



# Genetic algorithms in probabilistic finite element analysis of geotechnical problems

Lijie Cui, Daichao Sheng \*

*School of Engineering, The University of Newcastle, NSW 2308, Australia*

Received 14 June 2005; received in revised form 15 November 2005; accepted 21 November 2005

Available online 20 January 2006

---

## Abstract

In application to numerical analysis of geotechnical problems, the limit-state surface is usually not known in any closed form. The probability of failure can be assessed via the so-called reliability index. A minimization problem can naturally be formed with an implicit equality constraint defined as the limit-state function and optimization methods can be used for such problems. In this paper, a genetic algorithm is proposed and incorporated into a displacement finite element method to find the Hasofer–Lind reliability index. The probabilistic finite element method is then used to analyse the reliability of classical geotechnical systems. The performance of the genetic algorithm (GA) is compared with simpler probability methods such as the first-order-second-moment Taylor series method. The comparison shows that the GA can produce the results fairly quickly and is applicable to evaluation of the failure performance of geotechnical problems involving a large number of decision variables.

© 2006 Elsevier Ltd. All rights reserved.

**Keywords:** Genetic algorithms; Finite element method; Performance function; Probability of failure; FOSM; Hasofer–Lind reliability index

---

## 1. Introduction

The finite element method is increasingly used in the design of geotechnical systems. It can provide information on stability and displacements over time and, in many respects, is the most general method in geotechnical analysis and design. The finite element method currently used in practice is, however, largely a deterministic method that does not deal with the stochastic nature of design parameters. In reality, uncertainties of many types pervade the practice of geotechnical engineering, and estimation of geotechnical design parameters inevitably involves treatment of these substantial uncertainties and assessment of their implications on performance. Natural subsurface conditions may vary in space and time. Site characterisation is often based on limited information from sampling or boreholes. Models and methods used to predict the performance

of geotechnical systems are simplified representations of reality. Due to these uncertainties, one cannot guarantee that a design based on deterministic analysis using averaged values of the design parameters will perform successfully. While a sensitivity analysis provides valuable information, more insight is desirable. Increasingly, geotechnical engineers are being asked to quantify their degree of uncertainty by estimating a probability of failure. Therefore, there is a growing need for methods and tools that are able to incorporate the stochastic approach into geotechnical design for more realistic predictions of reliability. Methods are sought to account explicitly for the uncertainty and variability associated with soil parameters and to incorporate this uncertainty in geotechnical analysis.

Probabilistic methods are not new to geotechnical engineering [32,17,18,8]. Because of the significant amount of additional computational effort involved in combining stochastic approaches with advanced numerical methods, it is, however, not until very recently that these methods have emerged as a new tool for geotechnical engineering [30,23,12,11,6,7,19]. They provide not only a systematic

---

\* Corresponding author. Fax: +61 2 49216991.

E-mail address: [Daichao.Sheng@newcastle.edu.au](mailto:Daichao.Sheng@newcastle.edu.au) (D. Sheng).

evaluation of the uncertainties, but also a means for assessing the impact of these uncertainties on the likelihood of satisfactory performance of an engineering system. In the limited published research on probabilistic finite element methods, most applications have used simple soil models with typically one random parameter. Real geotechnical problems are characterised by complex soil behaviour, with large number of soil parameters, and by coupled physical processes like consolidation. With recent developments in efficient numerical algorithms, coupled with the rapid improvement of computer performance, probabilistic finite element methods that use more realistic soil and physical models are now feasible.

The genetic algorithm (GA) is one such probabilistic search method which is well suited to solving large combinatorial design problems. It has been widely used in various engineering optimization problems. For example, Simpson and Priest [29] demonstrated the application of a GA for identifying the maximum discontinuity frequency in a complex rock structure for three different problem sizes. Pal et al. [24] demonstrated the application of a GA in determining the material parameters when calibrating a constitutive model. Goh [10] incorporated a GA to search for the critical slip surface in multi-wedge stability analysis. The search was found to be sufficiently robust to handle layered soils with weak, thin layers and as efficient and accurate as the conventional pattern search method. Javadi et al. [14] used a GA to identify material parameters in a constitutive relationship describing the time dependency of air permeability of shotcrete tunnel lining. Deng and Lee [4] applied a GA for displacement back analysis of a steep slope at the Three Gorges Project site. McCombie and Wilkinson [21] applied a simple GA to the search for the minimum factor of safety in slope stability analysis using Bishop's simplified method, and showed that the GA can perform better than some traditional methods, such as a 'brute force' approach or a Monte-Carlo approach. Macari et al. [20] used a GA to find optimal material parameters in a constitutive model, and suggested that the use of a GA may overcome the limitations of having a large number of material parameters. Zolfaghari et al. [34] presented a GA to search for the critical non-circular failure surface in slope stability analysis and incorporated it into the Morgenstern-Price method to find the factor of safety for a variety of slope geometries and loading conditions. These applications indicate that the GA can be useful for solving various geotechnical problems where an appropriate (optimal) solution must be found from a large number of alternatives.

In the present paper, it is shown how a genetic algorithm can be incorporated into the displacement finite element method to evaluate the probability of failure using the Hasofer-Lind reliability index. The formulation for evaluating the probability failure is first outlined. A general framework of the genetic algorithm is then presented. The finite element method with the genetic algorithm is then used to study the reliability of classical geotechnical

systems, and its performance is compared with simpler methods such as those based on the first-order-second-moment Taylor series method.

## 2. Probabilistic finite element method

Traditionally, geotechnical analysis uses a 'factor of safety' to indicate the stability of a structure. In foundation analysis, for example, the ultimate collapse load that the foundation soil can support is divided by a factor of safety to give the allowable loading levels for design. The finite element method can be used to determine the collapse loads for complex geological conditions with nonlinear soil behaviour. Alternatively, the finite element method can be used to determine the actual displacement the structure will experience during its service life and this can be compared with the allowable displacement for the structure, resulting in a factor of safety on displacement. In either case, the factor of safety represents a blanket factor that implicitly includes all sources of variability and uncertainty inherent in the analysis. One of the most important, yet quantifiable, uncertainties is due to the variability of the soil and design parameters. In reality, these parameters are random variables that can be represented by some distribution of probability (i.e. probability density functions). Therefore, the outcome of the analysis should be interpreted in terms of probability instead of a deterministic factor of safety. The stability of the structure is then more meaningfully assessed by examining its probability of failure.

To assess the probability of failure of a geotechnical structure, a performance function can first be defined as:

$$G(\mathbf{x}) = R(\mathbf{x}) - S(\mathbf{x}),$$

where  $R$  represents the resistance to failure of the structure,  $S$  represents the action causing failure,  $G$  is the performance function with  $G > 0$  indicating satisfactory performance (stability),  $G < 0$  indicating unsatisfactory performance (failure) and the boundary defined by  $G = 0$  separating the stable and failure domains [18]. In the foundation example,  $R$  can be the ultimate collapse load or the allowable differential displacement and  $S$  can be the allowable load or the actual differential displacement, respectively. All these quantities are likely to be functions of a set of design and soil parameters  $\mathbf{x}$ . Without changing its meaning, it is also possible to define  $G$  as the ratio between  $R$  and  $S$ . However, such a change of definition of  $G$  is less preferable, because it is easier to work with the difference of two non-linear functions than with their ratio.

If the parameters  $\mathbf{x}$  are random variables, the probability of failure is then defined as:

$$P_f = P[G(\mathbf{x}) \leq 0] = \int_{G \leq 0} f(\mathbf{x}) d\mathbf{x}, \quad (1)$$

where  $f(x)$  is the probability density function of the vector formed by the variables  $\mathbf{x}$  (and is usually unknown).

## 2.1. Integration of the performance function

To determine the probability of failure, we usually need to determine the expected value and the standard deviation of the performance function. Methods such as Taylor's series, point estimation methods, and Monte-Carlo simulation are available for calculating the mean and standard deviation of the performance function. Once the expected values and standard deviation of the performance function have been determined, the reliability index  $\beta$ , which is assumed to be the number of standard deviations by which the expected value of a normally distributed performance function exceeds zero, can be calculated in the standard procedure of reliability analysis. The probability of failure can then be calculated using the cumulative distribution function of the standard normal distribution evaluated at  $-\beta$ .

The Taylor's series method is one of several methods to estimate the moments of a performance function based on moments of the input variables. It is based on a Taylor's series expansion of the performance function about, for example, the expected values of the random variables. As only first-order (linear) terms of the series are retained and only the first two moments (mean and standard deviation) are considered, the method is often termed the first-order, second-moment (FOSM) method [31]. The derivatives of the performance function with respect to a random variable can be estimated numerically by conducting finite element calculations using different values of the same random variable. Therefore, for  $n$  random variables, we need at least  $2n + 1$  calculations to approximate the first-order derivatives. Even though the Taylor's series method usually uses only the first-order terms of the series, in theory, it can be modified to include higher order terms. The main disadvantage of using higher-order terms is that the number of FE calculations is increased dramatically.

The point estimation method (PEM) is a procedure where probability distributions for continuous random variables are modelled by discrete equivalent distributions having two or more values. The elements of these discrete distributions (point estimates) have specific values with defined probabilities, such that the first few moments of the discrete distribution match those of the continuous random variables. As there are only a few values over which we need to integrate, the moments of the performance function are easily obtained. The point estimation method does not require the derivatives of the performance function.

The Taylor's series method and the PEM are simple to use, but do not simulate the random variation of the variables and hence their accuracy depends very much on various approximations. On the other hand, the Monte-Carlo method simulates the random variation of the variables, but requires a huge number of finite element calculations if there are low failure probabilities (often 10,000) and can only be realised using parallel computing techniques.

## 2.2. Hasofer and Lind reliability index

An alternative method is to determine the first-order reliability index according to [13, (H-L)]. The H-L reliability index can be related to the failure probability if all the variables are statistically independent and normally distributed

$$\beta = \min_{\mathbf{x}' \in F} \sqrt{\{\mathbf{x}'\}^T [\mathbf{C}]^{-1} \{\mathbf{x}'\}}, \quad (2)$$

$$x'_i = \frac{x_i - m_i}{\sigma_i}$$

where  $\mathbf{x}'$  is the vector representing the set of reduced random variables with zero mean and unit standard deviation,  $x_i$  is the  $i$ th random variable of  $\mathbf{x}'$ ,  $m_i$  is its mean,  $\sigma_i$  is its standard deviation,  $\mathbf{C}$  is the covariance matrix, and  $F$  is the failure domain defined by  $G(\mathbf{x}) < 0$ . Instead of trying to find the expected value and the standard deviation of the performance function, the H-L method endeavours to determine the minimum distance from the origin of the reduced coordinate system to the limit-state surface  $G(\mathbf{x}) = 0$  (see Fig. 1). Eq. (2) actually represents a typical optimisation problem with an implicit equality constraint defined as the limit state function

$$\text{Minimise } \beta \text{ subject to } G(\mathbf{x}) = 0. \quad (3)$$

In practical applications to geotechnical problems, the limit-state surface is usually not known in any closed form. Instead, it can be approximated by a number of discrete values obtained, for example, from repeated finite element analyses. The number of repeated finite element analyses is usually significant (e.g. 1000–10,000), and hence the cost of the probability analysis would be very high. To reduce this cost, a probabilistic search algorithm is introduced in the following section.

## 3. Genetic algorithms

Genetic algorithms (GA) are a class of search algorithms that simulate the process of natural selection and adaptation inspired by Darwin's theory. They are also referred to as stochastic optimization techniques and have been

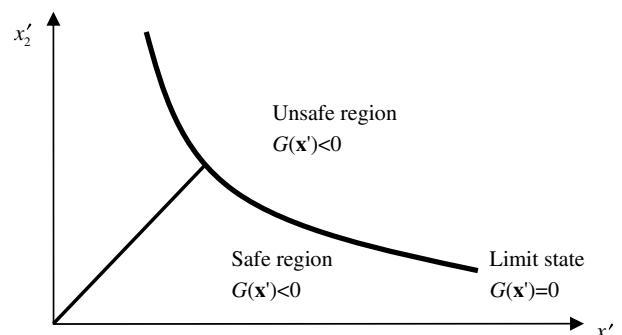


Fig. 1. Hasofer–Lind reliability index: performance function with two random variables.

shown to work well in many different disciplines. Stochastic optimization designates a family of optimization techniques in which the solution space is searched by generating candidate solutions with the help of a pseudo-random number generator. Being a stochastic method, the GA does not need specific information to guide the search, and requires only an evaluation of the objective function value for each decision variable set in order to proceed. It typically works with a coding of the decision variables, not with the decision variables themselves. It searches simultaneously using a population of decision variable sets, not a single set of decision variables [9]. Many variations of the genetic algorithms have been proposed and in general, the following components are required to implement a GA [2]:

1. A representation of potential decisions for the problem.
2. An objective function that measures the aggregated consequence of the decision variables, which is usually expressed in terms of maximising or minimising a criterion.
3. A set of constraints which describes the feasible set of combinations for the decisions.
4. GA operators (normally crossover and mutation).

To apply a GA to an engineering problem, it is necessary to accommodate the GA paradigm. In keeping with genetics terminology, the decision space is referred to as the environment, the potential solutions to the optimisation problem are called chromosomes (or strings, solutions that represent a set of decision variables), and the total number of solutions is called the population size. One iteration of the optimisation process is called a generation ( $G$ ).

In general, reproduction, crossover, mutation, and an elitist method are the essence of a standard genetic algorithm. The GA proceeds by evaluating each set of decision variables in the population at each generation. Fig. 2 illustrates the generic framework for a basic genetic algorithm. The first step in a GA is to produce an initial population of  $p$  points (sets) randomly with each consisting of  $n$  decision variables in the solution space. These decision variables are usually encoded as strings of binary digits or real numbers. The objective function values (fitness) are then calculated for each set of decision variables. Two points are then ran-

domly picked from the existing population and compared using the objective function value. The one with the better objective function value will have a higher chance of survival. For minimisation problems, individuals with lower objective function values will have a higher probability of being selected for the next generation. The two selected points are then mated to produce a new point using the genetic operators of crossover and mutation. This prediction is repeated until a population of  $p$  new points are generated. A kind of elitist method is then used to determine the final points for the new generation  $G + 1$ . The new generation is thus a collection with better fitness values from both the newly generated  $p$  points and the ones in the  $G$ th generation. When the new generation is created, each individual's objective function value is then evaluated. Over a number of generation cycles, the GA guides the population in the search space toward solutions of improved performance, yielding solutions more and more concentrated in the vicinity of the optima. The GA terminates if some stopping criterion (for example, a specified number of evaluations of solutions, or improvement of the best solution over the successive generations) is satisfied.

### 3.1. Description of a customised genetic algorithm

The GA represents a class of evolutionary search algorithms. Many researchers have put forward different formulations and refinements to the GA method. Because there is considerable scope for customising the GA, this section presents the detailed GA operators described in Fig. 2 and the adaptations to the basic GA used in this study. A key feature of the customised GA is the aggressive use of crossover along with the use of two lesser-used genetic operators, inversion and a population replacement strategy, to avoid premature convergence.

#### 3.1.1. Representation of decisions

The first question to explore is how to represent solutions in the GA population. Or in other words, how should the solutions be encoded? The genetic algorithms require encoding schemes to transform the vectors of decision variables to a structure that permits genetic operations. In this study, the most commonly used encoding binary scheme, i.e. strings of binary 0 and 1 bits [9], is used. Each decision variable is encoded into a binary string of fixed length  $k$ . The integer of the decoded binary variable ranges from 0 to  $2^k - 1$  and can be mapped linearly to the variable range ( $A_i, B_i$ ). The entire population of such designs constitutes a generation.

#### 3.1.2. Reproduction

The next question regarding the GA implementation is how to select parents for crossover. Different reproduction methods, such as rank selection, roulette wheel selection, or tournament selection, can be used. As mentioned in Section 2, the assessment of the probability of failure of a geotech-

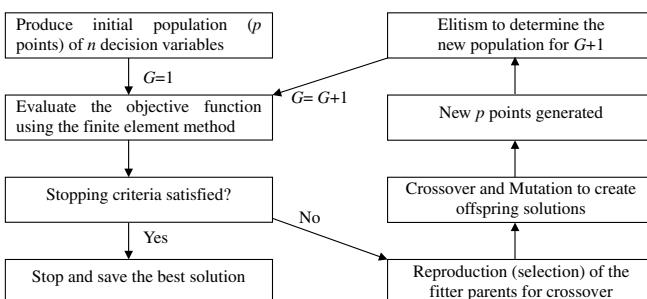


Fig. 2. A generic framework for a simple genetic algorithm.

nical structure can be viewed as a constrained optimization problem. A GA that uses only the traditional selection or operators would appear to be ineffective because of the limited power of generating feasible solutions [1]. Although there are many constraint-handling methods in the literature, most of them either require a large number of fitness function evaluations, complex encoding or mappings, or are limited to solving certain problems. Jimenez and Verdegay [15] and Deb and Agrawal [3] proposed a similar approach to handle constraints. The main idea is to use the tournament selection method to decide the solution according to the following three criteria:

1. If the two solutions are both feasible, then the selection is made according to the minimum value of the objective function ( $\beta$ ).
2. If a feasible solution pits against an infeasible solution, the feasible solution wins the tournament.
3. If both solutions are infeasible, the winner depends on the amount of constraint violation, and the one with the lower violation wins.

In this paper, a slight modification is made to the above procedure, i.e. if two infeasible solutions are compared, the one with the lower absolute value of  $G(\mathbf{x})$  will be chosen. The reason for this is that in this constrained problem, all the solutions that make  $G(\mathbf{x})$  approach zero need to be identified, from which the minimum  $\beta$  can be found.

### 3.1.3. GA operators

A crossover is the partial exchange of bits between two parent strings to form two offspring or child strings. Goldberg [9] described many different methods of performing the GA crossover operation. The most straightforward is the one point crossover. The crossover occurs if  $r < P_{\text{cross}}$  where  $r$  is a random number uniformly distributed between 0 and 1, and  $P_{\text{cross}}$  is a predefined probability for crossover. It begins by selecting two strings at random from the population using tournament selection. A crossover point  $L$  is randomly selected,  $2 \leq L \leq k - 1$ , where  $k$  is the length of the binary string. Two child strings are then created by swapping the parent substrings. Table 1(a) illustrates how the one point crossover operation works.

In this paper, traditional one-point crossover of binary strings is further modified [16]. Once the cut is made the offspring strings are swapped between the parents using either a direct (as shown in Table 1(a)) or an indirect swap (as

shown in Table 1(b)), another predefined probability  $P_{\text{swap}}$  is needed to decide if the indirect swap is used. The indirect swap occurs if  $r < P_{\text{swap}}$ , otherwise a direct swap is used. Table 1(b) illustrates the indirect diagonal swap operation. The indirect swap introduces more mixing than the traditional direct swap.

Mutation is simply an insurance policy against the irreversible loss of genetic material [9]. The GA considers each string bit-by-bit in the new generation formed as a result of reproduction and crossover. A probability of mutation,  $P_{\text{mutate}}$ , is usually selected to be quite small. For each bit of a binary string in the population, a random number,  $r$ , uniformly distributed between 0 and 1 is generated. If  $r < P_{\text{mutate}}$ , that bit is mutated by changing its value from either 0–1 or 1–0, as appropriate. This changes randomly the new offspring and helps to prevent all solutions in a population from converging to a local optimum prematurely.

Another GA operator is called inversion, and it is applied independently after the crossover and mutation. The idea behind inversion is to produce orderings in which beneficial genetic material is more likely to survive. It provides a type of non-destructive noise that helps crossover to escape local maxima [33]. For each string, if a random number is greater than the user defined probability  $P_{\text{inverse}}$ , the string remains unchanged, otherwise it is modified by the inversion operator. Under the inversion, two different bits along the decision string are chosen at random and swapped.

### 3.1.4. Replacement (elitism)

Once offsprings are produced, a decision has to be made about which of the current members of the population, if any, should be replaced by the new solutions. It forces several of the fittest members at each generation to survive until they are replaced by even fitter members. Note that one of the most important issues in genetic algorithms is the trade-off between exploitation and exploration [33]. Exploitation means that the decision variables from the already discovered promising search areas are exploited, while exploration refers to the promising new areas of the search space ready to be explored [9]. Traditional GA theory almost completely places the burden of exploitation and exploration on the crossover, forcing a trade-off between population diversity and selective pressure. Increasing the selective pressure tends to reduce diversity but increases search speed, whereas decreasing the selective pressure helps to maintain diversity but results in a slower, though more robust, search. Various replacement strategies have been proposed to deal with this issue. Eshelman and Schaffer [5] introduced a simple but suitable method, merging the child and parent populations and selecting the best individuals to make up the parent population for the next generation. This ensures the preservation of good genetic material from the parent population thereby allowing the crossover to concentrate on maintaining diversity. With this strategy in place a very high crossover probability

Table 1  
Illustration of genetic crossover

(a) Direct			
Parent string 1	1000 0101001	Offspring string 1	1000 1101001
Parent string 2	1111 1101001	Offspring string 2	1111 0101001
(b) Indirect			
Parent string 1	1000 0101001	Offspring string 1	1101001 1000
Parent string 2	1111 1101001	Offspring string 2	0101001 1111

$P_{\text{cross}} = 1$  can be used to vigorously explore the decision space so that there is no need to specify a value for  $P_{\text{cross}}$ . This method is adopted in this study.

### 3.2. The GA model

In summary the customised GA used in this study consists of the following steps:

1. An initial population of  $s$  designs, each consisting of  $n$  random variables ( $\mathbf{x}$ ), is randomly and uniformly sampled. The random variables  $x$  are encoded into binary strings. The finite element model evaluates the objective function for each initial decision variable. The objective function is defined as the minimum value of  $\beta$  when the performance function is zero, i.e.  $G(\mathbf{x}) = 0$ .
2. A tournament selection is applied to generate the decision variables for the next generation. The GA terminates if all the new solutions are the same as the solutions from the previous population.
3. A crossover is performed for each pair of new decision variables generated by the tournament selection. Mutations are carried out with low probability on a bit-by-bit basis in the strings. An inversion is then performed with probability  $P_{\text{invert}}$  for each variable.
4. The strings are then decoded into their real value equivalents and the objective function of the new design is evaluated.
5. The replacement method is used to determine the final members for the new generation from the child and parent populations.
6. Steps 2–5 are repeated until the stopping criteria are satisfied. The search stops if the best solution satisfies the tolerance of the performance function, or if the best solution does not improve over six successive generations.

## 4. Applications

### 4.1. Example 1

In order to test the accuracy of the genetic algorithm, we first consider a simple analytical problem with a known solution. The problem is defined as:  $G(\mathbf{x}) = x_1x_2 - 100$ , where  $x_1$  is a random variable with  $m = 10$ ,  $\sigma = 2$  and  $x_2$  is a random variable with  $m = 12$ ,  $\sigma = 4$ . The optimal solution to this problem is located at (10, 10) in the  $x_1 - x_2$  space. Using this optimal solution for  $x_1$  and  $x_2$  and Eq. (2), we can find the Hasofer–Lind reliability index,  $\beta = 0.5$ , for the two unrelated random variables ( $\mathbf{C} = \mathbf{I}$ ). On the other hand, the reliability index according to the FOSM method can be obtained with five discrete values.

$$G(x_1, x_2) = 20$$

$$G(x_1^+, x_2) = 44, \quad G(x_1^-, x_2) = -4$$

$$G(x_1, x_2^+) = 60, \quad G(x_1, x_2^-) = -20$$

with

$$x_1 = 10, \quad x_1^+ = 12, \quad x_1^- = 8$$

$$x_2 = 12, \quad x_2^+ = 16, \quad x_2^- = 8$$

The variance of the performance function and the reliability index can then be calculated as

$$\begin{aligned} \text{Var}(G) &= \left( \frac{G(x_1^+, x_2) - G(x_1^-, x_2)}{2\sigma_{x_1}} \right)^2 \sigma_{x_1}^2 \\ &\quad + \left( \frac{G(x_1, x_2^+) - G(x_1, x_2^-)}{2\sigma_{x_2}} \right)^2 \sigma_{x_2}^2 = 2176 \end{aligned}$$

$$\sigma_G = \sqrt{\text{Var}(G)} = 46.65$$

$$\beta = \frac{E(G)}{\sigma_G} = \frac{20}{46.65} = 0.43$$

The value of  $\beta$  based on the FOSM method is thus about 14% smaller than the Hasofer–Lind index.

The optimal solution in the genetic algorithm depends on the tolerance of function  $G = 0$ . For this simple problem, we can set a relatively stringent tolerance with an affordable GA search. For example, setting the tolerance to  $10^{-4}$ , will require 19 generations of GA computation and will lead to the exact solution of the Hasofer–Lind reliability index of 0.496, while the two decision variables  $x_1$  and  $x_2$  are found at (9.98, 10.01). Setting the tolerance to  $10^{-2}$ , will require only six generations of GA computation and will lead to a reliability index of 0.49 with the decision variables being located at (9.93, 10.06). Setting the tolerance to  $10^{-1}$ , will require only four generations of GA computation and will lead to a reliability index of 0.501 with the decision variables being located at (10.01, 9.99). Compared to the FOSM method using five discrete values to obtain the  $\beta$  value, the GA is able to find relatively consistent and accurate result to this simple analytical problem.

### 4.2. Example 2

In the second example, we study a simple strip footing problem. The problem geometry and the finite element mesh are shown in Fig. 3. The soil profile consists of two layers, with the upper layer having higher undrained shear strength. Both layers are simulated by the Tresca model for undrained loading. The soil properties are also given in Fig. 3. The design load for the strip footing is 150 kPa.

The finite element code, SNAC, developed at the University of Newcastle, is used to obtain the numerical solutions in this example as well as in the next example. SNAC contains a number of advanced numerical algorithms for various nonlinearities (see e.g. [28,27]), and has been verified for various elasto-plastic soil models (see e.g. [26,25]). In this and the next example, six-noded triangular elements are used to find the numerical collapse loads. This type of element has proved to give accurate predictions of the collapse load for plane strain problems [25,28].

Using the mean values of the undrained shear strength, we can find a numerical solution to the collapse load for the

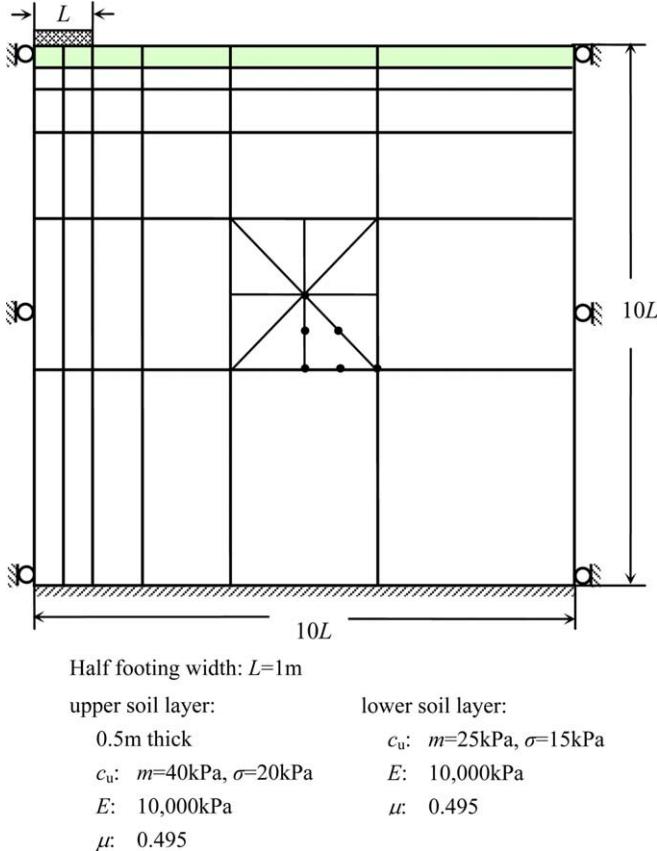


Fig. 3. Rigid strip footing: finite element mesh, boundary conditions and soil properties (each box consists of eight triangular elements. In total there are totally 288 elements, 625 nodes and 1143 degrees of freedom. Notation used in the figure:  $c_u$ : undrained shear strength,  $E$ : Young's modulus,  $\mu$ : Poisson's ratio,  $m$ : mean,  $\sigma$ : standard deviation).

footing, which is 161.5 kPa. Note this value is not guaranteed to be exact, but may contain various discretisation and linearisation errors in the finite element method. Nevertheless, we obtain a factor of safety around 1.08, which, according to engineering practice, is too small for bearing capacity safety. For this simple problem with only two random variables, we can also use the FOSM method and the finite element analysis to obtain a probability of failure. The performance function can be defined conveniently as:  $G(\mathbf{x}) = F(\mathbf{x}) - 150 = 0$ , in which  $F(\mathbf{x})$  is the numerical collapse load as a function of variables vector  $\mathbf{x}$  and 150 kPa is the design load. The two variables include the undrained shear strength  $c_{u1}$  and  $c_{u2}$  for the two soil layers. For simplicity, the two variables are assumed to be independently uncorrelated with normal distributions.

First, we run five finite element analyses using discrete values of the undrained shear strength and obtain five values of the performance function:

$$\begin{aligned} G(c_{u1}, c_{u2}) &= 11.5 \\ G(c_{u1}^+, c_{u2}) &= 29.1, \quad G(c_{u1}^-, c_{u2}) = -33.8 \\ G(c_{u1}, c_{u2}^+) &= 69.1, \quad G(c_{u1}, c_{u2}^-) = -65.6 \end{aligned}$$

with

$$\begin{aligned} c_{u1} = 40 \text{ kPa}, \quad c_{u1}^+ = 40 + 20 = 60 \text{ kPa}, \quad c_{u1}^- = 40 - 20 = 20 \text{ kPa} \\ c_{u2} = 25 \text{ kPa}, \quad c_{u2}^+ = 25 + 15 = 40 \text{ kPa}, \quad c_{u2}^- = 25 - 15 = 10 \text{ kPa} \end{aligned}$$

With these values, we can estimate the variance of the performance function and then the reliability index:

$$\begin{aligned} \text{Var}(G) &= \left( \frac{G(c_{u1}^+, c_{u2}) - G(c_{u1}^-, c_{u2})}{2\sigma_{c1}} \right)^2 \sigma_{c1}^2 \\ &\quad + \left( \frac{G(c_{u1}, c_{u2}^+) - G(c_{u1}, c_{u2}^-)}{2\sigma_{c2}} \right)^2 \sigma_{c2}^2 = 5525.1 \\ \sigma_G &= \sqrt{\text{Var}(G)} = 74.3 \\ \beta &= \frac{E(G)}{\sigma_G} = \frac{11.5}{74.3} = 0.15 \end{aligned}$$

This reliability index is very small and its corresponding probability of failure is 44%.

Alternatively, we can use the Genetic Algorithm to find this reliability index. In doing this, a key issue is to define a tolerance on the performance function, as illustrated in Fig. 4. The tolerance needs to be used in numerical computation and, if undefined, represents a machine rounding to zero. In general, a smaller tolerance represents a more accurate estimation of the reliability index, but may result in more computation. However, due to the random nature of the GA search, a smaller tolerance does not always mean a smaller  $|G(\mathbf{x})|$ . The tolerance range of the performance function studied in this example is between 0.1 and 0.0001.

The GA is first fine-tuned and the following GA parameters are used as default values for this case: population size = 60;  $P_{\text{mutate}} = 1/(\text{population size})$ ;  $P_{\text{inverse}} = 0.7$ ;  $P_{\text{swap}} = 0.1$ . The GA stops if the best solution satisfies the tolerance of the performance function.

The GA is run four times using the same random number seed value based on four different tolerances. Table 2 presents the results obtained from the GA finite element method. Compared to the result of  $\beta = 0.15$  obtained from the FOSM method, the reliability index obtained from the GA method is in the range of 0.45–0.51, significantly higher than the former. Table 2 also reveals that the decision variables can be different depending on the tolerance set for the performance function. Of interest here is the affordable

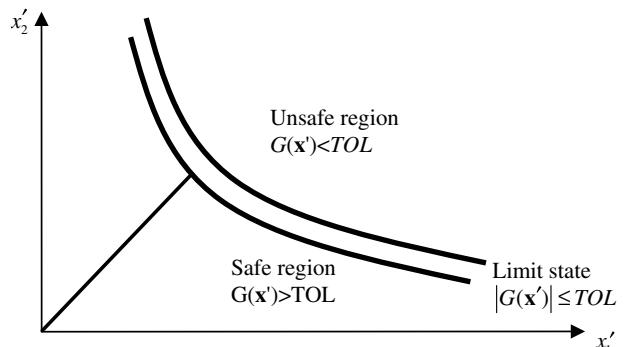


Fig. 4. Tolerance on the performance function with two random variables.

Table 2  
GA results for example 2

Tolerance	Actual $G(\mathbf{x})$	$\beta$	$c_{u1}$	$c_{u2}$	CPU times (min)	Number of GA evaluations
0.1	0.005	0.454	18.19	39.40	14.0	2
0.01	0.005	0.454	18.19	39.40	7.19	2
0.001	0.0008	0.516	23.13	29.97	107.8	15
0.0001	0.00003	0.487	17.69	40.30	122.28	17

number of GA evaluations and hence the CPU time. The CPU time was taken on a Sony Vaio Laptop with 512 GB RAM and a 2.8 GHz Pentium 4 processor.

To determine how consistently the GA is able to converge to a solution, the GA was run five times using different initial random seeds with the tolerance was set to 0.0001. All other GA parameters were the same as those given above except that the  $P_{\text{inverse}}$  was set to 0.9. As shown in Table 3, the GA obtains relatively consistent results for the five consecutive runs, with an average evaluation of 18.4 generations. Although it cannot be said that the best  $\beta$  value is found using the GA method, it at least indicates that the GA is able to find a more accurate  $\beta$  value than the FOSM method.

#### 4.3. Example 3

In the third example, we use the associated Mohr–Coulomb model to represent the soil behaviour in the same strip footing problem as shown in Fig. 3. In this case, each soil layer has two strength parameters: the cohesion  $c'$  and the friction angle  $\phi'$ . Their means and standard deviations are given below:

$$\begin{aligned} \text{Upper soil } c'_1 : m = 5 \text{ kPa}, \quad \sigma = 2 \text{ kPa}; \quad \phi'_1 : m = 30^\circ, \quad \sigma = 3^\circ \\ \text{Lower soil } c'_2 : m = 20 \text{ kPa}, \quad \sigma = 5 \text{ kPa}; \quad \phi'_2 : m = 25^\circ, \quad \sigma = 2^\circ \end{aligned}$$

Supposing the design load of the rigid footing is 200 kPa, the performance function of the limit state surface can then be expressed as:  $G(\mathbf{x}) = F(\mathbf{x}) - 200 = 0$ , with  $F(\mathbf{x})$  being the

numerical collapse load. The four variables include  $c'_1$  and  $\phi'_1$  for the upper soil and  $c'_2$  and  $\phi'_2$  for the lower soil. Similar to example 2, the four variables are considered to be independently uncorrelated with normal probability distributions.

Using the mean values of the cohesion and the friction angle, a numerical collapse load of 323 kPa was found, giving a deterministic factor of safety of 1.6. Again, the reliability index can be obtained from the FOSM method with nine discrete finite element analyses. The resulting value of 1.39 gives a probability of failure of 8%.

$$G(\text{mean}) = 123.56$$

$$\begin{aligned} G(c_1^+) = 161.97, \quad G(c_1^-) = 62.86, \quad G(\phi_1^+) = 164.58, \quad G(\phi_1^-) = 74.02 \\ G(c_2^+) = 169.89, \quad G(c_2^-) = 67.72, \quad G(\phi_2^+) = 152.02, \quad G(\phi_2^-) = 97.19 \end{aligned}$$

Similar to example 2, the GA was run four times using four different tolerances and the GA results are summarized in Table 4. The  $\beta$  values are found to lie in the range 0.996–1.10 for the four different tolerances. It may also be seen that the GA reached the solutions fairly quickly for tolerances up to 0.001. Again, it also shows that the GA is robust and efficient in obtaining the optimal result. It therefore suggests that the GA could be used as an alternative method for probabilistic finite element analysis for identifying optimal solutions to constrained problems in geotechnical engineering applications with stochastic inputs.

Note that the reliability indices obtained from the GA for the four tolerances are all lower than the ones obtained using the FOSM method. However, it does not necessarily mean that the FOSM method is more accurate simply because its solution is closer to the factor of safety of 1.6. In fact, the relationship between the factor of safety and the probability of failure is not monotonic, and some geotechnical structures with a factor of safety of 2 have a probability failure of  $10^{-7}$ , whereas others with factors of safety of 4–5 have probabilities of failure greater than  $10^{-1}$  [22]. Moreover, the FOSM method only performs  $2n + 1$  calculations to approximate the first-order derivatives for  $n$

Table 3  
Five GA runs for example 2

Runs	Actual $G(\mathbf{x})$	$\beta$	$c_{u1}$	$c_{u2}$	Number of GA evaluations
1	0.00003	0.461	21.81	31.70	24
2	0.00007	0.452	18.28	38.68	31
3	0.00007	0.444	21.17	32.73	7
4	0.00003	0.468	17.97	39.53	27
5	0.00001	0.480	19.74	33.42	3

Table 4  
GA results for example 3

Tolerance	Actual $G$	$\beta$	$c'_1$	$\phi'_1$	$c'_2$	$\phi'_2$	CPU times (min)	Number of GA generations
0.1	0.005	1.05	22.03	6.00	23.97	28.02	27	1
0.01	0.0068	1.07	17.24	6.00	24.08	28.14	54	2
0.001	0.0004	1.10	21.11	3.63	24.88	32.48	161	5
0.0001	0.00003	0.996	22.43	4.62	23.33	30.51	813	25

random variables, whereas numerous combinations of decision variables can be tested during the GA search. Therefore, it is reasonable to conclude that the GA should be able to find more reliable results as compared to the FOSM method, in particular, when more decision variables are involved.

## 5. Conclusions

In this paper, a GA method has been proposed and incorporated into the displacement finite element method for probabilistic failure analysis by calculating the Hasofer–Lind reliability index of geotechnical structures. Two cases were considered and the performance of the genetic algorithm was compared with the first-order second moment Taylor series method. Although it cannot be guaranteed that the GA locates the ‘best’ reliability index  $\beta$ , the results show that the GA gives more accurate estimations than the simpler probability method. In this sense, the GA approach offers a new opportunity for probabilistic finite element analysis of geotechnical problems. Future work will involve the application of the GA to geotechnical problems with a large number of decision variables.

## References

- [1] Chu PC, Beasley JE. Constraints handling in genetic algorithms: the set of partitioning problem. *J Heuristics* 1998;11:323–57.
- [2] Coello CA, Montes EM. Constraint handling in genetic algorithms through the use of dominance-based tournament selection. *Adv Eng Informatics* 2002;16(3):193–203.
- [3] Deb K, Agrawal S. A niched-penalty approach for constraint handling in genetic algorithms. In: Proceedings of the international conference on artificial neural networks and genetic algorithms, Portoroz, Slovenia; 1999. p. 235–43.
- [4] Deng JH, Lee CF. Displacement back analysis for a steep slope at the Three Gorges Project site. In: International journal of rock mechanics and mining sciences, vol. 38 (2). Oxford, UK: Elsevier; 2001. p. 259–68.
- [5] Eshelman LJ, Schaffer JD. Preventing premature convergence in genetic algorithm by preventing incest. In: Proceedings of the fourth international conference on genetic algorithm. La Jolla, CA: Morgan Kaufmann; 1991. p. 115–22.
- [6] Fenton GA, Vanmarcke EH. Simulation of random fields via local average subdivision. *J Eng Mech ASCE* 1990;116(8):1733–49.
- [7] Fenton GA, Griffiths DV, Williams MB. Reliability of traditional retaining wall design. *Geotechnique* 2005;55(1):55–62.
- [8] Gilbert RB. Geotechnical reliability. In: Senneset KG, Shackelford CD, Strout JM, editors. Proceedings of the US – Scandinavian workshop on geotechnical research collaboration. Department of Geotechnical Engineering, Norwegian Institute of Technology, Norway; 1994. p. 87–95.
- [9] Goldberg DE. Genetic algorithms in search, optimization and machine learning. Reading, MA: Addison-Wesley; 1989.
- [10] Goh ATC. Genetic algorithms search for critical slip surface in multi-wedge stability analysis. *Can Geotech J* 1999;36(4):382–91.
- [11] Griffiths DV, Fenton GA, Tveten DE. Probabilistic geotechnical analysis: how difficult does it need to be? Keynote Lecture, UFE Conference on Probabilistics in Geotechnics, Graz, Austria, September 15–19, 2002.
- [12] Griffiths DV, Fenton GA. Bearing capacity of spatially random soil: the undrained clay problem revisited. *Geotechnique* 2001;51(4):351–9.
- [13] Hasofer AM, Lind NC. Exact and invariant second moment code format. *J Eng Mech Div, ASME* 1974;100(EM1):111–21.
- [14] Javadi AA, Farmani R, Toropov VV, Snee CPM. Identification of parameters for air permeability of shotcrete tunnel lining using a genetic algorithm. *Comput Geotech* 1999;25(1):1–24.
- [15] Jimenez F, Verdegay JL. Constrained multi-objective optimization by evolutionary algorithms. In: Proceedings of the international ICSC symposium on engineering of intelligent systems (EIS'98), University of La Laguna, Tenerife, Spain; 1998, p. 266–71.
- [16] Kuczera GA. Efficient subsurface probabilistic parameter optimization for catchment models. *J Water Res Res* 1997;33(1):177–85.
- [17] Li KS, White W. Probabilistic characterisation of soil profiles. Research Report No. 19, Department of Civil Engineering, University College, University of New South Wales, Canberra, Australia; 1987.
- [18] Li KS, Lee IK, Lo S-CR. Limit state design in geotechnics. In: Li, Lo, editors. Proceedings of the Conference on probabilistic methods in geotechnical engineering. Balkema, Rotterdam; 1993. p. 29–42.
- [19] Low BK. Reliability-based design applied to retaining walls. *Geotechnique* 2005;55(1):63–75.
- [20] Macari EJ, Samarajiva P, Wathugala W. Selection and calibration of soil constitutive model parameters using genetic algorithms. In: Yamamoto JA, Kaliakin VN, editors. Proceedings of the sessions of the Geo-Frontiers 2005 Congress. Austin, TX, USA; 2005. p. 310–32.
- [21] McCombie P, Wilkinson P. The use of the simple genetic algorithm in finding the critical factor of safety in slope stability analysis. *Comput Geotech* 2002;29(8):699–714.
- [22] Mostyn GR, Li KS. Probabilistic slope analysis-state-of-play. In: Li, Lo, editors. Proceedings of the conference on probabilistic methods in geotechnical engineering. Rotterdam: Balkema; 1993. p. 89–109.
- [23] Paice GM, Griffiths DV. Reliability of an undrained clay slope formed from spatially random soil. In: Yuan JX, editor. Computer methods and advances in geomechanics, vol. 1. Rotterdam: Balkema; 1997. p. 543–8.
- [24] Pal S, Wathugala GW, Kundu S. Calibration of a constitutive model using genetic algorithms. *Comput Geotech* 1996;19(4):325–48.
- [25] Sheng DC, Sloan SW. Load stepping methods for critical state models. *Int J Numer Meth Eng* 2001;50(1):67–93.
- [26] Sheng DC, Sloan SW, Yu HS. Aspects of finite element implementation of critical state models. *Comput Mech* 2000;26(2):185–96.
- [27] Sloan SW, Abbo AJ, Sheng DC. Refined explicit integration of elastoplastic models with automatic error control. *Eng Computation* 2001;18:121–54. see also Erratum in *Engineering Computation*, 19, 594, 2002.
- [28] Sloan SW, Sheng DC, Abbo AJ. Accelerated initial stiffness methods for elastoplasticity. *Int J Numer Analy Meth Geomech* 2000;24:579–99.
- [29] Simpson AR, Priest SD. The application of genetic algorithms to optimisation problems in geotechnics. *Comput Geotech* 1993;15(1):1–19.
- [30] Thurner R, Schweiger HF, Pottler R, Mattle B. A practical approach for risk assessment employing reliability analysis and finite elements. In: Yuan JX, editor. Computer methods and advances in geomechanics, vol. 1. Rotterdam: Balkema; 1997. p. 235–40.
- [31] US Army Corps of Engineers. Risk-based analysis in geotechnical engineering for support of planning studies, ETL 1110-2-556; 1999.
- [32] Vanmarcke EH. Probabilistic modelling of soil profiles. *J Geotech Eng Div, ASCE* 1977;103:1227–46.
- [33] Whitley D. Using reproductive evaluation to improve genetic search and heuristic discovery. In: Genetic algorithms and their applications: Proceedings of the second international conference on genetic algorithms. New Jersey: Hillsdale Publishers; 1987. p. 115–22.
- [34] Zolfaghari AR, Heath AC, McCombie PF. Simple genetic algorithm search for critical non-circular failure surface in slope stability analysis. *Comput Geotech* 2005;82(3):139–52.