A NOVEL PENALTY SCHEME IN GENETIC ALGORITHMS FOR STRUCTURAL DESIGN OPTIMIZATION

Pruettha Nanakorn and Konlakarn Meesomklin

Sirindhorn International Institute of Technology,
Thammasat University, Thailand

ABSTRACT

In genetic algorithms, constraints are mostly handled by using the concept of penalty functions, which penalize infeasible solutions by reducing their fitness values in proportion to the degrees of constraint violation. In most of the available penalty schemes, some coefficients or constants have to be specified at the beginning of the calculation. Since these coefficients usually have no clear physical meanings, it is nearly impossible to estimate the appropriate values of these coefficients even by experience. Moreover, most of the schemes employ constant coefficients throughout the entire calculation. This may result in too weak or too strong a penalty during different phases of the evolution. In this study, a new penalty scheme that is free from the aforementioned disadvantages is developed. The proposed penalty function will be able to adjust itself during the evolution in such a way that the desired degree of the penalty is always obtained. The coefficient used in the proposed scheme will have a clear physical meaning. Thus, it will not be difficult to set the value of the coefficient by using experience.

1. INTRODUCTION

It is commonly known that Genetic Algorithms (GAs) are directly applicable only to unconstrained optimization. Nevertheless, many researchers have proposed solutions that can eliminate this limitation. Constraints are mostly handled by using the concept of penalty functions, which penalize infeasible solutions by reducing their fitness values in proportion to their degrees of constraint violation. In all available penalty schemes, the degree of the penalty can be further controlled by means of setting values of various coefficients in the penalty functions^{1,2,3}. Most of these coefficients are treated as constants during the calculation and their values have to be specified at the beginning of the calculation^{4,5,6}. These coefficients usually have no clear physical meanings. Thus, it is nearly impossible to know the appropriate values of the coefficients even by experience. This is because it is very hard to understand the correlation between the values of the coefficients and the characteristics of the problems without physical meanings of the coefficients. Consequently, for all problems with either similar or different natures, the appropriate values of the coefficients are generally obtained by trial and error. Many researchers, however, have tried to suggest different ranges of appropriate values for these coefficients, for various types of problem. Most of these suggestions are obviously doubtful. The reason is simply

that the appropriate values are usually given without any reference to the units used in the problems although the coefficients may have units and the appropriate values should vary with the units used. Another important concern is that these conventional penalty schemes do not usually adjust the strength of the penalty during the calculation, as the coefficients used are always kept constant. As a result, too weak or too strong penalty during different phases of the evolution may occur. This may lead to inaccurate solutions. Actually, there are some penalty schemes that vary the values of the coefficients to adjust the strength of the penalty during the calculation^{7,8,9}. However, this kind of scheme usually requires the varying values of these coefficients to be manually specified. It, therefore, becomes even more difficult to judiciously select appropriate values for different phases of the calculation.

Keeping these facts in mind, we develop a new penalty scheme that is free from the above disadvantages in this study. The proposed penalty function will be able to adjust itself during the evolution in such a way that the desired degree of the penalty is always obtained. The coefficient used in the proposed scheme will have a clear physical meaning and it will have no unit. Thus, it will not be difficult to set the value of the coefficient by experience.

2. GENETIC ALGORITHMS FOR CONSTRAINED OPTIMIZATION

In GAs, an optimization problem can be generally written as

$$F(x) = F[f(x)] \quad (1)$$

under constraints defined as

$$\begin{aligned} g_i(x) &\leq 0, & i = 1, \dots, K, \\ h_i(x) &= 0, & i = 1, \dots, P \end{aligned} \quad (2)$$

For the structural design optimization, x is an N -dimensional vector called the design vector, representing design variables of N structural components to be optimized and $f(x)$ is the objective function. In addition, $g_i(x)$ and $h_i(x)$ are inequality and equality constraints, respectively. They represent constraints, which the design must satisfy, such as stress and displacement limits. Finally, $F(x)$ is the fitness function which is defined as a figure of merit¹⁰.

It is not possible to directly utilize GAs to solve the above problem due to the presence of the constraints. In GAs, constraints are mostly handled by using the concept of penalty functions, which penalize infeasible solutions, i.e.,

$$\begin{aligned} F^a(x) &= F(x) & x \in \tilde{F} \\ F^a(x) &= F(x) - P(x) \end{aligned} \quad (3)$$

where $\tilde{F}$ denotes the feasible search space. Here, $P(x)$ is a penalty function whose value is greater than zero. In addition, $F^a(x)$ represents the augmented fitness function after

the penalty. Several forms of penalty functions have been proposed in the literature^{11,12,13}. Nevertheless, most of them can be written in the following general form, i.e.,

$$P(x) = \sum_{j=1}^k (\lambda_G)_j [G_j(x)]^\beta + \sum_{j=1}^p (\lambda_H)_j [H_j(x)]^\beta \quad (4)$$

where

$$\begin{aligned} G_j(x) &= \max [0, g_j(x)] \\ H_j(x) &= \text{abs} [h_j(x)] \end{aligned} \quad (5)$$

Here, $G_j(x)$ and $H_j(x)$ represent the degrees of the inequality and equality constraint violations, respectively. In addition, $(\lambda_G)_j$, $(\lambda_H)_j$ and β are constants. In most cases, the same value is used for all $(\lambda_G)_j$'s and $(\lambda_H)_j$'s. As for β , it is usually set to be 1 or 2.

In the first operator in GAs, the reproduction operator, a mating pool is created by letting individuals with higher fitness values have higher chance to be selected into the mating pool. Many reasonable selection algorithms are possible. However, the most widely used technique is the proportional selection. In this technique, the probability of the individual to be selected into the mating pool is

$$p(x_i) = \frac{F^a(x_i)}{\sum_{j=1}^n F^a(x_j)} \quad (6)$$

where x_i represents the i^{th} individual in the population and n is the population size. Clearly, in the above equation, it is essential that all fitness values must be positive. Therefore, the obtained fitness function after the penalty $F^a(x)$ may not be directly usable as its values may be negative. Moreover, the difference between the fitness values of the best feasible individuals and the average individuals varies generation by generation. In early generations, the difference can be very large and the best individuals become relatively too strong. As a result, the premature convergence may be obtained. In later generations, the difference can be very small and the average individuals become almost as strong as the best individuals. As a result, the search may become a random walk. To prevent all of these problems, the augmented fitness function is usually scaled into a specified positive range.

The penalty schemes used in GAs play a very important role in the performance of GAs. This role becomes even more important when the optimal solution lies on or close to the boundary between feasible and infeasible search spaces, which is very usual for the structural design optimization. In this study, we propose a new penalty scheme. The aim of the development is to create a scheme that is free from the disadvantages of the existing schemes, mentioned earlier.

3. ADAPTIVE PENALTY FUNCTION

To make the scheme simple, we employ a simple form of the penalized fitness function, i.e.,

$$F_i^a = F^a(x_i) = F(x_i) - P(x_i) = F(x_i) - \lambda(t)E(x_i) \quad (7)$$

where F_i^a represents the fitness function of the i^{th} individual after the penalty. Here, $\lambda(t)$ is a factor multiplied to $E(x_i)$ that is an error term. The factor $\lambda(t)$ varies with generation and the generation number is denoted by t . In this study, the error term $E(x_i)$ is defined as

$$E(x_i) = \sum_{j=1}^K G_j(x_i) + \sum_{j=1}^P H_j(x_i) \quad (8)$$

where $G_j(x_i)$ and $H_j(x_i)$ have already been defined in Eq. (5).

Now, the question is what the magnitude of the factor $\lambda(t)$ should be. It is not difficult to imagine that if the factor is too small, infeasible individuals with high original fitness values may have penalized fitness values higher than the fitness value of the feasible optimal individual being searched. If this happens, the population in subsequent generations will move toward the false peaks that appear in the infeasible region. On the contrary, if $\lambda(t)$ is too large, good characteristics in some infeasible individuals will have no chance to survive and will disappear rapidly. This may lead to premature convergence and the obtained solution can be quite wrong.

To avoid the above problems, the degree of the penalty must be enough to make the feasible optimal solution have the maximum fitness value, compared with all individuals (feasible and infeasible) after the penalty. However, the penalty must not be made too much stronger than that. To this end, we introduce the following condition, i.e.,

$$F^a(x_i) \leq \phi F_{avg}^{a,\tilde{F}} \quad \text{for } \forall x_i \in \tilde{U} \quad (9)$$

in which $\tilde{U}$ represents the infeasible search space. Here, $F_{avg}^{a,\tilde{F}}$ denotes the average fitness value of all feasible individuals in the generation and ϕ is a constant.

The above condition sets the maximum fitness value of the infeasible individuals in the generation to be at most equal to $\phi F_{avg}^{a,\tilde{F}}$, but not more than that. At this moment, it is not useful to consider the physical meaning of the constant ϕ yet because the penalized fitness function will have to be scaled afterwards. Therefore, it is enough to simply say that the factor ϕ is used to adjust the strength of the penalty. A way to obtain the value of this constant will be explained shortly. To satisfy the condition in Eq. (9), we calculate the factor $\lambda(t)$ by computing, for each infeasible individual, the factor $\lambda(t)$ that makes the penalized fitness value of that infeasible individual exactly equal to $\phi F_{avg}^{a,\tilde{F}}$. After that, the values of the factor obtained from all infeasible individuals are compared and the

maximum one is selected as $\lambda(t)$. If the maximum value is negative, zero is used instead. In short, we can express $\lambda(t)$ as

$$\lambda(t) = \max \left(0, \max_{\forall x_i \in \bar{U}} \left[\frac{F^a(x_i) - \phi F_{avg}^{a, \bar{F}}}{E(x_i)} \right] \right) \quad (10)$$

Eq. (10) insures that Eq. (9) is satisfied.

In this study, we employ a modified bilinear scaling technique shown in Figure 1. The minimum scaled fitness is set to zero to avoid negative fitness values while the scaled

Figure 1: Bilinear fitness scaling.

fitness of the average fitness of all feasible individuals $F_{avg}^{a, \bar{F}}$ is set to one. Furthermore, the maximum scaled fitness that is to be obtained from the best feasible members is set to C . Thus, the chance of the best feasible members being selected into the mating pool is equal to C times that of the average feasible members. All together, we have

$$\begin{aligned} F^s(x) &= \frac{C-1}{F_{max}^{a, \bar{F}} - F_{avg}^{a, \bar{F}}} F^a(x) + \frac{F_{max}^{a, \bar{F}} - C F_{avg}^{a, \bar{F}}}{F_{max}^{a, \bar{F}} - F_{avg}^{a, \bar{F}}} & \text{if } F^a(x) \geq F_{avg}^{a, \bar{F}} \\ F^s(x) &= \frac{1}{F_{avg}^{a, \bar{F}} - F_{min}^a} F^a(x) + \frac{F_{min}^a}{F_{min}^a - F_{avg}^{a, \bar{F}}} & \text{if } F^a(x) \leq F_{avg}^{a, \bar{F}} \end{aligned} \quad (11)$$

where $F^s(x)$ denotes the scaled fitness function. In addition, F_{min}^a denotes the minimum fitness value after the penalty while $F_{max}^{a, \bar{F}}$ denotes the fitness value of the best feasible members. This scaled fitness function $F^s(x)$ will be used in Eq. (6) instead of $F^a(x)$.

For all generations, we set the chance of the best infeasible members being selected into the mating pool to be equal to ϕ times that of the average feasible members, i.e.,

$$F^s(x_i) \leq (\phi F_{avg}^{s,\tilde{F}} = \phi) \quad \text{for } \forall x_i \in \tilde{U} \quad (12)$$

where $F_{avg}^{s,\tilde{F}}$ is the scaled value of the average fitness of all feasible individuals which is equal to 1. From the above condition, we can express ϕ in Eq. (9) in terms of ϕ as

$$\begin{aligned} \phi &= \frac{CF_{avg}^{a,\tilde{F}} + F_{max}^{a,\tilde{F}}(\phi - 1) - \phi F_{avg}^{a,\tilde{F}}}{(C - 1)F_{avg}^{a,\tilde{F}}} \quad \text{for } \phi \geq 1 \\ \phi &= \frac{F_{min}^a + \phi F_{avg}^{a,\tilde{F}} - \phi F_{min}^a}{F_{avg}^{a,\tilde{F}}} \quad \text{for } \phi \leq 1 \end{aligned} \quad (13a, b)$$

In the real calculation, the coefficient ϕ will be set at the beginning of the calculation. This coefficient has a very clear physical meaning, i.e., the chance to be selected into the mating pool of the best infeasible members compared with that of the average feasible members. In addition, the coefficient does not have any unit. Due to these reasons, it is possible to set this coefficient by using experience. Knowing ϕ , we can compute ϕ and, subsequently, the factor $\lambda(t)$. In case of $\phi \geq 1$, ϕ can be obtained from Eq. (13a) directly because all parameters in the equation are readily available. In this case, the parameters $F_{avg}^{a,\tilde{F}}$ and $F_{max}^{a,\tilde{F}}$ can be obtained directly from the original fitness values of the feasible individuals without any penalty consideration. On the contrary, if $\phi < 1$, ϕ cannot be obtained without iteration since one of the parameters, i.e., F_{min}^a , is not readily available. Note that F_{min}^a is the minimum fitness in the generation after the penalty and it is most likely that F_{min}^a will belong to the infeasible members. This F_{min}^a can be obtained from Eq. (7), which, in turn, requires the value of ϕ [see Eq. (10)]. Nevertheless, the required iteration is very simple and takes almost no time to perform.

Figure 2: Bilinear fitness scaling for cases where no feasible individual is available.

In short, the purpose of the scheme is to fix, throughout the calculation, the relative chance of the best infeasible members being selected into the mating pool compared with that of the average feasible members. This means that the penalty is always adjusted so that the aforementioned purpose is achieved in all generations. This guarantees that the desired degree of the penalty is obtained throughout the evolution. Consequently, the problem of too weak or too strong penalty during different phases of the evolution is removed.

Since the proposed penalty scheme requires the average fitness value over all feasible individuals, it is necessary to have at least one feasible individual in the population. In the case that there is none, the fitness values of the infeasible individuals will be given based on the magnitudes of the error they have. The idea is to strongly encourage the population to move toward the feasible region. Here, a bilinear scaling scheme shown in Figure 2 is used. The fitness is scaled in such a way that the scaled fitness values of the individuals with the highest error are equal to zero and the scaled fitness values of the individuals with the average error are equal to one. In addition, the scaled fitness values of the individuals with the smallest error are set to be Z . Thus, the chance of the individuals with the smallest error being selected into the mating pool is equal to Z times that of the individuals with the average error. In summary, we have

$$\begin{aligned}
 F^s(x) &= \frac{Z-1}{E_{\min} - E_{avg}} E(x) + \frac{E_{\min} - ZE_{avg}}{E_{\min} - E_{avg}} & \text{if } E(x) \leq E_{avg} \\
 F^s(x) &= \frac{1}{E_{avg} - E_{\max}} E(x) + \frac{E_{\max}}{E_{\max} - E_{avg}} & \text{if } E(x) > E_{avg}
 \end{aligned} \tag{14}$$

4. RESULTS

To investigate the validity and efficiency of the proposed penalty scheme, the scheme is used in the design optimization of two different structures, i.e., six-bar and ten-bar trusses. To be able to see clearly the advantages of the proposed scheme over the conventional schemes, the obtained results are compared with those from a selected conventional scheme. Since most of the conventional schemes are based on the same concept with slightly different details, comparison with one selected conventional scheme is sufficient to show the advantages of the proposed scheme over the conventional schemes. Finally, the results are also compared with the existing results in the literature.

4.1. Six-Bar Truss

The first problem to be considered is the six-bar truss shown in Figure 3. Here, we consider only the sizing optimization. Thus, the design variables are six sectional areas of the six members of the truss. The cross-sectional area of each member is taken from the following 32 discrete values, i.e., 1.62, 1.80, 2.38, 2.62, 2.88, 3.09, 3.13, 3.38, 3.63, 3.87, 4.18, 4.49, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.5, 13.5, 13.9, 14.2, 15.5, 16.0, 18.8, 19.9, 22.0, 22.9, 26.5, 30.0, and 33.5 in². Therefore, a five-bit string is required for each design variable. There are two types of constraint in this problem, i.e., the stress and displacement constraints. The design parameters used in the problem are shown in Table 1.

Figure 3: Six-bar truss.

For comparison, the most popular conventional penalty scheme is also used to solve the problem. The conventional form is expressed as

$$F_i^a = F^a(x_i) = F(x_i) - P(x_i) = F(x_i) - \lambda E(x_i) \quad (15)$$

where the coefficient λ is constant and the error term $E(x_i)$ is the same as that defined in Eq. (8). In both proposed and conventional schemes, the fitness function $F(x_i)$ is defined as

$$F(x_i) = \frac{1}{1 + \text{Weight}(x_i)} \quad (16)$$

where two different units of weight, i.e., pound (lb) and newton (N) are used. Two units are used in order to investigate the effect of unit on the results from both schemes. Since it is impossible to judiciously estimate the appropriate value of the coefficient λ in the conventional scheme, a wide range of values will be used. All GA parameters used can be found in Table 1. To start the calculation, the initial population is generated at random. The type of crossover operator used here is the one-point crossover¹.

Figure 4 shows the results obtained from the proposed and conventional schemes. Each point in the graph represents an average weight of the best feasible designs obtained from 200 different runs. The results obtained by using newtons in Eq. (16) are converted into pounds for comparison. In the conventional scheme, the coefficient λ is varied

Table 1: Design and GA parameters for the six-bar truss problem.

Design parameters		GA parameters	
Item	Value	Item	Value
Modulus of elasticity	10 ⁷ psi	Maximum number of generations	100
Weight density	0.1 lb/in ³		
Allowable tensile stress	25,000 psi	Population size	70
Allowable compressive stress	25,000 psi	Crossover probability	0.8
Maximum y-displacement	2 in.	Mutation probability	0.001
		ϕ	0.25 – 1.75
		γ	0.000001 – 100
		C	2
		Z	5

Table 2: Comparison of the results for the six-bar truss problem.

Item	Proposed	Rajan [8]
Area 1 (in ²)	30.0	30.0
Area 2	19.9	19.9
Area 3	15.5	15.5
Area 4	7.22	7.22
Area 5	22.0	22.0
Area 6	22.0	22.0
Total Weight (lb)	4962.1	4962.1

Figure 4: Average weight of the best feasible designs obtained from 200 runs — six-bar truss.

exponentially from 0.000001 to 100 while in the proposed scheme the coefficient is varied from 0.25 to 1.75. Note that the value of ϕ should be varied between 0 to 2.0 since the maximum scaled fitness value C is set to be 2.0 (see Table 1). It can be clearly seen from the results that the proposed scheme is more robust than the conventional scheme. In the proposed scheme, changing the unit has little effect on the results while in the conventional scheme the effect is much more noticeable. Moreover, in the proposed scheme, it is easier to notice the trend of the results when ϕ is varied. It can be reasonably said that good results are obtained with the value of ϕ around 0.75. On the contrary, in the conventional scheme, this kind of trend is not very obvious, considering both results obtained with newtons and pounds. Nevertheless, we may say that good results are obtained with the value of λ around 0.0001.

The best result obtained from the proposed scheme is also compared with the best result reported by Rajan⁸. They are exactly the same. The details of the result are shown in Table 2. It must be noted that in this study, except for the new penalty algorithm, the rest of the algorithms are standard. This is not the case for the work by Rajan⁸, which employs more complicated GAs.

4.2 Ten-Bar Truss

The next problem to be considered is the ten-bar truss shown in Figure 5. This problem is one of the benchmark problems used to test optimization methods. In this problem, we also consider only the sizing optimization. Therefore, the design variables are ten sectional areas. The cross-sectional areas of members 1, 3, 4, 7, 8 and 9 are taken from the following 32 discrete values, i.e., 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.5, 13.5, 13.9, 14.2, 15.5, 16.0, 16.9, 18.8, 19.9, 22.0, 22.9, 26.5, 30.0, and 33.5 in². For the rest of the members, the cross-sectional areas are taken from the following 32 discrete values, i.e., 1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.5, 13.5, 13.9, and 14.2 in². Similar to the previous problem,

Figure 5: Ten-bar truss.

Table 3: Design and GA parameters for the ten-bar truss problem.

Design parameters		GA parameters	
Item	Value	Item	Value
Modulus of elasticity	10 ⁷ psi	Maximum number of generations	100
Weight density	0.1 lb/in ³		
Allowable tensile stress	25,000 psi	Population size	40
Allowable compressive stress	25,000 psi	Crossover probability	0.8
Maximum x , y -displacement	2 in.	Mutation probability	0.001
		ϕ	0.25 – 1.75
		γ	0.000001 – 100
		C	2
		Z	5

a five-bit string is required for each design variable. The design parameters and genetic parameters are shown in Table 3.

The results obtained from the proposed and conventional schemes are shown in Figure 6. Similar to the previous problem, each point in the graph represents an average weight of the best feasible designs obtained from 200 different runs. The robustness of the proposed scheme is again obvious. The effect of the unit used on the results from the proposed scheme is noticeably less than that on the results from the conventional scheme. Moreover, the results from the proposed scheme also exhibit clear tendency with respect to the value of the coefficient used while those from the conventional scheme do not. In the proposed

Figure 6: Average weight of the best feasible designs obtained from 200 runs — ten-bar truss.

Table 4: Comparison of the results for the ten-bar truss problem.

Item	Proposed	Rajeev and Krishnamoorthy [e beginning of the calculation ⁴]	Camp et al. [6]	Galante [Error! Reference source not found.]
Area 1 (in ²)	33.5	33.5	30.0	33.5
Area 2	1.62	1.62	1.62	1.62
Area 3	22.9	22.0	26.5	22.0
Area 4	15.5	15.5	13.5	14.2
Area 5	1.62	1.62	1.62	1.62
Area 6	1.62	1.62	1.62	1.62
Area 7	7.22	14.2	7.22	7.97
Area 8	22.9	19.9	22.9	22.9
Area 9	22.0	19.9	22.0	22.0
Area 10	1.62	2.62	1.62	1.62
Total Weight (lb)	5,499.3	5,613.8	5,556.9	5,458.3

scheme, it can be said that good results are obtained with the value of ϕ around 0.5. In the conventional scheme, although it is not very obvious, we may barely say that good results are obtained with the value of λ around 0.1.

The best result obtained from the proposed scheme is also compared with the best results reported by Rajeev and Krishnamoorthy⁴, Camp et al.⁶, and Galante¹⁰ in Table 4. It can be seen that the result obtained from the proposed penalty scheme is relatively very good although Rajeev and Krishnamoorthy⁴, Camp et al.⁶, and Galante¹⁰ employ more complicated GAs.

In the previous problem, the appropriate value of ϕ is around 0.75 and the appropriate value of λ is around 0.0001. Since the two problems are quite similar, similar values of the coefficients from the two problems are expected. In this aspect, the proposed scheme evidently outperforms the conventional scheme. Having similar appropriate values of the coefficient for similar problems allows the coefficient to be set by experience. Since the coefficient in the proposed scheme has a physical meaning, which directly corresponds to the understandable degree of the penalty, the characteristics of the problems being solved can be directly related to the appropriate degree of the penalty. This kind of advantage cannot be found in the conventional schemes.

5. CONCLUSION

This paper presents a new adaptive penalty scheme in GAs for structural design optimization. The existing penalty schemes generally require the values of some coefficients to be specified at the beginning of the calculation and these coefficients usually have no clear physical meanings. Consequently, it is very difficult to select the appropriate values of these coefficients even by experience. Moreover, most of the existing schemes employ constant coefficients throughout the entire calculation. This may result in too weak or too strong penalty during different phases of the evolution. To avoid these drawbacks, a new penalty scheme is proposed. The main concept of the proposed scheme is to fix, throughout all generations, the chance to be selected into the mating pool of the best infeasible members compared with that of the average feasible members. The parameter that has to be set is the ratio between the fitness value of the best infeasible members and the fitness value of the average feasible members. This ratio has a very clear physical meaning. Therefore, it can be set easily, based on experience of different types of problem. In addition, under this concept, the penalty is always adjusted so that the desired degree of the penalty is achieved in all generations.

The proposed scheme is tested by using two optimization problems of truss structures. The comparisons with a representative conventional scheme clearly show the advantages of the proposed method. From the results, it can be seen that the proposed scheme is robust and the required parameter can be obtained by experience. In addition, the comparisons with the results from the literature also show that the proposed penalty scheme yields relatively good results although, except for the new penalty algorithm, the proposed technique employs very standard GAs.

6. ACKNOWLEDGEMENTS

The authors are grateful to the Thailand Research Fund (TRF) for providing the financial support to this study.

7. REFERENCES

1. Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley. Reading, Massachusetts.
2. Deb, K. 1995. *Optimization for Engineering Design*. Prentice-Hall. Englewood Cliffs. N.J.
3. Michalewicz, Z. 1996. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer. New York.
4. Rajeev, S. and Krishnamoorthy, C. S. 1992. Discrete optimization of structures using genetic algorithms. *Journal of Structural Engineering*. 118(5):1233-1249.
5. Jenkins, W. M. 1997. On the application of natural algorithms to structural design optimization. *Engineering Structures*. 19(4):302-308.

6. Camp, C., Pezeshk, S. and Cao, G. 1998. Optimized design of two-dimensional structures using a genetic algorithm. *Journal of Structural Engineering*. 124(5):551-559.
7. Adeli, H. and Cheng, N. 1993. Integrated genetic algorithm for optimization of space structures. *Journal of Aerospace Engineering*. 6(4):315-328.
8. Rajan, S. D. 1995. Sizing, shape, and topology design optimization of trusses using genetic algorithm. *Journal of Structural Engineering*. 121(10):1480-1487.
9. Rafiq, M. Y. and Southcombe, C. 1998. Genetic algorithms in optimal design and detailing of reinforced concrete biaxial columns supported by a declarative approach for capacity checking. *Computers and Structures*. 69:443-457.
10. Galante, M. 1996. Genetic algorithms as an approach to optimize real-world trusses. *International Journal for Numerical Methods in Engineering*. 39:361-382.