Lecture Notes in Artificial Intelligence, Vol. 2358, 2002, pp. 119-128

Genetic Algorithms for Design of Liquid Retaining Structure

K.W. Chau[1] and F. Albermani[2]

[1]Department of Civil & Structural Engineering, Hong Kong Polytechnic University,
Hunghom, Kowloon, Hong Kong
cekwchau@polyu.edu.hk
[2]Department of Civil Engineering, University of Queensland, QLD 4072, Australia

Abstract. In this paper, genetic algorithm (GA) is applied to the optimum design of reinforced concrete liquid retaining structures, which comprise three discrete design variables, including slab thickness, reinforcement diameter and reinforcement spacing. GA, being a search technique based on the mechanics of natural genetics, couples a Darwinian survival-of-the-fittest principle with a random yet structured information exchange amongst a population of artificial chromosomes. As a first step, a penalty-based strategy is entailed to transform the constrained design problem into an unconstrained problem, which is appropriate for GA application. A numerical example is then used to demonstrate strength and capability of the GA in this domain problem. It is shown that, only after the exploration of a minute portion of the search space, near-optimal solutions are obtained at an extremely converging speed. The method can be extended to application of even more complex optimization problems in other domains.

1 Introduction

In solving practical problems of design optimization, owing to the availability of standard practical sizes and their restrictions for construction and manufacturing purposes, the design variables are always discrete. In fact, it is more rational to use discrete variables during the evaluation process of optimization since every candidate design is a practically feasible solution. This may not be the case when the design variables are continuous, since some of the designs evaluated in the optimization procedures are solely mathematically feasible yet not practically feasible. However, most of the programming algorithms developed for the optimum design of structural systems during the last few decades assume continuous design variables and simple constraints, which are not always correct. Only very few algorithms have dealt with the optimization of structures under the actual design constraints of code specifications. Many of them require the approximation of derivative information and yet may only attain local optima.

Recently, there has been a widespread interest in the use of genetic algorithms (GAs), which are applications of biological principles into computational algorithm, to accomplish the optimum design solutions [1]. They are specifically appropriate to solve discrete optimum design problems and apply the principle of survival of the fittest into the optimization of structures. Although GAs require only objective function value to direct the search and do not need any derivatives of functions often necessary in other optimization programming methods, they are able to search through large spaces in a short duration. In particular, GAs have a much more global perspective than many other methods. Yet, literature review shows that only steel structures are often considered in structural optimization field [2-5].

This paper delineates a genetic algorithm (GA) for the optimum design of reinforced concrete liquid retaining structures using discrete design variables. Whereas either minimum weight or minimum cost can represent the objective function equally in the optimization of steel structures, their counterparts of reinforced concrete structures must be minimum material cost. Reinforced concrete structures involve more design variables as it involves both concrete and steel reinforcement, which have rather different unit costs.

2 Genetic Algorithms

GAs, being search techniques based on the mechanism of natural genetics and natural selections [6], can be employed as an optimization method so as to minimize or maximize an objective function. They apply the concept on the artificial survival of the fittest coupled with a structured information exchange using randomized genetic operators taken from the nature to compose an efficient search mechanism. GAs work in an iterative fashion successively to generate and test a population of strings. This process mimics a natural population of biological creatures where successive generations of creatures are conceived, born, and raised until they are ready to reproduce.

2.1 Comparisons with Conventional Algorithms

GAs differ from traditional optimization algorithms in many aspects. The following are four distinct properties of GAs, namely, population processing, working on coded design variables, separation of domain knowledge from search, and randomized operators. Whilst most common engineering search schemes are deterministic in nature, GAs use probabilistic operators to guide their search. Whereas other optimization methods often need derivative information or even complete knowledge of the structure and parameters, GAs solely entail objective function value information for each potential solution they generate and test. The population-by-population approach of GA search climbs many peaks in parallel simultaneously whilst the more common point-by-point engineering optimization search techniques often locates false local peaks especially in multi-modal search spaces.

2.2 Features of GAs

GAs are not limited by assumptions about search space, such as continuity or existence of derivatives. Through a variety of operations to generate an enhanced population of strings from an old population, GAs exploit useful information subsumed in a population of solutions. Various genetic operators that have been identified and used in GAs include, namely, crossover, deletion, dominance, intra-chromosomal duplication, inversion, migration, mutation, selection, segregation, sharing, and translocation.

2.3 Coding Representation

A design variable has a sequence number in a given discrete set of variables in GAs, which require that alternative solutions be coded as strings. Successive design entity values can be concatenated to form the length of strings. Different coding schemes have been used successfully in various types of problems. If binary codes are used for these numbers, individuals in a population are finite length strings formed from either 1 or 0 characters. Individuals and the characters are termed chromosomes and artificial genes, respectively. A string may comprise some substrings so that each substring represents a design variable.

2.4 Selection Operator

The purpose of selection operator is to apply the principle of survival of the fittest in the population. An old string is copied into the new population according to the fitness of that string, which is defined as the non-negative objective function value that is being maximized. As such, under the selection operator, strings with better objective function values, representing more highly fit, receive more offspring in the mating pool. There exist a variety of ways to implement the selection operator and any one that biases selection toward fitness can be applicable.

2.5 Crossover Operator

The crossover operator leads to the recombination of individual genetic information from the mating pool and the generation of new solutions to the problem. Several crossover operators exist in the literature, namely, uniform, single point, two points and arithmetic crossover.

2.6 Mutation Operator

The mutation operator aims to preserve the diversification among the population in the search. A mutation operation is applied so as to avoid being trapped in local optima. This operator is applied to each offspring in the population with a predetermined probability. This probability, termed the mutation probability, controls the rate of mutation in the process of selection. Common mutation operation is simple, uniform, boundary, non-uniform and Gaussian mutation.

3 Problem Formulation

The following depicts the optimization design of a reinforced concrete liquid retaining structure, subjected to the actual crack width and stress constraints in conformance to the British Standard on design of concrete structures for retaining aqueous liquid, BS 8007 [7]. The set of design variables is determined so that the total material cost of the structure comprising n groups of member,

$$\min C(x) = \sum_{i}^{n} U_i * V_i + R_i * W_i \tag{1}$$

is minimized subject to the constraints. In eq. (1), U_i and V_i represent the unit cost and the concrete volume of member i respectively. R_i and W_i are the unit cost and the weight of steel reinforcement of member i respectively.

The serviceability limit state or crack width constraint is

$$W_a - W_{\max} \le 0 \tag{2}$$

where W_a is the actual crack width and W_{max} is the prescribed maximum crack width, which will be 0.1mm or 0.2 mm depending on the exposure environment. W_a, is determined using the following formula:-

$$W_a = \frac{3a_{cr}\varepsilon_m}{1 + 2(\frac{a_{cr} - c}{h - x})} \tag{3}$$

where a_{cr} is the distance from the point considered to the surface of the nearest longitudinal bar, ε_m is the average strain for calculation of crack width allowing for concrete stiffening effect, c is the minimum cover to the tension reinforcement, h is the overall depth of the member and x is the depth of the neutral axis.

The stress constraints, representing the ultimate limit states of flexure and shear resistance, are expressed in terms of the following equations for members subject to bending and shear force [8]:

$$M_{au} - M_{ult} \le 0 \tag{4}$$

$$V_a - V_{ult} \le 0 \tag{5}$$

where M_{au} is the actual ultimate bending moment, M_{ult} is the nominal ultimate moment capacity of the reinforced concrete section, V_a is the actual ultimate shear force and V_{ult} is the nominal ultimate shear capacity of the section. The ultimate moment capacity is determined by the following equations, depending on whether concrete or steel stresses is more critical.

$$M_{ult} = \frac{F_y}{1.15} A_s Z \text{ or } M_{ult} = 0.157 F_{cu} b d^2 \text{ whichever is the lesser} \tag{6}$$

where F_y is the yield strength of reinforcement, A_s is area of tension steel, Z is the lever arm, F_{cu} is the characteristic concrete strength, b is the width of section and d is the effective depth of section. Ultimate shear capacity of the section ($V_{ult} = v_c b_v d$) is represented by shear strengths v_c for sections without shear reinforcement, which depend upon the percentage of longitudinal tension reinforcement $[100A_s/(b_v d)]$ and the concrete grade:-

$$v_c = 0.79[100 A_s /(b_v d)]^{1/3} (400/d)^{1/4} / \gamma_m \tag{7}$$

where b_v is breadth of section, γ_m is a safety factor equal to 1.25, with limitations that $[100A_s/(b_v d)]$ should not be greater than three and that (400/d) should not be less than one. For characteristic concrete strengths greater than 25 N/mm^2, the values given by the above expression is multiplied by $(F_{cu}/25)^{1/3}$.

Prior to applying GA, a transformation on the basis of the violations of normalized constraints [9] is employed to change the constrained problem to become unconstrained. The normalized form of constraints is expressed as:

$$\frac{W_a}{W_{\max}} - 1 \le 0 \quad \text{i=1,…..n} \tag{8}$$

$$\frac{M_{au}}{M_{ult}} - 1 \le 0 \quad \text{i=1,…..n} \tag{9}$$

$$\frac{V_a}{V_{ult}} - 1 \le 0 \quad \text{i=1,…..n} \tag{10}$$

The unconstrained objective function φ(x) is then written as

$$\varphi(x) = C(x)[1 + K \sum_i^n \{(\frac{W_a}{W_{\max}} - 1)^+ + (\frac{M_{au}}{M_{ult}} - 1)^+ + (\frac{V_a}{V_{ult}} - 1)^+\}] \tag{11}$$

where K is a penalty constant and

$$(\frac{W_a}{W_{\max}} - 1)^+ = \max(\frac{W_a}{W_{\max}} - 1{,}0) \tag{12}$$

$$(\frac{M_{au}}{M_{ult}} - 1)^+ = \max(\frac{M_{au}}{M_{ult}} - 1{,}0) \tag{13}$$

$$(\frac{V_a}{V_{ult}} - 1)^+ = \max(\frac{V_a}{V_{ult}} - 1{,}0) \tag{14}$$

The penalty parameter largely depends upon the degree of constraint violation, which is found to be amenable to a parallel search employing GAs. Values of 10 and 100 have been attempted here and it is found that the results are not quite sensitive.

In order to ensure that the best individual has the maximum fitness and that all the fitness values are non-negative, the objective function is subtracted from a large constant for minimization problems. The expression for fitness here is

$$F_j = [\varphi(x)_{\max} + \varphi(x)_{\min}] - \varphi_j(x) \tag{15}$$

where F_j is the fitness of the j-th individual in the population, $\varphi(x)_{max}$ and $\varphi(x)_{min}$ are the maximum and minimum values of $\varphi(x)$ among the current population respectively and $\varphi_j(x)$ is the objective function value computed for the j-th individual. The calculation of the fitness of an individual entails the values of crack width and stresses, which are obtained from the finite element structural analysis.

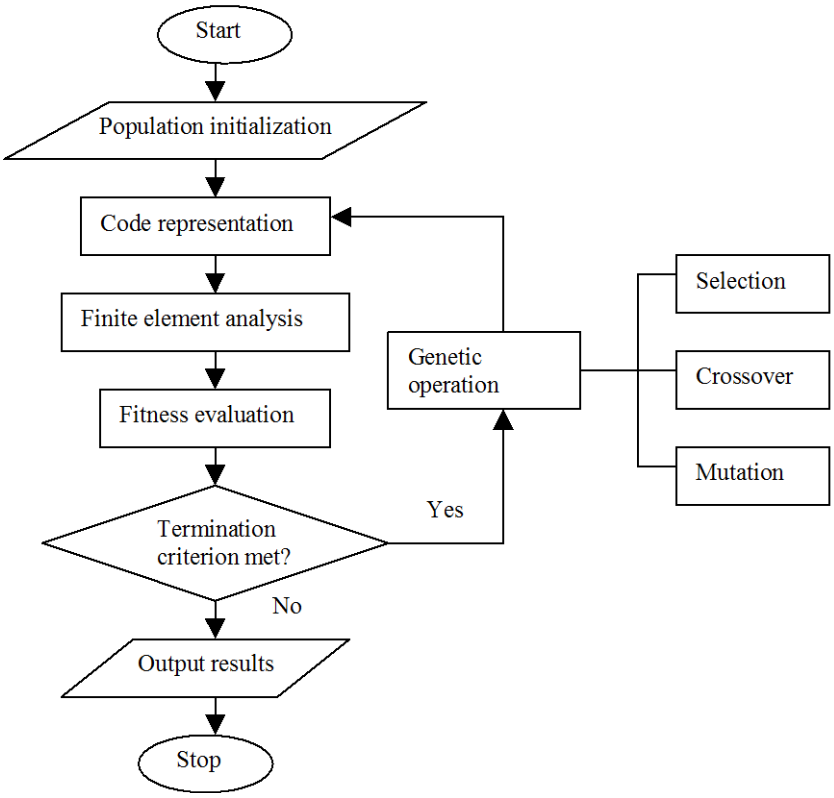

Fig. 1. Flow chart of the GA for design of liquid retaining structures

4 GA Optimization Procedure

In this study, the GA has been implemented under Microsoft Visual Basic programming environment. Figure 1 shows the flow chart of the GA for design of liquid retaining structures. The initial population containing binary digits is first randomly generated. The binary codes for the design variables of each individual are decoded and their sequence numbers are determined. The finite element analysis of the structure is performed and the unconstrained function φ(x) for each individual is computed. From the maximum and minimum values of this function in the population, the fitness value for each individual is then found.

The population for the next generation is then reproduced by applying the selection operator. On the basis of their fitness, that is, the best strings make more copies for mating than the worst, the individuals are copied into the mating pool. In order to emulate the survival of the fittest mechanism, a rank-based scheme is employed to facilitate the mating process [10]. The chromosomes are ranked in descending order of their fitness and those with higher fitness values have a higher probability of being selected in the mating process. As the number of individuals in the next generation remains the same, the individuals with small fitness die off. Besides, the concept of elitism is also incorporated into the selection process [11]. This strategy keeps the chromosome with the highest fitness value for comparison against the fitness values of chromosomes computed from the next generation. If the ensuing generation fails to enhance the fitness value, the elite is reinserted again into the population.

A two-site crossover is employed to couple individuals together to generate offspring. It involves the random generation of a set of crossover parameters, comprising a match and two cross sites. A match is first allocated for each individual. A fixed crossover probability ($p_{crossover}$) is established beforehand so that the genetic operation of crossover is performed on each mated pair with this probability. It is necessary to find the crossover sites and to perform the crossover in the following manner. Suppose that two strings X and Y of length 11 are the mating pair with the following genes

$$X = x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11} \tag{16}$$

$$Y = y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}, y_{11} \tag{17}$$

For each mated pairs subjected to crossover operation, two cross sites cs_1 and cs_2 are randomly generated. Two new strings are created by swapping all characters between positions cs_1 and cs_2 inclusively from one individual in the pair to the other. The cross sites for the matching pairs have to be the same. For instance, if the cross sites generated are 2 and 7, the resulting crossover yields two new strings X' and Y' following the partial exchange.

$$X' = x_1, x_2, | y_3, y_4, y_5, y_6, y_7, | x_8, x_9, x_{10}, x_{11} \tag{18}$$

$$Y' = y_1, y_2, | x_3, x_4, x_5, x_6, x_7, | y_8, y_9, y_{10}, y_{11} \tag{19}$$

The genetic operation of mutation is performed on the chromosomes with a preset mutation probability. It is applied to each offspring in the newly generated population. It operates by flipping the gene of an offspring from 1 to 0 or vice versa at random position. After the operation, the initial population is replaced by the new population.

An iteration cycle with the above steps is then continued until the termination criterion is reached. It may occur when the distance between the maximum and the average fitness values of the current population becomes less than a certain threshold, or a preset maximum number of generation is attained. At that moment, the optimum design solution is represented by the individual with the maximum fitness value in the current population.

Table 1. Composition of the individual string

Variable number	Design variable	Substring length
1	Slab thickness	4
2	Reinforcement diameter	3
3	Reinforcement spacing	4

5 Numerical Example

The test case is concerned with the optimum design of an underground circular shaped liquid retaining structure. The volume and height are 100 m^3 and 5 m, respectively. The exposure condition is very severe so that the designed crack width equals to 0.1 mm. The grades of concrete and reinforcement are 40 and high yield deformed bar respectively. The concrete cover is 40 mm and the aggregate type is granite or basalt with a temperature variation of 65 °C in determining the coefficient of expansion for shrinkage crack computation. The unit costs of concrete and reinforcement are \$500 per m^3 and \$3000 per tonne, respectively. A surcharge load of 10 kN/m^2 is specified. The level of liquid inside the tank, the ground level and the level of water table above the bottom of the tank are 5 m, 5 m and 4 m, respectively. The specific weight of soil is 20 kN/m^2 with active soil pressure coefficient of 0.3. Since it is an underground structure and the wind load is not applicable. Load combinations for both serviceability and ultimate limit states are in compliance with BS 8110.

As a demonstration, the wall and slab sections are classified under the same member group. The practically available values of the design variables are given in the lists T, D and S, representing slab thickness, bar diameter and bar spacing respectively.

$$T = (200, 225, 250, 300, 350, 400, 450, 500, 600, 700, 800, 900, 1000) \tag{20}$$

$$D = (10, 12, 16, 20, 25, 32, 40). \tag{21}$$

$$S = (100, 125, 150, 175, 200, 225, 250, 275, 300) \tag{22}$$

Since GAs work on coded design variables, now it is necessary to code the design variables into a string. Owing to the simplicity of manipulation, a binary coding is adopted here. Table 1 shows the composition of an individual string with a total length of eleven. The population size (P_{size}), the crossover probability and the mutation probability are selected as 10, 0.95 and 0.01, respectively. The population size is chosen to provide sufficient sampling of the decision space yet to limit the computational burden simultaneously. It is found that GAs are not highly sensitive to these parameters. These values are also consistent with other empirical studies.

Figure 2 shows the relationship between the minimum cost versus the number of generations for both this GA (modified GA) and the original GA used in [1] (simple GA), representing an average of 20 runs of the algorithms. It can be seen that this modified GA is slightly better than the simple GA for this numerical example. The minimum cost of $37522, representing a reinforced concrete section of member thickness 250 mm with reinforcement diameter 25 mm at spacing 200 mm, is found after 6 generations. As the generations progress, the population gets filled by more fit individuals, with only slight deviation from the fitness of the best individual so far found, and the average fitness comes very close to the fitness of the best individual. It is recalled that each generation represents the creation of P_{size} = 10 strings where P_{size}* $p_{crossover}$ = 10*0.95 = 9.5 of them are new. It is noted that only a simple demonstration is shown here with the solution of this problem in just 6 generations. However, if the wall and slab sections are classified under different member groups, the application will become much more complicated.

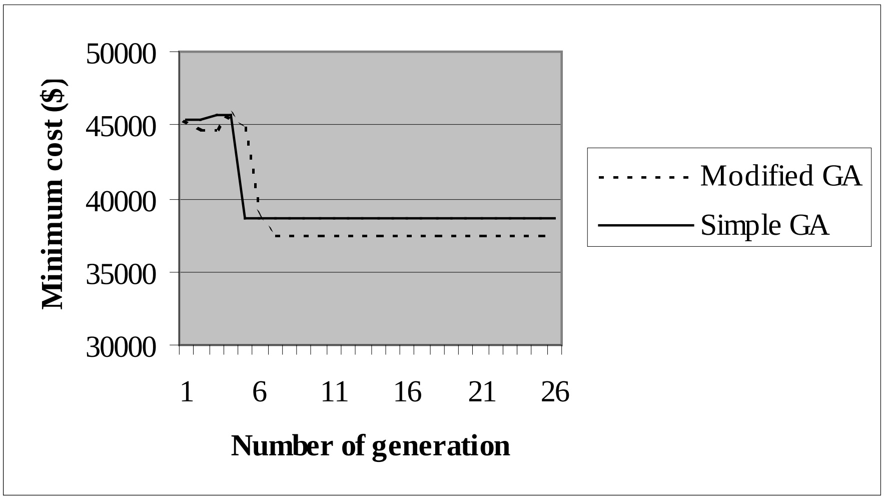

Fig. 2. Minimum cost versus number of generation for different GAs

6 Conclusions

In this paper, a GA has been successfully implemented for the optimum design of reinforced concrete liquid retaining structures involving discrete design variables. Only after examining a minute portion of the design alternatives, it is able to locate the optimal solution quickly. GAs acquire real discrete sections in a given set of standard sections. It should be noted that design variables are discrete in nature for most practical structural design problems. In most other mathematical programming techniques and in the optimality criteria approach, an approximation is often made by assigning the acquired optimum continuous design variables to the nearest standard sections. Approximate relationships may also be used between the cross-sectional properties of real standard sections in these usual methods. However, in this application, GAs remove these approximations and do not depend upon underlying continuity of the search space.

References

1. Goldberg, D.E., Kuo, C.H.: Genetic Algorithms in Pipeline Optimization. Journal of Computing in Civil Engineering ASCE **1(2)** (1987) 128-141
2. Adeli, H. and Cheng, N.T.: Integrated Genetic Algorithm for Optimization of Space Structures. Journal of Aerospace Engineering ASCE **6(4)** (1993) 315-328
3. Gutkowski, W., Iwanow, Z., Bauer, J.: Controlled Mutation in Evolutionary Structural Optimization. Structural and Multidisciplinary Optimization **21(5)** (2001) 355-360
4. Rajeev, S., Krishnamoorthy, C.S.: Genetic Algorithms-Based Methodologies for Design Optimization of Trusses. Journal of Structural Engineering ASCE **123(3)** (1997) 350-358
5. Yang, J., Soh, C.K.: Structural Optimization by Genetic Algorithms with Tournament Selection. Journal of Computing in Civil Engineering ASCE **11(3)** (1997) 195-200
6. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
7. British Standards Institution: BS 8007: Design of Concrete Structures for Retaining Aqueous Liquids. British Standards Institution, London (1987)
8. British Standards Institution: BS 8110: Structural Use of Concrete. British Standards Institution, London (1985)
9. Hayalioglu, M.S.: Optimum Load and Resistance Factor Design of Steel Space Frames using Genetic Algorithm. Structural and Multidisciplinary Optimization **21(4)** (2001) 292-299
10. Wu, S.J., Chow, P.T.: Steady-State Genetic Algorithms for Discrete Optimization of Trusses. Computers and Structures **56(6)** (1995) 979-991.
11. Koumousis, V.K., Georgiou, P.G.: Genetic Algorithms in Discrete Optimization of Steel Truss Roofs. Journal of Computing in Civil Engineering ASCE **8(3)** (1994) 309-325