as *statical basis* [12, 13].

The complementary strain energy ($\boldsymbol{U}^c$) is calculated as

$$\boldsymbol{U}^c = \frac{1}{2}\{\mathbf{r}\}^{\mathrm{T}}[\mathbf{F}_{\mathrm{m}}]\{\mathbf{r}\} \tag{4}$$

where $[\mathbf{F}_{\mathrm{m}}]$ is the unassembled flexibility matrix of the structure. Now $\{\mathbf{q}\}$ should be calculated such that $\boldsymbol{U}^c$ becomes minimum. Substituting $\{\mathbf{r}\}$ from Equation (3) in Equation (4) leads to

$$U^c = \frac{1}{2}[\mathbf{p}\ \mathbf{q}][\mathbf{H}]\begin{bmatrix}\mathbf{p}\\ \mathbf{q}\end{bmatrix}, \qquad \text{where } [\mathbf{H}] = [\mathbf{B}_0\ \ \mathbf{B}_1]^{\mathrm{T}}[\mathbf{F}_{\mathrm{m}}][\mathbf{B}_0\ \ \mathbf{B}_1] \tag{5}$$

Decomposing the matrix $[\mathbf{H}]$ into four submatrices $[\mathbf{H}_{\mathrm{pp}}]$, $[\mathbf{H}_{\mathrm{pq}}]$, $[\mathbf{H}_{\mathrm{qp}}]$, and $[\mathbf{H}_{\mathrm{qq}}]$, we obtain U^c as

$$U^c = \frac{1}{2}\left(\{\mathbf{p}\}^{\mathrm{T}}[\mathbf{H}_{\mathrm{pp}}]\{\mathbf{p}\} + \{\mathbf{p}\}^{\mathrm{T}}[\mathbf{H}_{\mathrm{pq}}]\{\mathbf{q}\} + \{\mathbf{q}\}^{\mathrm{T}}[\mathbf{H}_{\mathrm{qp}}]\{\mathbf{p}\} + \{\mathbf{q}\}^{\mathrm{T}}[\mathbf{H}_{\mathrm{qq}}]\{\mathbf{q}\}\right) \tag{6}$$

In order to satisfy the compatibility conditions, relative displacements in the released positions (member cuts) for the basic structure must be zero. By considering the second theorem of Castigliano, the partial derivative of the complementary strain energy U^c with respect to the redundants $\{\mathbf{q}\}$ must be equal to a zero vector.

$$\frac{\partial U^c}{\partial \mathbf{q}} = 0 \Rightarrow [\mathbf{H}_{\mathrm{qp}}]\{\mathbf{p}\} + [\mathbf{H}_{\mathrm{qq}}]\{\mathbf{q}\} = 0 \tag{7}$$

The equation above is obtained by considering the fact that $[\mathbf{H}]$ is symmetric and therefore $[\mathbf{H}_{\mathrm{qp}}]^{\mathrm{T}} = [\mathbf{H}_{\mathrm{pq}}]$.

In conventional design methods, $\{\mathbf{q}\}$ is obtained by solving the Equation (7), resulting in

$$\{\mathbf{q}\} = -[\mathbf{H}_{\mathrm{qq}}]^{-1}[\mathbf{H}_{\mathrm{qp}}]\{\mathbf{p}\} \tag{8}$$

In the present approach, finding the inverse of $[\mathbf{H}_{qq}]$ is not required. Thus $\{\mathbf{q}\}$ should be selected such that Eq. (7) holds. The left-hand side of this equation is a zero vector, and should be changed to a scalar. The best is to find its norm. For this norm to be zero, all the entries must be zero [10].

Thus the equality constraint will then be

$$h(\mathbf{x}): \quad \mathrm{norm}\left([\mathbf{H}_{qq}]\{\mathbf{q}\} + [\mathbf{H}_{qp}]\{\mathbf{p}\}\right) = 0 \tag{9}$$

The inequality constraints: In order to control the stability and yield conditions for all members of a frame structure, the following constraints must be satisfied:

$$g_i(\mathbf{x}) = \left[\frac{f_a}{F_a} + \frac{c_m f_b}{(1\text{-}f_a/F_e')F_b}\right]_i -1 \le 0 \quad \text{where } F_e{}'_i = \frac{105\times 10^5}{(k_i l_i / r_i)^2} \tag{10.1}$$

$$g_i(\mathbf{x}) = \left[\frac{f_a}{0.6F_y} + \frac{f_b}{F_b}\right]_i -1 \le 0 \qquad \text{if} \left(\frac{f_a}{F_a}\right)_i > 0.15 \tag{10.2}$$

$$g_i(\mathbf{x}) = \left[\frac{f_a}{F_a} + \frac{f_b}{F_b}\right]_i -1 \le 0 \qquad \text{if} \left(\frac{f_a}{F_a}\right)_i \le 0.15 \quad i = 1,\dots,m \tag{10.3}$$

$$g_i(\mathbf{x}) = \frac{\Delta_{\mathrm{drift}}}{\Delta_{\mathrm{allowable}}} -1 \le 0 \tag{10.4}$$

In which f_a and f_b are the axial and flexural member stresses, respectively. F_y is the yield stress; The corresponding allowable stresses F_a and F_b in relations (10.1) to (10.3) are calculated based on the design code provisions AISC [14]. Δ_{drift} is the lateral displacement at the top level of the frame, and $\Delta_{\mathrm{allowable}}$ is the displacement permitted by the code of practice, which is taken as 0.005 times the height of the frame. The constants k and c_m are taken as 1.5 and 1.0, respectively.

The inequality constraint for the solution will be

$$g(\mathbf{x}) = \sum_{i=1}^{m} \max\left[g_i(\mathbf{x}), 0\right] \tag{11}$$

Penalty Function: The penalty function is defined as follows:

$$f_{penalty}(\mathbf{x}) = \boldsymbol{W}(\mathbf{b}) \times \left(\alpha\, h(\mathbf{x}) + \beta\, g(\mathbf{x})\right) \tag{12}$$

Where α and β are user defined coefficients. Therefore, the main goal will be to minimize the following function:

$$\boldsymbol{F}(\mathbf{x}) = \boldsymbol{W}(\mathbf{b}) + f_{penalty}(\mathbf{x}) = \boldsymbol{W}(\mathbf{b}) \times \left(1 + \alpha\, h(\mathbf{x}) + \beta\, g(\mathbf{x})\right) \tag{13}$$

3. GENETIC ALGORITHMS AND PARTICLE SWARM OPTIMIZATION

a) Basic concepts of genetic algorithms

Genetic algorithm (GA) is a well-known and frequently used evolutionary computation technique. This method was originally developed by John Holland [15] and his PhD students Hassan *et al*. [16]. The idea was inspired from Darwin's natural selection theorem which is based on the idea of the survival of the fittest. The GA is inspired by the principles of genetics and evolution, and mimics the reproduction behavior observed in biological populations.

In GA, a candidate solution for a specific problem is called an individual or a chromosome and consists of a linear list of genes. GA begins its search from a randomly generated population of designs that evolve over successive generations (iterations), eliminating the need for a user-supplied starting point. To perform its optimization-like process, the GA employs three operators to propagate its population from one generation to another. The first operator is the "selection" operator in which the GA considers the principal of "survival of the fittest" to select and generate individuals (design solutions) that are adapted to their environment. The second operator is the "crossover" operator, which mimics mating in biological populations. The crossover operator propagates features of good surviving designs from the current population into the future population, which will have a better fitness value on average. The last operator is "mutation", which promotes diversity in population characteristics. The mutation operator allows for global search of the design space and prevents the algorithm from getting trapped in local minima [16].

b) Basic concepts of PSO

Particle Swarm Optimization (PSO) is one of the recent evolutionary optimization methods. This technique was originally developed by Kennedy & Eberhart [17] in order to solve problems with continuous search space. PSO is based on the metaphor of social interaction and communication, such as bird flocking and fish schooling. This algorithm can be easily implemented and it is computationally inexpensive, since its memory and CPU speed requirements are low [18].

PSO shares many common points with GA. It conducts the search using a population of particles which correspond to individuals in GA. Both algorithms start with a randomly generated population. PSO does not have a direct recombination operator. However, the stochastic acceleration of a particle towards its previous best position, as well as towards the best particle of the swarm (or towards the best in its neighborhood in the local version), resembles the recombination procedure in evolutionary computation [19-21].

Compared to GA, the PSO has some attractive characteristics. It has memory, so knowledge of good solutions is retained by all particles, whereas in GA, previous knowledge of the problem is destroyed once the population changes. PSO does not use the filtering operation (such as selection in GAs), and all the members of the population are maintained through the search procedure to share their information effectively.

PSO uses social rules to search in the design space by controlling the trajectories of a set of independent particles. The position of each particle, $\mathbf{x}_i$, representing a particular solution of the problem, is used to compute the value of the fitness function to be optimized. Each particle may change its position, and consequently may explore the solution space, simply varying its associated velocity. In fact, the main PSO operator is the velocity update, that takes into account the best position, in terms of fitness value reached by all the particles during their paths, $\mathbf{p}_g^{\mathrm{t}}$, and the best position that the agent itself has reached during its search, $\mathbf{p}_i^{\mathrm{t}}$, resulting in a migration of the entire swarm towards the global optimum.

At each iteration, the particle moves around according to its velocity and position; the cost function to be optimized is evaluated for each particle in order to rank the current location. The velocity of the particle is then stochastically updated according to

$$\mathbf{v}_i^{\mathrm{t}+1} = \omega \mathbf{v}_i^{\mathrm{t}} + \mathrm{C}_1\, r_1^{\mathrm{t}} \left(\mathbf{p}_i^{\mathrm{t}} - \mathbf{x}_i^{\mathrm{t}} \right) + \mathrm{C}_2\, r_2^{\mathrm{t}} \left(\mathbf{p}_g^{\mathrm{t}} - \mathbf{x}_i^{\mathrm{t}} \right) \tag{14.1}$$

$$\mathbf{x}_i^{\mathrm{t}+1} = \mathbf{x}_i^{\mathrm{t}} + \mathbf{v}_i^{\mathrm{t}+1} \tag{14.2}$$

Equation (14.1) describes how the velocity is dynamically updated and Eq. (14.2) is used to update the position of the "flying" particles.

$\mathbf{v}_i^{\mathrm{t}}$ is the velocity vector at iteration t, r_1 and r_2 represent random numbers in the range [0,1]; $\mathbf{p}_i^{\mathrm{t}}$ denotes the best ever particle position of particle *i*, and $\mathbf{p}_g^{\mathrm{t}}$ corresponds to the global best position in the swarm up to iteration t [19].

The remaining terms are problem dependent parameters; for example, C_1 and C_2 represent ''trust'' parameters indicating how much confidence the current particle has in itself (C_1 or cognitive parameter) and how much confidence it has in the swarm (C_2 or social parameter), and ω is the inertia weight. The latter term plays an important role in the PSO convergence behavior since it is employed to control the exploration abilities of the swarm. It directly affects the current velocity, which in turn is based on the previous history of velocities. Large inertia weights allow for wide velocity updates providing the global exploration of the search space, while small inertia values concentrate the velocity updates to nearby regions of the design space.

4. HYBRID OF GA AND PSO (HGAPSO)

Although GAs have been successfully applied to a wide spectrum of problems, using GAs for large-scale optimization could be very expensive due to its requirement of a large number of function evaluations for convergence. This would result in a prohibitive cost for computation of function evaluations even with the best computational facilities available today.

Considering the efficiency of the PSO, and the compensatory property of GA and PSO, combining the searching abilities of both methods in one algorithm seems to be a logical approach. In this paper, the hybrid of GA and PSO named HGAPSO, originally presented by Juang [22], is used. This algorithm consists of four major operators: enhancement, selection, crossover, and mutation.
The flowchart of the HGAPSO is shown in Fig. 1.

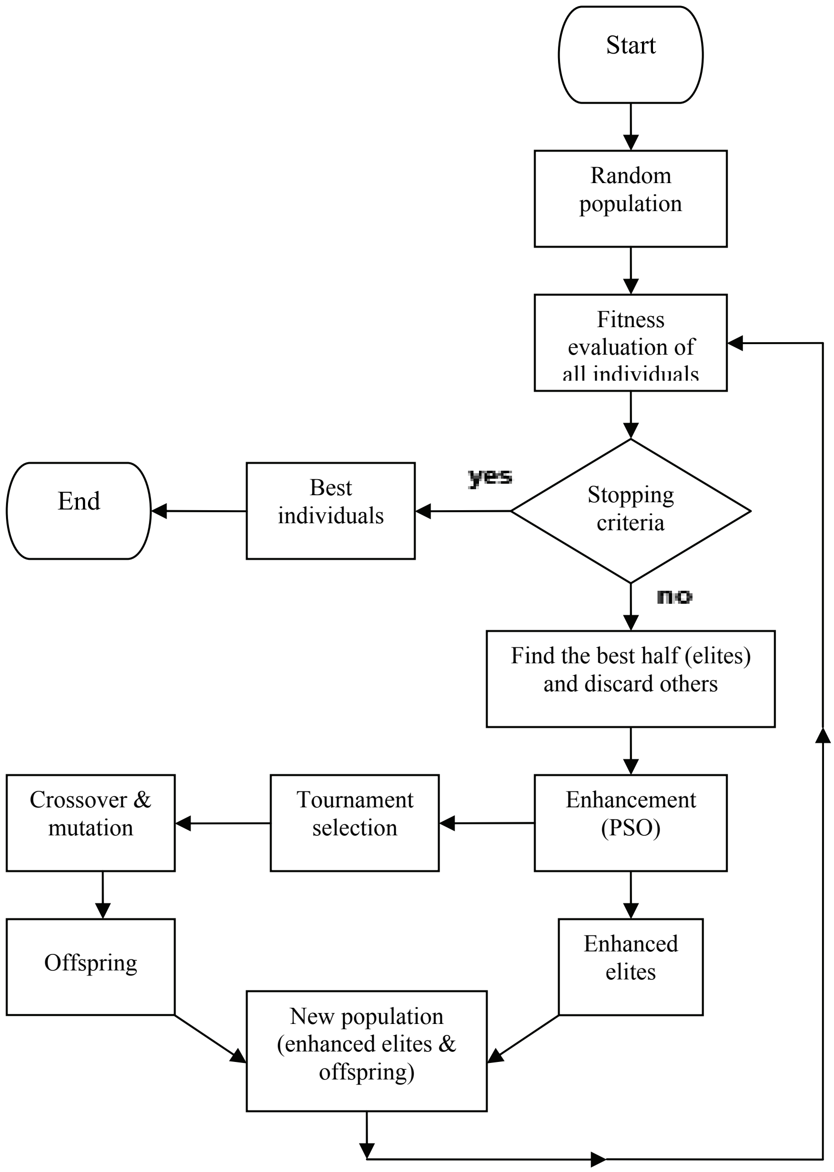

Fig. 1 The flow chart of the HGAPSO

Before describing the details of the algorithm and its operators, the issue of coding is presented.

It is obvious that the feasible region in constrained optimization problems may be of any shape (convex or concave and connected or disjointed). In real-parameter constrained optimization using GAs, schemata specifying contiguous regions in the search space (such as 110*…*) may be considered to be more important than schemata specifying discrete regions in the search space (such as (*1*10*...*), in general. Since, any arbitrary contiguous region in the search space cannot be represented by single Holland's schema and since the feasible search space can usually be of any arbitrary shape, it is expected that the single-point crossover operator used in binary GAs will not always be able to create feasible children solutions from two feasible parent solutions.

The floating-point representation of variables in a GA and a search operator that respects contiguous regions in the search space may be able to eliminate the above two difficulties associated with binary coding and single-point crossover.

Hence, a floating point coding scheme is adopted here for all of the GA, PSO and HGAPSO. For the frame structures where design variables must have discrete values, the solutions are achieved by rounding the design variables to the nearest permissible integer number.

An elitist strategy has been used in the algorithms of this work, where the best solution is given the opportunity to be directly carried over to the next generation.

Enhancement: In each generation, after the fitness values of all the individuals in the population are calculated, the top-half best performing ones are marked. These individuals are regarded as elites. Instead of reproducing the elites directly to the next generation as elite GAs do, we first enhance the elites. The enhancement operation tries to mimic the maturing phenomenon in nature, where individuals will become more suitable to the environment after acquiring knowledge from the society. Furthermore, by using these enhanced elites as parents, the generated offspring will achieve better performance than those bred by original elites [22]. The enhancement of the elites is performed by the velocity and position update procedures in PSO (Eqs. (14.1) and (14.2)).

By analyzing the PSO, Perez and Behdinan [23] presented the stability conditions for the algorithm as shown below:

$$C_1 r_1 + C_2 r_2 > 0 \tag{15.1}$$

$$\frac{C_1 r_1 + C_2 r_2}{2} - \omega < 1 \tag{15.2}$$

$$\omega < 1 \tag{15.3}$$

Knowing that $r_1, r_2 \in [0,1]$, the following parameter selection heuristic was derived:

$$0 < C_1 + C_2 < 4 \tag{16.1}$$

$$\frac{C_1 + C_2}{2} - 1 < \omega < 1 \tag{16.2}$$

It was concluded that if ω, C_1, and C_2 are selected with the heuristics specified in (16a) and (16b), the system will guaranteed convergence to an equilibrium point.

In this article the parameters C_1, C_2 and ω are chosen considering the above conditions and according to the searching ability needed to solve each of the problems.

Due to the importance of the inertia weight in controlling the global/local search behavior of the PSO, a dynamic improvement has proven useful by forcing an initial global search with a high inertia weight ($\omega \approx 1$), and subsequently narrowing down the algorithm exploration to feasible areas of the design space by decreasing its value towards local search values ($\omega < 0.5$).

In this approach, a dynamic decrease of ω value has been suggested based on a fraction multiplier (k_w) as shown in Eq. (17). When no improvements had been made for a predefined number of consecutive design iterations [24]:

$$\omega^{t+1} = k_w \omega^{t} \tag{17}$$

It should be noted that the elites in each generation can be from both groups of the previous generation, i.e., the enhanced elites or the produced offspring. If the elite *i* is an offspring produced by the parents of the previous generation, then $\mathbf{v}_i^t$ is set to zero, and $\mathbf{p}_i^t$ is set to $\mathbf{x}_i^t$, i.e., the newly generated individual itself. Otherwise, $\mathbf{p}_i^t$ records the best solution of individual *i* evolved so far.

Selection: In the HGAPSO, the GA operations are performed on the enhanced elites achieved by PSO.

In order to select parents for the crossover operation, the tournament selection scheme is used. Two enhanced elites are selected randomly, and their fitness values are compared to select the one with better fitness as a parent and place it in the mating pool.

This scheme is used as the selection operator in the GA as well.

Crossover: Parents are selected randomly from the mating pool in groups of two and two offspring are created by performing crossover on the parent solutions. In this paper, a simulated binary crossover (SBX) is used. SBX operator is particularly suitable here because the spread of children solutions around parent solutions can be controlled using a distribution index, η_c. With this operator any arbitrary contiguous region can be searched, provided there is enough diversity maintained among the feasible parent solutions [24].

Here, we have used small values for η_c at the first generations in order to have offspring solutions away from parents and to provide a global search. The value of η_c is increased with the increase of generation, guiding the algorithm towards a local search.

Mutation: The final genetic operator is mutation. It can create a new genetic material in the population to maintain the population's diversity.

In this article, mutation is not applied to all of the population, and a mutation probability ($\boldsymbol{P}_m$) is assigned to every individual according to its fitness value:

$$\boldsymbol{P}_{m\,i} = 0.5 \times \left[\frac{\mathbf{F}_{\max} - \mathbf{F}_i}{\mathbf{F}_{\max} - \mathbf{F}_{\text{ave}}} \right] \qquad \text{if } \mathbf{F}_i \geq \mathbf{F}_{\text{ave}} \tag{18.1}$$

$$\boldsymbol{P}_{m\,i} = \left[\frac{\mathbf{F}_{\text{ave}} - \mathbf{F}_i}{\mathbf{F}_{\max} - \mathbf{F}_{\text{ave}}} \right] \qquad \text{if } \mathbf{F}_i < \mathbf{F}_{\text{ave}} \tag{18.2}$$

$\mathbf{F}_i$ is the fitness value of the individual *i*, $\mathbf{F}_{\max}$ and $\mathbf{F}_{\text{ave}}$ are the maximum and average fitness values of the population in each generation. After assigning the $\boldsymbol{P}_m$ s, a random number in the range [0, 1] is created for each generation. The individuals having a $\boldsymbol{P}_m$ greater than this number are mutated.

The mutation operator used here is a variable dependent random mutation. In the random mutation operator, a solution is created in the vicinity of the parent solution with a uniform probability distribution [25]:

$$x_i^{(1,\,t+1)} = x_i^{(1,\,t)} + (r_i - 0.5)\,\Delta_i \tag{19}$$

r_i is a random number in [0, 1]. Δ_i is the user defined maximum perturbation allowed in the *i*th decision variable(x_i).Care should be taken to check if the above calculation takes $x_i^{(1,\,t+1)}$ outside of the specified lower and upper limits.

In this approach, at each generation, Δ_i for a variable x_i is calculated using the average of that variable or the difference between its maximum and minimum in the population, i.e.

$$\Delta_i = 0.5 \times (\max(x_i) - \min(x_i)) \tag{20a}$$

$$\Delta_i = (0.025 \sim 0.075) \times \text{ave}(x_i) \tag{20b}$$

Decision on the formula and the multipliers to be used depends on the structure as well as the variables, especially for the SAND problem where the design and state variables have different natures and values.

After applying the GA operators, the offspring and the enhanced elites from PSO, form the new population and their fitness is evaluated and compared in order to select the elites for the next generation.

5. NUMERICAL EXAMPLES

Example 1 (A 10-bar truss): The first example considers a well-known test problem corresponding to a 10-bar truss non-convex optimization as shown in Fig. 2. In this problem the cross-sectional area for each of the 10 members in the structure are being optimized towards the minimization of total weight. The cross-sectional area varies between 0.1 in.^{2} (0.645 cm^2) and 35.0 in.^{2} (225.806 cm^2). Constraints are specified in terms of stress and displacement of the truss members. The allowable stress for each member is 25 ksi (172368.93 kN/m^2) for both tension and compression, and the allowable displacement on the nodes is ±2 in. (5.08 cm) in the x and y directions. The density of the material is 0.1 lb/in.^{3} (27.1447 kN/m^3), Modulus of elasticity is E = 10^4 ksi (68947572.93 kN/m^2) and vertical downward loads of 100 kips (444.822 kN) are applied at nodes 2 and 4. In total the problem has a variable dimensionality of 12 (10 design variables and 2 redundant, all in a continuous search space) and constraint dimensionality of 22 (10 stress constraints and 12 displacement constraints).

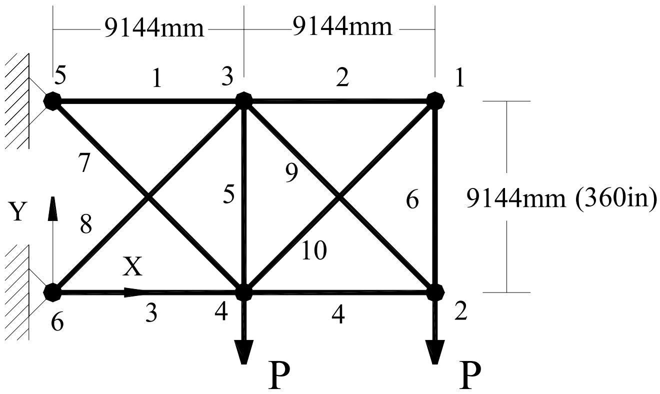

Fig. 2. The 10-bar truss

The problem is solved by the algorithm presented in the previous section. The population size considered is 150. The results related to the three algorithms are presented in Table 1 and compared with the results in the literature in Table 2. The highlighted sections present constraint violations.

Table 1. Results for the 10-bar truss

Member Areas-in.^{2}(cm^2)	GA	PSO	HGAPSO
A1	30.4802 (196.646)	30.7081 (198.116)	30.6395 (197.674)
A2	0.1 (0.645)	0.1 (0.645)	0.1 (0.645)
A3	23.2578 (150.050)	23.9127 (154.275)	23.0607 (148.778)
A4	15.1859 (97.973)	14.7259 (95.006)	15.0192 (96.898)
A5	0.1 (0.645)	0.1 (0.645)	0.1 (0.645)
A6	0.546752 (3.527)	0.1 (0.645)	0.591063 (3.813)
A7	7.4523 (48.079)	8.54672 (55.140)	7.49068 (48.327)
A8	2.0453 (13.195)	21.0104 (135.551)	21.108 (136.180)

Table 1 Continued.

A9	21.5427 (138.985)	20.8107 (134.262)	21.5653 (139.131)
A10	0.1 (0.645)	0.1 (0.645)	0.1 (0.645)
Weight-lb (kN)	5060.87 (22.511)	5076.68 (22.581)	5061.4 (22.513)
σ_{max}-ksi (kN/m^2)	24.99999 (172368.863)	20.38199 (140528.874)	24.9621 (172107.621)
Δ_{max}-inch (cm)	1.9999981 (5.079995)	1.99999944 (5.079999)	1.9999703 (5.079925)

The GA presents the best answer, while the answer obtained by PSO is the poorest in quality.

Table 2. Results in the literature for the 10-bar truss

Member Areas-inch^2 (cm^2)	Kaveh & Rahami [10]	Haftka & Grdal [26]	El-Sayed & Jang [27]	Galante [24]	Memari & Fuladgar [28]	Rizzi [29]	Haug & Arora [30]
A1	30.6677 (197.856)	30.52 (196.903)	32.966 (212.683)	30.44 (196.387)	30.561 (197.167)	30.731 (198.264)	30.031 (193.748)
A2	0.1 (0.645)	0.1 (0.645)	0.1 (0.645)	0.1 (0.645)	0.1 (0.645)	0.1(0.645)	0.1 (0.645)
A3	22.8722 (147.562)	23.2 (149.677)	22.799 (147.090)	21.79 (140.580)	27.946 (180.296)	23.934 (154.413)	23.274 (150.155)
A4	15.3445 (98.997)	15.22 (98.193)	14.146 (91.264)	14.26 (92.000)	13.619 (87.864)	14.733 (95.051)	15.286 (98.619)
A5	0.1 (0.645)	0.1 (0.645)	0.1 (0.645)	0.1 (0.645)	0.1 (0.645)	0.1(0.645)	0.1 (0.645)
A6	0.4635 (2.99)	0.551 (3.555)	0.739 (4.768)	0.451 (2.910)	0.1 (0.645)	0.1 (0.645)	0.557 (3.594)
A7	7.4796 (48.255)	7.457 (48.109)	6.381 (41.168)	7.628 (49.213)	7.907 (51.013)	8.542 (55.110)	7.468 (48.181)
A8	20.9651 (135.258)	21.04 (135.742)	20.912 (134.916)	21.63 (139.548)	19.345 (124.806)	20.954 (135.187)	21.198 (136.761)
A9	21.7026 (140.016)	21.53 (138.903)	20.978 (135.342)	21.36 (137.806)	19.273 (124.342)	21.836 (140.877)	21.618 (139.471)
A10	0.1 (0.645)	0.1 (0.645)	0.1 (0.645)	0.1 (0.645)	0.1 (0.645)	0.1(0.645)	0.1 (0.645)
Weight lb (kN)	5061.9 (22.515)	5060.8 (22.510)	5013.24 (22.299)	4987 (22.182)	4981.1 (22.156)	5061.6 (22.514)	5060.92 (22.514)
σ_{max} ksi (kN/m^2)	24.99468 (172332.3)	25.00271 (172387.7)	31.28779 (215721.8)	25.08667 (172966.6)	20.60011 (142032.8)	20.35507 (140343.3)	24.92077 (171822.7)
Δ_{max} inch (cm)	1.999988 (5.0800)	1.999965 (5.0799)	2.013124 (5.1133)	2.027979 (5.1511)	2.060467 (5.2336)	1.98233 (5.0351)	1.999989 (5.0800)
Member Areas-in.^{2} (cm^2)	Ghasemi et al. [31]	Perez & Behdinan [23]	Gellatly & Berke [32]	Schmit & Miura [33]	Schmit & Miura [33]	Schmit & Farshi [34]	Dobbs & Nelson [35]
A1	25.73 (166.000)	33.5 (216.129)	31.35 (202.258)	30.67 (197.871)	30.57 (197.225)	33.432 (215.690)	30.5 (196.774)
A2	0.109 (0.703)	0.1 (0.645)	0.1 (0.645)	0.1 (0.645)	0.369 (2.381)	0.1 (0.645)	0.1 (0.645)
A3	24.85 (160.322)	22.766 (146.877)	20.03 (129.226)	23.76 (153.290)	23.97 (154.645)	24.26 (156.516)	23.29 (150.258)
A4	16.35 (105.484)	14.417 (93.013)	15.6 (100.645)	14.59 (94.129)	14.73 (95.032)	14.26 (92.000)	15.428 (99.535)

Table 2 Continued.

A5	0.106 (0.684)	0.1 (0.645)	0.14 (0.903)	0.1 (0.645)	0.1 (0.645)	0.1 (0.645)	0.1 (0.645)
A6	0.109 (0.703)	0.1 (0.645)	0.24 (1.548)	0.1 (0.645)	0.364 (2.348)	0.1 (0.645)	0.21 (1.355)
A7	8.700 (56.129)	7.534 (48.606)	8.350 (53.871)	8.578 (55.342)	8.547 (55.142)	8.388 (54.116)	7.649 (49.348)
A8	21.41 (138.129)	20.467 (132.045)	22.21 (143.290)	21.07 (135.935)	21.11 (136.193)	20.74 (133.8060	20.98 (135.355)
A9	22.3 (143.871)	20.392 (131.561)	22.06 (142.322)	20.96 (135.226)	20.77 (134.000)	19.69 9127.032)	21.818 (140.761)
A10	0.122 (0.787)	0.1 (0.645)	0.1 (0.645)	0.1 (0.645)	0.32 (2.065)	0.1 (0.645)	0.1 (0.645)
Weight-lb (kN)	5095.65 (22.665)	5024.21 (22.348)	5112 (22.738)	5076.85 (22.582)	5107.3 (22.717)	5089 (22.636)	5080.0 (22.596)
σ_{max}-ksi(kN/m^2)	18.52549 (127728.8)	25.01711 (172486.9)	22.93687 (158144.2)	20.3494 (140304.2)	20.39588 (140624.6)	21.19186 (146112.7)	24.06749 (165939.5)
Δ_{max}-inch(cm)	0.01259 (0.0320)	2.0389 (5.1788)	1.99999 (5.0800)	1.99989 (5.0797)	1.99998 (5.0799)	1.99982 (5.0795)	1.99997 (5.0799)

The convergence for the three algorithms is illustrated in Fig. 3.

It took about 3812 and 2483 generations for the GA and the PSO algorithms to converge, respectively. However the HGAPSO algorithm converged in only 1855 generations. The number of required generations for reaching a solution by GA is greater than the other two. Instead, the last result of this algorithm is the best.

Although the HGAPSO does not result in the best answer, compared with the two others, the smaller generation and lower function evaluation of this algorithm is noticeable, Fig. 3.

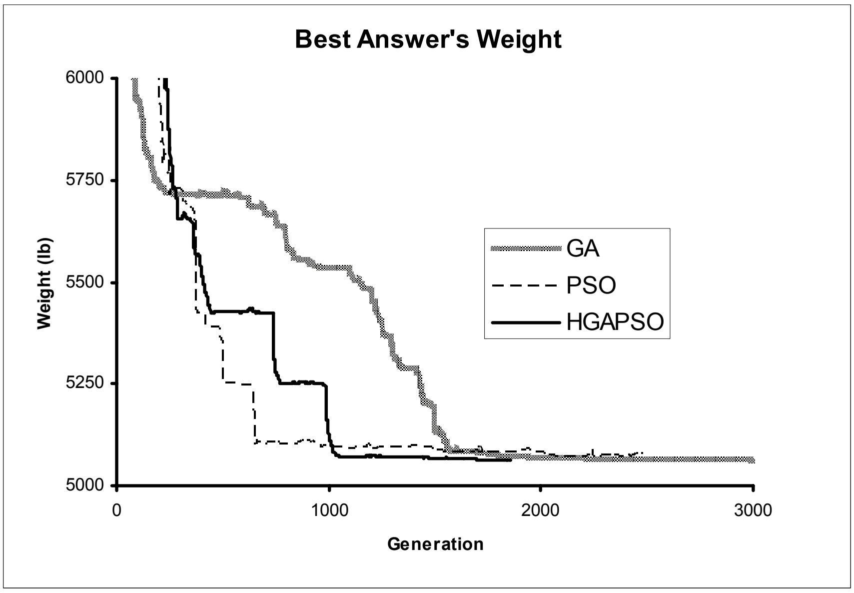

Fig. 3. Convergence of the three algorithms for the 10-bar truss

For this truss the last generation for GA, PSO and HGAPSO were 3793, 2434 and 1956, leading to average weights of 5066.05lb, 5084.5lb and 5067.6lb, respectively, Fig. 4.

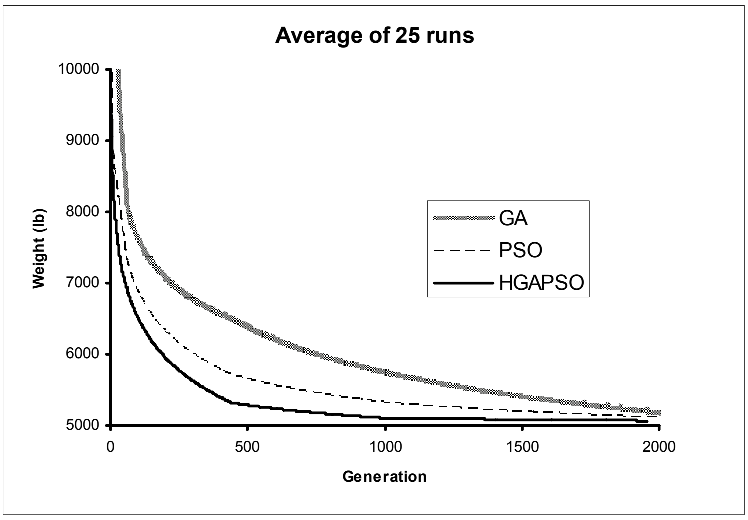

Fig. 4. Best weight versus the number of generations for the 10-bar truss

Example 2 (A one-bay eight-storey steel frame): Figure 5 shows the configuration and applied loads of a one-bay eight-story frame structure. The 24 members of the structure have been categorized into eight groups, as indicated in the figure. The lateral drift at the top of the structure is the only performance constraint (no more than 2 in.). The modulus of elasticity is taken as E=200 GPa (29000 ksi) and for the material density, ρ=76.8 kN/m^3 (2.83×10^{-4} kips/in.^{3}).

A set of 267 discrete W-sections from the American Institute of Steel Construction (AISC) are used for the possible cross-sectional areas for each member. In total the problem has a variable dimensionality of 32 (8 discrete design variables and 24 redundant forces in a continuous search space) and constraint dimensionality of 1 (displacement constraint).

The algorithms require a population size of 500 to converge to the best answers possible.

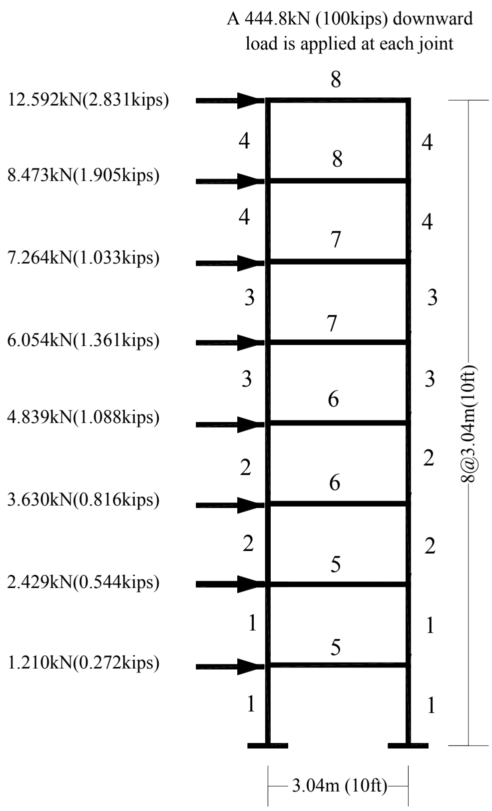

Fig. 5. A one-bay eight-storey frame

Optimal values of the 8 design variables obtained by the three algorithms are listed in Table 3. The results obtained by the other researchers are illustrated in Table 4.

Table 3. Results for the one-bay eight-storey frame

Group number	GA	PSO	HGAPSO
1	W21×44	W21×44	W18×35
2	W18×35	W16×26	W18×35
3	W14×22	W21×44	W14×22
4	W12×14	W12×16	W12×16
5	W16×26	W14×30	W16×31
6	W18×40	W21×44	W21×44
7	W18×35	W14×22	W18×35
8	W12×22	W16×26	W16×26
Weight (kN)	31.3786	33.9814	31.243
Δ_{drift} (cm)	5.06415	4.99796	5.06645

The HGAPSO algorithm found the optimal weight of the one-bay eight-story frame to be 31.243 kN (7.025 kips). Like the previous example, the PSO seems to be a weak algorithm for reaching an optimal solution.

Table 4. Results in the literature for the one-bay eight-storey frame

Group Number	Kaveh & Shojaee [36]	Kaveh & Talatahari [37]	Kaveh & Talatahari [37]	Khot et al. [38]	Camp et al. [39]
1	W 21×50	W 18×35	W 18×35	W 14×34	W 18×46
2	W 16×26	W 16×31	W 16×31	W 10×39	W 16×31
3	W 16×26	W 16×26	W 14×22	W10×33	W 16×26
4	W 12×14	W 14×22	W 12×16	W 8×18	W 12×16
5	W 16×26	W 16×31	W 21×48	W 21×68	W 18×35
6	W 18×40	W 18×40	W 18×40	W 24×55	W 18×35
7	W 18×35	W 16×26	W 16×31	W 21×50	W 18×35
8	W 14×22	W 14×22	W16×36	W 12×40	W 16×26
Weight (kN)	31.68	30.91	32.29	41.02	32.83
Δ_{drift} (cm)	5.00245	5.19599	5.29819	5.052143	5.26538

Figure 6 compares the convergence rate of the three algorithms. Although the PSO does not take as many generations as the GA, the quality of the solution obtained by this method is poor and the advantage of the GA is its better solution. Although the GA provides an answer with almost the same quality as the HGAPSO, the convergence rate of the algorithms show the superiority of the HGAPSO. It took the HGAPSO 4968 generations to converge to the solution, while the GA and PSO algorithms converged in 9318 and 6427 generations, respectively.

For this frame the last generation for GA, PSO and HGAPSO were 9284, 6416 and 4953, leading to average weights of 31.78kN, 35.15kN and 31.63kN, respectively, Fig. 7.

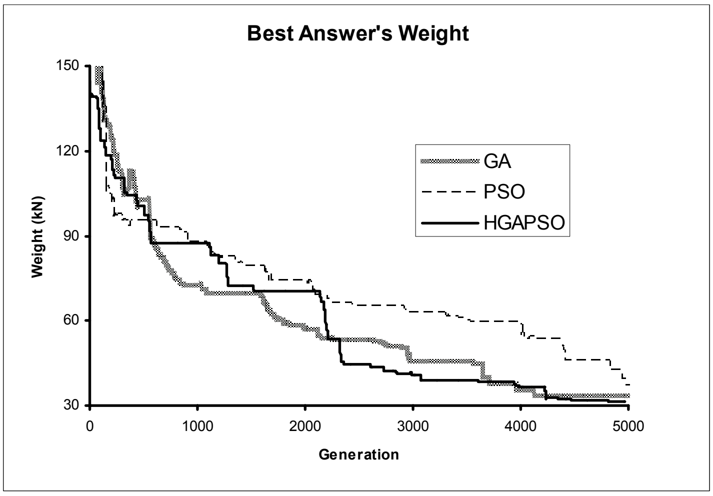

Fig. 6. Convergence of the three algorithms for the one-bay eight-storey frame

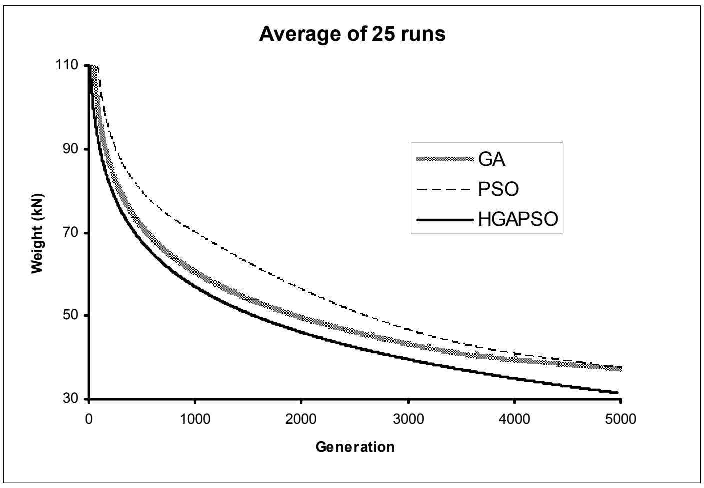

Fig. 7. Best weight versus the number of generations for the one-bay eight-storey frame

Example 3 (A two-bay three-storey steel frame): This frame was optimized by Kameshki and Saka [40] considering semi-rigid joints. Fig. 8 shows the configuration and applied loads of the frame. The 15 members of the structure have been categorized into seven groups, as indicated in the figure.

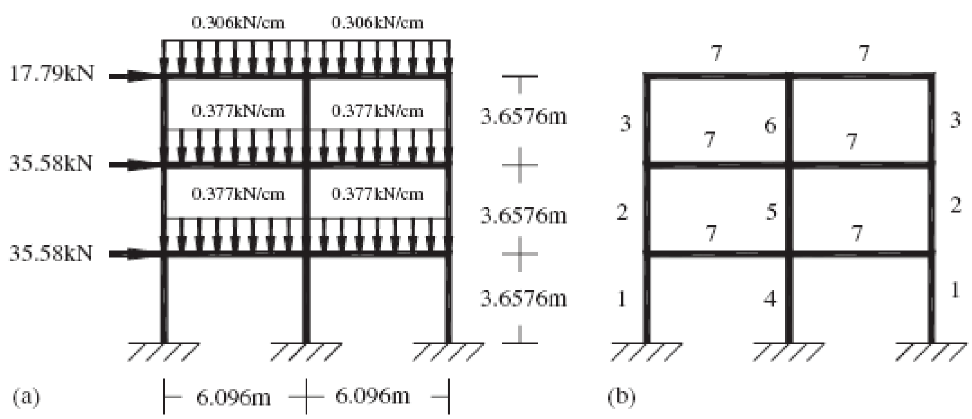

Fig. 8. The two-bay three-storey frame: a) loading of the frame; b) member grouping

The material properties are those of DIN-ST 37:
Weight density of the material, ρ = 76.9724 kN/m^3

Modulus of elasticity, E= 2.0594×10^8 kN/m^2
Yield stress is taken as F_y = 2400 kgf/cm^2 (23535.96 kN/m^2)

A set of 267 discrete W-sections from the American Institute of Steel Construction (AISC)[39] are used for the possible cross-sectional areas for each member.

In total the problem has a variable dimensionality of 25 (7 discrete design variables and 18 redundant forces in a continuous search space).

The constraints include the design specifications of AISC-ASD presented in the previous section for each member. The allowable inter-storey drift is 1.219 cm and allowable sway of the top storey is 3.65 cm. The population size considered for the problem was 425.

Optimal values of the seven design variables obtained by the three algorithms are listed in Table 5.

Table 5. Results for the two-bay three-storey frame

Group Number	GA	PSO	HGAPSO
1	W16×31	W18×40	W16×31
2	W18×35	W12×26	W16×36
3	W10×19	W16×26	W6×9
4	W21×55	W14×48	W18×55
5	W16×31	W16×40	W16×31
6	W10×17	W8×13	W14×26
7	W21×44	W21×44	W21×44
Weight (kN)	38.22043	38.87062	37.75168
Interaction ratio	0.99669	0.99026	0.99754
Δ_{storey} (cm)	0.45348	0.50146	0.46614
Δ_{drift} (cm)	1.2675	1.12211	1.3045

It is apparent from the table that stress constraints were dominant in the design problem. The HGAPSO obtained the best solution (37.75168 kN), the GA result was acceptable, but the PSO converged to a local optimum again.

Figure 9 compares the convergence rate of the three algorithms.
It took the GA 8925 generations to converge to the solution, while the PSO and HGAPSO algorithms converged in 4396 and 4262 generations, respectively.

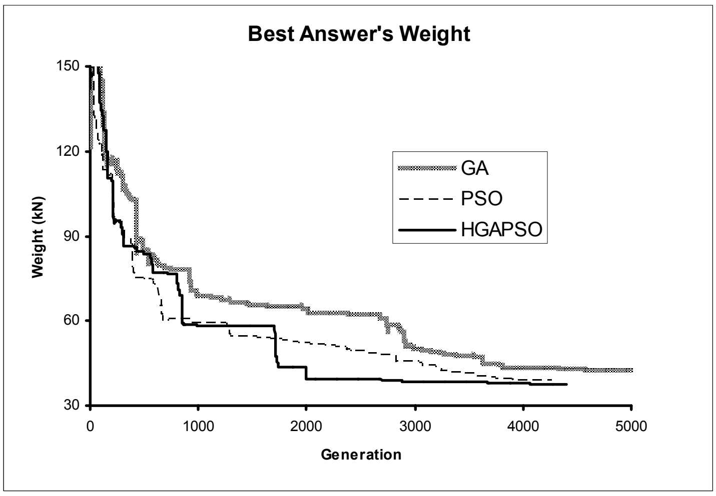

Fig. 9. Convergence of the three algorithms for the two-bay three-storey frame

For this frame the last generation for GA, PSO and HGAPSO were 8925, 4298 and 4380, leading to average weights of 38.8kN, 39.83kN and 37.98kN, respectively (see Fig. 10).

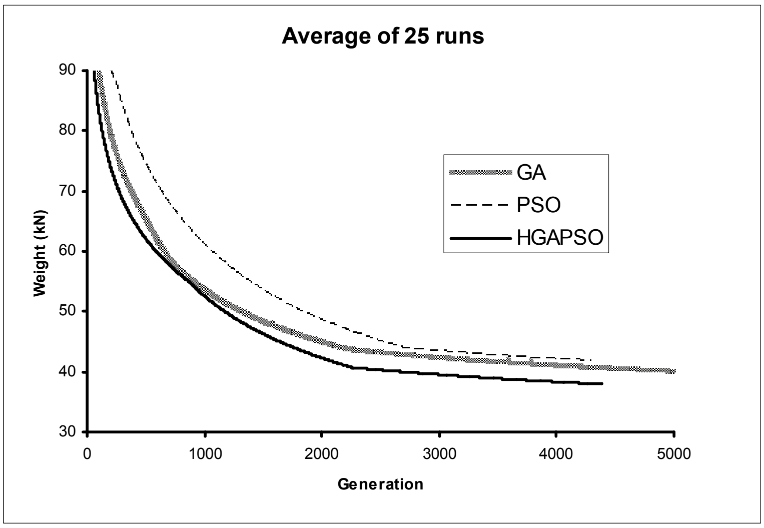

Fig. 10. Best weight versus the number of generations for the two-bay three-storey frame

Example 4 (A three-bay three-storey steel frame): Figure 11 shows the configuration and applied loads of the frame. The 21 members of the structure have been categorized into 12 groups, as indicated in the figure.

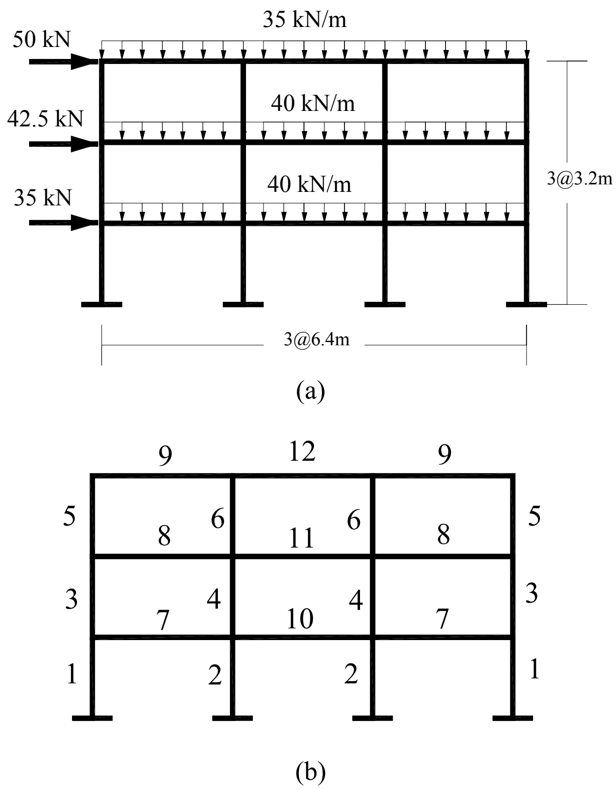

Fig. 11. The three-bay three-storey frame; (a) loadings (b) member grouping

The material properties and the sections used are the same as the previous example.
In total the problem has a variable dimensionality of 39 (12 discrete design variables and 27 redundant forces in a continuous search space).

The constraints include the design specifications of AISC-ASD for each member. The allowable sway of the top storey is 3.17 cm. The population size considered for the problem was 700.

Optimal values of the 12 design variables obtained by the three algorithms are listed in Table 6.

Table 6. Results for the three-bay three-storey frame

Group Number	GA	PSO	HGAPSO
1	W21×48	W8×24	W10×26
2	W18×35	W21×50	W18×40
3	W18×35	W8×24	W18×35
4	W21×48	W18×60	W24×55
5	W14×34	W16×31	W14×34
6	W10×19	W16×26	W10×19
7	W21×48	W21×50	W21×48
8	W21×44	W21×50	W21×44
9	W18×40	W18×50	W18×40
10	W21×44	W21×48	W21×50
11	W18×40	W16×50	W18×40
12	W18×40	W18×40	W16×40
Weight (kN)	56.83275	61.04345	56.48630
Interaction ratio	0.9977	0.9928	0.9959
Δ_{drift} (cm)	1.06	0.99	1.07

The HGAPSO results in the best answer again, while the GA and PSO are placed second and third in obtaining an acceptable solution.

The convergence rate of the algorithms in Fig. 12 shows the efficiency of the HGAPSO compared with the two others as well.

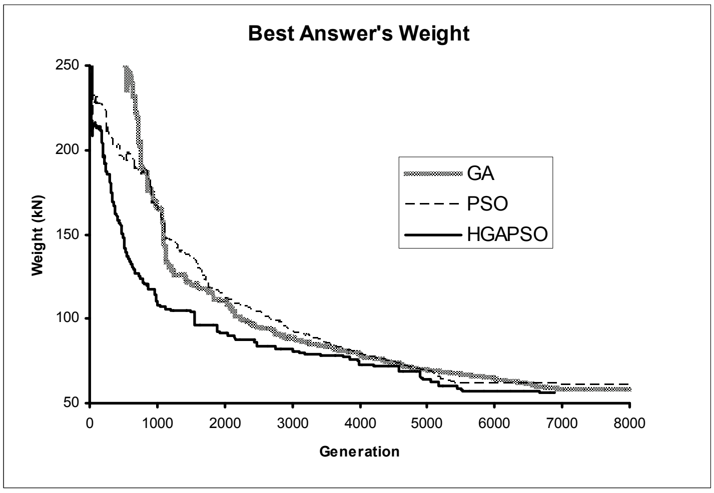

Fig. 12. Convergence of the three algorithms for the three-bay three-storey frame

It took the GA 13563 generations to converge, while the PSO and the HGAPSO converged in 8253 and 6876 generations, respectively.

For this frame, the last generation for GA, PSO and HGAPSO were 13487, 8234 and 6841, leading to average weights of 57.65kN, 63.65kN and 56.95kN, respectively, Fig. 13.

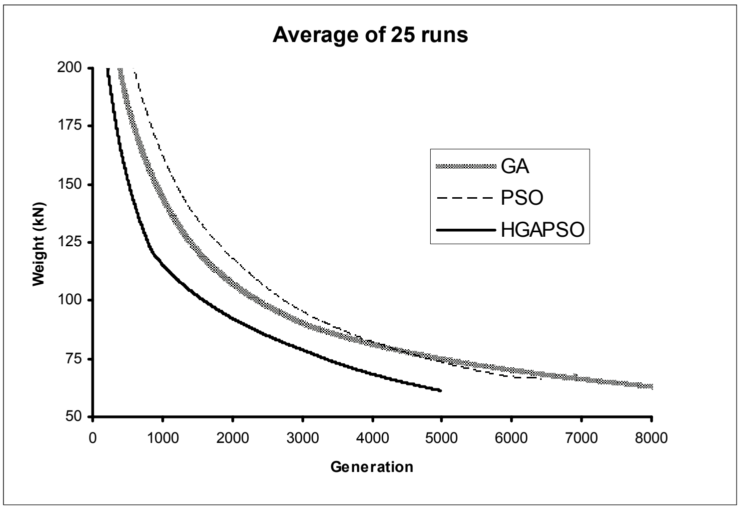

Fig. 13. Best weight versus the number of generations for the three-bay three-storey frame

6. CONCLUDING REMARKS

In this paper, different structures are optimized based on the SAND formulation by a hybrid algorithm of GA and PSO. HGAPSO introduces the concept of the maturing phenomenon in nature into the evolution of individuals originally modelled by GA. The maturing phenomenon is mimicked by PSO, where individuals enhance themselves based on social interactions and their private cognition. From the perspective of PSO, HGAPSO introduces crossover operation into the society. Thus, the evolution of individuals is no longer restricted to be in the same generation.

In order to demonstrate the performance of HGAPSO, SAND of different structures is carried out by this algorithm as well as simple GA and PSO. The results show the superiority of the HGAPSO, especially in larger problems with higher degrees of statical indeterminacy.

Acknowledgement- The first author is grateful to the Iran National Science Foundation for the support.

REFERENCES

1. Schmit, L. A. & Fox, R. L. (1965). An integrated approach to structural synthesis and analysis. *AIAA J.*, Vol. 3, pp. 1104–1112.
2. Arora, J. S. & Wang, Q. (1992). Review of formulations for structural and mechanical system optimization. *Struct. Multidisc. Optim.*, Vol. 30, pp. 251-272.
3. Orozco, C. E. & Ghattas, O. N. (1992). Sparse approach to simultaneous analysis and design of geometrically nonlinear structures. *AIAA J.*, Vol. 30, pp. 1877-1885.
4. Orozco, C. E. & Ghattas, O. N. (1997). A reduced SAND method for optimal design of nonlinear structures. *Int. J. Numer. Methods Eng.*, Vol. 40, pp. 2759-2774.
5. Wu, C. C. & Arora, J. S. (1987). Simultaneous analysis and design optimization of nonlinear response. *Engineer. Comput.*, Vol. 2, pp. 53-63.
6. Haftka, R. T. & Kamat, P. (1989). Simultaneous nonlinear structural analysis and design. *Comput. Mech.*, Vol. 4, pp. 409–416.
7. Ringertz, U. T. (1989). Optimization of structures with nonlinear response. *Eng. Optim.*, Vol. 14, pp. 179-188.
8. Kirsch, U. & Rozvany, G. I. N. (1994). Alternative formulations of structural optimization, *Struct. Multidisc. Optim.*, Vol. 7, pp. 32–41.

9. Larsson, T. & Ronnqvist, M. (1995). Simultaneous structural analysis and design based on augmented Lagrangian duality. *Struct. Multidisc. Optim.*, Vol. 9, pp. 1-11.
10. Kaveh, A. & Rahami, H. (2006). Analysis, design and optimization of structures using force method and genetic algorithm. *Int. J. Numer. Methods Eng.*, Vol. 45, pp. 1570-1584.
11. Kaveh, A. & Rahami, H. (2006). Nonlinear analysis and optimal design of structures via force method and genetic algorithm. *Comput. Struct.*, Vol. 84, pp. 770-778.
12. Kaveh, A. (2006). *Optimal structural analysis.* Wiley, 2nd edition, Chichister, UK.
13. Kaveh, A. & Rahami, H. (2004). Algebraic Graph theory for suboptimal cycle bases of graphs for an efficient force method, *Iranian J. Sci. Tech., Trans. B: Tech.*, Vol. 28, pp. 529-536.
14. AISC, (1989). *Allowable stress design and plastic design specifications for structural steel buildings*. 9th edition, American Institute of Steel Construction, Chicago, IL.
15. Holland, J. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI., The University of Michigan Press.
16. Hassan, R., Cohanim, B. & Weck, O. (2005). A comparison of particle swarm optimization and the genetic algorithm. *Proceedings of 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, Austin, Texas, pp. 18-21.
17. Kennedy, J. & Eberhart, R. C. (1995). Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks*, Piscataway, NJ, USA, pp. 1942-1948.
18. Eberhart, R. C., Simpson, P. K. & Dobbins, R. W. (1966). Computational intelligent PC tools. Boston, MA, Academic Press Professional.
19. Eberhart, R. C. & Shi, Y. (1998). Comparison between genetic algorithms and particle swarm optimization. In *Evolutionary Programming* VII, Porto, Springer, pp. 611–616.
20. Rechenberg, I. (1994). Evolution strategy, In *Computational Intelligence: Imitating Life*. IEEE Press, Piscataway, NJ.
21. Schwefel, H. P. (1995). *Evolution and optimum seeking*. Wiley, New York.
22. Juang, C. F. (2004). A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. Vol. 34, pp. 997-1006.
23. Perez, R.E. & Behdinan, K. (2007). Particle swarm approach for structural design optimization. *Comput. Struct.*, Vol. 85, pp. 1579–1588.
24. Galante, M. (1992). Structures optimization by a simple genetic algorithm. In: *Numerical Methods in Engineering and Applied Sciences*, Barcelona, Spain, pp. 862–870.
25. Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Wiley, New York.
26. Haftka, R. & Grdal, Z. (1992). *Elements of structural optimization*. 3rd edition, Dordrecht: Kluwer Academic Publishers.
27. El-Sayed, M. & Jang, T. (1994). Structural optimization using unconstrained non-linear goal programming algorithm. *Comput. Struct.,* Vol. 52. pp. 723–727.
28. Memari, A. & Fuladgar, A. (1994). Minimum weight design of trusses by Behsaz program. *2nd International Conference on Computational Structures Technology*, Athens, Greece.
29. Rizzi, P. (1976). Optimization of multi-constrained structures based on optimality criteria. *17th Structures, Structural Dynamics, and Materials Conference*, King of Prussia, PA.
30. Haug, E. & Arora, J. (1979). *Applied optimal design*. Wiley, New York.
31. Ghasemi, M., Hinton, E. & Wood, R. (1997). Optimization of trusses using genetic algorithms for discrete and continuous variables. *Eng. Comput.*, Vol. 16, pp. 272–301.
32. Gellatly, R. & Berke, L. (1971). Optimal structural design. Tech Rep AFFDLTR-70-165. Air Force Flight Dynamics Laboratory (AFFDL).

33. Schmit, L. & Miura, H. (1976). A new structural analysis synthesis capability: access 1. *AIAA J.*, Vol. 14, pp. 661–671.
34. Schmit, L. and Farshi, B. Some approximation concepts in structural synthesis. *AIAA J.*, Vol. 12, pp. 692–699(1974).
35. Dobbs, M. & Nelson, R. (1976). Application of optimality criteria to automated structural design. *AIAA J.*, Vol. 14, pp. 1436–1443.
36. Kaveh, A. & Shojaee, S. (2007). Optimal design of skeletal structures using ant colony optimization, *Int. J. Numer. Methods Eng.*, Vol. 70, pp. 563-581.
37. Kaveh, A. & Talatahari, S. (2007). A discrete particle swarm ant colony optimization for design of steel frames. *Asian J. Civil Eng.*, Vol. 9, pp. 563-575.
38. Khot, N. S., Venkayya, V. B. & Berke, L. (1976). Optimum structural design with stability constraints. *Int. J. Numer. Methods Eng.*, Vol. 10, pp. 1097–1114.
39. Camp, C. V., Pezeshk, S. & Cao, G. (1998). Optimized design of two dimensional structures using a genetic algorithm, *J. Struct. Eng., ASCE*, Vol. 124, pp. 551-559.
40. Kameshki, E. S. & Saka, M. P. (2001). Optimum design of nonlinear steel frames with semi-rigid connections using a genetic algorithm. *Comput. Struct.,* Vol. 79, pp. 1593-1604.