Applied Soft Computing 9 (2009) 387–392

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Locating the critical failure surface in a slope stability analysis by genetic algorithm

Aniruddha Sengupta *, Anup Upadhyay

Department of Civil Engineering, Indian Institute of Technology, Kharagpur 721302, India

ARTICLE INFO

Article history:
Received 21 July 2005
Received in revised form 11 January 2006
Accepted 23 April 2008
Available online 15 May 2008

Keywords:
Slope stability
Critical failure surface
Evolutionary algorithm
Genetic algorithm

ABSTRACT

The slope stability analysis is routinely performed by engineers to evaluate the stability of embankment dams, road embankments, river training works, excavations and retaining walls. Locating the critical failure surface of a soil slope is rendered erroneous and cumbersome due to the existence of local minima points. In case of large soil slopes, engineers face with a search space too large to employ the trial and error method in a computationally efficient fashion. A genetic algorithm is proposed to locate the critical surface under general conditions with general constraints. Convergence to any prescribed degree of precision was achieved with the algorithm. The algorithm has been demonstrated to be computationally superior to other optimization routines, like, Monte-Carlo method and grid-points approach.

1. Introduction

The slope stability analysis is routinely performed by engineers to evaluate the stability of embankment dams, road embankments, river training works, excavations and retaining walls. The slides that occurred during the construction of the Panama Canal and construction of railways in Sweden spurred the engineers all over the world into a lot of research on various aspects of the stability analysis. Determination of the potential failure surface (slip surface) and the corresponding forces tending to cause slip and to restore or stabilize the sliding mass, and the computation of available margin of safety are the essential steps in a stability analysis. The margin of safety or factor of safety for a failure surface is computed as a ratio of the restraining force (shear strength) over sliding force (shear stress). Thus, the determination of the critical failure surface (a failure surface for which factor of safety is minimum) is central to a slope stability analysis. The presence of several local minima points in the search space proves to be the chief problem in a slope stability analysis as demonstrated by Chen and Shao [1]. The authors of this paper have tried different search techniques like the "Grid" method and found that the existing methods fall for local minima even for slopes with simple geometry. Chen and Morgenstern [2] have also reported the same findings. In the present paper, a genetic algorithm (GA) has been successfully employed to locate the critical failure surface in a soil slope. Unlike other approaches, which have been demonstrated to fall for local minima, the GA always zeroed in on the global minima and found to be computationally efficient even for large slopes.

2. Existing methods for determining critical failure surface

The initial work on locating the critical failure surface in a slope stability analysis was done by Fellenius [3]. In Fellenius method the center of the initial critical circle, O, is assumed to be the intersection of two lines set off from the base and top of the slope. A point, P, is then fixed $2H$ below the top of the slope and $4.5H$ horizontally from the toe of the slope, H being the height of the slope. According to Fellenius, the center of the critical failure surface lies along a line joining the points P and O, and is obtained by trial and error. The procedure is valid for cohesive soils only. In 1962, Jumikis extended the procedure for general soils. The procedure is found to be less reliable for irregular and long slopes. Most of the existing computer programs UTEXAS3 [Edris and Wright [43]], STABL [Siegel [50]], SLOPE/W [Geo-Slope Intl [44]] however use a method of search popularly known as 'Grid' method. In this method, the location of the center of the critical surface is usually found by a 3 by 3, 9-point square grid. The center point of the first grid and the spacing between the grid points used for the initial mode of search are to be specified and it should represent a best estimate of the center of the critical failure surface. The location of the center point of the grid is shifted during the search until the center of the grid corresponds to the center of a surface,

* Corresponding author.
E-mail address: Sengupta@civil.iitkgp.ernet.in (A. Sengupta).

1568-4946/$ – see front matter
doi:10.1016/j.asoc.2008.04.015

which has a lower value for the factor of safety than any of the eight other failure circles whose centers are located on the perimeter of the grid. The spacing between the grid points is also reduced simultaneously with the progress of the search. The search is terminated when the grid spacing has been reduced to a specified distance and the center of the grid corresponds to the lowest factor of safety. The main problem with this method is that it is quite rigorous and there is no guarantee that the critical failure surface as obtained is the global minimum.

With the advent of fast computers, optimization based techniques have become an effective mean of searching for the critical slip surface in the slope stability analysis. Baker and Garber [4,5] used the calculus of variations to locate the critical slip surface and to calculate associated factor of safety. However, the existence of a minimum in their results were found to be questionable (Luceno and Castillo [6]). Celestino and Duncan [7], and Li and White [8] used alternating variable methods to locate the critical slip surface. This approach is also not practical, as it gets complicated even for a simple slope. Baker [9] used dynamic programming to determine the critical slip surface. Chen [10] postulated that using a random trail search would lead to the global minimum factor of safety. Nguyen [11] and De Natale [12] used the simplex method, and Chen and Shao [1] used simplex, steepest descent, and Davidson–Fletcher–Powell (DFP) methods in conjunction with grid search solution. This approach too is inefficient if a high accuracy is required which makes it unsuitable for real problems. Even though these methods may work for simple problems, there are many limitations associated with these methods, which have been addressed by Li and White [13]. Greco [14] presented Monte-Carlo based techniques of the random walk type to locate the critical slip surface. The trial solutions are randomly generated and then compared with the best solution for improvement. However, implementation of this method in an automatic search requires too many constraints. Husein et al. [15] also developed an approach for locating the critical slip surface based on Monte-Carlo techniques. Monte-Carlo based methods (random walk and random jumping) are simply structured, random searching and optimization techniques. A large number of trial surfaces are usually generated to ensure minimum factor of safety. The search space is demarcated to reduce the amount of unproductive computation, but there is no guarantee of finding the lowest factor of safety. This can be effective when the search space is tightly controlled, but necessitates the analysis of a large number of solutions. The slope failure potential has been also evaluated by fuzzy logic [Mathada et al. [16], Dodagoudar and Venkatachalam [17], Rubio et al. [18] and Giasi et al. [45]]. But this method is applied to very simple slopes only.

The genetic algorithm has been receiving a lot of attention now days because of its elegance and efficiency. The genetic algorithm is being applied to solve a large spectrum of problems numerically, like, optimization of traffic signal control [Anderson et al. [19]], subsonic wing design [Shigeru [20]], hydraulic actuator design [Andersson [21]], city planning [Balling and Wilson [22]], wire-antenna design [Caswell and Lamont [23]], control systems [Fleming and Purshouse [24]], ground water monitoring [Reed and Minsker [25]], water supply system design [Rouhiainen and Tade [26]], concert hall design [Sato et al. [27]], piled raft foundation [Ganeshwadi and Dodagoudar [28]], crack detection in structures [Vakil-Baghmisheh et al. [29], Sahoo and Maity [30]], etc. In this paper, a new application of the genetic algorithm has been explored. A search technique based on genetic algorithm has been proposed in this paper as an alternative to the traditional methods of search for critical failure surface and minimum factor of safety in a slope stability problem. As demonstrated in this paper, the proposed method is computationally efficient and always converges to the global minima.

3. Genetic algorithm

A genetic algorithm (GA) [Back [31], Dasgupta and Michalewicz [32], Michalewicz [33], Holland [34], Coello [35], Hedberg [36], Goldberg [37], Zoffaghari et al. [38], Yang et al. [39], Nian and Zheng [40]] is a search technique used to find approximate solutions to optimization and search problems. Genetic algorithms are a particular class of evolutionary algorithms (EA) that use techniques inspired by evolutionary biology such as inheritance, mutation, crossover and natural selection. Genetic algorithms are typically implemented in which a population of abstract representations (called chromosomes) of feasible solutions (called individuals) to an optimization problem evolves toward better solutions. The evolution starts from a population of completely random individuals and happens in generations. In each generation, the fitness of the whole population is evaluated, multiple individuals are stochastically selected from the current population based on their fitness, and modified (mutated or recombined) to form a new population, which becomes current in the next iteration. A solution to a problem is represented by a list of parameters, called chromosomes which are typically represented by simple strings of data and instructions. Initially several such individuals are randomly generated to form the first initial population. During each successive generation, each individual is evaluated, and a value of fitness is returned by a fitness function. The pool is sorted, with those having better fitness (representing better solutions to the problem). For each individual to be produced in the next step, a pair of parent organisms is selected for breeding. Selection is biased towards elements of the initial generation which have better fitness, though it is usually not so biased that poorer elements have no chance to participate, in order to prevent the population from converging too early to a sub-optimal or local solution. There are several well-defined organism selection methods, like roulette wheel selection and tournament selection. Following selection, the crossover operation is performed upon the selected chromosomes. Organisms are recombined by this probability. Crossover results in two new child chromosomes, which are added to the next generation population. The chromosomes of the parents are mixed during crossover, typically by simply swapping a portion of the underlying data. This process is repeated with different parent organisms until there are an appropriate number of candidate solutions in the next generation population. The next step is to mutate the newly created offspring. Typical genetic algorithms have a fixed, very small probability of mutation on the order of 0.01 or less. Based on this probability, the new child organism's chromosome is randomly mutated. These processes ultimately result in the next generation population of chromosomes that is different from the initial generation. Generally the average fitness will have increased by this procedure for the population, since only the best organisms from the first generation are selected for breeding. This generation process is repeated until a termination condition (like, maximum number of generation reached, minimum criteria satisfied or successive iterations do no longer produce better results) has been reached.

3.1. Molding the slope stability problem

To employ the genetic algorithm, the factor of safety is defined as the fitness function which, must be expressed in terms of (a) the coordinates of the center of the circular failure surface (X, Y) and (b) the radius of the circular failure surface (R) (Table 1).

The Method of Slice (Bishop [41], Fellenius [3]) is the most commonly used soil slope stability method, where the soil mass above an assumed slip surface is divided into vertical slices (refer

Table 1
Range of control variables

Control variable	Range
X	$\{h^*\cot(B), h^*\cot(B) - 10\}$
Y	$\{h, h + 10\}$
R	$\{0, 30\ \text{m}\}$

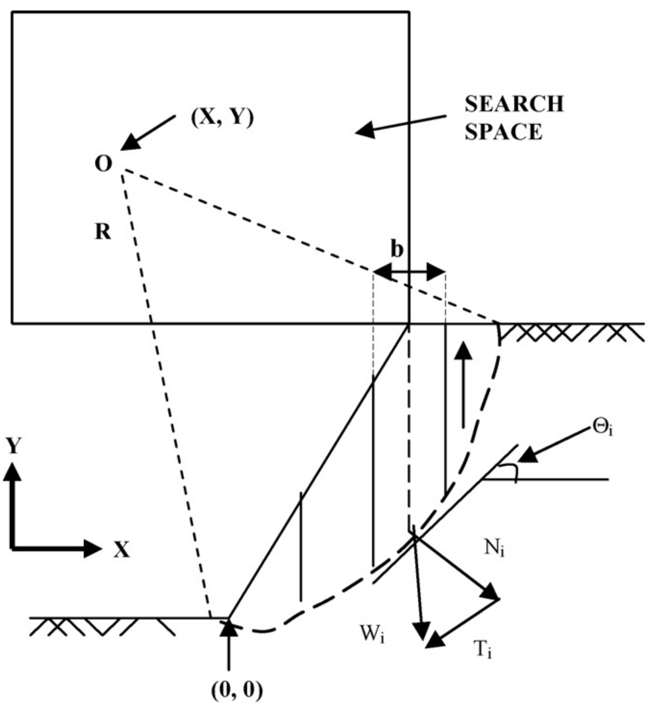

Fig. 1. Failure circle (in the Method of Slice). (X, Y): center of the failure circle; R: radius of the failure circle; B: angle of inclination of the slope; h: height of the slope; B: width of a slice; W_i: weight of ith slice; N_i: normal component of forces in ith slice along failure circle; T_i: tangential component of forces in ith slice along failure circle; Θ_i: direction of tangent in ith slice.

to Fig. 1). The forces (W_i, T_i and N_i) on a typical slice are shown in Fig. 1. The Factor of safety (F) in the Method of Slice is expressed as:

$$F = \frac{[c' * L + \tan(\Phi') * \Sigma(N - U)]}{(\Sigma T)} \tag{1}$$

where L is the length of the failure arc. The c'(effective cohesion), and Φ'(effective angle of internal friction) are known constants. The variables which depend on the coordinates of the center (X, Y) and radius (R) of the failure surface are:

1. N (normal force at each slice)
2. U (force due to pore-pressure at each slice) and
3. T (tangential force at each slice)

3.2. Derivation of (N, U, T) in terms of (X, Y, and R)

First, we determine the points of intersection of the failure circle with the boundaries of the slope.

Let the x coordinate of the point of intersection at the foot of the slope be X_L and the x coordinate of the point of intersection at the top of the slope be X_U.

The equation of the failure circle is:

$$(x - X)^2 + (y - Y)^2 = R^2 \tag{2}$$

The equation for the foot of the slope is: $y = 0$ and the equation for the top of the slope is:

$$Y = h \tag{3}$$

From (2) and (3) we get:

$$X_L + X + \sqrt{}(R^2 - Y^2) \tag{4}$$

$$X_U = X + \sqrt{}[R^2 - (h - Y)^2] \tag{5}$$

The width of each slice is

$$\left(X_U - \frac{X_L}{n}\right) = b \tag{6}$$

where n = number of slices.

3.2.1. Deriving Θ_i

Θ_i is the angle between the tangent to the circle at the point of intersection of the circle with the centerline of each slice (Fig. 2).

The equation of the centerline of a slice is given by

$$\begin{aligned} x &= \frac{(x_i + x_{i+1})}{2} \\ &= X + \sqrt{}(R^2 - Y^2) + b * \left(i + \frac{1}{2}\right) \end{aligned} \tag{7}$$

Solving (7) with the equation of the failure circle, the coordinates of the point of intersection of the circle with the centerline of each slice are obtained.

$$X_{Mi} = X + \sqrt{}(R^2 - Y^2) + i * b \tag{8}$$

$$Y_{Mi} = Y + \sqrt{}\left[R^2 - \left\{\sqrt{}(R^2 - Y^2) + b * \left(i + \frac{1}{2}\right)\right\}\right. \tag{9}$$

The slope of the tangent of a circle at a point (x_1, y_1) is given by

$$Xx_1 + yy_1 + g(x + x_1) + f(y + y_1) + c = 0 \tag{10}$$

where, the equation of the circle is

$$x^2 + y^2 + 2gx + 2fy + c = 0$$

The equation of the failure circle is:

$$x^2 + y^2 - 2X * x - 2Y * y - R^2 = 0 \tag{11}$$

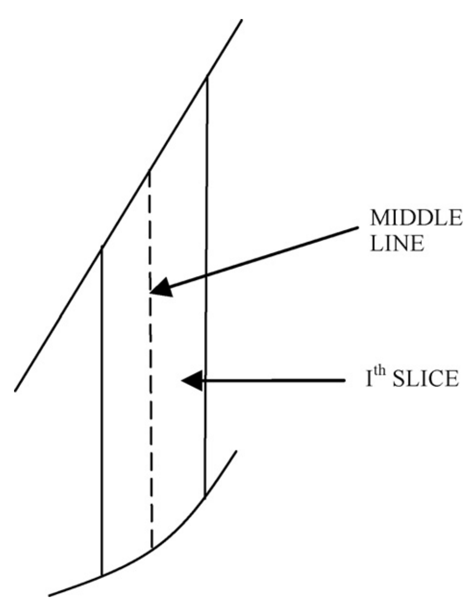

Fig. 2. Ith slice.

Thus, the equation of the tangent is

$$y = x * \frac{(X - X_{Mi})}{(Y - Y_{Mi})} + \frac{(X * X_{Mi} + Y * Y_{Mi} + R^2)}{(Y - Y_{Mi})} \tag{12}$$

Thus,

$$\Theta_i = \tan^{-1}\left[\frac{(X - X_{Mi})}{(Y_{Mi} - Y)}\right] \tag{13}$$

Now W can be resolved into T and N as

$$T = W * \sin(\Theta_i) = \gamma * h * b * \sin(\Theta_i) \tag{14}$$

$$N = W * \cos(\Theta_i) = \gamma * h * b * \cos(\Theta_i) \tag{15}$$

where, $W = \gamma * h * b$

In case of slopes with multiple layers having different index properties, the same formulation holds true. Since one slice is considered at a time, the index properties can be varied from slice to slice.

3.2.2. Total angle (TA)

The total angle is the angle between the two lines joining the points of intersection of the failure circle with the base and top of the slope.

$$TA = \tan^{-1}\left[\frac{(M_1 - M_2)}{(1 + M_1 * M_2)}\right] \tag{16}$$

and the length of each slice is given by

$$L_{slice} = \left(\frac{TA}{10}\right) * R \tag{17}$$

where,

$$M_1 = \frac{Y}{(X - X_L)}$$

$$M_2 = \frac{(Y - h)}{(X - X_U)}$$

Thus, all the terms that come into play in the factor of safety equation are expressed in terms of (X, Y, and R).

4. Working of the genetic algorithm (GA)

The proposed genetic algorithm has 2 steps: (1) choosing and minimizing the fitness function and (2) fine-tuning the parameters of the algorithm.

The fitness function is chosen to be the factor of safety reduced by the factor of safety of the worst chromosome, that is,

$$F = F - F_{WORST} \tag{18}$$

This is often referred to as the technique of windowing. It eliminates the weakest chromosome and stimulates the strongest ones.

Each of the three variables (X, Y, and R) in the Eqs. (11) and (12), is converted to its corresponding binary representation. The binary strings corresponding to each of the three control variables is then concatenated. Eight bits have been employed for each of the three variables. The fitness function corresponding to each chromosome is calculated. The chromosomes having lower fitness function values are assigned a higher probability of being copied to the next generation. The biased roulette wheel approach has been employed for this purpose.

Pairs of chromosomes are picked up randomly and the 2-point crossover has been employed to increase the variation in the population. In 2-point crossover any two positions, say kth and lth are decided randomly and the parts of the chromosome from 1 to ($k - 1$) and ($l + 1$) to 24 are swapped. The flip-bit mutation has been employed in this algorithm. In this mutation, the value of the chosen bit is simply inverted (0 becomes 1 and 1 becomes 0). The position at which this inversion takes place is chosen randomly.

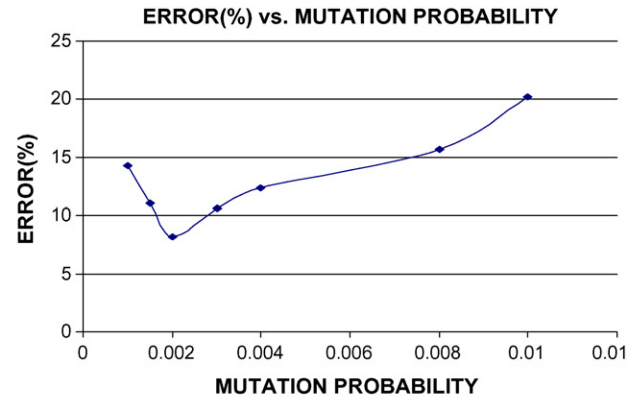

Fig. 3. Error (%) vs. mutation probability.

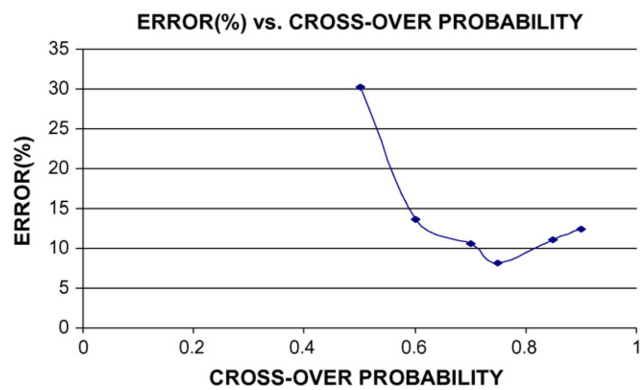

Fig. 4. Error (%) vs. crossover probability.

To fine-tune the parameters, the Grid method was implemented and for a data set of 100 slopes, the average error was calculated. The Fig. 3 shows the error versus mutation probability and Fig. 4 shows the error versus the crossover probability.

The following parameters were found out to be the best for the present problem (Table 2).

5. Results

Comparison of results in terms of minimum factor of safety from the genetic algorithm, Monte-Carlo and grid approach is summarized below:

As can be seen from the above figure (Fig. 5), the factor of safety stabilizes at 1.2 at first. This represents the local minima. The Grid method or the Monte-Carlo technique would have erroneously assumed 1.2 to be the lowest factor of safety. But in GA, upon increasing the number of generations, the factor of safety stabilizes finally at around 1.16 (at 300 generations) which represents the

Table 2
Parameters of the GA

Mutation probability	2/1000
Crossover probability	75/100
Population size	200
Chromosome length	24
Crossover	Two point

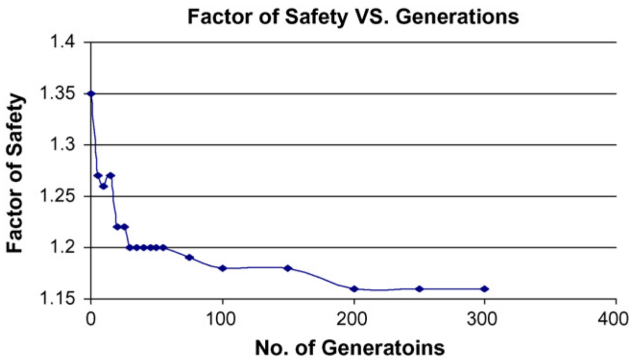

Fig. 5. Factor of safety vs. number of generations.

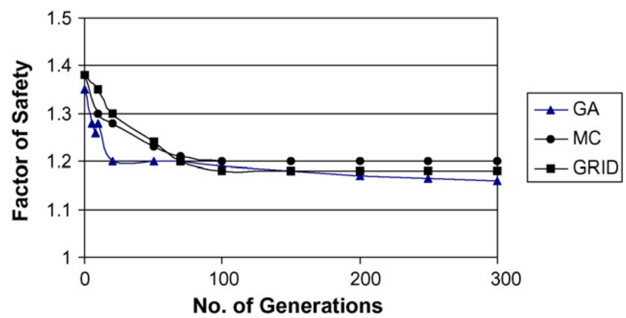

Fig. 6. Comparison of factor of safety by GA, Monte-Carlo and Grid methods.

global minima. This ability of the genetic algorithm to zero in on the global minima is its central strength.

In Fig. 6, the data points represented by triangles show the variation of the factor of safety with the number of generations as determined by GA. The data points represented by dashed lines correspond to the Grid method and Monte-Carlo methods.

The grid approach falls for the local minima of 1.2 but the GA zeroes in on the global minima of 1.16 after 300 generations.

6. Comparison of CPU memory usage

The above three methods (GA, Monte-Carlo (MC) and Grid methods) were tested for CPU memory usage on a Pentium-4 processor, 384 Mb RAM (Fig. 7).

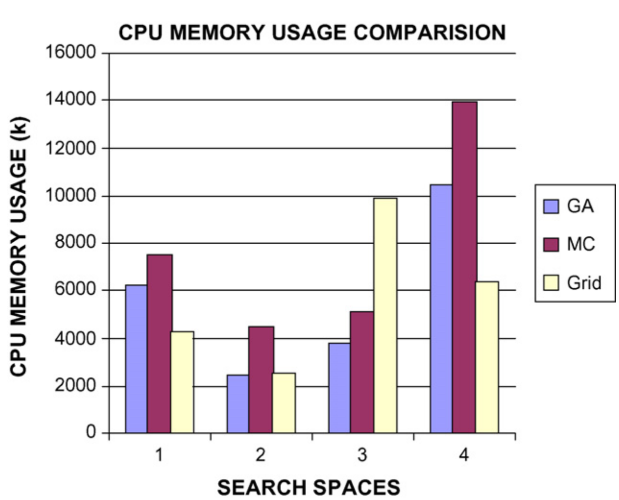

Fig. 7. Comparison of CPU memory usage for GA, Monte-Carlo and Grid method.

The dimensions of the search space were varied in each of the four search spaces. The genetic algorithm consumed less memory than Monte-Carlo method for each of the four slopes. The Grid method exhibited irregular memory consumption because it is prone to fall for local minima. If the local minimum is achieved early in the search process, the CPU memory consumption is lowest for Grid method.

7. Conclusions

The results show that a genetic algorithm (GA) can be successfully employed to locate the critical failure surface in a soil slope. Unlike the grid approach, which has been demonstrated to fall for local minima, the GA zeroed in on the global minima. The Monte-Carlo approach being a randomized hunt cannot be relied upon in the case of large soil slopes. It also turns out to be computationally inefficient. The distinguishing advantage of GA is its ability to combine the advantages of a randomized approach such as the Monte Carlo technique (the ability to find unexpected solutions and to find the region of true minimum) with those of a systematic approach (to hunt for a local minimum). The genetic algorithm consumed less memory than Monte-Carlo method for each of the four slopes. In some of the cases the Grid method consumed less CPU memory because it is prone to fall for local minima. If the local minimum is achieved early, the CPU memory consumption is lowest for the Grid method.

References

[1] Z. Chen, C. Shao, Evaluation of minimum factor of safety in slope stability analysis, Can. Geotech. J. 25 (4) (1983) 735–748.

[2] Z. Chen, N.R. Morgenstern, Extension to the generalized method of slices for stability analysis, Can. Geotech. J. 20 (1) (1983) 104–109.

[3] W. Fellenius, Calculation of the stability of earth dams, in: Transactions of the 2nd Congress on Large Dams, vol. 4, 1936, 445–459.

[4] R. Baker, M. Gaber, Variational approach to slope stability, in: Proceedings of the 9th International Conference on Soil Mechanics and Foundation Engineering. vol. 2, Japanese Society for Soil Mechanics and Foundation Engineering, Tokyo, (1977), pp. 9–12.

[5] R. Baker, M. Garber, Theoretical analysis of the stability of slopes, Geotechnique 28 (1978) 341–395.

[6] A. Luceno, E. Castillo, Evaluation of variational methods in slope analysis, in: Proceedings of the 1st International Symposium on Landslides, vol. 1, Meerut, Sarita Prakashan, (1980), pp. 255–258.

[7] T.B. Celestino, J.M. Duncan, Simplified search for noncircular slip surface, in: Proceedings of the 10th International Conference on SMFE, 1981, pp. 391–394.

[8] K.S. Li, W. White, Rapid evaluation of the critical slip surface in slope stability problems, Int. J. Numer. Anal. Methods Geomech. 11 (1987) 449–473.

[9] R. Baker, Determination of the critical slip surface in slope stability computations, Int. J. Numer. Anal. Methods Geomech. 4 (1980) 333–359.

[10] Z.Y. Chen, Random trails used in determining global minimum factors of safety of slopes, Can. Geotech. J. 25 (1992) 735–748.

[11] V.U. Nguyen, Determination of critical slope failure surfaces, J. Geotech. Eng. ASCE 111 (2) (1985) 238–250.

[12] J.S. De Natale, Rapid identification of critical slip surface structure, J. Geotech. Eng. ASCE 117 (10) (1991) 1568–1589.

[13] K.S. Li, W. White, Rapid evaluation of the critical slip surface in slope stability analysis, Report No. 9, Australian Defense Force Army, University of New South Wales, Australia (1986).

[14] V.R. Greco, Efficient Monte Carlo Technique for locating critical slip surface, J. Geotech. Eng. ASCE 122 (7) (1996) 517–525.

[15] M.A.I. Husein, W.F. Hassen, S.K. Sarma, A global search method for locating general slip surface using Monte Carlo techniques, J. Geotech. Geoenviron. Eng. ASCE 127 (8) (2001) 688–698.

[16] V.S. Mathada, G. Venkatachalam, A. Srividya, Slope stability assessment—a comparison of probabilistic, possibilistic and hybrid approaches, Int. J. Performability Eng. 3 (2) (2007) 11–21.

[17] G.R. Dodagoudar, G. Venkatachalam, Reliability analysis of finite earth slopes, Indian Geotech. J. 36 (4) (2006) 348–372.

[18] E. Rubio, J.W. Hall, M.G. Anderson, Uncertainty analysis in a slope hydrology and stability model using probabilistic and imprecise information, Comput. Geotech. 31 (7) (2004) 529–536.

[19] J.M. Anderson, T.M. Sayers, M.G.H. Bell, Optimization of a fuzzy logic signal controller by a multiobjective genetic algorithm, in: Proceedings of the 9th International Conference on Road Transport Information and Control, London, 186–190, April, 1998.

[20] O. Shigeru, Pareto solutions of multipoint design of supersonic wings using evolutionary algorithms, in: I.C. Parmee (Ed.), Adaptive Computing in Design and Manufacture V, Springer-Verlag, London, 2002, pp. 3–15.
[21] J. Andersson, P. Krus D. Wallace, Multi-objective optimization of hydraulic actuator systems, Proceedings of the 2000 ASME Design Automation Conference, Paper DAC-14512, Baltimore, Maryland, September (2000).
[22] R. Balling, S. Wilson, The maximum fitness function for multi-objective evolutionary computation: application to city planning, in: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001), Morgan Kaufmann Publishers, San Francisco, California, 2001 July, pp. 1079–1084.
[23] D.J. Caswell, G.B. Lamont, Wire-Antenna Geometry Design with Multiobjective Genetic Algorithms Cong. on Evolutionary Computation (CEC'2002), 1, IEEE Service Center, Piscataway, New Jersey, 2002 May, pp. 103–108.
[24] P. Fleming, R.C. Purshouse, Evolutionary algorithms in control systems engineering: a survey, Control Eng. Pract. 10 (2002) 1223–1241.
[25] P.M. Reed, B.S. Minsker, Discovery and negotiation using multiobjective genetic algorithms: a case study in groundwater monitoring design, in: Proceedings of Hydroinformatics, Cardiff, UK, 2002.
[26] C. Rouhiainen, M.O. Tade, Genetic algorithms for optimal scheduling of chlorine dosing in water distribution systems, in: Proceedings of the 20th Convention of the Australian Water Association, Perth, Australia, 6–10 April, 2003.
[27] S. Sato, K. Otori, A. Takizawa, H. Sakai, Y. Ando, H. Kawamura, Applying genetic algorithms to the optimum design of a concert hall, J. Sound Vib. 258 (3) (2002) 517–526.
[28] U.A. Ganeshwadi, G.R. Dodagoudar, Reliability analysis of piled raft foundation, Indian Geotech. J. 36 (2) (2006) 69–85.
[29] M. Vakil-Baghmisheh, P. Mansour, M.H. Sadeghi, M.M. Ettefagh, Crack detection in beam-like structures using genetic algorithms, J. Appl. Soft Comput. 8 (2) (2008) 1150–1160.
[30] B. Sahoo, D. Maity, Damage assessment of structures using hybrid neuro-genetic algorithm, J. Appl. Soft Comput. 7 (1) (2007) 89–104.
[31] T. Back, Evolutionary Algorithms in Theory and Practice, Oxford University press, New York, New York, 1996.
[32] D. Dasgupta, Z. Michalewicz, Evolutionary Algorithms in Engineering Applications, Springer, Berlin, 1997.
[33] Z. Michalewicz, A survey of constraint handling techniques in evolutionary computation methods, in: Proceeding of the 4th Annual Conference on Evolutionary Programming, MIT Press, Cambridge, MA, 1995, pp. 135–155.
[34] J. Holland, Genetic algorithms, Sci. Am. (1992) 66–72.
[35] C. Coello, An updated survey of GA-based multiobjective optimization techniques, ACM Comput. Surveys 32 (2) (2000) 109–143.
[36] S.R. Hedberg, Evolutionary computing: the rise of electronic breeding, IEEE Intell. Syst. 20 (6) (2005) 12–15 (November/December).
[37] D.E. Goldberg, The Design of Innovation: Lessons from and for Competent Genetic Algorithms, Addison-Wesley, Reading, MA, 2002.
[38] A.R. Zoffaghari, A.C. Heath, P.E. Mc Combie, Simple genetic algorithm search for critical non-circular failure surface in slope stability analysis, Comput. Geotech. 32 (3) (2005) 139–152.
[39] C.X. Yang, L.G. Tham, X.T. Feng, P.K.K. Lee, Two-stepped evolutionary algorithm and its application to stability analysis of slopes, J. Comput. Civil. Eng. 18 (2) (2004) 145–153.
[40] T. Nian, D. Zheng, Application of a simple genetic algorithm to the stability of slopes subjected to pore-water pressure, in: Proceedings of 6th World Congress on Intelligent Control and Automation, Dalian, China, (2006), pp. 3668–3671.
[41] A.W. Bishop, The use of slip circle in the stability analysis of earth slopes, Geotechnique 5 (1955) 7–17.
[43] E.V. Edris, S.G. Wright, UTEXAS3 slope stability package, Instruction Report GL-87-1, Computer Applications in Geotechnical Engineering (CAGE) and Geotechnical Aspects of the Computer Aided Structural Engineering (G-CASE) Project, USACE, Washington DC (1992).
[44] Geo-Slope Intl. SLOPE/W, A Slope Stability Analysis Package, Alberta Canada (2004).
[45] C.J. Giasi, P. Masi, C. Cherubini, Probabilistic and fuzzy reliability analysis of a sample slope near Aliano, Eng. Geol. 67 (3) (2003) 391–402.
[50] R.A. Siegel, STABL, Joint Highway Research Project, JHRP-75-9, Department of Civil Engineering, Purdue University, West Lafayette, Indiana (1978).