

Dynamic selective pressure using hybrid evolutionary and ant system strategies for structural optimization

A. Kaveh^{1,*},[†] and M. Shahrouzi²

¹*Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Narmak, Tehran 16, Iran*

²*International Institute of Earthquake Engineering and Seismology, 26 Arghavan, North Dibajee, Farmanieh, Tehran 19, Iran*

SUMMARY

Genetic algorithms have already been applied to various fields of engineering problems as a general optimization tool in charge of expensive sampling of the coded design space. In order to reduce such a computational cost in practice, application of evolutionary strategies is growing rapidly in the adaptive use of problem-specific information. This paper proposes a hybrid strategy to utilize a cooperative dynamic memory of more competitive solutions combining indirect information share in ant systems with direct constructive genetic search. Some proper coding techniques are employed to enable testing the method with various sets of control parameters. As a challenging field of interest, its application to structural layout optimization is considered while an example of a traveling salesman problem is also treated as a combinatorial benchmark. Copyright © 2007 John Wiley & Sons, Ltd.

Received 14 June 2006; Revised 23 February 2007; Accepted 2 April 2007

KEY WORDS: evolutionary strategies; genetic algorithm; ant colony metaphor; direct index coding; variable mutation band; structural topology and sizing optimization

1. INTRODUCTION

Evolutionary search methods in numerical fields are basically considered as attempts to simulate the following Darwinian principles [1–3]:

1. Living creatures in nature include various species in considerably large groups called the *population*.
2. Their search to *limited* resources leads them to *compete* with each other and *fitter* ones are more likely to *survive*.

*Correspondence to: A. Kaveh, Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Narmak, Tehran-16, Iran.

[†]E-mail: alikaveh@iust.ac.ir

3. Every current population has a level of *diversity* in species probably due to random and environment-affected *mutation* in some *descendent* species.

In this regard, artificial simulation utilizes evolutionary features: *population-based* search, *fitness-based selection* and *reproduction*. Two common fields of such search methods are classified as, but not limited to, *evolutionary strategies* (ESs) [1] and *genetic algorithms* (GAs) [3]. Reproduction in GAs is more unified and consists of *crossover* as an exploitative and *mutation* as an explorative operator [4, 5]. Since in GAs information transfer between phenotypic and genotypic spaces takes place only during fitness evaluation, it is a more general optimization tool, while in ESs utilizing some *adaptive techniques* has made its search more efficient in finding local optima [6–8]. Superiority of exploration in GA, is related to its population-based strategy combined with non-biased exchanging between the search region islands by means of proper encoding and the employed genetic operators [9–11].

Because of many simplifying assumptions and techniques with respect to the real natural evolution process, such numerical simulations are very sensitive to fine tuning their control parameters to provide a proper balance between exploration and exploitation. A number of investigators have indicated that population size and mutation rate are the parameters that most affect algorithm efficiency and effectiveness [12, 13].

In most of the problems, sensitivity to the mutation rate is even higher and a few works have addressed adaptive control of this parameter [14–18]. However, the authors utilized a proper integer coding that enables definition and control of the *mutation band* in a self-adaptive manner [19–21].

Another way to maintain diversity of the population as a key point in an explorative search is to control population size and quality. It is considered in the ES framework to introduce the concept of *selective pressure* [22–24]. Only some of these children in (λ, μ) strategy or some of the whole parents and children in $(\lambda + \mu)$ strategy are transferred to the next generation as new parent individuals, and the corresponding ratio is limited to the proposed selective pressure [25, 26]. It is difficult to tune the penalty coefficients in a single run to provide a proper balance between several constraints in the fitness function. Using only the fitness ranks for selection criteria may increase the risk of losing some ‘good’ but not necessarily fittest individuals and may cause premature convergence during further generations. In this regard, employing the elitist strategy may affect the effectiveness of the genetic search that is best suited to the unconstrained form [27]. Hence, there is a need to improve the selection method with additional criteria to preserve good solutions during the genetic search. A number of investigators have already proposed an interesting method called SASEGASA to prevent premature convergence using various crossover types in parallel to utilize feature finding *via* segregated subpopulations and then sharing them by a delayed reunification. They also offered temporarily population expansion with self-adopted selective pressure steering based on the simulated annealing approach to maintain the desired diversity in GAs [28].

The present work is another attempt to this aim, combining the explorative search capability of GAs with the distributed share of *a posteriori* information in ant colony approaches [29–31]. The method is based on the alternate appearance of more competitive individuals in the elite portion of the population [32] utilizing a dynamic colony of them [10, 33, 34]. The challenging/rewarding field of structural optimal design includes several local optima in addition to a number of design variables and several types of constraints, making the size of feasible space narrow compared to the whole design space [9]. Hence, such an interesting design field is selected here to verify the proposed method by a number of illustrative benchmark examples.

After this introduction, the elements of Ant Colony Optimization (ACO)-based search and also the behavior of elite individual competition in the genetic search are discussed in Section 2 in order to be used in Section 3 to sample the best genetic material for the additional part of the mating pool. An algorithm for dynamic variation of selective pressure is proposed to transfer such sampled genotypes to the next generations for improving effectiveness and efficiency of the search. The algorithm is further evolved by introducing the dynamic mutation band control for structural layout optimization in Section 4. A number of truss and frame examples are treated to test the proposed method of dynamic selective pressure in Section 5, and a comparative study is also made. Concluding remarks are in Section 6.

2. DYNAMIC SAMPLING OF THE BEST GENERATIVE INDIVIDUALS USING ANT STRATEGIES

Proper evolutionary search is governed by interaction between exploration and exploitation agents to simulate natural competition of individuals. In the coded genotypic space, crossover as an exploitative agent tries to search permutations possible for chromosomes with some common genes. On exchanging these genes *via crossover*, only those with different alleles will produce new permutations. In other words, the crossover operator together with *fitness-based selection* sweeps all possible permutations between any set of chromosomes with some common schemata. Every such set may be considered a local search region/island and the efficiency of such local search is governed by direct information share in crossover. An exploration agent such as a *mutation* operator will then be needed to introduce new search regions to such a process to preserve access to the region of global optimum. As new islands are added to and old ones gradually are omitted from the current population as the *search window*, fitter representatives of them will compete with the current elite ones alternatively. Such a process is called *alternate appearance of competitive individuals* [10, 33].

Despite GAs, ant strategies use an indirect type of information share between individual solution trials. Each ant completes its trial solution *state* by state in a stochastic manner with two resources of guidelines: *pheromone trail* and *attractiveness* [29, 30]. Attractiveness is a local guide for exchanging between states and can be taken as analogous to fitness in GA notation. But the indirect information share *via* evaporating trail is an extra feature to such evolutionary search.

The present work uses such an ant system strategy to sample more competitive individuals in the population. The width of such a portion is dominated by the number of sampling ants, N' , while its depth is affected by the pheromone deposit packet, $\Delta\tau$. In the sampling process offered by the authors [33, 34], at every generation of the evolutionary search, N' ants sample N' individuals of the current population that are fitter than the others. These sampled solution states are then copied to the colony of more frequent successive individuals. Subsequently, their pheromone trail is updated in a discrete manner, and as soon as the trail of any individual reaches zero it is omitted from the colony. In this way, a dynamic short-term memory of sampled individuals during genetic search is provided based on indirect share of genetic data.

3. DYNAMIC SELECTIVE PRESSURE USING THE COLONY OF MORE COMPETITIVE SOLUTIONS

As mentioned before, the fitness-based selection tends to gradually fill the population with fitter individuals. In addition, the population size in traditional GAs is taken as fixed, in which most

of the parent individuals are replaced with their children due to a high rate of crossover that is used to maintain speed of convergence. Consequently, it may cause losing some good parents that are necessary for access to new search islands, thus weakening the diversity of the population or even causing premature convergence. A simple and common tool to reduce such an effect is the *tournament selection* method, which allows choosing a parent out of even two less fit individuals. It is somehow covering the concept of *fitness sharing* in an implicit way [4, 35]. An interesting approach has already been reported based on feature extraction during subpopulations and recombining them to delay the loss of good genetic information or diversity of population using simulated selective pressure variation [28].

Another solution is offered in the present work using the dynamic colony of more frequent competitive individuals. In this strategy, a colony of more frequent and competitive solutions during previous generations is added to the current population in the mating pool to decrease the chance of losing good genetic material. The size of such a colony is adaptively altered during the search, while the population size is constant. Such a proposed method is hereinafter called *dynamic selective pressure*, DSP. The whole procedure is given by the following algorithm:

1. Initiate a population of individuals for the first generation. For a small population size, some initial individuals may be designed as feasible in addition to the other randomly generated ones.
2. After reproduction, decoding and fitness evaluation, use N' ants to try the same number of more attractive individuals and select feasible ones among them.
3. Transfer any of these individuals to a database of more attractive solutions as a design colony *via* the following steps:
 - a. Check whether the same individual is already in the design colony; if so store its location number in it, if not add the colony size, M , by unity and consider the last location to add the newcomer.
 - b. For any added chromosome, the corresponding k th ant deposits a fixed amount of pheromone packet, $\Delta\tau$; hence the trail will be

$$\tau_k = \tau_k + \Delta\tau \quad (1)$$

4. Update the trail of individuals in the colony by discrete pheromone evaporation as
- 5.

$$\tau_i = \tau_i - 1, \quad i = 1, \dots, M \quad (2)$$

6. Eliminate any member whose trail value has reached $\tau_k = 0$ from the camp and consequently decrease the colony size.
7. Add the remaining members of the colony to the mating pool provided they are not in the tabu list of previously added members.
8. Select N_P individuals as parents of the next generation out of such an updated mating pool. N_P is the initial population size.
9. Continue the GA operations with the new population and repeat steps (2)–(7) until the stopping criteria are satisfied.

According to this algorithm, the number of descendents, μ , in the selection stage out of mating pool, is thus taken as N_P while the generators' number in such $\lambda + \mu$ strategy is $N_P + N_C$. Because

the colony size, N_C , in this method is dynamically tuned in an adaptive way, the corresponding selective pressure will be dynamic, measured by the inverse of the following factor:

$$s = \frac{\mu}{\lambda + \mu} \quad (3)$$

The factor s is a normalized number between 0 and 1 and a lower s indicates higher selective pressure.

The proposed method allows using smaller population size than usual in which the exploration due to the mutation operator is expected to be more effective. In this way, the frequency of alternate appearance of more attractive individuals during generations can be employed as a secondary selection criterion in addition to the fitness ranks.

Releasing the tabu list check in the recent algorithm is possible; however, it leads to more severe selective pressure variance and may affect convergence rate, especially for small population sizes.

4. ADAPTIVE TUNING MUTATION OPERATOR FOR DISCRETE STRUCTURAL SIZE AND TOPOLOGY OPTIMIZATION

In many structural problems such as skeletal frameworks, selecting design variables is limited to discrete variants among a practical list of standard/predetermined sections. For this purpose, the authors developed a special type of integer coding that deals with section indices as the character alphabet assignable to genes [19, 20]. For the case of topology optimization, a zero index is added to such a list to indicate elimination of the corresponding member group in the structural model [20]. Another interesting feature of such a *direct index coding* (DIC), is revealing the definition of *mutation band* and then controlling it in an adaptive manner [21]. It is defined as the index range assignable to each gene in every generation of the search, which can be at least set to 2 and at most set to the maximum available indices in the section list plus one for topological elimination. A *dynamic variable band mutation* (DVBM), is then utilized in the following manner [21]:

1. For the first generation, initiate the mutation band with its maximum available for every gene.
2. For any gene of a chromosome in every new generation, if the delay in fitness improvement of the elitist individual is less than its maximum experienced during previous generations then decrease the mutation band by unity; otherwise increase it up to the maximum number of variants assignable to that gene. In structural layout optimization, perform such an increase/decrease of the mutation band with the center of the index currently assigned to that gene in the section list to allow the possibility of preserving improved genetic material among previous generations.

5. ILLUSTRATIVE EXAMPLES

A number of truss and frame examples are treated here to test the proposed method of DSP. Tournament selection is also employed to avoid the rapid loss of diversity in the population. According to the applied elitist strategy, the least fit individual is replaced with the fittest feasible

one of every previous generation. Such a strategy is activated after the first elitist individual becomes feasible. In each run, the set of initial chromosomes is saved so that comparison with other methods becomes possible starting from the same initial population.

The structural optimization problem is formulated as

$$\text{Minimize } w = \rho \cdot \sum_{i=1}^M A_i \cdot L_i \quad (4)$$

$$\text{Subject to } \mathbf{KU} = \mathbf{P} \quad (5)$$

$$g_j^d = \left| \frac{\Delta^j}{\Delta^{j,a}} \right| - 1 \leq 0 \quad (6)$$

$$g_i^s = \left| \frac{\sigma_i}{\sigma^a} \right| - 1 \leq 0 \quad (7)$$

$$\text{for } i = 1, \dots, ne, \quad j = 1, \dots, n\text{DOF}$$

where \mathbf{K} , \mathbf{U} and \mathbf{P} are the stiffness matrix, nodal displacement and force vectors, respectively. ρ is the material density, and L_i and A_i denote the length and cross-sectional area of the i th member. Δ^j and σ_i indicate the nodal displacement and axial member stress response while the number of elements and degrees of freedom in the problem are denoted by ne and $n\text{DOF}$, respectively.

Subsequently, the fitness function is considered as

$$F = F_0 - w * (1 + K_p \cdot V) \quad (8)$$

$$V = \sum_{n\text{LC}} (\max(g^d, 0) + \max(g^s, 0)) \quad (9)$$

K_p is the penalty constant taken as 10 for this class of treated examples [36] and V denotes the total constraints' violation regarding all $n\text{LC}$ loading conditions. F_0 is taken as 0 in the case of the tournament selection scheme. The static equilibrium constraint (5) is automatically satisfied during the structural analysis to derive stress and displacement responses.

5.1. Layout optimization for the 6-node planar truss

As a benchmark example, the well-known 6-node truss of Figure 1 is treated here [20, 37–43]. Material properties are taken as $E = 68.95 \text{ GN/m}^2$, $\rho = 27.1 \text{ kN/m}^3$, allowable axial stress is $\sigma^a = \pm 172 \text{ MN/m}^2$ and nodal displacements are confined to $|\Delta^{a,y}| \leq 0.0508 \text{ m}$. In this example, optimal integrated topology and size is searched provided that each of P_1 and P_2 is applied as a separate loading condition [43]. Member cross sections are to be selected among the profile list S_1 of Table I.

Member groups are taken identical to member IDs, thus the minimal length chromosome has 10 genes, and the size of discrete search space is $(16 + 1)^{10}$, i.e. its minimum required cardinality.

The example is first treated with a random set of initial individuals once using a fixed selective pressure of 1 and the other time altering it by the present method. Control parameters are given in Table II including the number of ants and trail deposit packet selected for this example. In the first attempt, two-point crossover and a population of size 40 were employed. Both these first and

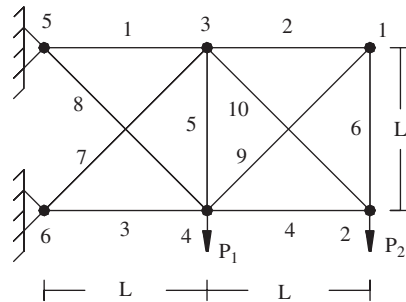


Figure 1. Boundary conditions, dimensions and loading of the 6-node truss example: $L = 9.14$ m, $P_1 = P_2 = 445.4$ kN.

Table I. Section index lists used for treated examples.

Profile index	Profile list		
	S_1	S_2	S_3
	Section area (10^{-4} m ²)	Section area (10^{-4} m ²)	Section names (AISC-ASD manual [44])
1	6.45	0.774	W6X25
2	19.35	1.255	W8X10
3	32.26	2.142	W8X13
4	51.61	3.348	W8X18
5	64.51	4.065	W8X21
6	67.74	4.632	W8X24
7	77.42	6.542	W8X28
8	96.77	7.742	W8X31
9	109.68	9.032	W10X15
10	141.94	10.839	W10X17
11	154.84	12.671	W10X19
12	167.74	14.581	W10X22
13	180.64	21.483	W10X26
14	187.10	34.839	W10X30
15	200.00	44.516	W12X14
16	225.81	52.903	W12X16
17	—	60.258	W12X19
18	—	65.226	W12X22
19	—	—	W12X26
20	—	—	W12X30
21	—	—	W14X22
22	—	—	W14X26
23	—	—	W14X30
24	—	—	W16X26
25	—	—	W16X31
26	—	—	W16X36
27	—	—	W24X55
28	—	—	W24X62
29	—	—	W24X68

Table II. Control parameters applied to Example 1.

Method	Parameter							
	Initial population	Population size	Crossover type	Crossover probability	Mutation type	Mutation probability	Number of ants	Pheromone deposit packet
Present Work 1	Set-1	40	Two-point	0.90	DVBM	0.10	—	—
Present Work 2	Set-1	40	Two-point	0.90	DVBM	0.10	5	5
Present Work 3	Set-2	30	Uniform (40%)	0.90	DVBM	0.10	—	—
Present Work 4	Set-2	30	Uniform (40%)	0.90	DVBM	0.10	5	5

Table III. Result of layout optimization as achieved set of indices for the 6-node truss of Example 1.

Member ID	1	2	3	4	5	6	7	8	9	10	Structure's weight (N)
Present Work 3 and Reference [43]	13	0	8	9	2	0	2	9	10	0	19266.8
Present Works 1, 2, 4 and Reference [20]	12	0	8	8	2	0	2	9	10	0	19266.5

Table IV. Resulting computational effort in Example 1 with a search space size of 17^{10} .

Method	Parameter	
	Number of fitness evaluations to achieve optimum	Number of generations to achieve optimum
Present Work 1	5688	190
Present Work 2	3266	109
Present Work 3	2875	72
Present Work 4	2872	72

second sample runs resulted in the global optimum mentioned in the literature [20] (Table III), but it took less computational effort for the proposed method of DSP as indicated in Table IV. For the second attempt, uniform crossover and a smaller population size were applied to a randomly generated set of individuals as initial population. Again the DSP resulted in the global optimum more efficiently, and even in lesser number of generations and also lesser computational efforts. But the genetic search with constant selective pressure factor; $s = 1$ led to premature convergence to a heavier structure as shown in Figure 2. Some topologically unstable trusses are detected by controlling the *degree of static indeterminacy* of truss and consequently are highly penalized without any structural analysis [45]. This may be considered the reason for the minor difference in computational effort of the third and fourth present works in Table IV.

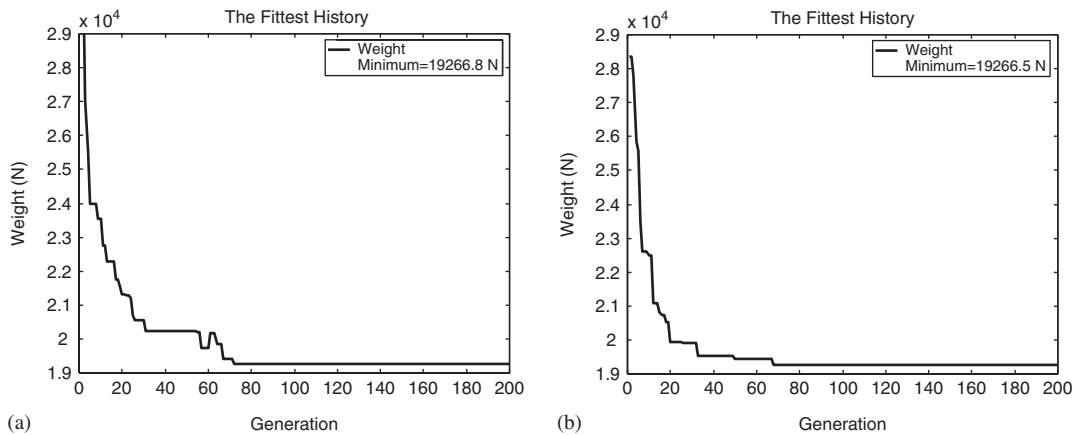


Figure 2. History of structural weight in: (a) Present Work 3 and (b) Present Work 4.

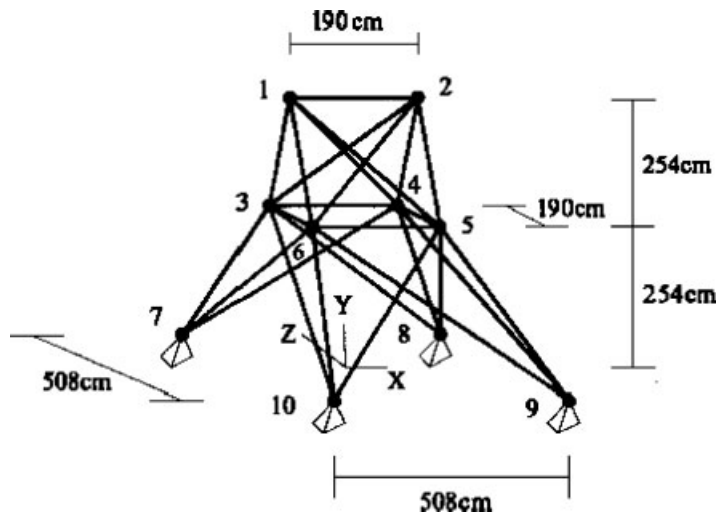


Figure 3. Boundary conditions and dimensions of the 10-node space truss example.

5.2. Layout optimization of the 10-node space truss

The capability of the proposed method with respect to structural layout optimization in three-dimensional problems for grouped members is tested in this example (Figure 3). The loading state is given in Table V. Material properties are taken as $E = 68.95 \text{ GN/m}^2$ and $\rho = 27.1 \text{ kN/m}^3$, while the constraint on nodal displacements is $|\Delta^a| \leq 0.0089 \text{ m}$. Table VI has listed allowable stresses for distinct member groups. The topological groups are taken identical to the eight profile groups (Table VII), allowing application of a direct indexed chromosome with the same number of genes. The character alphabet for every such gene is taken as 18 section indices in list S_2 of Table I and as one 0 index for the topological part of optimization.

Table V. Loading state applied to the 6-node truss of Example 2.

Loading condition	Node number	P_x (kN)	P_y (kN)	P_z (kN)
1	1	4.45	-22.25	44.5
1	2	0	-22.25	44.5
1	3	2.225	0	0
1	6	2.225	0	0
2	1	0	-22.25	89
2	2	0	-22.25	-89

Table VI. Allowable tension and compression stresses for profile groups of Example 2.

Profile group	1	2	3	4	5	6	7	8
Allowable tension stress (MPa)	275.90	275.90	275.90	275.90	275.90	275.90	275.90	275.90
Allowable compression stress (MPa)	242.04	79.94	119.36	242.04	242.04	46.62	46.62	76.64

Table VII. Member list of the protomorph used for Example 2 and corresponding member groups.

Member IDs	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Node 1	1	1	2	1	2	2	2	1	1	3	4	3	5	3	6	4	5	3	4	6	5	3	4	5	6
Node 2	2	4	3	5	6	5	4	3	6	6	5	4	6	10	7	9	8	8	7	9	10	7	8	9	10
Profile group	1	2	2	2	2	3	3	3	3	4	4	5	5	6	6	6	6	7	7	7	7	8	8	8	8
Topology group	1	2	2	2	2	3	3	3	3	4	4	5	5	6	6	6	6	7	7	7	7	8	8	8	8

Table VIII. Control parameters applied to Example 2.

Method	Parameter							
	Initial population	Population size	Crossover type	Crossover probability	Mutation type	Mutation probability	Number of ants	Pheromone deposit packet
Present Work 5	Set-3	30	Uniform (40%)	0.90	DVBM	0.10	—	—
Present Work 6	Set-3	30	Uniform (40%)	0.90	DVBM	0.10	3	10

Control parameters for sample runs with the same random set of individuals in the initial population are listed in Table VIII. In the first run, DVBM is added to traditional genetic search with a fixed size of mating pool and population, while selective pressure is dynamically altered in the second run using three ants and a trail packet of 10. Member grouping of this example results in more severe topological instabilities, which might delay convergence. The history of achieved weight is depicted in Figure 4 after the first reproduction.

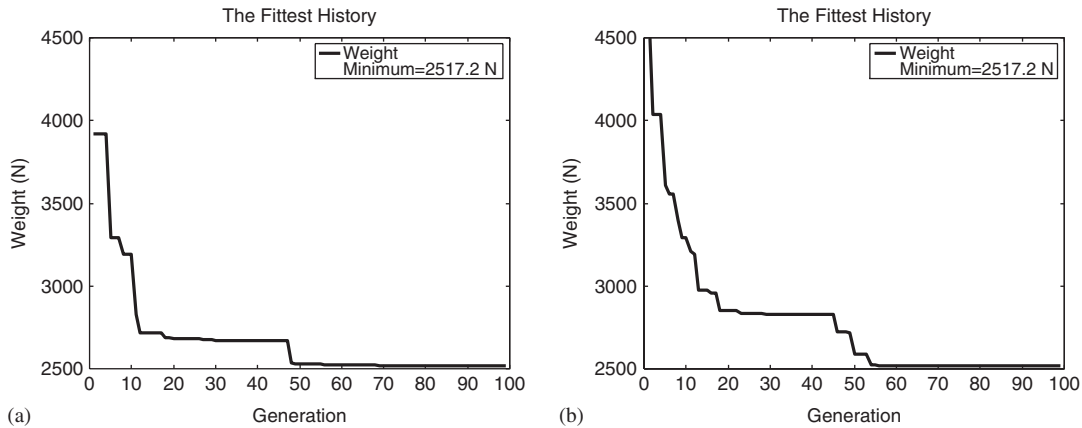


Figure 4. History of structural weight in: (a) Present Work 5 and (b) Present Work 6.

Table IX. Optimal layout of member groups for the 10-node truss in Example 2 [43].

Member group	1	2	3	4	5	6	7	8	Structure's weight (N)
Profile index in list S_2	0	10	13	0	0	7	11	12	2517

Table X. Resulting computational effort in Example 2 with a search space size of 19^8 .

Method	Parameter	
	Number of fitness evaluations to achieve optimum	Number of generations to achieve optimum
Present Work 5	2054	69
Present Work 6	1663	56

Table IX demonstrates the layout of the global optimum compared to those previously achieved in the literature [43, 46]. It was captured by DSP in less number of generations and computational effort than the canonical GA even with mutation band control (Table X).

5.3. Example 3: Optimal sizing design for a two-bay six-story planar frame

The two-bay six-story frame of this example has been previously treated by Huang and Arora [47] under floor loads of 14.59 kN/m (dead + live loads) and lateral wind loads of 40 kN at each story.

The applied loading conditions are:

1. Dead + live
2. 0.75 (dead + live + wind)

Material properties are taken as $E = 207 \text{ GN/m}^2$ and $\rho = 76.8 \text{ kN/m}^3$ while the yield stress is $F_y = 248.2 \text{ MN/m}^2$.

The genotype search space of this size optimization problem will have 29^{18} variants when direct index encoding is employed.

Here, the simple stress constraint of relation (7) is replaced with

$$g_i^s = \left| \frac{f_a}{F_a} + \frac{f_b}{F_b} \right| - 1 \leq 0, \quad \frac{f_a}{F_a} \leq 0.15 \quad (10)$$

otherwise consider the maximum of

$$g_i^s = \left| \frac{f_a}{0.6F_y} + \frac{f_b}{F_b} \right| - 1 \leq 0 \quad (11)$$

$$g_i^s = \left| \frac{f_a}{F_a} + \frac{C_m f_b}{(1 - f_a/F_e) \cdot F_b} \right| - 1 \leq 0 \quad (12)$$

for $i = 1, \dots, ne$

in which f_a, f_b are the axial and flexural member stresses. The corresponding allowable stresses F_a, F_b and other related coefficients in relations (10)–(12) are calculated based on the design code provisions [44]. In order to estimate slenderness buckling ratio k , the following relation is employed [48]:

$$k = \max \left(0.9, \left(1 - \frac{G_m}{20} \right) \right) \times \sqrt{1 + G_m} \quad (13)$$

$$G_m = \min \left(\left(\frac{\sum EI_c/L_c}{\sum EI_g/L_g} \right) \Big|_{n_1}, \left(\frac{\sum EI_c/L_c}{\sum EI_g/L_g} \right) \Big|_{n_2} \right) \quad (14)$$

where n_1, n_2 denote the end nodes of each frame member and I_c, I_g are moments of inertia for each intersecting column and girder to that joint.

In this example, 29 sections from list S_3 in Table I are available for 18 symmetric member groups, resulting in a search space of minimal required cardinality using DIC, that is 29^{18} . The axis of symmetry is taken as the middle column in the frame (Figure 5).

Control parameters and results for a sample set are shown in Tables XI–XIV. In many structural problems the ratio of feasible solutions over infeasible ones is considerably high [9]. It is more crucial in this example because of its higher cardinality of the search space with respect to the previous ones. Therefore, in this attempt half the set of initial individuals is generated randomly, while for the other half the corresponding allele for each gene is set to its maximum assignable (index of heavier section) in order to include a sufficient number of feasible individuals in the initial population.

Regarding structural weight response in Table XIII and Figure 6, it is realized that the proposed method of DSP, i.e. Present Work 7, not only has more convergence rate but also shows better

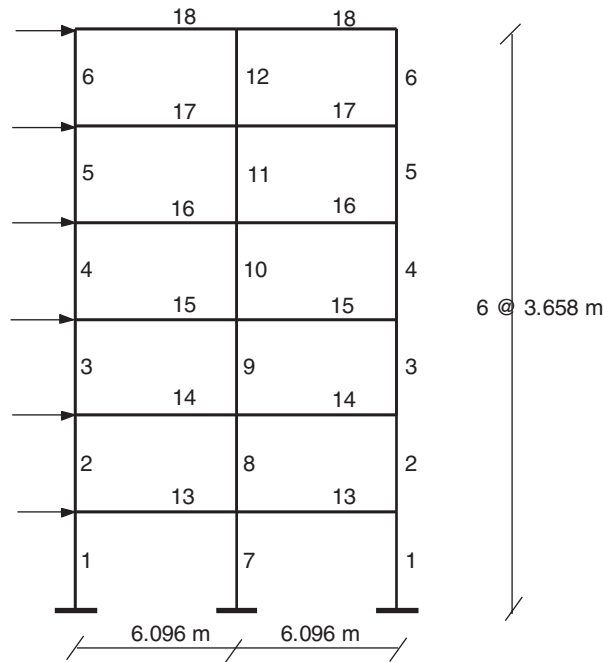


Figure 5. Member groups for the two-bay six-story frame of Example 3 and exertion points of the lateral loading.

Table XI. Control parameters for Example 3.

Method	Parameter							Number of ants	Pheromone deposit packet
	Initial population	Population size	Crossover type	Crossover probability	Mutation type	Mutation probability			
Present Work 7	Set-4	30	Two-point	0.90	DVBM	0.10	—	—	
Present Work 8	Set-4	30	Two-point	0.90	DVBM	0.10	3	10	

Table XII. Achieved layouts after 300 generations of sizing with GA for Example 3.

Method	Member group																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Present Work 7	28	27	24	24	17	9	26	18	26	10	21	15	23	26	20	24	10	16
Present Work 8	29	23	20	13	17	15	3	29	25	12	17	15	26	20	25	22	21	15

Table XIII. Selected results for achieved layouts in Example 3.

Method	Objective		
	Structural weight (N)	Maximum nodal displacement (m)	Maximum stress ratio
Present Work 7	32 886	0.0500	0.994
Present Work 8	32 587	0.0507	0.994

Table XIV. Resulting computational effort in Example 3 with a search space size of 29^{18} .

Method	Parameter	
	Number of fitness evaluations to achieve optimum	Number of generations to achieve optimum
Present Work 7	6780	226
Present Work 8	3270	109

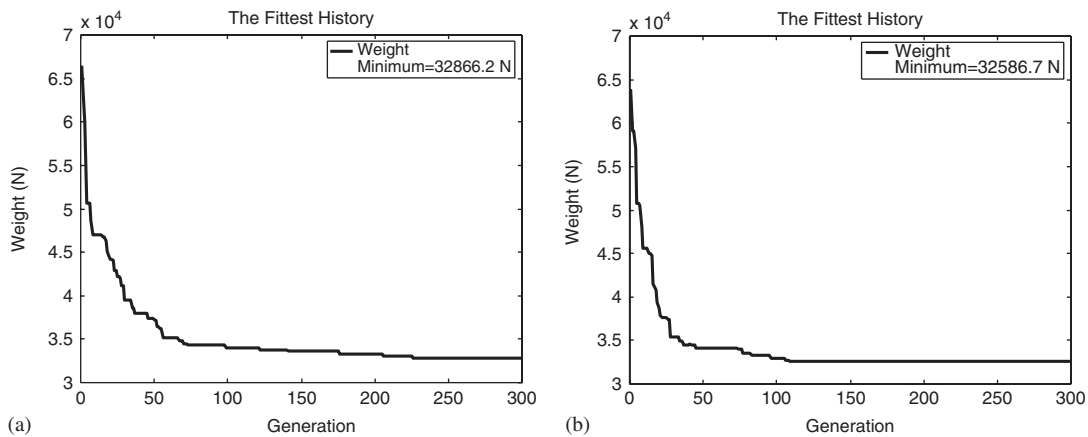


Figure 6. History of structural weight in: (a) Present Work 7 and (b) Present Work 8.

results within the same number of generations with respect to Present Work 8. It shows that more efficiency and effectiveness are expected by DSP than traditional genetic search with constant selective pressure.

5.4. Example 4: Symmetrical TSP for the 76-city problem

Application of the proposed method for hard combinatorial problems is studied here using an example of the *Traveling Salesman Problem* (TSP) (Figure 7) (Table XV). In this regard, the symmetric 76-city example is selected from the literature [49, 50]. The TSP is defined as looking

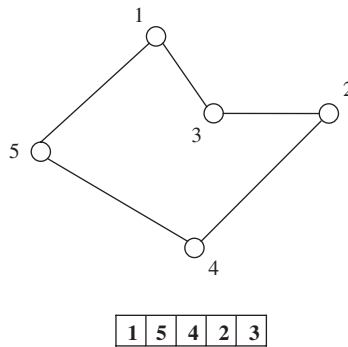


Figure 7. Representation of a tour in a sample five-city TSP by the corresponding chromosome.

Table XV. Control parameters for TSP runs in Example 4.

Method	Parameter						
	Initial population	Population size	Crossover probability	Mutation probability	Number of ants	Pheromone deposit packet	Selective pressure
Present Works 9–11	Set-5	76	0.90	0.20	—	—	Constant
Present Works 12–14	Set-5	76	0.90	0.20	3	5	Dynamic
Present Works 15	Set-5	76	0.90	0.20	3	5	Dynamic*

*Promoted algorithm for TSP.

(a) Parents 1 5 4 2 3 1 2 4 3 5
 (b) Children 5 4 2 3 1 1 3 2 4 5

Figure 8. Sample chromosomes: (a) before and (b) after the employed crossover for the TSP.

(a) 1 3 2 4 5
 (b) 1 5 2 4 3

Figure 9. Sample chromosome: (a) before and (b) after the employed mutation for the TSP.

for the minimal length cycle subgraph of a clique graph that covers all of its nodes with no repeated cities visited in each tour. The tour length is calculated based on Euclidian distances associated with the edges between city pairs. Every tour alternative is thus represented by a chromosome string of non-repeated city numbers as its alleles. Ordinary GA operators may cause repeated nodes in the child chromosome that does not form a cycle; therefore, special crossover and mutation operators such as two-opt are employed for this problem (Figures 8 and 9).

Table XVI. Results of TSP runs in Example 4 achieved in 3000 generations.

Method	Parameter	
	Minimum achieved tour length	Corresponding generation number
Present Work 9	648	2093
Present Work 10	723	2853
Present Work 11	696	2290
Present Work 12	617	1933
Present Work 13	679	1135
Present Work 14	693	2412
Present Work 15	653	2977

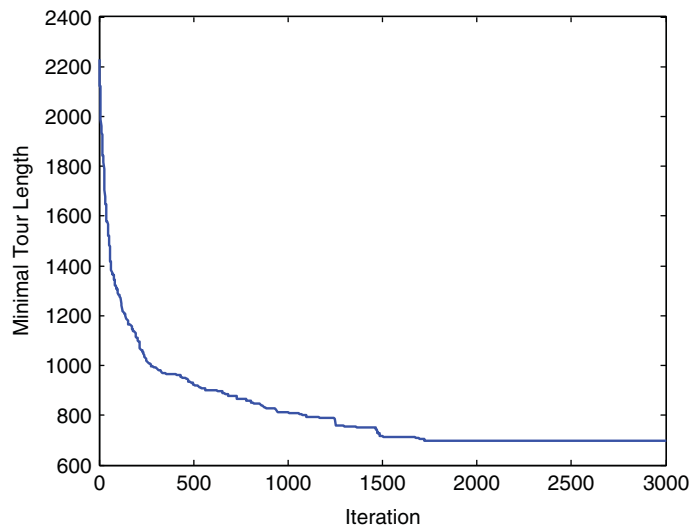


Figure 10. History of standard GA for a 76-city TSP example in Present Work 11.

The problem is solved with the same initial set of individuals by each method for a number of runs to reduce stochastic effect in conclusion. As shown in Table XVI, most of the times the proposed method has converged faster with a better average compared with those achieved by the standard GA.

In the last trail, a stochastic promotion is offered to insure that ‘good but not the fittest’ individuals are given a chance to enter the design colony and consequently avoid their early loss in the mating pool. Due to such a procedure, every j th individual of the design colony is replaced with a randomly chosen one from the whole population with the probability as in the state transition rule of the ACO strategies.

As can be realized from Figure 10, standard GA for TSP may lead to premature convergence. On the other hand, Figure 11 shows that the recent modification in the developed method of search

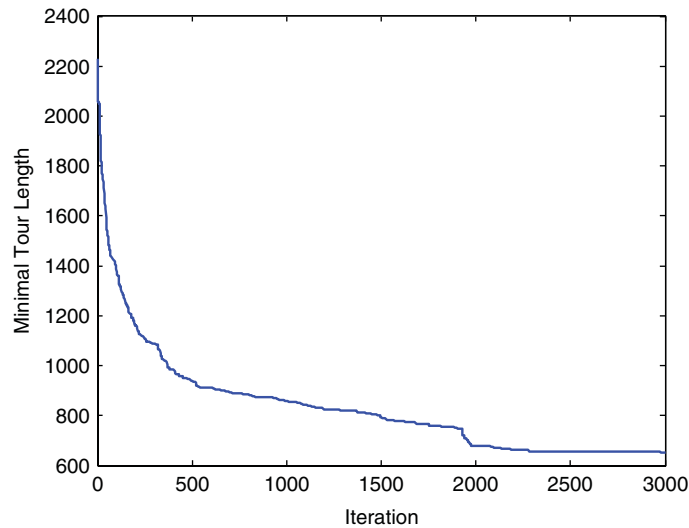


Figure 11. History of the proposed search method with dynamic selective pressure for a 76-city TSP example in Present Work 15.

can enable it to escape from the local optima trap and avoid premature convergence. The effect is somehow like a delayed partial niching in the population.

Discussion: The best performance of the proposed method belongs to problems in which most of the individuals with similar higher fitness are close points in the search space. This is true in the structural problems considered as the subject of this paper. It may also be noted that for complicated TSPs as a class of NP-hard problems, more improvement techniques may be employed such as the following ones.

Extending the capture zone of the ants to the whole population (by a small probability threshold) is an attempt to avoid early loss of *good but not the fittest* individuals during generations of the search. However, according to our trials a better way is to combine a partial niching as follows with the proposed colony of more frequent individuals to avoid the whole population being crowded with representatives of limited specific genetic islands.

1. In each generation, let the ants capture the fitter feasible competitive individuals out of the population and enter them to the corresponding colony.
2. For every individual in such a colony, if a random number falls below a pre-determined probability threshold, add its complement to the colony. The threshold is called *PartialNichingRate* and is preferably taken to be no more than the *MutationRate*.

For the TSP, every allele in the complement of a chromosome is obtained by subtracting the corresponding allele (city number) of the main chromosome from $n + 1$ (n is the number of cities).

3. Sort individuals of the new colony based on their evaluated fitness.
4. Add the predetermined portion of the achieved colony to the mating pool and continue the genetic search.

In case such a pre-determined portion is taken as 100%, then Step 3 is suppressed.

The size of the proposed colony dynamically alters so that the above subroutine is then regarded as a type of partial niching that will not disturb the whole population. In this way, when the fitter individuals tend to limit the search only to their own islands, the colony size will consequently increase and partial niching will be activated to add the corresponding complement chromosomes to the mating pool. According to our trials (even with TSPs more than 200 cities), it is expected that the high genotypic distance between such individuals and their complements helps in avoiding undesired loss of population diversity without excessive disturbance in the current genetic memory and even allows the use of less mutation rate for a more refined search toward the global optimum. However, a full study of such effects in NP-hard problems such as TSP goes beyond the scope of the present paper and requires considerably more hardware and computational effort/time.

6. CONCLUDING REMARKS

In order to increase the efficiency and effectiveness of the genetic search, the concept of selective pressure from evolutionary strategies is studied in this article. The proposed method empowers the genetic search not only in the width of fixed population size but also in the depth of previous generations by utilizing a dynamic colony of more attractive experienced designs.

For tuning the mutation operator, a special class of integer coding is employed. This is the most suitable representation for discrete search problems such as integrated size and topology optimization of structural skeletal frameworks, empowering GA to tune the mutation band in an adaptive manner.

The method is further developed, combining the direct share of genetic material during crossover with indirect information share using ant strategies. In this way, a dynamic colony of more attractive solution alternatives is constructed during the genetic search and added to the mating pool to improve its quality/diversity. The corresponding selective pressure is then altered in accordance with the proposed algorithm using the automatically extracted problem-specific information.

As a result, such a short-term memory of more attractive designs during the search preserves to keep good genetic material. It is further tested with examples of two- and three-dimensional truss layout optimization and also a frame sizing example with more complicated constraints/search space. In all these examples, the results obtained showed outstanding efficiency of the proposed method over the canonical genetic search with constant selective pressure. The result of the second attempt in the first example also demonstrates the capability of DSP in overcoming premature convergence in spite of the simple GA even with mutation band control. It is related to the proposed feature for indirect information share of more attractive solutions from previous generations to the current one. The present method also allows using a more limited search window than usual population size to capture the global optimum, thus leading to more efficient/effective search with respect to the constant selective pressure GAs.

REFERENCES

1. Rechenberg I. *Evolutionstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*. Frommann-Holzboog: Stuttgart, 1973.
2. Hu J, Goodman ED. Hierarchical fair competition model for parallel evolutionary algorithms. *Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002, IEEE World Congress on Computational Intelligence*, Honolulu, HI, 2002.
3. Fraser AS. Simulation of genetic systems by automatic digital computers. I. Introduction. *Australian Journal of Biological Sciences* 1857; **10**:484–491.

4. Holland HJ. *Adaptation in Natural and Artificial Systems, an Introductory Analysis with Application to Biology, Control and Artificial Intelligence*. The University of Michigan Press: Ann Arbor, MI, 1975.
5. Goldberg DE, Samtani MP. Engineering optimization via genetic algorithm. *Proceedings of the 9th Conference on Electronic Computation*, University of Alabama, Birmingham. ASCE: New York, NY, 1986; 471–484.
6. Whitley D, Rana S, Heckendorn RB. Representation issues in neighborhood search and evolutionary algorithms. In *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science*, Quagliarella D, Periaux J, Poloni C, Winter G (eds). Wiley: New York, 1998; 39–57.
7. Coello Coello CA. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**:1245–1287.
8. Kauffman SA. *The Origins of Order: Self-organization and Selection in Evolution*. Oxford University Press: New York, 1993.
9. Hiroyasu T, Miki M, Shimosaka H, Tanimura Y. Structural optimization by real-coded probabilistic model-building GA. *Proceedings of the 2002 IEEE International Conference on Systems, Man and Cybernetics (SMC 2002)*. *Academic Journal* 2002.
10. Kaveh A, Shahrouzi M. A hybrid approach for genetic search in structural optimization with minimal initial population. In press.
11. Rothlauf F. *Representations for Genetic and Evolutionary Algorithms*. Studies in Fuzziness and Soft Computing, vol. 104. Springer: Heidelberg, 1st edition 2002, 2nd printing, 2003.
12. Camp CV, Pezeshk S, Cao G. Design of framed structures using a genetic algorithm. In *Advances in Structural Optimization*, Frangopol DM, Cheng FY (eds). ASCE: New York, 1997; 19–30.
13. Schmitt LM. Fundamental study: theory of genetic algorithms. *Theoretical Computer Science* 2001; **259**:1–61.
14. Yuan B, Gallagher M. A hybrid approach to parameter tuning in genetic algorithms. *Proceedings of the IEEE Congress in Evolutionary Computation (CEC'05)*, vol. 2, Napier University, Edinburgh, U.K., 2–5 September 2005; 1096–1103.
15. Petrovski A, Brownlee A, McCall J. Statistical optimisation and tuning of GA factors. *Proceedings of the IEEE Congress in Evolutionary Computation (CEC'05)*, vol. 1, Napier University, Edinburgh, U.K., 2–5 September 2005; 758–764.
16. Lis J, Lis M. Self-adapting parallel genetic algorithm with the dynamic mutation probability, crossover rate and population size. In *Proceedings of the 1st Polish National Conference on Evolutionary Computation*, Oficina Wydawnicza Politechniki, Warszawskiej, Arabas J (ed.), 1996; 324–329.
17. Thierens D. Adaptive mutation rate control schemes in genetic algorithms. *Proceedings of the 2002 IEEE World Congress on Computational Intelligence: Congress on Evolutionary Computation*, Honolulu, HI. IEEE Press: New York, 2002; 980–985.
18. Thierens D. Scalability problems of simple genetic algorithms. *Evolutionary Computation* 2000; **7**(3):331–352.
19. Kaveh A, Shahrouzi M. Direct index coding for discrete size optimization of structures by genetic algorithms. *International Journal of Civil Engineering* 2006; **3**(3–4):166–181.
20. Kaveh A, Shahrouzi M. Simultaneous topology and size optimization of structures by genetic algorithm using minimal length chromosome. *Engineering Computations* 2006; **6**(5):664–674.
21. Kaveh A, Shahrouzi M. Simulated annealing and adaptive dynamic variable band mutation for structural optimization by genetic algorithms. *Asian Journal of Civil Engineering* 2006; **7**(6):655–674.
22. Baeck T. Selective pressure in evolutionary algorithms: a characterization of selection mechanisms. *Proceedings of the First IEEE Conference on Evolutionary Computation*, Orlando, Florida, U.S.A., 27–29 June 1994; 57–62.
23. Affenzeller M. Transferring the concept of selective pressure from evolutionary strategies to genetic algorithms. *Proceedings of the 14th International Conference on Systems Science*, vol. 2, Wrocaw, Poland, 11–14 September 2001; 346–353.
24. Affenzeller M. A generic evolutionary computation approach based up on genetic algorithms and evolution strategies. *Journal of Systems Science* 2002; **28**(4):59–71.
25. Affenzeller M, Wagner S. SASEGASA: an evolutionary algorithm for retarding premature convergence by self-adaptive selection pressure steering. *IWANN 2003*, Lecture Notes in Computer Science. Springer: Berlin/Heidelberg, 2003.
26. Beyer H-G. Toward a theory of evolution strategies: the (μ, λ) -theory. *Evolutionary Computation* 1995; **2**(4): 381–407.
27. Beker T, Hadany L. Noise and elitism in evolutionary computation. In *Soft Computing Systems- Design, Management and Applications*, Abraham A *et al.* (eds). IOS Press: Amsterdam, 2002.

28. Affenzeller M, Wagner S. SASEGASA: a new generic parallel evolutionary algorithm for achieving highest quality results. *Journal of Heuristics, Special Issue on New Advances on Parallel Meta-Heuristics for Complex Problems* 2004; **10**(3):239–263.
29. Colomi A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies. In *Proceedings of the 1st European Conference on Artificial Life*, Paris, France, Varela F, Bourgine P (eds). Elsevier Publishing: Amsterdam, The Netherlands, 1991; 134–142.
30. Dorigo M, Gambardella LM. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation* 1997; **1**:53–66.
31. Maniezzo V, Carbonaro A. Ant colony optimization: an overview. In *Essays and Surveys in Metaheuristics*, Ribeiro C (ed.). Kluwer Academic Publishers: Dordrecht, 2001; 21–44.
32. De Jong KA. Genetic algorithms are not function optimizers. In *Foundations of Genetic Algorithms 2*, Whitley LD (ed.). Morgan Kaufmann: Los Altos, CA, 1993.
33. Kaveh A, Shahrouzi M. A hybrid ant strategy and genetic algorithm to tune the population size for efficient structural optimization. *Engineering Computations* 2007; **24**(3):237–254.
34. Kaveh A, Shahrouzi M. Extracting optimal structural design family in genetic search incorporating ant colony approach, submitted.
35. Goldberg DE, Richardson J. Genetic algorithms with sharing for multimodal function optimization. *Proceedings of the 2nd International Conference on Genetic Algorithms and their Application*, Lawrence Erlbaum Associates, 1987; 41–49.
36. Rajeev S, Krishnamoorthy CS. Discrete optimization of trusses using genetic structures. *Journal of Structural Engineering* (ASCE) 1992; **118**:1233–1350.
37. Cai J, Thierauf G. Discrete structural optimization using evolution strategies. *Technical Report*, Department of Civil Engineering, University of Essen, Essen, Germany, 1993.
38. Deb K, Gulati S. Design of truss-structures for minimum weight using genetic algorithms. *Finite Elements in Analysis and Design* 2001; **37**:447–465.
39. Coello Coello CA. Discrete optimization of trusses using genetic algorithms. *EXPERTSYS-94: Expert Systems Applications and Artificial Intelligence*, Gournaysur-Marne, France, IITT International, 1994; 331–336.
40. Galante M. Genetic algorithms as an approach to optimize real-world trusses. *International Journal for Numerical Methods in Engineering* 1996; **39**:361–382.
41. Ghasemi MR, Hinton E, Wood RD. Optimization of trusses using genetic algorithms for discrete and continuous variables. *Engineering Computations* 1999; **3**:272–301.
42. Turkkan N. Discrete optimization of structures using a floating-point genetic algorithm. *Annual Conference of the Canadian Society for Civil Engineering*, Moncton, NB, Canada, 2003.
43. Kaveh A, Kalatjari V. Topology optimization of trusses using genetic algorithm force method and graph theory. *International Journal for Numerical Methods in Engineering* 2003; **58**:771–791.
44. AISC. *Allowable Stress Design and Plastic Design Specifications for Structural Steel Buildings* (9th edn). American Institute of Steel Construction: Chicago, IL, 1989.
45. Kaveh A. *Structural Mechanics: Graph and Matrix Methods* (3rd edn). Research Studies Press (Wiley): Somerset, U.K., 2004.
46. Chai S, Shi LS, Sun HC. An application of relative difference quotient algorithm to topology optimization of truss structures with discrete variables. *Structural Optimization* 1999; **18**:48–55.
47. Huang M-W, Arora JS. Optimal design of steel structures using standard sections. *Structural Optimization* 1997; **14**:24–35.
48. Furlong RW. Column slenderness and charts for design. *AISC Journal, Proceedings* 1971; **68**(1):9–18.
49. Reinlet G. TSPLIB—A traveling salesman problem library. *ORSA Journal on Computing* 1991; **3**:376–384.
50. Reinlet G. TSPLIB. <http://www.iwr.uni-heidelberg.de/iwr/comopt/soft/TSPLIB95/TSPLIB.html>