Indian Journal of Engineering & Materials Sciences
Vol. 16, August 2009, pp. 245-258

# Bridge truss optimization under moving load using continuous and discrete design variables in optimization methods

Vedat Toğan* & Ayşe T Daloğlu
Department of Civil Engineering, Karadeniz Technical University, 61080 Trabzon, Turkey

*Received 17 January 2007; accepted 23 March 2009*

The aim of this work is to minimize the weight of steel bridge trusses under the limitations imposed on its behaviour by the design codes when it is subjected to moving load (HS20-truck). Adaptive manner operators are included to enhance the capability of GA. Design examples are solved to illustrate the applicability of proposed algorithm. Results are compared with those evaluated by various optimization methods with continuous design variables such as sequential convex programming (SCP), sequential quadratic programming (SQP) and evolution strategy (EVOL). It can be concluded that the results obtained by GA is meaningful, more suitable for practice and GA performs well to find minimum weight of the bridge trusses under moving load.

**Keywords:** Bridge truss; Optimization methods; Evolutionary computation; Genetic algorithm

The weight and the volume of a structural system can be minimized under certain limitations imposed on its behaviour by the design codes. Optimum design methods to be used in the structural engineering can be deterministic or stochastic. In the deterministic search methods such as mathematical programming, the objective function and stated constraints are used as a linear or nonlinear combination of the design variables, or a single cross-sectional property is used as the design variable and then all other properties are expressed as a function of that design variable[1]. On the other hand, a random component is usually introduced in the stochastic search methods such as genetic algorithms (GA) and evolution strategy (EVOL). Both GA and EVOL use of the solution principles and mechanisms of the biological evolution processes, in which numerous optimization mechanisms are embedded. In contrast to the GA, EVOL imitate the effects of genetic procedures on the phenotype.

In contrast to deterministic techniques that try to reach optimum step by step, stochastic techniques work on a population of potential solutions for a given problem. This population is generated randomly. And then genetic operations, especially crossover and mutation, are performed on old population to generate a new population. GA and EVOL get their power from the genetic operators.

*For correspondence (E-mail: togan@ktu.edu.tr)

Bridges can be designed more elegant and economical through the development of computer technologies and optimum design approaches. The self-weight of the structure increases quickly as its span expands. That is why it is very important to design the structures with a possible minimum self-weight satisfying certain design requirements. The aim of this study fulfills this target using GA with discrete design variables and compares the design obtained continuous design variables.

## Sequential Convex and Quadratic Programming

An efficient and accurate solution to any optimization problem depends not only on size of the problem in terms of the number of constraints and design variables but also on characteristics of the objective function and constraints. When an optimum design problem has both the linear objective function and constraints in the design variables, the problem is known as a linear programming (LP). Quadratic programming (QP) concerns with the minimization or maximization of a quadratic objective function that is linearly constrained. Nonlinear programming problem in which the objective function and constraints can be nonlinear functions of the design variables is more difficult to solve. A solution of the NP problem generally requires an iterative procedure to determine the search direction at each iteration[2,3]. If the problem is a so-called convex programming then the Kuhn Tucker (KT) equations are both necessary and sufficient for a global solution point[4,5]. The solution of the KT equations forms the basis for many nonlinear programming algorithms. These algorithms attempt to compute the Lagrange multipliers directly. Constrained quasi-Newton methods guarantee super linear convergence by accumulating second order information regarding the KT equations using a quasi-Newton updating procedure. These methods are commonly referred to as sequential quadratic programming (SQP) methods.

## Evolution Strategy

Rechenberg in 1973 developed first evolution strategy, so-called (1+1)-evolution strategy. The denotation (1+1)-evolution strategy means that one "offspring" individual is created from one "parent(s)" by a mutation. Parents and offspring individuals represent potential solutions to an optimization problem encoded as a set of numerical parameters. Either only the offspring or both parents plus offspring individuals compete for survival into the next generation. The mutation rate can be adjusted between successive generations to optimize the search progress. Mutation is a domain specific operator and can be implemented, for example, as a Gaussian distributed increment of a numerical parameter. The evolution strategy has some characteristics as follows[4,6]: (i) Reproduction in evolution strategy is not proportional to the fitness value; (ii) The original evolution strategy does not make a distinction between genotype (the bit strings which are manipulated by genetic operators) and phenotype (the decoded value, e.g., an integer value interpreted as a parameter of the objective function, i.e., the fitness) of an individual. In the original evolution strategy is both coincide; (iii) In evolution strategy, both parents and offspring may compete to survive into the next generation; and (iv) Mutation is the main force to drive an evolution strategy.

## Genetic Algorithms

Genetic algorithm can be divided into a problem-independent genetic operations part and problem-dependent function evaluation part. The coding scheme representing the variables of the optimization problems and fitness evaluation that reflects how good the solution is compared to the other solutions in the population serve as a link between these two parts[7]. Genetic algorithms do not require gradient information; they can be effective regardless of the nature of the objective functions and constraints. They combine the use of random numbers and information from previous iterations to evaluate and improve a population of points (a group of potential solutions) rather than a single point at a time[8]. The GA initiates a search to find the optimum in a discrete space by first selecting a number of individuals randomly to constitute the initial population. And then, Genetic operators such as selection, crossover and mutation are applied to produce a new generation.

The process that the genetic operators are applied on the population sequentially in order to obtain a better population than the previous one is known as the genetic loop. The loop is repeated until a satisfying solution to the problem is obtained ensuring certain design criteria.

Various types of genetic operators have been proposed by researchers to improve the performance of simple genetic algorithm (SGA) introduced by Goldberg[9]. The basic idea behind proposing various applications of the genetic operators is to prevent the population from getting stuck on a local optimum, and to maintain diversity. However, both the operators in SGA and improved versions of crossover and mutation operators are applied with pre-defined rates that are imposed on the algorithm by the user. Whereas, the choice of mutation and crossover probabilities, $p_m$ and $p_c$, critically effect the behaviour and performance of GA. There are various suggestions for $p_m$ and $p_c$ in the literature, but not an exact value is given[10]. In addition Pawlowsky concluded that there is no "best" unique operator setting in all circumstances[11].

The adaptive probabilities of crossover and mutation proposed by Srinivas and Patnaik[10] and improved by Toğan and Daloğlu[12] are used in this study to decide the probability of mutation and crossover according to the fitness value of the solutions and to relieve the user.

## Adaptive mutation and crossover

Recently, researchers have applied new GA implementations for better performance or to enhance GA's capabilities[10-14]. One of the new implementations is adaptive technique. The adaptive procedure proposed by Srinivas and Patnaik[10] and improved by Toğan and Daloğlu[12] is used in the study as follows:

The expressions for $p_m$ and $p_c$ take the forms of

$$p_c = k_1\,(f_{max} - f') \,/\, (f_{max} - f_{aveg}) \qquad f' \geq f_{aveg}$$

$p_c = k_3 \qquad f' < f_{aveg} \quad \ldots (1)$

$p_m = k_2\,(f_{max} - f) / (f_{max} - f_{aveg}) \qquad f \geq f_{aveg}$

$p_m = k_4 \qquad f < f_{aveg} \quad \ldots (2)$

Here, $f$ is the fitness of an individual, $f_{aveg}$ is the average fitness value of the population, $f_{max}$ is the maximum fitness value of the population, and $f'$ is the larger of the fitness value of the solutions to be crossed, Srinivas and Patnaik[10]. Coefficients $k_1$, $k_2$, $k_3$, and $k_4$ have to be less than or equal to 1.0.

When Srinivas and Patnaik[10] used a value of 0.5 for $k_2$ and $k_4$, they assigned a value of 1.0 for $k_1$ and $k_3$ to disrupt all the solutions with a fitness value less than $f_{aveg}$. However, the following is recommended for the case of $f < f_{aveg}$ in Eq. (2).

$p_m = (f_{aveg} - f) / (f_{aveg} - f_{min}) \qquad f < f_{aveg} \quad \ldots (3)$

where, $f_{min}$ is the minimum fitness value of the population. Using the above expression, solutions are disrupted depending on the fitness value, instead of destroying all the solutions with the fitness value less than $f_{aveg}$ with a constant probability of mutation. The probability of mutation increases as the fitness value tends to get closer to $f_{min}$, so a constant probability of mutation is not employed for all of the solutions having a fitness value of less than $f_{aveg}$. Thus, some infeasible individuals with good characteristics especially having exact or near average fitness value of the population will have a chance to survive.

Rates of mutation and crossover are calculated by multiplying $p_m$ and $p_c$ with string length for each solution in the population. These rates state the number of design variables to be disrupted for an individual. For adaptive mutation, design variables in the individual are arranged according to the level of violation. And then design variables are renewed with the mutation rate starting with the most violated one. This is useful because modification of bit is not being done randomly and therefore good solutions are being kept as they are. Thus, only the solution with fitness value of $f_{min}$ exposes mutation completely and diversity maintains.

For adaptive crossover, on the other hand, since $p_c$ changes from couple to couple, information exchange between the pairs can be done with flexible point crossover. That is, the crossover is performed with various crossover points changing from 1 to the string length of the individual. $f'$ represents the solution with the low fitness value in this study, on the contrary of the one described by Srinivas and Patnaik[10]. This is because if the lowest value of fitness is bigger than the average fitness value of population, the crossover will take place between the pairs having good fitness value. Whereas, when $f'$ represents the best fitted one, there is a possibility that the crossover to take place among the pairs with bad fitness value.

Adaptive manner proposed by Toğan and Daloğlu[12] is applied in the study as given below:

$p_m = 0.5\,(f_{max} - f) / (f_{max} - f_{aveg}) \qquad f \geq f_{aveg}$

$p_m = (f_{aveg} - f) / (f_{aveg} - f_{min}) \qquad f < f_{aveg} \quad \ldots (4)$

$p_c = (f_{max} - f') / (f_{max} - f_{aveg}) \qquad f' \geq f_{aveg}$

$p_c = 1.0 \qquad f' < f_{aveg} \quad \ldots (5)$

In contrast to mutation, rearranging of Eq. (5) is not necessary here because of the fact that the crossover is not in progress for renewing the bits between the couples. Hence, the formulation of the unconstrained optimization problem which is based on the violations of normalized constraints proposed by Rajeev and Krishnamoorthy[15] is given as:

$$min.\ W = \sum_{k=1}^{NG} A_k \sum_{i=1}^{m} \rho_i\, L_i \times \left(1 + K \sum_{r=1}^{M+N} C\right) \quad \ldots (6)$$

Subject to the following constraints:

$$\sigma_{il} \big/ (\sigma^{t}_{allowable} \text{ or } \sigma^{c}_{allowable}) - 1 \leq 0 \qquad i = 1,\ldots,M \quad \ldots (7)$$

and

$$d_{jl} \big/ d_{allowable} - 1 \leq 0 \qquad j = 1,\ldots,N \quad \ldots (8)$$

where $M$ is the total number of members in the structure, and $N$ is the number of restricted displacements. $\sigma_{il}$ is the stress in member $i$ for the load case $l$. $\sigma^{t}_{allowable}$ and $\sigma^{c}_{allowable}$ are the allowable stresses in tension and compression under load case $l$. $d_{jl}$ displacement of joint $j$, and $d_{allowable}$ is its upper bound under load case $l$. NG is the total number of groups in the structure; $A_k$ is the area of members belonging to group $k$; $m$ is the number of members in group $k$. $L_i$ and $\rho_i$ are the length and the density of member $i$. $K$ in Eq. (7) is known as a penalty coefficient. It is a constant to be selected depending on the problem. A value of 10 was found suitable for the optimization of truss systems[15] and it is taken 10 in the current work. $C$ is the violation coefficient of the constraints and it is computed in the following manner.

$$C = \begin{cases} \max\left(0, \sigma_{il} / (\sigma^{t}_{allowable} \text{ or } \sigma^{c}_{allowable}) - 1\right) & i = 1, \ldots, M \\ \max\left(0, d_{jl} / d_{allowable} - 1\right) & j = 1, \ldots, N \end{cases} \quad \ldots (9)$$

The value encoding is also adapted in the study.

## Application Examples

In any optimum design problem, some certain criteria must be established to evaluate a reasonable solution. For a structure, typical criteria may be (i) minimum cost; (ii) minimum weight; (iii) minimum construction time; (iv) minimum labor; (v) minimum cost of owner's products[16].

The self-weight of the structure increases quickly as its span expands. That is why it is very important to design the structures, i.e., the bridges that have long span, with a possible minimum self-weight satisfying certain design requirements. So, the criterion of minimum weight is generally used in optimum design. In this study, the moving load which is a series of concentrated truck-wheel loads given in AASTHO (Fig. 1) is considered. It is treated as a series of static loads moving on the bridge deck.

The first concentrated load in the series, 36 kN in Fig. 1, takes three positions between any two adjacent nodal joints of the bridge truss. Initially the first concentrated load in the series is placed on the first joint of the truss while the other two wheels are still of the bridge (Fig. 2a), and this is the first load case for the truss. Next the front wheel of the truck is placed at the center of the two adjacent joints. This situation requires the first load in the series to be placed on the bridge deck between the two nodal joints. The transference of this load from the deck to the truss needs to be calculated. If the front wheel is placed in the middle of the two adjacent joints, the transference of this load to the each nodal joint will be 18 kN as shown in Fig. 2b, and this is the second load case for the truss. As the truck continues to move on, the front wheel gets on the second nodal joint of the truss (Fig. 2c), and this will be the third load case for the bridge truss. While the truck moves on the bridge, the contribution of the second and the third wheel loads also needs to be included since they will be on the deck, and the transference of the wheel loads to the truss joints are calculated according to the distance from the joints as explained above and as illustrated in Figs 2 (d), (e) and (f), as it is indirect loading for the truss.

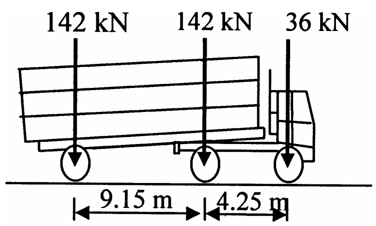


Fig. 1—Moving load (HS20)

Since vehicles travel directly on the superstructure, all of the parts are subjected to vibration and must be designed under impact load. AASHTO-1992 prescribes empirically that the impact factor expressed as a portion of live load[16].

$$I=50/(L+125) \leq 0.30 \quad \ldots (10)$$

where, $L$ is the span of the bridge expressed in feet and $I$ is the impact factor. The value of impact factor

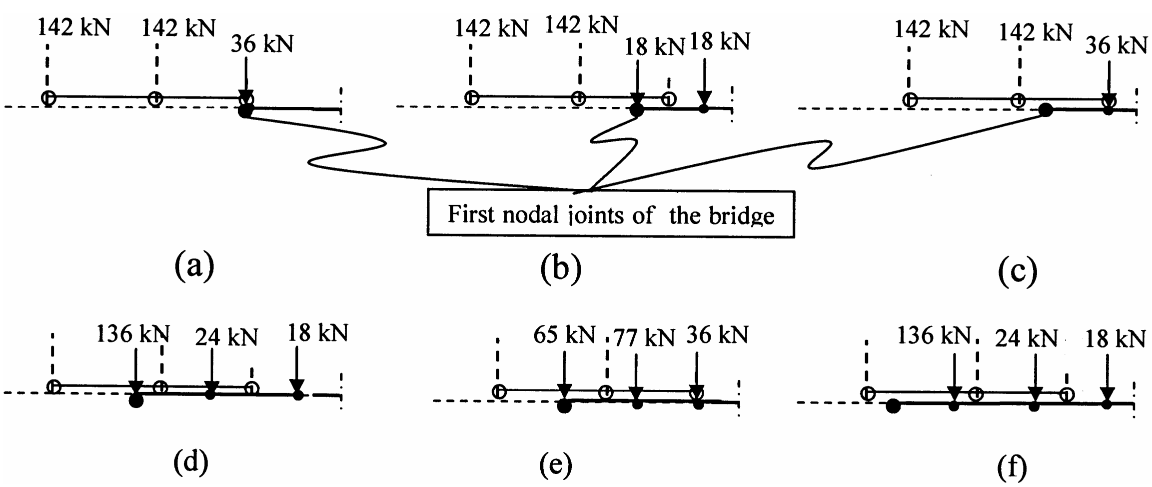


Fig. 2—HS20 Loading between any two adjacent Joints

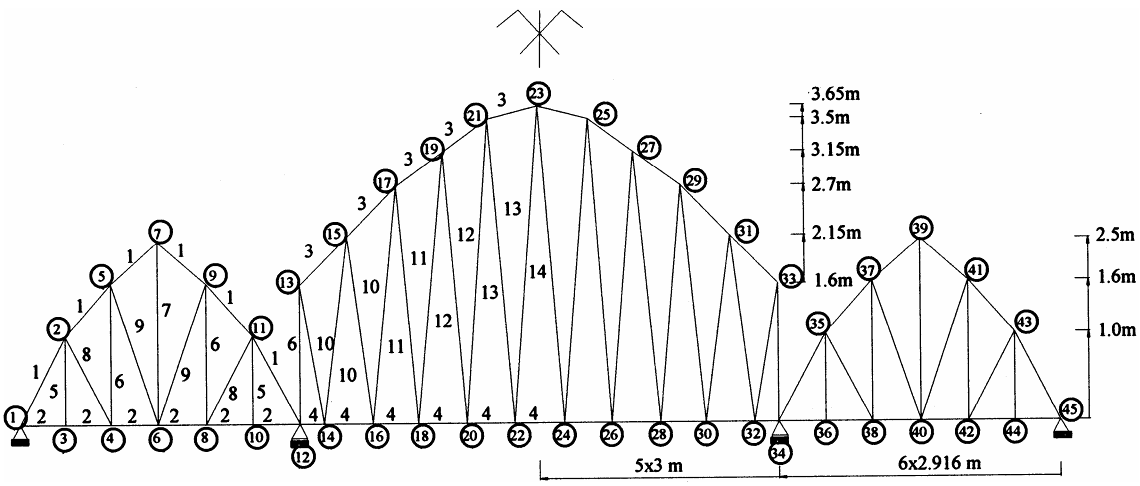


Fig. 3—Eighty-four-bar bridge truss

is increased by one and the load vector of the truss system is multiplied by it, and the system is analyzed.

In the design, the allowable displacement is limited to $L/800$ at the mid span of the bridge according to AASHTO. Allowable tensile stress, $\sigma^{t}_{allowable}$, and the young modulus, $E$, are taken as 212 $N/mm^2$ and 210 $kN/mm^2$ respectively. Allowable stresses for the compression members, $\sigma^{c}_{allowable}$, are calculated considering the slenderness ratio of the member according to the criteria given by AISC. The density, $\rho$, is taken as 7.85E-8 $kN/mm^3$. Double angles, angles, structural tees cut from S shapes and pipes given in AISC are adopted to form the structural members.

To design the bridge trusses with a minimum weight, the related optimization algorithms are run 20 times totally and the results presented in Tables and the best results among the all runs are given in the following sections. In addition to the optimization of bridge trusses with discrete design variables by GA, the same trusses are also optimized by SCP, SQP, and EVOL. The optimization methods tool of Institute for Computational Engineering, Faculty of Civil Engineering, Ruhr University Bochum for the SCP, SQP and EVOL are used with the default values for the purpose. The bridge trusses are optimized with different initial value of design variables and tables also show the initial values of design variables needed to get these results. For GA, a new initial population is created randomly for each try of 20 run, and the optimization process is repeated.

## Bridge Truss 1

Figure 3 shows a plane truss having 45 joints and 84 members. The areas of the members are taken as the design variables and they are collected in to 14 groups. The allowable displacement is 22 mm at joints 6 and 40 while it is 38 mm at joints 22 and 24. The truss is designed under 55 static load cases representing the moving load HS20 to cross the bridge once.

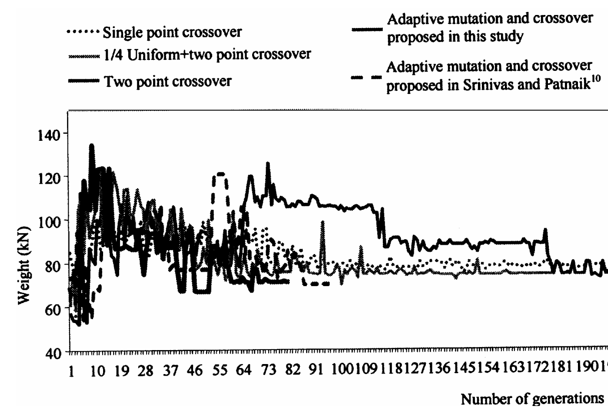


Fig. 4—Design history for bridge truss 1 with various operators

The optimal solution has a weight of 65.285 kN by GA. Figure 4 shows how the adaptive operators effect the solution. It also shows the exploring of the solution space and finding the optimum value of the weight with the number of generation for single point, two points, ¼ uniform + two point crossover, and adaptive mutation and crossover operators proposed in this study and in Srinivas and Patnaik[10]. For this case, the maximum number of generations is taken as 200 in the algorithm, and the process is terminated if the same individual constitutes 75% of the population. Also, a population size of 50 is taken in the optimization process. For this truss, the convergence is obtained at generation 79 when adaptive mutation and crossover are imposed in the algorithm. It is observed that the convergence is faster when the adaptive operators are used in the algorithm compare to those by single point, two points, ¼ uniform + two point crossover, and adaptive mutation and crossover operators proposed in Srinivas and Patnaik[10], and number of iteration to reach the optimum design is much smaller with the adaptive operators.

As known if any member in the structural system is in compression, the stress constraint turns out to be buckling constraint. In that case, the allowable stress is obtained by computing its critical stress. In contrast to allowable tensile stress, the critical stress is not constant but it is a function of the slenderness ratio of the compression member. So, it is related to radius of gyration of the cross-section. Since GA works with discrete design variable it is very simple to compute allowable compression stress of the member in each design step since all the geometric properties of the cross-sections are included in the algorithm as input data. But gradient based methods and EVOL use continues design variables, and it is not possible to give radius of gyration of member as an input data. One way to overcome this problem is to calculate the radius of gyration in terms of the area of the cross-section. According to Saka[17] this relationship has the form of

$$R = aA^b \qquad \ldots (11)$$

where $a$ and $b$ are the constants depending on the types of section adopted for the members and their values are given for some sections in Table 1. $A$ is cross-section area of the members and $r$ is the radius of gyration. The relationship given in Eq. (11) is used in the study to calculate the allowable compression stresses of the members to consider the buckling effect in the optimum design by SCP, SQP, and EVOL, whereas the real value of radius of gyration are used as they are given in the codes for the optimum design by GA.

Table 1—Values of the constants of Eq.(11) for various section

| | Angle | Pipe | Tee | Double Angle |
|---|---|---|---|---|
| $a$ | 0.8338 | 0.4993 | 0.2905 | 0.584 |
| $b$ | 0.5266 | 0.6777 | 0.8042 | 0.524 |

The results are summarized in Table 2 and Figs 5 (a), (b) and (c) present the variation of design variables, objective functions and constraints by SQP, SCP, and EVOL respectively.

Studying Figs 5 (a), (b), and (c) it can easily be said that the values of the objective function obtained by SCP and SQP increase as the number of iteration increases. Whereas the value of the objective function obtained by EVOL decreases as the number of iteration increases. Also, iteration numbers of 69, 63, for SQP and SCP are necessary to find optimum weight while 102 iterations are needed for EVOL. All constraints are satisfied and become zero when the process is terminated. The results summarized in Table 2 show that the minimum weight obtained by GA is higher than the weight obtained by the other

Table 2—Results of optimization processes for bridge truss 1

| Number of groups | SCP ($mm^2$) | SQP ($mm^2$) | EVOL ($mm^2$) | $SCP^+$ ($mm^2$) | $SQP^+$ ($mm^2$) | $EVOL^+$ ($mm^2$) | GA ($mm^2$) | Section | Initial values |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 5235 | 5309 | 5423 | 5625 | 5625 | 5625 | 5625 | Double angle | 900 |
| 2 | 4840 | 4831 | 4722 | 5625 | 5625 | 4838 | 4838 | " | 700 |
| 3 | 5587 | 5312 | 5682 | 6258 | 6258 | 6258 | 6258 | Tee cut from S | 700 |
| 4 | 3542 | 3810 | 3201 | 3870 | 4070 | 3322 | 2406 | " | 300 |
| 5 | 837 | 1138 | 905 | 954 | 1438 | 1096 | 954 | Pipe | 300 |
| 6 | 1167 | 1429 | 1509 | 1438 | 1451 | 1729 | 1729 | " | 300 |
| 7 | 1713 | 1350 | 1717 | 1948 | 1438 | 1948 | 1729 | " | 300 |
| 8 | 762 | 931 | 772 | 954 | 1096 | 954 | 2045 | " | 300 |
| 9 | 759 | 762 | 765 | 954 | 954 | 954 | 2045 | " | 300 |
| 10 | 2271 | 2354 | 2516 | 2374 | 2774 | 2774 | 2045 | " | 300 |
| 11 | 1640 | 1599 | 1517 | 1729 | 1729 | 1729 | 1729 | " | 300 |
| 12 | 871 | 1357 | 998 | 954 | 1438 | 1096 | 1729 | " | 300 |
| 13 | 936 | 1140 | 1005 | 1096 | 1438 | 1438 | 1729 | " | 300 |
| 14 | 1222 | 913 | 1221 | 1438 | 1096 | 1438 | 1719 | " | 300 |
| Weight (kN) | 60.16 | 61.31 | 60.48 | 67.14 | 68.94 | 64.98 | 65.28 | | |

$^+$ shows the results converted to the standard section

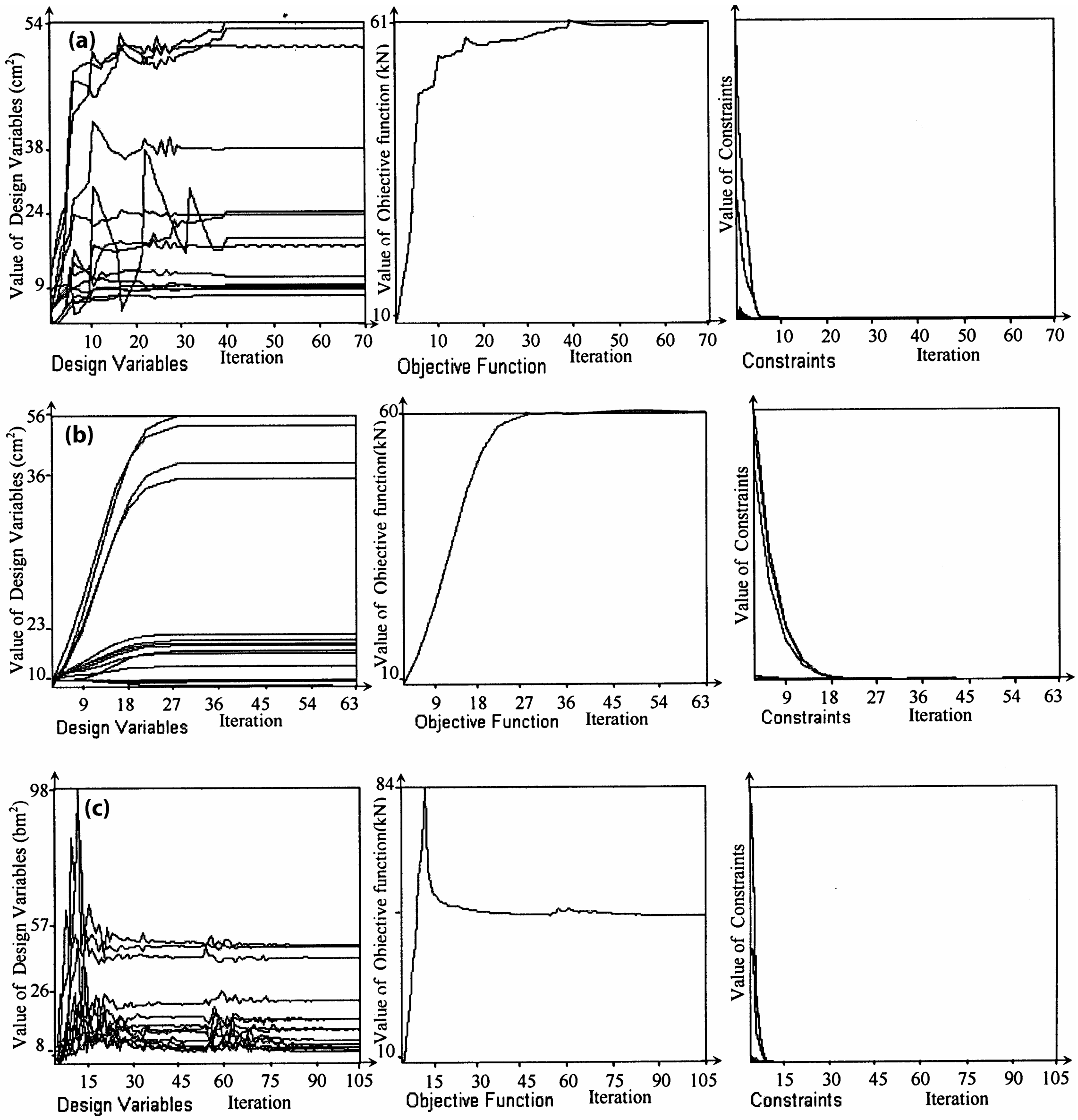


Fig. 5—Variation of the design variables, objective function, and constraints in SQP, SCP, and EVOL by iterations

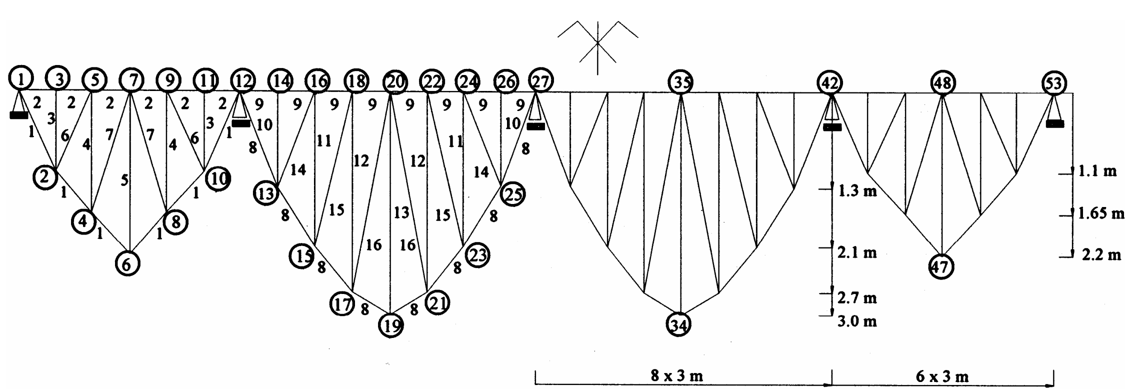


Fig. 6—Hundred-bar bridge truss

methods. This is because the continues design variables are used for SCP, SQP, and EVOL as mentioned above and it is impossible to include any standard section properties as they are available in practice. From the practical point of view design variables obtained by SCP, SQP, and EVOL are not real and this optimization process is not applicable. They must be converted to the standard sections given in the codes to make the results by SCP, SQP, and EVOL practical and feasible. Table 2 also presents the results converted to the standard sections for SCP, SQP, and EVOL in order to make a comparison with GA. It can be stated that the result obtained by GA is lighter than those produced by SCP and SQP while it slightly bigger than that obtained by EVOL. However, it is worthy to state that converting the results of SCP, SQP and EVOL to next higher available standard sections results with infeasible solution. In other words, the converted results for SCP, SQP, and EVOL do not satisfy the constraints, and a process is needed to update the result to satisfy the constraints.

## Bridge Truss 2

Another bridge truss shown in Fig. 6 is optimized as a second application. The bridge truss has 53 joints and 100 members. Displacements are limited to 23 mm at joints 7 and 48 and 30 mm at joints 20 and 35.

The design variables of the structure are collected into 16 groups. The truss is investigated under 67 static load cases representing HS20 crossing the bridge once. The size of the population is taken as 50; the optimum weight of bridge truss is determined as 83.67 kN. Figure 7 shows how the adaptive mutation and crossover affect the solution depending on the fitness value. It is seen in Fig. 7 that the effects of the probabilities of both the adaptive operators lose theirs effect as the fitness value of the solution increases. It is also seen that the probability of crossover is high while probability of mutation is low for some generations. This is because probability of mutation works on the fitness value of each solution while probability of crossover works on the partner having the lower fitness value. As the solution having the high fitness is disrupted with low rate of mutation, its partner is crossed with high rate of crossover if its partner has lower fitness than solution and near average fitness of the population. This will not be the case in the following generations since the majority of the individuals in the generation will be identical.

The design procedure initiates the optimization cycles for SCP, SQP and EVOL by deciding the initial values of the variables being the areas of cross sections first. They can be selected in any way as desired. They can be selected from a feasible domain without violating any of the constraints of the optimization problem, or from an infeasible domain that violate the constraints. However, it is observed that the selection of the initial value of area affects the design considerably. Especially in the case of the research domain has multiply strict local and global points. This situation is illustrated in Fig. 8. If any initial value is selected randomly it is most possible to reach a strict local or local optimum point in the search domain. Hence the design process should be repeated with different initial value in order to check whether the point reached is global optimum or not. It must be also repeated several times because of the fact that the GA initiates a search to find the optimum result in a discrete space by selecting a number of individuals randomly to constitute the initial population. Table 3 summarizes the best solution of all the runs.

## Bridge Truss 3

Another truss shown in Fig. 9 is considered as the final example. It has 99 joints and 193 members. 100 static load cases are applied on the truss representing the moving load of HS20 crossing the bridge. Cross-sections of the members are collected into 19 groups, and therefore there are 19 design variables since one

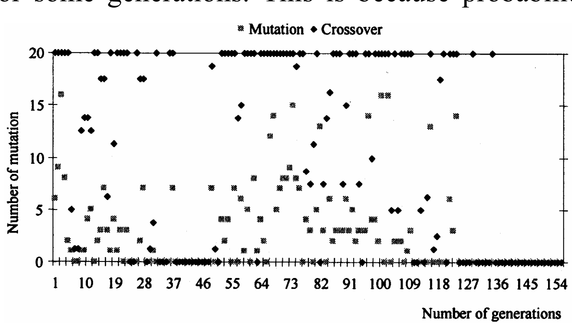


Fig. 7—Working of adaptive mutation and crossover

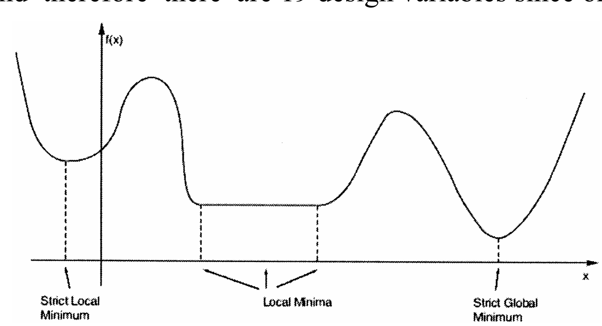


Fig. 8—Unconstrained local and global minima in one dimension

Table 3—Results of optimization processes for bridge truss 2

| Number of groups | SCP ($mm^2$) | SQP ($mm^2$) | EVOL ($mm^2$) | $SCP^+$ ($mm^2$) | $SQP^+$ ($mm^2$) | $EVOL^+$ ($mm^2$) | GA ($mm^2$) | Section | Initial values |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 5379 | 5831 | 5638 | 6258 | 6258 | 6258 | 5187 | Tee cut from S | 650 |
| 2 | 5140 | 4904 | 4911 | 6258 | 5187 | 5187 | 5187 | " | 650 |
| 3 | 559 | 749 | 619 | 605 | 877 | 703 | 1251 | Angle | 750 |
| 4 | 550 | 631 | 567 | 605 | 703 | 605 | 1251 | " | 750 |
| 5 | 1902 | 1319 | 1813 | 2096 | 1451 | 1954 | 1845 | " | 750 |
| 6 | 955 | 1265 | 970 | 1090 | 1348 | 1090 | 2354 | " | 750 |
| 7 | 1024 | 1636 | 1047 | 1116 | 1774 | 1116 | 1251 | " | 750 |
| 8 | 5465 | 5624 | 6034 | 5625 | 5948 | 6129 | 4838 | Double angle | 750 |
| 9 | 4287 | 4614 | 3789 | 4658 | 4838 | 3903 | 4658 | " | 750 |
| 10 | 564 | 749 | 596 | 605 | 877 | 703 | 1251 | Angle | 750 |
| 11 | 601 | 717 | 631 | 703 | 767 | 703 | 1845 | " | 750 |
| 12 | 712 | 712 | 773 | 767 | 767 | 877 | 1954 | " | 750 |
| 13 | 1595 | 1015 | 941 | 1774 | 1116 | 1090 | 1954 | " | 750 |
| 14 | 1237 | 1524 | 1290 | 1348 | 1600 | 1348 | 2339 | " | 750 |
| 15 | 1059 | 1413 | 1078 | 1116 | 1548 | 1116 | 1954 | " | 750 |
| 16 | 1206 | 1419 | 1212 | 1348 | 1548 | 1348 | 2329 | " | 750 |
| Weight (kN) | 77.05 | 81.38 | 77.41 | 85.73 | 86.46 | 81.60 | 83.67 | | |

$^+$ shows the results converted to the standard section

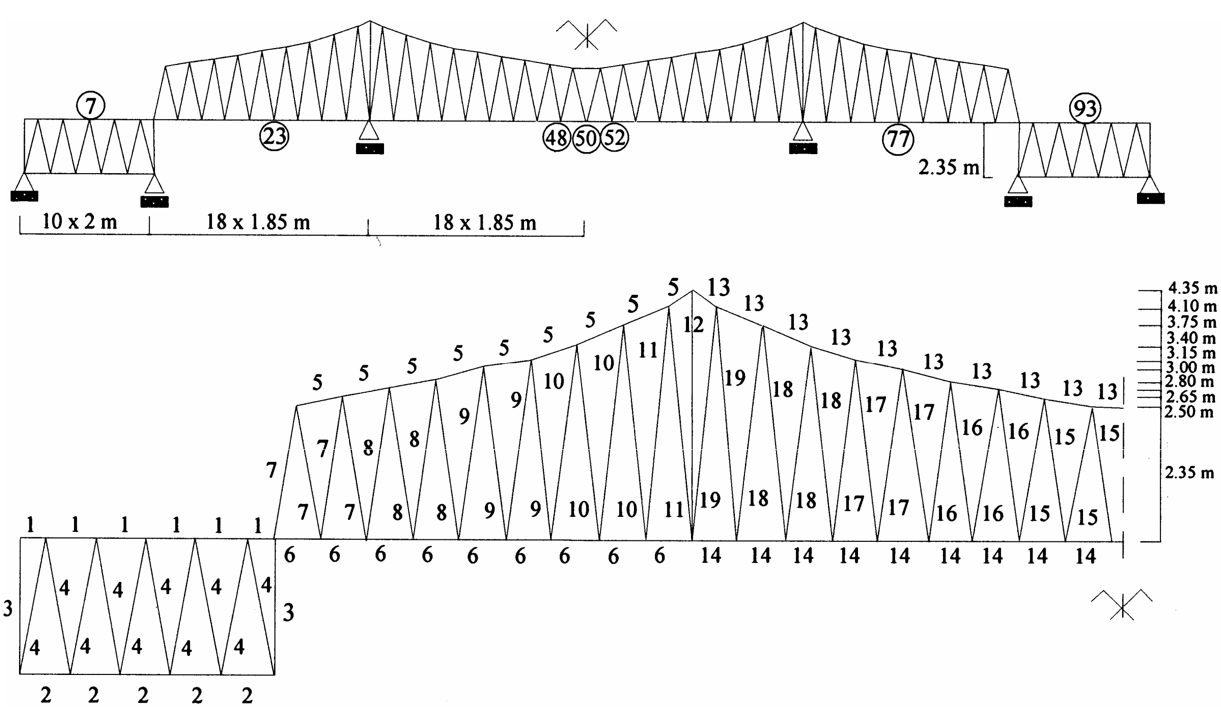


Fig. 9—Hundred and ninety-three-bar bridge truss

bit is required to represent each cross-section in the value encoding; the string length of coded part of the chromosome is 19 bits.

The allowable values for vertical deflections are 25 mm at nodes 7 and 93, 42 mm at nodes 23 and 77, and 83 mm at nodes 48, 50 and 52. The size of population is taken as 60 in the design process. The variation of the weight of the truss with the number of generation, and the maximum displacement at each generation are plotted in Fig. 10. Studying both Figs 10 and 11 that illustrate the variation of violation of displacements and stresses with the number of generations, it can be said that the displacement constraints are dominant in the design process because the weight of the truss is getting heavier while the mid-span displacement is getting closer to the allowable displacement. Minimum weight of the truss without violating any of the constraints is obtained as 465.08 kN. The program is terminated when 85% of the population consists of the same individual and the number of iteration is 118.

The same truss is designed one more time ignoring the displacement constraints to see how it affects the result. The minimum weight of the truss is obtained as 347.557 kN for this run under the stress and stability constraints only, and the convergence is obtained faster. This clearly shows that the displacement constrains are active and dominant over stress and stability constraints for this bridge truss as well as other bridge trusses designed in the study.

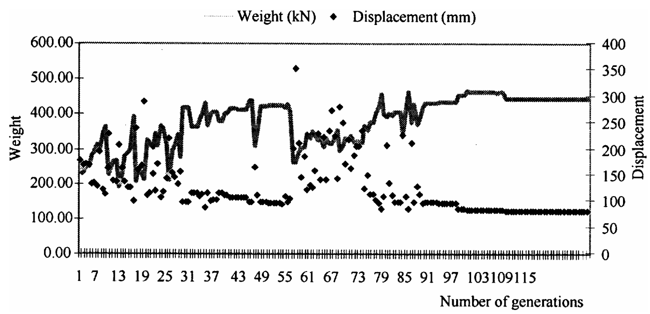


Fig. 10—Variation of the weight and maximum displacement with generations

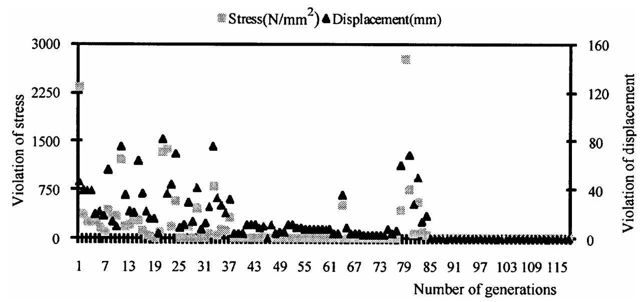


Fig. 11—Variation of the violation of stress and displacement with generations

Table 4—Results of optimization processes for bridge truss 3

| Number of groups | SCP ($mm^2$) | SQP ($mm^2$) | EVOL ($mm^2$) | SCP$^+$ ($mm^2$) | SQP$^+$ ($mm^2$) | EVOL$^+$ ($mm^2$) | GA ($mm^2$) | Section | Initial values |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 5789 | 4254 | 4304 | 5948 | 4658 | 4658 | 6129 | Double angle | 5000 |
| 2 | 5767 | 4328 | 4341 | 5948 | 4658 | 4658 | 3903 | " | 5000 |
| 3 | 2518 | 2550 | 2753 | 2696 | 2696 | 2812 | 3064 | Angle | 2000 |
| 4 | 2501 | 1766 | 1870 | 2696 | 1845 | 1954 | 2812 | " | 2000 |
| 5 | 14999 | 15000 | 14860 | 15161 | 15161 | 15161 | 14258 | Tee cut from W | 2000 |
| 6 | 14920 | 15000 | 14694 | 15161 | 15161 | 15161 | 14258 | " | 2000 |
| 7 | 4855 | 4249 | 5602 | 5419 | 5419 | 7677 | 5419 | Pipe | 1000 |
| 8 | 3997 | 3581 | 3954 | 5419 | 3941 | 5419 | 5419 | " | 1000 |
| 9 | 3575 | 2945 | 2983 | 3941 | 3599 | 3599 | 3941 | " | 1000 |
| 10 | 3377 | 1519 | 1745 | 3599 | 1729 | 1948 | 3599 | " | 1000 |
| 11 | 2959 | 1545 | 2329 | 3599 | 1729 | 2774 | 3599 | " | 1000 |
| 12 | 4549 | 7419 | 7137 | 5419 | 7677 | 7677 | 5419 | " | 1000 |
| 13 | 14999 | 15000 | 14999 | 15161 | 15161 | 15161 | 14258 | Tee cut from W | 2000 |
| 14 | 14999 | 15000 | 14976 | 15161 | 15161 | 15161 | 14258 | " | 2000 |
| 15 | 6009 | 6735 | 6523 | 7677 | 7677 | 7677 | 7677 | Pipe | 1000 |
| 16 | 7582 | 9236 | 8960 | 7677 | 9419 | 9419 | 7677 | " | 1000 |
| 17 | 7662 | 8094 | 8517 | 8258 | 8258 | 9419 | 7677 | " | 1000 |
| 18 | 7056 | 6238 | 6931 | 7677 | 7677 | 7677 | 7677 | " | 1000 |
| 19 | 5111 | 4692 | 3953 | 5419 | 5419 | 5419 | 5419 | " | 1000 |
| Weight (kN) | 470.51 | 453.01 | 457.06 | 490.06 | 471.77 | 484.42 | 465.08 | " | 1000 |

$^+$ shows the results converted to the standard section

The optimum results obtained by all four methods are presented in Table 4. It can be concluded that the optimum weight of bridge trusses obtained by GA is slightly bigger than those obtained by the other three methods with continuous design variables for all cases. However when the results of SCP, SQP and EVOL is converted the standard sections GA produces the lighter design (Table 4).

It can be possible to use the exact value of the radius of gyration in the design process by GA while this cannot be possible with the other methods in which the radius of gyration is expressed in terms of the cross-section areas in the design process. And Eq. (11) does not work well all the time. For example a member having 1754 $mm^2$ area of cross-section and consist of the double angle section is adopted in the

Table 5—Results of optimization processes for bridge truss 3 with new approximation

| Number of groups | SCP ($mm^2$) | SQP ($mm^2$) | EVOL ($mm^2$) | SCP$^+$ ($mm^2$) | SQP$^+$ ($mm^2$) | EVOL$^+$ ($mm^2$) | Initial values |
|---|---|---|---|---|---|---|---|
| 1 | 4074 | 5897 | 4150 | 4658 | 7419 | 4658 | 5000 |
| 2 | 3278 | 5891 | 3490 | 3690 | 7419 | 3690 | 5000 |
| 3 | 2205 | 2541 | 2931 | 2329 | 2696 | 3709 | 2000 |
| 4 | 3434 | 4687 | 3541 | 3709 | 5445 | 3709 | 2000 |
| 5 | 15071 | 15099 | 14926 | 15161 | 15161 | 15161 | 2000 |
| 6 | 15071 | 14427 | 14867 | 15161 | 15161 | 15161 | 2000 |
| 7 | 4365 | 4638 | 4468 | 5419 | 5419 | 5419 | 1000 |
| 8 | 3651 | 4213 | 4535 | 5419 | 5419 | 5419 | 1000 |
| 9 | 3270 | 3816 | 3532 | 3599 | 5419 | 3599 | 1000 |
| 10 | 2101 | 3038 | 2352 | 2774 | 3599 | 2774 | 1000 |
| 11 | 2466 | 2752 | 4004 | 2774 | 2774 | 5419 | 1000 |
| 12 | 7597 | 4794 | 3927 | 7677 | 5419 | 5419 | 1000 |
| 13 | 15071 | 15099 | 15015 | 15161 | 15161 | 15161 | 2000 |
| 14 | 15071 | 15099 | 15047 | 15161 | 15161 | 15161 | 2000 |
| 15 | 6962 | 5980 | 7886 | 7677 | 7677 | 9419 | 1000 |
| 16 | 7290 | 7492 | 8769 | 7677 | 7677 | 9419 | 1000 |
| 17 | 8317 | 7569 | 8101 | 9419 | 7677 | 9419 | 1000 |
| 18 | 6336 | 6788 | 6575 | 7677 | 7677 | 7677 | 1000 |
| 19 | 4792 | 4638 | 4196 | 5419 | 5419 | 5419 | 1000 |
| Weight (kN) | 460.67 | 478.62 | 466.22 | 483.69 | 513.47 | 493.84 | 1000 |

$^+$shows the results converted to the standard section

Table 6—Extracting profile list

| Section Types | | | | | | | |
|---|---|---|---|---|---|---|---|
| Double Angle | | Single Angle | | Tee | | Pipe | |
| Area ($mm^2$) | $r$ (mm) | Area ($mm^2$) | $r$ (mm) | Area ($mm^2$) | $r$ (mm) | Area ($mm^2$) | $r$ (mm) |
| 2722.58 | 23.19 | 1567.74 | 14.86 | 1896.77 | 18.64 | 690.32 | 15.37 |
| 3206.45 | 27.18 | 1600 | 17.45 | 2180.64 | 20.27 | 954.84 | 19.46 |
| 3690.32 | 31.24 | 1845.16 | 20.02 | 2406.45 | 24.23 | 1096.77 | 24.05 |
| 3903.22 | 39.88 | 1954.83 | 25.25 | 3019.35 | 25.4 | 1438.71 | 29.46 |
| 4658.06 | 39.62 | 2329.03 | 25.15 | 3870.96 | 26.92 | 1729.03 | 34.04 |
| 5625.8 | 47.75 | 2696.77 | 25.04 | 4070.96 | 27.18 | 2045.16 | 38.35 |
| 7419.34 | 47.24 | 3709.67 | 29.97 | 5187.09 | 28.96 | 2774.19 | 47.75 |
| 14774.16 | 62.74 | 4148.38 | 29.97 | 6258.05 | 30.23 | 3599.99 | 57.15 |
| 17096.74 | 62.23 | 4587.09 | 29.97 | 7548.37 | 34.04 | 5419.34 | 74.68 |
| 19354.8 | 61.98 | 5445.15 | 29.72 | 8193.53 | 34.54 | 7677.4 | 93.22 |
| 21612.86 | 61.47 | | | 11483.85 | 38.86 | 9419.34 | 111.25 |
| | | | | 14258.03 | 62.73 | | |
| | | | | 15161.26 | 63.5 | | |

$r$, exact radius of gyration

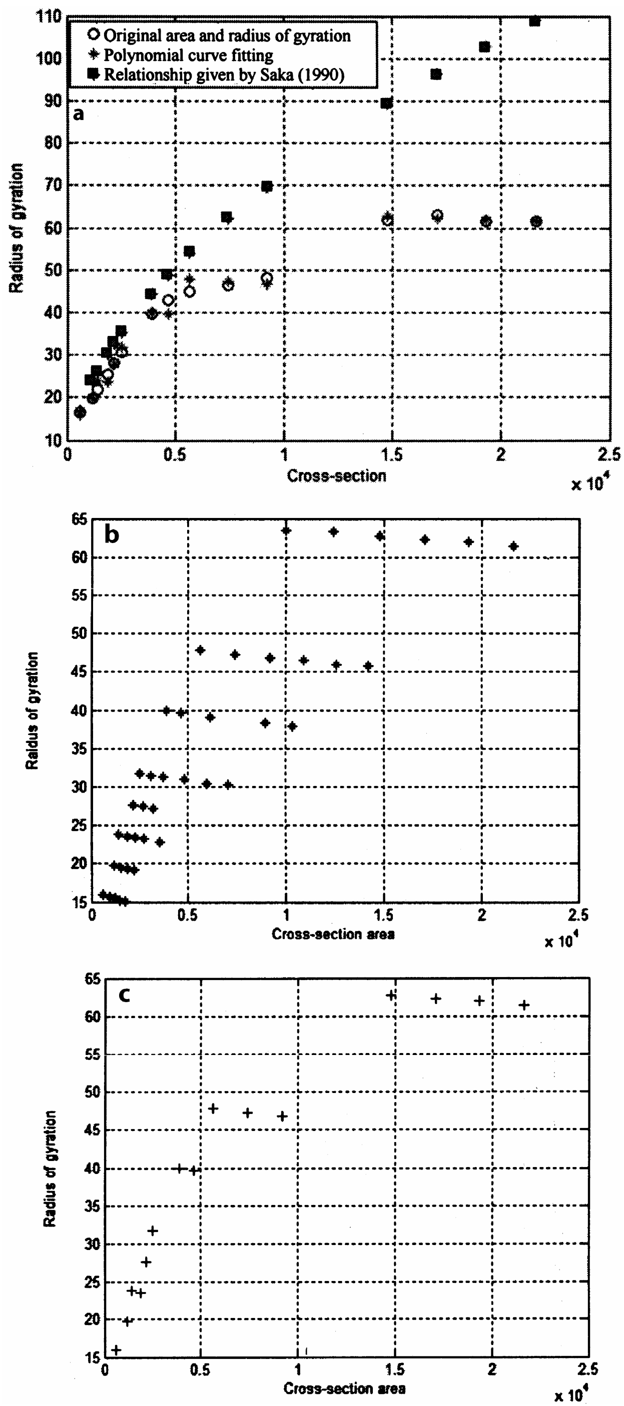


Fig. 12—Visualization of the polynomial fitting approximation (a) graphical symbolization of results, (b) before extracting, and (c) after extracting

code. The radius of gyration is calculated as 26.2 mm by Eq. (11) for the same member. And this value is used in the design process by SCP, SQP, and EVOL. But according to the double angle section list in the codes, the real radius of gyration of the member is equal to 15.09 mm which is used in the design process by GA. So it can easily be seen that this cause a higher value of the allowable compression stress to be calculated obeying the AISC-ASD. Therefore, one will end up with a larger cross-section by GA.

In this study a new relationship is proposed between the radius of gyration and the area of cross-section. A polynomial curve fitting approximation using MATLAB is offered instead of the relationship in Eq. (11). Then Bridge Truss 3 is designed to find the optimum weight once again by SCP, SQP, and EVOL in order to see how the minimum weight of truss changes depending on the radius of gyration. As presented in Table 5, the results show that the new relationship by the polynomial curve fitting approximation is more effective than old one (Fig. 12a). Further, it can be said that even this relationship is not very effective due to the fluctuation on radius of gyration (Fig. 12b).

There is not a regular relationship between the actual area of cross-sections and the radius of gyrations. Hence, some cross-sections are extracted from the list in order to get a smooth curve for the data given (Fig. 12c).

Naturally, this curve is not enough to represent the whole data. This is clearly shown in Table 5. The polynomials obtained for some cross-section types are as follows:

For double angle;

$$r= -18.5621\, z^7 +46.3275\, z^6 +18.4652\, z^5 -99.9004\, z^4 +16.4648\, z^3 +48.2871\, z^2+5.5338\, z+46.0816 \quad \ldots (12)$$

For single angle;

$$r= -8.9001\, z^7 +15.3680\, z^6 +21.0658\, z^5 -35.9631\, z^4 -11.5443\, z^3 +18.2282\, z^2 +5.9197\, z +25.2508 \quad \ldots (13)$$

For Tee;

$$r= -1.0952\, z^7 -3.6734\, z^6 +15.0258\, z^5 +1.7861\, z^4 -16.1685\, z^3 -0.8382\, z^2 +11.0013\, z+31.5137 \quad \ldots (14)$$

For Pipe;

$$r= 3.1701\, z^3 -9.0935\, z^2 +32.5148\, z +54.8731 \quad \ldots (15)$$

The value of $z$ in these polynomials is normalized according to mean and standard deviation that are calculated from extracting sections list presented in Table 6. Means and standard deviations related with the extracting sections list are presented in Table 7. The value of $z$ is calculated using Eq. (16).

Table 7—The means and standard deviations for the extracting profile list

| Section Types | Mean (mm$^2$) | Standard Deviation (mm$^2$) |
|---|---|---|
| Double angle | 9460 | 7230 |
| Single Angle | 2990 | 1390 |
| Tee | 6580 | 4540 |
| Pipe | 3350 | 2930 |

$$z = (x - mn)/std \quad \ldots (16)$$

where, $x$ represents the value of design variables (area of cross-sections), $mn$ stands for the mean value and $std$ symbolizes standard deviation.

## Observations and conclusions

Value encoding provides a comfort because the string length is independent of the number of sections list adopted for the design variables of optimization problem, and programming gets simpler since decoding scheme is not necessary.

Generally in the GA based optimization problems $p_m$ takes a value between 0.0 - 0.5, while the value of $p_c$ varies between 0.0-1.0 in the literature. However, the adaptive operators, in the current work, provide various ways for the consideration of $p_m$ and $p_c$ depending on the fitness of the solutions and the population. Hence, the adaptive operators used in the current work improve performance of GA.

For the examples solved, it is observed that the displacement constraints are dominant over stress and stability constraints. Therefore, it can be concluded that the displacement constraints play an important role in the optimization process for the bridge trusses. A lighter truss can be obtained with less number of iteration if displacement constraints are kept out in the design process.

It can be said that the results by GA are meaningful, more suitable for practice. GA performs well to find minimum weight of bridge trusses under moving load for optimization with discrete design variables. The optimizations with continuous design variables confirm the optimal designs reported.

## Acknowledgement

The authors thank Prof. Hartmann and Dr Baitsch, Institute for Computational Engineering, Faculty of Civil Engineering, Ruhr University Bochum, Germany, for their help and interest.

## Nomenclature

| | | |
|---|---|---|
| $GA$ | = | genetic algorithm |
| $SCP$ | = | sequential convex programming |
| $SQP$ | = | sequential quadratic programming |
| $EVOL$ | = | evolution strategy |
| $LP$ | = | linear programming |
| $QP$ | = | quadratic programming |
| $NP$ | = | nonlinear programming |
| $KT$ | = | Kuhn tucker |
| $SGA$ | = | simple genetic algorithm |
| $p_m$ | = | probability of mutation |
| $p_c$ | = | probability of crossover |
| $f$ | = | fitness of an individual |
| $\bar{f}$ | = | average fitness value |
| $f_{max}$ | = | maximum fitness value |
| $f_{min}$ | = | minimum fitness value |
| $f'$ | = | larger of the fitness value |
| $k_1, k_2, k_3, k_4$ | = | coefficients |
| $W$ | = | Weight of the bridge truss |
| $NG$ | = | number of groups |
| $M$ | = | total number of members |
| $N$ | = | number of restricted displacement |
| $\sigma^t_{allowable}$ | = | allowable tensile stress |
| $\sigma^c_{allowable}$ | = | allowable compression stress |
| $\sigma_{il}$ | = | stress in member $i$ |
| $l$ | = | load case |
| $d_{jl}$ | = | displacement of joint $j$ |
| $j$ | = | joint number |
| $d_{allowable}$ | = | upper bound of displacement |
| $A_k$ | = | area of members belonging to group $k$ |
| $k$ | = | group number |
| $m$ | = | number of members in group $k$ |
| $L_i$ | = | length of member $i$ |
| $\rho_i$ | = | density of member $i$ |
| $K$ | = | penalty coefficient |
| $C$ | = | violation coefficient |
| $I$ | = | impact factor |
| $L$ | = | span of the bridge |
| $r$ | = | radius of gyration |
| $a$ | = | constant |
| $b$ | = | constant |
| $z$ | = | variable of polynomial |
| $x$ | = | design variable |
| $mn$ | = | mean value |
| $std$ | = | standard deviation |

## References

1 Camp C, Pezeshk S & Cao G, *ASCE J Struc Eng*, 124 (1998) 551-559.

2 Arora J S & Wang Q, *Struct Multidisp Optimiz*, 30 (2005) 251-272.

3 Baitsch M, *Optimization of imperfect structures using distributed components*, paper presented at Seventh International Conference on Computational Structures Technology, Civil-Comp., Lisbon, Portugal, 2004.

4 Hartmann D, *Design Optimization*, Lecture notes; Computational Engineering, Ruhr University Bochum, 2004.

5 Arora J S, *Introduction to optimum design*, (McGraw-Hill, New York), 2004.

6 Dianati M, *An Introduction to Genetic Algorithms and Evolution Strategies*, (Technical Report, University of Waterloo), 2002.

7 Krishnamoorthy C S, Venkatesh P P & Sudarshan R, *ASCE J Comput Civil Eng*, 16 (2002) 66-75.
8 Marler R T & Arora J S, *Struct Multidisp Optimiz*, 26 (2004) 369-395.
9 Goldberg D E, *Genetic Algorithms in Search, Optimization and Machine Learning* {Addison- Wesley, New York}, 1989.
10 Srinivas M & Patnaik L M, *IEEE Trans Syst, Man & Cybern*, 24 (1994) 656-667.
11 Yang S, *Adaptive crossover in genetic algorithms using statistics mechanism*, edited by Bedan (MOT Press), 2002, 182.
12 Toğan V & Daloğlu T A, *Eng Struc*, 28 (2006) 1019-1027.
13 Jenkins W M, *Comput & Struct*, 80 (2002) 471-480.
14 Toğan V, *Size and Shape Optimization of Trusses under Dead and Moving Loads Using Genetic Algorithm* (in Turkish), M. Sc. Thesis, Karadeniz Technical University, Trabzon, 2004.
15 Rajaev S & Krishnamoorthy C S, *ASCE J Struc Eng.*, 118 (1992) 1233-1250.
16 Salmon C G & Johnson, J E, *Steel structure: Design and behavior*, 4th ed (Harper Collins Publishers, New York), 1996.
17 Saka M P, *ASCE J Struc Eng*, 116 (1990) 2599-2620.