PERGAMON

Computers and Structures 79 (2001) 2527–2539

Computers & Structures

www.elsevier.com/locate/compstruc

An adaptive penalty function in genetic algorithms for structural design optimization

Pruettha Nanakorn *, Konlakarn Meesomklin

Civil Engineering Program, Sirindhorn International Institute of Technology, Thammasat University, P.O. Box 22, Thammasat-Rangsit Post Office, Pathumthani 12121, Thailand

Received 4 July 2000; accepted 23 July 2001

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 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 appropriate values of these coefficients even by experience. Moreover, most 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 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.

Keywords: Genetic algorithms; Optimization; Structural design; Adaptive penalty function; Constrained optimization; Truss and frame structures

1. Introduction

When designing structures, engineers have to consider not only the load-carrying capacity of the structures but also the cost to construct them. Material cost is one of the major costs in construction. Designs that use a smaller amount of materials are therefore preferable, given that the construction methods do not become too expensive or impractical. To achieve this goal, optimization techniques have been employed in structural design [1–5]. There are many conventional optimization methods [6,7], each of which may work well for some specific problems. To select appropriate optimization methods for structural design, it is necessary to understand characteristics of this kind of optimization problem. The first important characteristic of structural design optimization is that, in structural design optimization, the solution sought is the global optimal solution. Moreover, in structural design, design variables are generally discrete variables. Finally, structural design optimization always contains constraints. These three major characteristics suggest that genetic algorithms (GAs) can be the choice. This is simply because this optimization technique is generally suitable for problems with discrete variables. Moreover, it searches for the global optimal point. Though GAs cannot be directly applied to problems with constraints, small modification can be used to incorporate constraints. Due to these facts, the technique is gaining popularity among researchers in the field of structural design optimization [8–14].

GAs are global probabilistic search algorithms inspired by Darwin's survival-of-the-fittest theory [15].

* Corresponding author. Tel.: +66-2-986-9009x1906; fax: +66-2-986-9009x1900.

E-mail address: nanakorn@siit.tu.ac.th (P. Nanakorn).

0045-7949/01/$ - see front matter
PII: S0045-7949(01)00137-7

They have received considerable attention because of their versatile applications in several fields [6,8,14–17]. GAs start their search from many points in search space at the same time. These starting search points are usually selected randomly. Through the consideration of fitness values of these search points, which are given based on their merit, and the randomized information exchange among the points, a new set of search points with higher merit is created. The process is then repeated until a satisfactory result is obtained. Since the technique utilizes information from many search points at the same time, there is less chance for the search to be trapped in any of the local optimal points. Another distinguishing characteristic of GAs is that the algorithms work with coding of the parameter set, not the parameters themselves. Generally, the binary code is used. Because of the discrete nature of coding, the algorithms are the perfect choice for those problems with discrete variables.

Since GAs are directly applicable only to unconstrained optimization, many researchers have proposed solutions that can eliminate this limitation. Constraints are mostly handled by using 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 penalty can be further controlled by means of setting values of various coefficients in penalty functions [6,8, 15,18]. Most of these coefficients are treated as constants during the calculation and their values have to be specified at the beginning of the calculation [19–21]. These coefficients usually have no clear physical meanings. Thus, it is nearly impossible to know 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 being solved without physical meanings of the coefficients. Consequently, for all problems with either similar or different natures, 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 appropriate values are usually given without any reference to the units used in the problems although the coefficients may have units and appropriate values should vary with the units used. Another important concern is that these conventional penalty schemes do not 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 a penalty during different phases of the evolution may occur. This will 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 [9,12,22]. However, these schemes require 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.

Several different ideas that are more sophisticated have been proposed to improve penalty function methods for handling constrained optimization problems [23]. Powell and Skolnick [24] re-mapped fitness values of both feasible and infeasible individuals in such a way that all feasible solutions have higher fitness than any infeasible solutions. The key concept of this approach is the assumption of the superiority of feasible solutions over infeasible ones. Unfortunately, this assumption rarely holds during the evolution since it always happens that some infeasible individuals process very good genes that can be very valuable for later generations. As a result, these infeasible individuals are more preferable during the evolution than many low fitness feasible individuals. For this reason, it is necessary to allow some infeasible individuals to have higher fitness than some feasible individuals. Le Riche et al. [25] proposed a segregated GA that uses two values of penalty parameters for each constraint instead of one. The population is split into two coexisting and cooperating groups, where individuals in each group are evaluated using either one of the two penalty parameters. During the evolution, the two groups interbreed. Since the two penalty parameters are different, the two groups converge in the design space along two different trajectories, which helps locate the optimal region faster. If a large value is selected for one of the penalty parameters and a small value for the other parameter, simultaneous convergence from both feasible and infeasible sides can be achieved. However, although the approach provides a new overall penalty scheme, the problem with this approach is still the way of choosing the penalty for each of the two groups.

Rasheed [26] proposed a penalty scheme with an adaptive penalty coefficient. The scheme considers two key individuals of the population, i.e., the point that has the least sum of constraint violations and the point that has the best fitness value. These two points are compared at every certain number of generations. If both points are the same then the penalty coefficient is assumed adequate; otherwise, the penalty coefficient is increased to make the two points have equal fitness values. In addition, the penalty coefficient is reduced if at some stage the population contains no infeasible points. The inconveniences of this technique are how to choose the initial value for the penalty coefficient and how to appropriately update it. In addition, the size of the generation gap for updating the penalty coefficient must reasonably be selected. Coello [27] proposed a technique based on the concept of co-evolution to create two populations that interact with each other in such a way that one population evolves the penalty factors to be used by the fitness function of the main population, which is responsible for optimizing the objective function. This technique is inconvenient because the approach requires evolution of two parallel populations instead of one. Therefore, it is computationally more expensive.

In this study, a new adaptive penalty scheme is proposed. The penalty function used in the scheme will be able to adjust itself automatically during the evolution in such a way that the desired degree of penalty is always obtained. The coefficient used in the proposed scheme will have a clear physical meaning that directly represents the degree of penalty employed. Therefore, for each particular problem, the appropriate value of the coefficient can be reckoned based on the appropriate degree of penalty for the problem. In addition, the coefficient in the proposed scheme will have no units. For each particular problem, if the same value of the coefficient is used, similar results can always be expected even when different units are employed in the problem. Since it is expected that similar structural optimization problems require similar degrees of penalty, with the proposed scheme, it is therefore possible to set the value of the coefficient by using experience. It must be noted that the main objective of this work is to obtain an adaptive penalty scheme that is robust and can still reproduce the same quality of results as ones obtained from GAs found in the literature, whose penalty parameters are carefully obtained for each specific problem by trial and error. In brief, the proposed scheme will be a scheme that can efficiently be used in different problems without a lot of guesswork.

2. Genetic algorithms for constrained optimization

An optimization problem using GAs can be generally expressed as

Maximize

$$F(\mathbf{x}) = F[f(\mathbf{x})], \quad \mathbf{x} = (x_1, x_2, \ldots, x_N) \in \mathbf{R}^N, \tag{1}$$

under constraints defined as

$$\begin{aligned} g_i(\mathbf{x}) &\leqslant 0, \quad i = 1, \ldots, K, \\ h_i(\mathbf{x}) &= 0, \quad i = 1, \ldots, P. \end{aligned} \tag{2}$$

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

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

$$\begin{aligned} F^{\mathrm{a}}(\mathbf{x}) &= F(\mathbf{x}) && \text{if } \mathbf{x} \in \widetilde{\mathbf{F}}, \\ F^{\mathrm{a}}(\mathbf{x}) &= F(\mathbf{x}) - P(\mathbf{x}) && \text{otherwise,} \end{aligned} \tag{3}$$

where $\widetilde{\mathbf{F}}$ denotes the feasible search space. Here, $P(\mathbf{x})$ is a penalty function whose value is greater than zero. In addition, $F^{\mathrm{a}}(\mathbf{x})$ represents an augmented fitness function after the penalty. Several forms of penalty functions have been proposed in the literature [6,8,15,18]. Nevertheless, most of them can be written in the following general form, i.e.,

$$P(\mathbf{x}) = \sum_{j=1}^{K} (\lambda_G)_j [G_j(\mathbf{x})]^{\beta} + \sum_{j=1}^{P} (\lambda_H)_j [H_j(\mathbf{x})]^{\beta}, \tag{4}$$

where

$$\begin{aligned} G_j(\mathbf{x}) &= \max[0, g_j(\mathbf{x})], \\ H_j(\mathbf{x}) &= \mathrm{abs}[h_j(\mathbf{x})]. \end{aligned} \tag{5}$$

Here, $G_j(\mathbf{x})$ and $H_j(\mathbf{x})$ represent the degrees of 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. The degree of penalty can be controlled by adjusting the values of the coefficients $(\lambda_G)_j$'s and $(\lambda_H)_j$'s. These coefficients do not have physical meanings. Clearly, it is impossible to judiciously select appropriate values for them. Even though in common practice, one value is used for all $(\lambda_G)_j$'s and $(\lambda_H)_j$'s, which significantly simplifies the situation, the appropriate value of this one coefficient is still not obvious.

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 proportional selection. In this technique, the probability of the ith individual to be selected into the mating pool is

$$p(\mathbf{x}_i) = \frac{F^{\mathrm{a}}(\mathbf{x}_i)}{\sum_{j=1}^{n} F^{\mathrm{a}}(\mathbf{x}_j)}, \tag{6}$$

where $\mathbf{x}_i$ represents the ith individual in the population and n is the population size. Clearly, in the above equation, it is essential that all fitness values be positive. Therefore, the obtained fitness function after the penalty $F^{\mathrm{a}}(\mathbf{x})$ may not be directly usable as its values may be negative. Moreover, the difference between the fitness values of the best individuals and 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, premature convergence may be obtained. In later generations, the difference can be very small and 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, an augmented fitness function is usually scaled into a specified positive range. Many fitness scaling schemes have been proposed in the literature [11,15,16,18,28].

3. Adaptive penalty function

It can be easily seen that 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 the feasible and infeasible search spaces, which is very usual for structural design optimization. In this study, a new penalty scheme that is free from the disadvantages of existing schemes discussed earlier is proposed. To make the scheme simple, a simple form of the penalized fitness function is employed, i.e.,

$$F_i^{\mathrm{a}} = F^{\mathrm{a}}(\mathbf{x}_i) = F(\mathbf{x}_i) - P(\mathbf{x}_i) = F(\mathbf{x}_i) - \lambda(t)E(\mathbf{x}_i), \tag{7}$$

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

$$E(\mathbf{x}_i) = \sum_{j=1}^{K} G_j(\mathbf{x}_i) + \sum_{j=1}^{P} H_j(\mathbf{x}_i), \tag{8}$$

where $G_j(\mathbf{x}_i)$ and $H_j(\mathbf{x}_i)$ 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. If this happens, the population in subsequent generations will move toward false peaks that appear in the infeasible region. On the contrary, if the factor 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 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 much stronger than that. To this end, the following condition is introduced, i.e.,

$$F^{\mathrm{a}}(\mathbf{x}_i) \leqslant \phi(t) F_{\mathrm{avg}}^{\mathrm{a},\tilde{\mathbf{F}}} \quad \text{for } \forall \mathbf{x}_i \in \widetilde{\mathbf{U}} \tag{9}$$

in which $\widetilde{\mathbf{U}}$ represents the infeasible search space. Here, $F_{\mathrm{avg}}^{\mathrm{a},\tilde{\mathbf{F}}}$ denotes the average fitness value of all feasible individuals in the generation and $\phi(t)$ is a factor of $F_{\mathrm{avg}}^{\mathrm{a},\tilde{\mathbf{F}}}$.

The above condition sets the maximum fitness value of infeasible individuals in the generation t to be equal to $\phi(t)F_{\mathrm{avg}}^{\mathrm{a},\tilde{\mathbf{F}}}$. At this moment, it is not useful to consider the physical meaning of the coefficient $\phi(t)$ yet because the penalized fitness function will have to be scaled afterwards. Therefore, it is enough to simply say that the coefficient $\phi(t)$ is used to adjust the strength of the penalty in the generation. A way to obtain the value of this coefficient will be explained shortly.

The condition in Eq. (9) is satisfied by employing an appropriate value of the factor $\lambda(t)$ in Eq. (7). For each infeasible individual, the factor $\lambda(t)$ that makes the penalized fitness value of that infeasible individual exactly equal to $\phi(t)F_{\mathrm{avg}}^{\mathrm{a},\tilde{\mathbf{F}}}$ is computed. After that, values of the factor $\lambda(t)$ obtained from all infeasible individuals are compared and the maximum one is selected as the real $\lambda(t)$. If the maximum value is negative, zero is used instead. In short, $\lambda(t)$ can be expressed as

$$\lambda(t) = \max\left(0, \max_{\forall \mathbf{x}_i \in \tilde{\mathbf{U}}} \left[\frac{F(\mathbf{x}_i) - \phi(t) F_{\mathrm{avg}}^{\mathrm{a},\tilde{\mathbf{F}}}}{E(\mathbf{x}_i)} \right] \right). \tag{10}$$

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

In this study, a modified bilinear scaling technique as shown in Fig. 1 is employed for fitness scaling. The minimum scaled fitness is set to be 0 to avoid negative fitness values while the scaled fitness of the average fitness of all feasible individuals is set to be 1. Furthermore, the maximum scaled fitness that is to be obtained from the best feasible members is set to be C. Thus, the chance of the best members being selected into the mating pool is equal to C times that of the average feasible members. All together, the scaled fitness can be written as

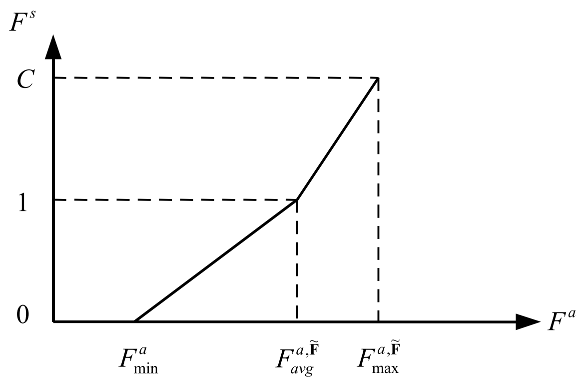

Fig. 1. Bilinear fitness scaling.

$$F^{s}(\mathbf{x}) = \frac{C-1}{F^{a,\tilde{\mathbf{F}}}_{max} - F^{a,\tilde{\mathbf{F}}}_{avg}} F^{a}(\mathbf{x}) + \frac{F^{a,\tilde{\mathbf{F}}}_{max} - CF^{a,\tilde{\mathbf{F}}}_{avg}}{F^{a,\tilde{\mathbf{F}}}_{max} - F^{a,\tilde{\mathbf{F}}}_{avg}} \quad \text{if } F^{a}(\mathbf{x}) \geqslant F^{a,\tilde{\mathbf{F}}}_{avg},$$
$$F^{s}(\mathbf{x}) = \frac{1}{F^{a,\tilde{\mathbf{F}}}_{avg} - F^{a}_{min}} F^{a}(\mathbf{x}) + \frac{F^{a}_{min}}{F^{a}_{min} - F^{a,\tilde{\mathbf{F}}}_{avg}} \quad \text{if } F^{a}(\mathbf{x}) \leqslant F^{a,\tilde{\mathbf{F}}}_{avg}, \tag{11}$$

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

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

$$F^{s}(\mathbf{x}_i) \leqslant (\varphi F^{s,\tilde{\mathbf{F}}}_{avg} = \varphi) \quad \text{for } \forall \mathbf{x}_i \in \widetilde{\mathbf{U}}, \tag{12}$$

where $F^{s,\tilde{\mathbf{F}}}_{avg}$ is the scaled value of the average fitness of all feasible individuals, which is equal to 1. Note that φ is constant for all generations. From the above condition, $\phi(t)$ in Eq. (9) is expressed in terms of φ as

$$\phi(t) = \frac{CF^{a,\tilde{\mathbf{F}}}_{avg} + F^{a,\tilde{\mathbf{F}}}_{max}(\varphi - 1) - \varphi F^{a,\tilde{\mathbf{F}}}_{avg}}{(C-1)F^{a,\tilde{\mathbf{F}}}_{avg}} \quad \text{for } \varphi \geqslant 1, \tag{13a}$$

$$\phi(t) = \frac{F^{a}_{min} + \varphi F^{a,\tilde{\mathbf{F}}}_{avg} - \varphi F^{a}_{min}}{F^{a,\tilde{\mathbf{F}}}_{avg}} \quad \text{for } \varphi \leqslant 1. \tag{13b}$$

In real calculations, 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. This physical meaning is directly related to the degree of penalty. In addition, the coefficient does not have any units. Due to these reasons, it is possible to set this coefficient by using experience. After φ is set, $\phi(t)$ and, subsequently, $\lambda(t)$ can be computed. In case of $\varphi \geqslant 1$, $\phi(t)$ can be obtained from Eq. (13a) directly because all parameters in the equation are readily available. In this case, the parameters $F^{a,\tilde{\mathbf{F}}}_{avg}$ and $F^{a,\tilde{\mathbf{F}}}_{max}$ can be obtained directly from original fitness values of feasible individuals without any penalty consideration. On the contrary, if $\varphi < 1$, $\phi(t)$ cannot be obtained without iteration since one of the parameters, i.e., F^{a}_{min}, is not readily available. Note that F^{a}_{min} is the minimum fitness in the generation after the penalty and it is most likely that F^{a}_{min} will belong to infeasible members. This F^{a}_{min} can be obtained from Eq. (7) which, in turn, requires the value of $\phi(t)$ (see Eq. (10)). Nevertheless, the iteration is very simple and takes almost no time to perform. To this end, the individual $\mathbf{x}_{F^{a}_{min}}$ that gives the minimum augmented fitness value is considered. Here, Eq. (7) yields

$$F^{a}_{min} = F^{a}\left(\mathbf{x}_{F^{a}_{min}}\right) = F\left(\mathbf{x}_{F^{a}_{min}}\right) - \lambda(t)E\left(\mathbf{x}_{F^{a}_{min}}\right). \tag{14}$$

Also, consider the individual $\mathbf{x}_{\lambda}$ that gives the value of $\lambda(t)$ in Eq. (10), i.e.,

$$\lambda(t) = \max\left(0, \max_{\forall \mathbf{x}_i \in \widetilde{\mathbf{U}}}\left[\frac{F(\mathbf{x}_i) - \phi(t)F^{a,\tilde{\mathbf{F}}}_{avg}}{E(\mathbf{x}_i)}\right]\right) = \frac{F(\mathbf{x}_{\lambda}) - \phi(t)F^{a,\tilde{\mathbf{F}}}_{avg}}{E(\mathbf{x}_{\lambda})}. \tag{15}$$

Using Eq. (15) in Eq. (14) gives

$$F^{a}_{min} = F\left(\mathbf{x}_{F^{a}_{min}}\right) - \left(\frac{F(\mathbf{x}_{\lambda}) - \phi(t)F^{a,\tilde{\mathbf{F}}}_{avg}}{E(\mathbf{x}_{\lambda})}\right)E\left(\mathbf{x}_{F^{a}_{min}}\right). \tag{16}$$

Substituting Eq. (16) into Eq. (13b) yields

$$\phi(t) = \left\{(\varphi - 1)E\left(\mathbf{x}_{F^{a}_{min}}\right)F(\mathbf{x}_{\lambda}) + E(\mathbf{x}_{\lambda})\left[F\left(\mathbf{x}_{F^{a}_{min}}\right) + \varphi F^{a,\tilde{\mathbf{F}}}_{avg} - \varphi F\left(\mathbf{x}_{F^{a}_{min}}\right)\right]\right\} \Big/ \left\{F^{a,\tilde{\mathbf{F}}}_{avg}\left[E(\mathbf{x}_{\lambda}) + (\varphi - 1)E\left(\mathbf{x}_{F^{a}_{min}}\right)\right]\right\}. \tag{17}$$

A problem is that the individuals $\mathbf{x}_{\lambda}$ and $\mathbf{x}_{F^{a}_{min}}$ are not known from the beginning and iteration is required. In the first step of the iteration, it is assumed that $F^{a}_{min} = F^{a,\tilde{\mathbf{F}}}_{avg}$. By using Eq. (13b), the intermediate value of $\phi(t)$ for this step of the iteration is obtained, i.e., $\phi(t) = 1$. After that, the intermediate value of $\lambda(t)$ is obtained from Eq. (10) and at the same time the individual $\mathbf{x}_{\lambda}$ can be identified. With the obtained $\lambda(t)$, the individual $\mathbf{x}_{F^{a}_{min}}$ can be subsequently identified from Eq. (7). Consequently, the value of $\phi(t)$ for the next step of the iteration is computed from Eq. (17). The process is repeated until the value of $\phi(t)$ becomes unchanging.

To be able to understand the proposed scheme better, let us consider an optimization problem of a uniaxial bar shown in Fig. 2. A uniaxial force of 10 lb is applied at the free end of the bar. Allowable stress is assumed to be 2 psi. Our task is to find the optimal area of the bar that yields minimum volume. For illustrative purpose, it is

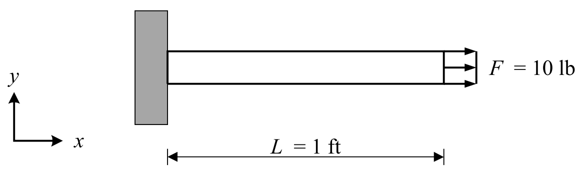

Fig. 2. Illustrative example—uniaxial problem.

assumed that the area of the bar is a continuous variable and, as a result, the optimal solution is simply equal to 5 in.^{2} Suppose that GAs are used to obtain the solution and the fitness function is defined as

$$F\ (\text{Area}) = \frac{1}{1 + \text{Volume (in.}^3)} = \frac{1}{1 + \text{Area (in.}^2) \times \text{Length (in.)}}. \tag{18}$$

From this fitness function, it is obvious that the smaller the area is, the larger the fitness value will be (see Fig. 3). Nevertheless, the area cannot be smaller than 5 in.^{2}; otherwise, the bar will violate the stress constraint. Therefore, fitness values of those individuals that violate the constraint have to be reduced. In this example, 19 individuals with different areas ranging from 1 to 19 in.^{2} are assumed (see Fig. 3). In the proposed penalty scheme, the average fitness of all feasible members $F_{\text{avg}}^{\text{a},\tilde{\mathbf{F}}}$ is calculated. If there are any individuals that have their fitness values exactly equal to $F_{\text{avg}}^{\text{a},\tilde{\mathbf{F}}}$, they are the average feasible members. Nevertheless, in real calculations, it does not matter whether there are any of them or not in the population since only the value of their fitness $F_{\text{avg}}^{\text{a},\tilde{\mathbf{F}}}$ is to be used. In the proposed scheme, infeasible members are penalized in such a way that the best infeasible members have scaled fitness values equal to φ times that of the average feasible members. Fig. 5 illustratively shows scaled fitness values after the penalty and scaling when $\varphi = 1.0$ and 1.5 while Fig. 4 shows fitness values just after the penalty but before the scaling. Note that, in this example, the maximum fitness is scaled to be 2.0 while the average fitness of feasible members is scaled to be 1.0. In addition, the minimum fitness is scaled to be 0. By adjusting the value of φ, the degree of penalty can be efficiently adjusted.

In fact, the purpose of the proposed 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 penalty is obtained throughout the evolution process. Consequently, the problem of too weak or too strong a penalty during different phases of the evolution is removed. Note that the relative scaled fitness values of the best feasible members and the average feasible members are set via fitness scaling (see Fig. 1). As a result, the relative chance of the best feasible members being selected into the mating pool compared with that of the best infeasible members can also be controlled. For example, when φ is set to be 1.0 in the current example, the chance of the best feasible members to be selected becomes two times that of the best infeasible members since, from the fitness scaling, the chance of the best feasible members is set to be two times that of the average feasible members.

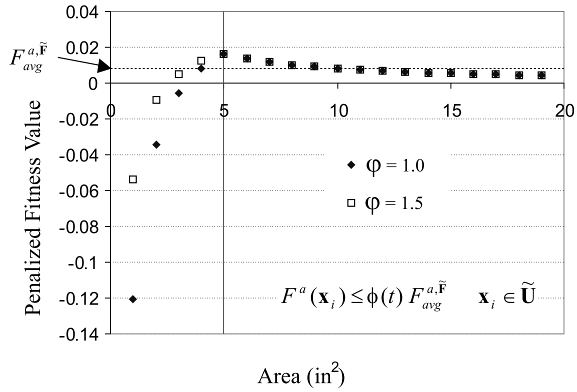

Fig. 4. Fitness value after the penalty—uniaxial problem.

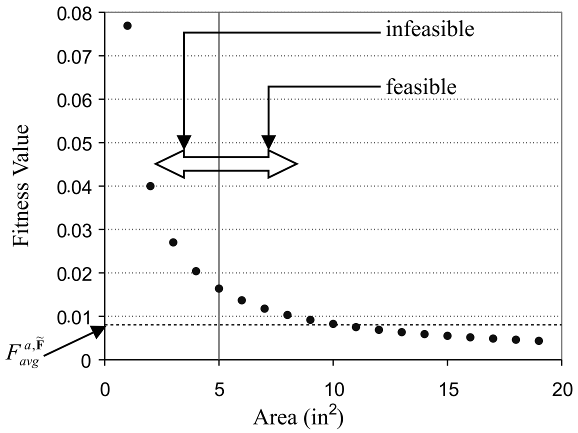

Fig. 3. Original fitness value—uniaxial problem.

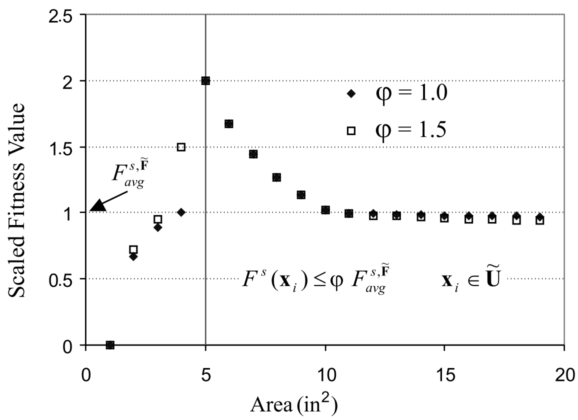

Fig. 5. Fitness value after the scaling—uniaxial problem.

In this study, since the fitness scaling in Fig. 1 is employed, acceptable values of φ therefore lie between 0 and C. Note that C is the scaled fitness of the best feasible individuals. Using only positive values for φ is obviously necessary because only positive scaled fitness values are acceptable. Setting φ exactly equal to zero is actually equivalent to using the death penalty scheme, which simply rejects infeasible solutions from the population. Using φ that is greater than C is in fact possible but it will mean that the best infeasible individuals will have a better chance to be selected into the mating pool than the best feasible ones. This is obviously too harsh a penalty. For this reason, the value of φ should not exceed C. For any value of φ between 0 and C, the best feasible individuals always have the maximum fitness value among all other individuals in the generation. Nevertheless, depending on the magnitude of φ, some infeasible members may have higher fitness than a certain number of feasible ones (see Fig. 5).

Actually, the key point in the development of the proposed scheme is that the user-specified penalty parameter φ is defined based on the relationship between two fitness values that are already scaled. Since scaled fitness values are directly used in the selection for the mating pool without further modification, the physical meaning of the proposed penalty parameter can be preserved. If penalty parameters are defined before the fitness scaling is performed, the fitness scaling will probably destroy the desired physical meanings of the parameters.

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 infeasible individuals will be given based on the magnitudes of error they have. The idea is to strongly encourage the population to move toward the feasible region. Here, a bilinear scaling scheme as shown in Fig. 6 is used. Fitness is scaled in such a way that scaled fitness values of individuals with the highest error are equal to 0 and scaled fitness values of individuals with average error are equal to 1. In addition, scaled fitness values of 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 average error. In summary, the scaled fitness is expressed as

$$F^{\mathrm{s}}(\mathbf{x}) = \frac{Z-1}{E_{\min}-E_{\mathrm{avg}}}E(\mathbf{x}) + \frac{E_{\min}-ZE_{\mathrm{avg}}}{E_{\min}-E_{\mathrm{avg}}} \quad \text{if } E(\mathbf{x}) \leqslant E_{\mathrm{avg}},$$
$$F^{\mathrm{s}}(\mathbf{x}) = \frac{1}{E_{\mathrm{avg}}-E_{\max}}E(\mathbf{x}) + \frac{E_{\max}}{E_{\max}-E_{\mathrm{avg}}} \quad \text{if } E(\mathbf{x}) > E_{\mathrm{avg}}. \tag{19}$$

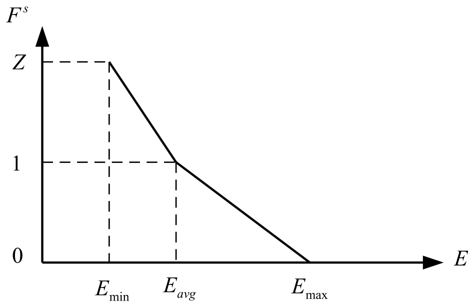

Fig. 6. Bilinear fitness scaling for the case when no feasible individual is available.

4. Results

To investigate the validity and efficiency of the proposed penalty scheme, the scheme is used in design optimization of three different two-dimensional structures, i.e., a six-bar truss, a ten-bar truss, and a one-bay eight-story frame. To be able to see clearly the advantages of the proposed scheme over conventional schemes, particularly in terms of robustness, obtained results are compared with those from a selected conventional scheme. Since most conventional schemes are based on the same concept with slightly different details, comparison with one selected conventional scheme is sufficient to show advantages of the proposed scheme over conventional schemes. As already mentioned, the main objective of this study is to develop an adaptive penalty scheme that is robust and still able to reproduce the same quality of results as ones obtained from GAs found in the literature. To show this comparison of the proposed method, results are also compared with existing results in the literature.

4.1. Six-bar truss

The first problem to be considered is the six-bar truss as shown in Fig. 7. Here, only sizing optimization is considered. Thus, 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, 3.84, 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 in.^{2} Therefore, a five-bit string is required for each design variable. There are two types of constraint in this problem, i.e., stress and displacement constraints. Design parameters used in the problem are shown in Table 1.

For comparison, the most popular conventional penalty form is selected, i.e.,

$$F_i^{\mathrm{a}} = F^{\mathrm{a}}(\mathbf{x}_i) = F(\mathbf{x}_i) - P(\mathbf{x}_i) = F(\mathbf{x}_i) - \lambda E(\mathbf{x}_i), \tag{20}$$

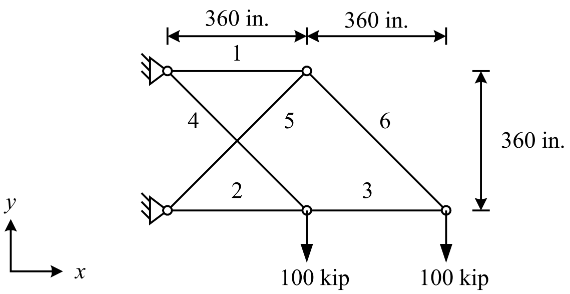

Fig. 7. Six-bar truss.

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

$$F(\mathbf{x}_i) = \frac{1}{1 + \text{Weight}(\mathbf{x}_i)}, \tag{21}$$

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 an appropriate value of the coefficient λ in the conventional scheme, a wide range of values will be used. All GA parameters used in this problem can be found in Table 1. To start the calculation, an initial population is generated at random. The type of crossover operator used here is the one-point crossover [15].

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
		Mutation probability	0.001
Maximum y-displacement	2 in.	φ	0.25–1.75
		λ	0.000001–100
		C	2.0
		Z	5.0

Fig. 8 shows 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. (21) are converted into pounds for comparison. In the conventional scheme, the coefficient λ is varied exponentially from 0.000001 to 100 while in the proposed scheme the coefficient φ 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 a trend in the results when φ is varied. It can be reasonably said that good results are obtained with values of φ around 0.75–1.0. On the contrary, it is much more difficult to observe this kind of trend in the results of the conventional scheme as they are very much scattered and exhibit no recognizable pattern. Although the trend in the results of the proposed scheme can be observed, it is also important to note that, when φ is varied, the results of the proposed scheme actually vary to a much lesser degree than the results of the conventional scheme do when λ is varied. Even though it may be argued that, in this study, λ is exponentially changed while φ is linearly changed, the same difference in the way that the parameters are varied and tried is expected in the real practice. This is because, in the real practice, it will also be impossible to estimate appropriate values of the coefficient λ in the conventional scheme, so a very wide range of values must be tested. With the proposed scheme, lesser sensitivity of results to the magnitude of the parameter ensures that even when the appropriate value of φ is not clearly known, a range of values of φ may be used and reasonable results can still be obtained. This fact really confirms the robustness of the proposed scheme.

To ensure that the proposed scheme is capable of giving results of the same quality as those GAs found in the literature, the best result obtained from the proposed scheme is also compared with the best result reported by Rajan [9]. They are exactly the same. The details of the results 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 Rajan [9], which employs more complicated GAs.

4.2. Ten-bar truss

The next problem to be considered is the ten-bar truss as shown in Fig. 9. This problem is one of the

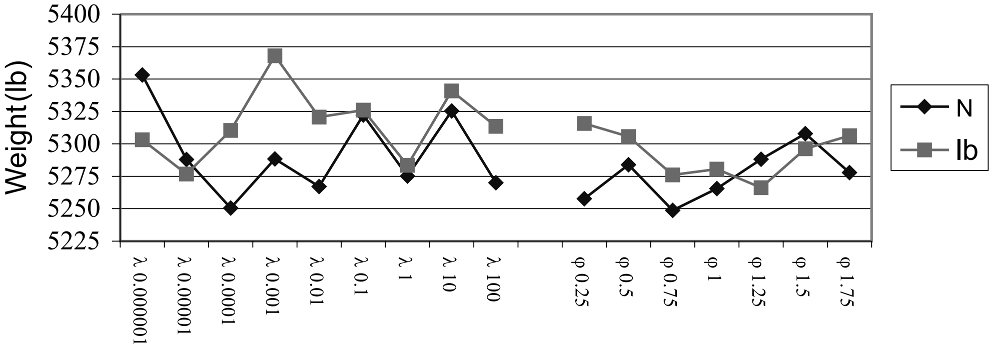

Fig. 8. Average weight of the best feasible designs obtained from 200 runs—six-bar truss.

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

Member	Size of member (in.^{2})	
	Proposed	Ref. [9]
1	30.0	30.0
2	19.9	19.9
3	15.5	15.5
4	7.22	7.22
5	22.0	22.0
6	22.0	22.0
Total weight (lb)	4962.1	4962.1

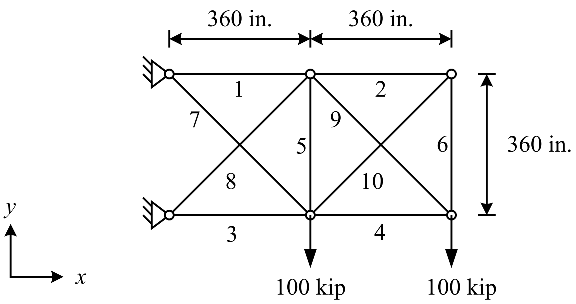

Fig. 9. Ten-bar truss.

benchmark problems used to test optimization methods. Also in this problem, only sizing optimization is considered. Therefore, design variables are ten sectional areas. 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 in.^{2} 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,

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 displacements	2 in.	Mutation probability	0.001
		φ	0.25–1.75
		λ	0.000001–100
		C	2.0
		Z	5.0

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 in.^{2} Similar to the previous problem, a five-bit string is required for each design variable. Design parameters and genetic parameters are shown in Table 3.

Results obtained from the proposed and conventional schemes are shown in Fig. 10. 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 a rather clear tendency with respect to the value of the coefficient used while those from the conventional scheme do not, and are quite scattered. In the proposed scheme, it can be reasonably said that good results are obtained with values of φ around 0.5–0.75. Similar to the previous problem, even though the trend in the

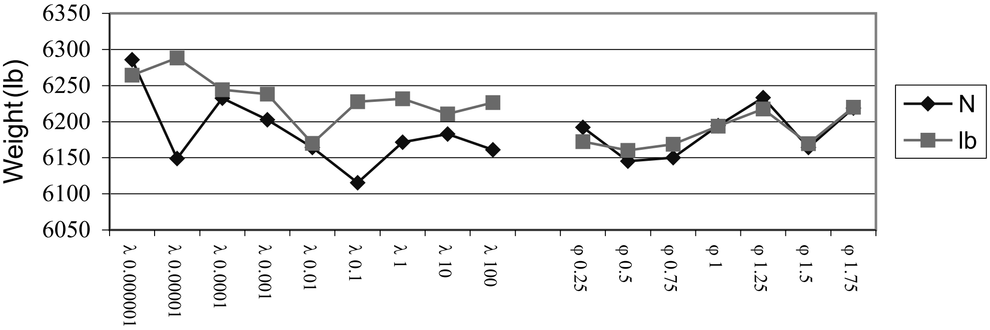

Fig. 10. Average weight of the best feasible designs obtained from 200 runs—ten-bar truss.

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

Member	Size of member (in.^{2})			
	Proposed	Ref. [19]	Ref. [21]	Ref. [29]
1	33.5	33.5	30.0	33.5
2	1.62	1.62	1.62	1.62
3	22.9	22.0	26.5	22.0
4	15.5	15.5	13.5	14.2
5	1.62	1.62	1.62	1.62
6	1.62	1.62	1.62	1.62
7	7.22	14.2	7.22	7.97
8	22.9	19.9	22.9	22.9
9	22.0	19.9	22.0	22.0
10	1.62	2.62	1.62	1.62
Total weight (lb)	5499.3	5613.8	5556.9	5458.3

results of the proposed scheme can be observed, the results are not that much sensitive to the magnitude of the penalty parameter when compared with the conventional scheme. Consequently, a range of values of φ may be used when the appropriate value is not known. The best result obtained from the proposed scheme is also compared with the best results reported by Rajeev and Krishnamoorthy [19], Camp et al. [21], and Galante [29] in Table 4. It can be seen that the result obtained from the proposed penalty scheme is relatively good although Rajeev and Krishnamoorthy [19], Camp et al. [21], and Galante [29] employ more complicated GAs.

In the previous six-bar truss problem, the appropriate value of φ is around 0.75–1.0, which is similar to the value obtained for the ten-bar truss problem. Since the two problems are quite similar, similar values of the coefficient from the two problems are expected. In this aspect, the proposed scheme evidently outperforms the conventional scheme, which does not exhibit any obvious similarity between these two problems. 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 penalty, the characteristics of the problems being solved can be directly related to the appropriate degree of penalty. This kind of advantage may not be found in existing conventional schemes.

4.3. One-bay eight-story frame

The last problem to be considered is the one-bay eight-story frame as shown in Fig. 11. Similar to the previous two problems, only sizing optimization is considered. The 24 members of the structure are categorized

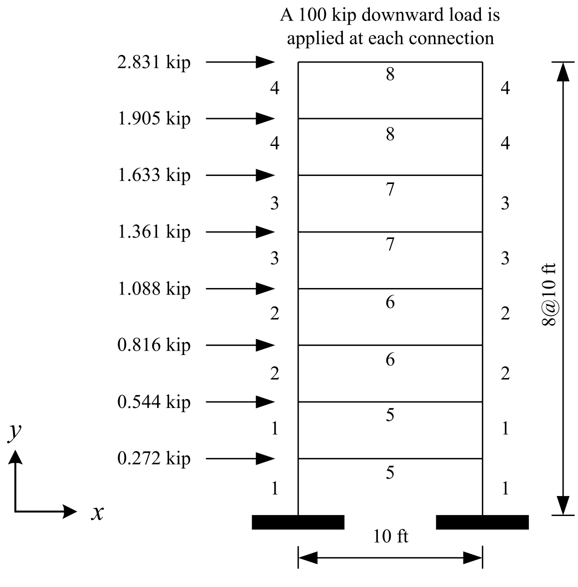

Fig. 11. One-bay eight-story frame.

Table 5
Design and GA parameters for the one-bay eight-story frame problem

Design parameters		GA parameters	
Item	Value	Item	Value
Modulus of elasticity	29×10^3 ksi	Maximum number of generations	100
Weight density	2.83×10^{-4} kip/in.^{3}		
Maximum x-displacement at the top of the structure	2 in.	Population size	50
		Crossover probability	0.85
		Mutation probability	0.05
		φ	0.25–1.75
		λ	0.000001–100
		C	2.0
		Z	5.0

Table 6
Comparison of the results for the one-bay eight-story frame problem

Group number	Proposed	GAs [21]	Optimality criteria [21]
1	W 12 × 45	W 18 × 46	W 14 × 34
2	W 14 × 34	W 16 × 31	W 10 × 39
3	W 12 × 35	W 16 × 26	W 10 × 33
4	W 10 × 19	W 12 × 16	W 8 × 18
5	W 18 × 35	W 18 × 35	W 21 × 68
6	W 18 × 40	W 18 × 35	W 24 × 55
7	W 16 × 36	W 18 × 35	W 21 × 50
8	W 16 × 26	W 16 × 26	W 12 × 40
Total weight (kip)	7.47	7.38	9.22

into eight groups (as indicated in Fig. 11). In this problem, 256 sections are selected from a list of 268 W-sections from the American Institute of Steel Construction Allowable Stress Design (AISC-ASD) specifications given in Ref. [30] by discarding the 12 biggest sections from the list. Thus, an eight-bit string is required for each design variable. There is only a displacement constraint in the problem that is the maximum x-displacement at the top of the structure. Design and genetic parameters are shown in Table 5.

Fig. 12 shows results obtained from the proposed and conventional schemes. In the figure, each point in the graph also represents an average weight of the best feasible designs obtained from 200 different runs. Once again, the robustness of the proposed scheme is confirmed. The effect of the unit on the results obtained from the proposed scheme is almost negligible. This conclusion is not true for the case of the conventional scheme, which exhibits large differences between the results from the two different units. In this problem, the insensitivity of the results to the value of the parameter is very apparent for the proposed scheme. On the contrary, the results from the conventional scheme show very high variation when the parameter is varied. This confirms the higher robustness of the proposed scheme over the conventional one. Although some of the averages of the best results from the conventional scheme shown in Fig. 12 may seem to be better than those from the proposed scheme, a comparison of the best result obtained from the proposed technique and results reported by Camp et al. [21] in Table 6 shows that the proposed method is actually acceptable. In their paper, Camp et al. [21] provide both results from their own GAs, which are not the standard GAs, and from the optimality criteria method [31].

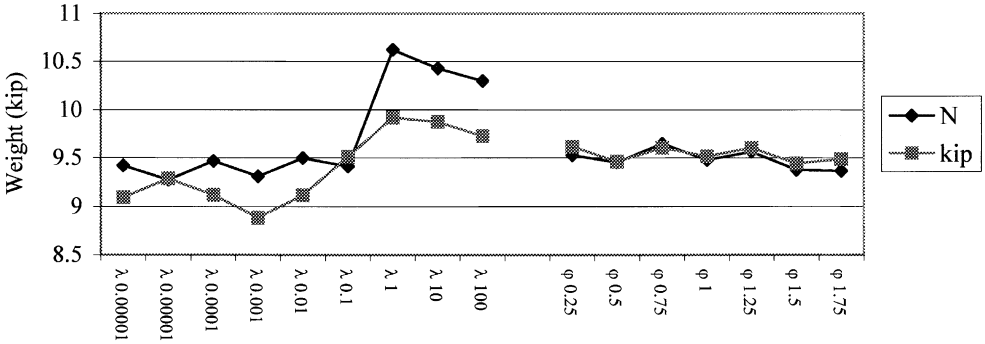

Fig. 12. Average weight of the best feasible designs obtained from 200 runs—one-bay eight-story frame.

5. Conclusion

This paper presents a new adaptive penalty scheme in GAs for structural design optimization. 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 appropriate values of these coefficients even by experience. Moreover, most existing 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. 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 penalty is achieved in all generations.

The proposed scheme is tested by using three optimization problems of truss and frame structures. 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 very robust. The results from the proposed scheme do not significantly depend on the units used. In addition, for some problems, it will be possible to observe trends in the results of the proposed scheme when the magnitude of the coefficient is varied. Moreover, it can be expected that, if similar problems are considered, appropriate values of the coefficient will be similar. All these characteristics encourage setting the value of the coefficient by experience. It is also observed that the results of the proposed scheme do not exhibit high fluctuation when the penalty coefficient is varied. As a result, even when appropriate values of the coefficient are not clearly known, a range of values may be used and reasonable results can still be obtained. Finally, comparisons with 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.

Acknowledgements

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

References

[1] Qian L, Zhong W, Sui Y, Zhang J. Efficient optimum design of structures-program DDDU. Comput Meth Appl Mech Eng 1982;30:209–24.
[2] Templeman AB. Discrete optimum structural design. Comput Struct 1988;30(3):511–8.
[3] Hall SK, Cameron GE, Grierson DE. Least-weight design of steel frameworks accounting for *P*–*Δ* effects. J Struct Eng 1989;115(6):1463–75.
[4] Adeli H, Park HS. A neural dynamics model for structural optimization-theory. Comput Struct 1995;57(3):383–90.
[5] Tzan S, Pantelides CP. Annealing strategy for optimal structural design. J Struct Eng 1996;122(7):815–27.
[6] Deb K. Optimization for engineering design: algorithms and examples. New Delhi: Prentice-Hall; 1995.
[7] Rao SS. Engineering optimization. New York: Wiley; 1996.
[8] Marcelin JL, Trompette P, Dornberger R. Optimization of composite beam structures using a genetic algorithm. Struct Optimiz 1995;9:236–44.
[9] Rajan SD. Sizing, shape, and topology design optimization of trusses using genetic algorithm. J Struct Eng 1995;121(10):1480–7.
[10] Ramasamy JV, Rajasekaran S. Artificial neural network and genetic algorithm for the design optimization of industrial roofs—a comparison. Comput Struct 1996;58(4):747–55.
[11] Kallassy A, Marcelin JL. Optimization of stiffened plates by genetic search. Struct Optimiz 1997;13:134–41.
[12] Rafiq MY, Southcombe C. Genetic algorithms in optimal design and detailing of reinforced concrete biaxial columns supported by a declarative approach for capacity checking. Comput Struct 1998;69:443–57.
[13] Shrestha SM, Ghaboussi J. Evolution of optimum structural shapes using genetic algorithm. J Struct Eng 1998; 124(11):1331–8.
[14] Marcelin JL. Evolutionary optimisation of mechanical structures: towards an integrated optimisation. Eng Comput 1999;15:326–33.
[15] Goldberg DE. Genetic algorithms in search, optimization, and machine learning. Reading, Massachusetts: Addison-Wesley; 1989.
[16] Grefenstette JJ. Optimization of control parameters for genetic algorithms. IEEE Trans Syst, Man, Cybernet 1986;SMC-16(1):122–8.
[17] Dawid H. Adaptive learning by genetic algorithms: analytical results and applications to economic models. New York: Springer; 1999.
[18] Michalewicz Z. Genetic algorithms + data structures = evolution programs. New York: Springer; 1996.
[19] Rajeev S, Krishnamoorthy CS. Discrete optimization of structures using genetic algorithms. J Struct Eng 1992;118(5):1233–49.
[20] Jenkins WM. On the application of natural algorithms to structural design optimization. Eng Struct 1997;19(4):302–8.
[21] Camp C, Pezeshk S, Cao G. Optimized design of two-dimensional structures using a genetic algorithm. J Struct Eng 1998;124(5):551–9.

[22] Adeli H, Cheng N. Integrated genetic algorithm for optimization of space structures. J Aerosp Eng 1993; 6(4):315–28.

[23] Michalewicz Z. Genetic algorithms, numerical optimization, and constraints. In: Eshelman LJ, editor. Proceedings of the sixth international conference on genetic algorithms. San Francisco: Morgan Kaufmann; 1995.

[24] Powell D, Skolnick MM. Using genetic algorithms in engineering design optimization with non-linear constraints. In: Forrest S, editor. Proceedings of the fifth international conference on genetic algorithms. San Mateo: Morgan Kaufmann; 1993.

[25] Le Riche RG, Knopf-Lenoir C, Haftka RT. A segregated genetic algorithm for constrained structural optimization. In: Eshelman LJ, editor. Proceedings of the sixth international conference on genetic algorithms. San Francisco: Morgan Kaufmann; 1995.

[26] Rasheed KM. GADO: a genetic algorithm for continuous design optimization (doctoral dissertation). New Brunswick, New Jersey: The State University of New Jersey; 1998.

[27] Coello CAC. Use of a self-adaptive penalty approach for engineering optimization problems. Comput Industry 2000;41:113–27.

[28] Rajeev S, Krishnamoorthy CS. Genetic algorithms-based methodologies for design optimization of trusses. J Struct Eng 1997;123(3):350–8.

[29] Galante M. Genetic algorithms as an approach to optimize real-world trusses. Int J Numer Meth Eng 1996;39:361–82.

[30] Burns T. Structural steel design—LRFD. New York: Delmar; 1995.

[31] Khot NS, Venkayya VB, Berke L. Optimum structural design with stability constraints. Int J Numer Meth Eng 1976;10:1097–114.