An Evolutionary Algorithm for Locating the Critical Failure Surface in a Soil Slope

Aniruddha Sengupta

Assistant Professor, Indian Institute of Technology, Kharagpur 721302 (India), Email: Sengupta@civil.iitkgp.ernet.in

and

Anup Upadhyay

Student, Indian Institute of Technology, Kharagpur, India

ABSTRACT

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. An evolutionary 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, Monte-Carlo method and grid-points approach.

Keywords: slope stability, critical failure surface, evolutionary algorithm.

INTRODUCTION

Determination of the critical failure surface is central to 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 (1983). The authors have tried different search techniques like the “Grid” method and found that the methods fall for local minima even for slopes with simple geometry. Chen and Morgenstern (1983) too has zeroed in on the same problem.

Baker and Garber (1978) have derived the Euler’s differential equation using calculus of variation. This approach is not practical as it gets complicated even for a simple slope. The internal stress distribution generated from the analysis may not be reasonable. Revilla and Castillo (1977) have applied similar approach using Janbu's simplified method. The random generation approach was tried by Boutrup and Lovell (1980) and also by Chen and Shao (1983). 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 (1986).

The most critical limitation to all these methods is that there is no guarantee that the critical failure surface as obtained is the global minimum. In the grid-points method a grid of circle centers is defined (circular failure surfaces are widely used in slope stability analysis) with the radii being defined in different ways as the search is complete only if the radius is varied too. In practice, this must be done in a systematic way, usually by starting with a coarse grid, then progressively finer grids as the region in which the critical case lies is identified. Another method is the 'Monte Carlo' approach [Husein, Hassen and Sarma (2002)], which is a randomized search within the specified search space. 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, as in the reference cited, but necessitates the analysis of a large number of solutions.

MOLDING THE SLOPE STABILITY PROBLEM

To employ the evolutionary algorithm we define the factor of safety as the fitness function which, must be expressed in terms of

1. The coordinates of the center of the circular failure surface (X, Y)

2. The radius of the circular failure surface (R).

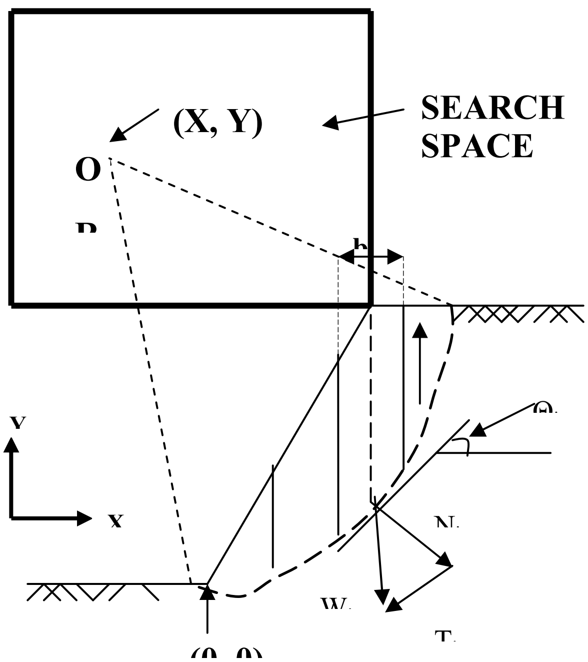

Figure 1. Failure circle (in the Method of Slices)

(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 i_{th} slice, N_i: Normal component of forces in i_{th} slice along failure circle, T_i: Tangential component of forces in i_{th} slice along failure circle, Θ_i : direction of tangent in i_{th} slice.

Table 1. Range of Control variables.

CONTROL VARIABLE	RANGE
X	{ h*cot(B), h*cot(B)-10 }
Y	{ h, h+10 }
R	{ 0, 30 meters }

The Method of Slice is the most commonly used soil slope stability analysis method, where the soil mass above an assumed slip surface is divided into vertical slices (refer to Figure 1). The forces (W_i, T_i and N_i) on a typical slice are shown in Figure 1. The Factor of safety (F) in the Method of Slice is expressed as**:**

$$\mathbf{F = [c'^*L + \tan(\Phi')^* \Sigma (N - U)] / (\Sigma T)} \qquad \mathbf{(1)}$$

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)

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:

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

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

the equation for the top of the slope is : **Y = h** **(3)**

From (2) and (3) we get:

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

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

The width of each slice is

$$(X_U - X_L/n) = b \qquad (6)$$

where, n = number of slices.

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.

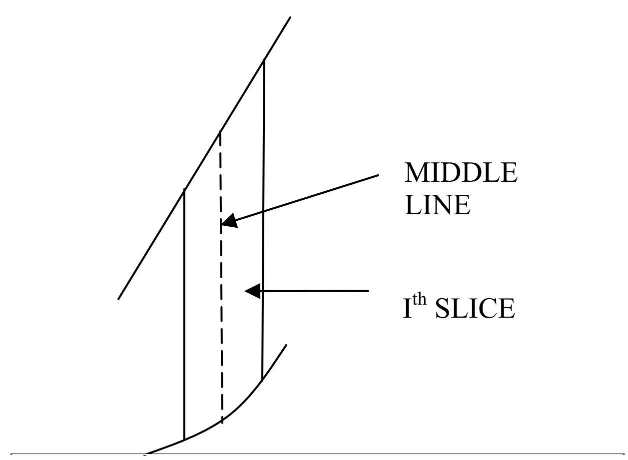

Figure 2. I^th Slice.

The equation of the centerline of a slice is given by

$$x = (x_i + x_{i+1})/2$$

$$= X + \sqrt{(R^2 - Y^2)} + b*(i+1/2) \qquad (7)$$

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

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

$$Y_{Mi} = Y + \sqrt{[R^2 - \{\sqrt{(R^2 - Y^2)} + b*(i+1/2)\}]} \qquad (9)$$

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

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

Where, the equation of the circle is

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

The equation of the failure circle is:

$$x^2 + y^2 - 2X*x - 2Y*y - R^2 = 0 \qquad (11)$$

Thus the equation of the tangent is

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

Thus

$$\Theta_i = \tan^{-1}[(X - X_{Mi})/(Y_{Mi} - Y)] \qquad (13)$$

Now we can resolve W into T and N as

$$T = W*\sin(\Theta_i) = \gamma*h*b*\sin(\Theta_i) \qquad (14)$$

$$N = W*\cos(\Theta_i) = \gamma*h*b*\cos(\Theta_i) \qquad (15)$$

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

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

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.

$$TA = \tan^{-1}[(M_1 - M_2)/(1+M_1*M_2)] \qquad (16)$$

- and the length of each slice is given by

$$\mathbf{L_{slice} = (TA/10)*R} \qquad \mathbf{(17)}$$

- where,

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

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

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

Working of the Evolutionary Algorithm (EA)

The proposed evolutionary algorithm is 3-stepped:

1. Choosing and minimizing the fitness function.
2. Fine-tuning the parameters of the algorithm:
    1. Cross-over probability
    2. Mutation probability
    3. Chromosome length
    4. Population size
    5. Choice between single point cross-over and two-point cross-over.

We have chosen the fitness function to be the factor of safety reduced by the factor of safety of the worst chromosome.

$$F = F - F_{WORST} \qquad (18)$$

This is referred as the technique of *windowing*. Windowing eliminates the weakest chromosome - the probability comes to zero - and stimulates the strongest ones

Generating a population

Each of the three variables, i .e. (X, Y, and R) is converted to its corresponding binary representation. For example, if the value of R is 25 meters it will be represented as 00110001. The binary strings corresponding to each of the three control variables is concatenated. The catch is to allocate the same number of bits for each of the three strings. We have employed 8 bits for each of the three variables. A typical chromosome looks like **00111001|00010001|00011000.** If the population size is 50, 50 such strings are generated randomly.

Reproduction

Reproduction is a process in which individual chromosomes are copied according to their fitness values. 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.

Crossover

Pairs of chromosomes are picked up randomly. In single point crossover any position in the chromosome, say k^{th} is decided at random and the parts of the chromosome from (k+1) to 24 are swapped. In 2-point crossover any 2 positions, say k^{th} and l^{th} are decided randomly and the parts of the chromosome from 1 to (k-1) and (l+1) to 24 are swapped. We have employed the 2-point crossover to increase the variation in the population.

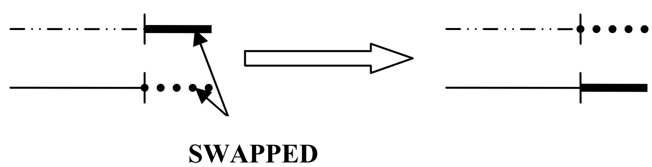

Figure 3. Crossover

Mutation

Mutation is a genetic operator that alters one ore more gene values in a chromosome from its initial state. This can result in entirely new gene values being added to the gene pool. With these new gene values, the genetic algorithm may be able to arrive at better solution than was previously possible. We have employed *flip-bit mutation*. In this mutation we simply invert the value of the chosen bit (0 becomes 1 and 1 becomes 0). The position at which this inversion takes place is chosen randomly.

Fine-Tuning the Parameters

To fine-tune the parameters the grid method was implemented and for a data set of 100 slopes average error was calculated.

The following parameters were found out to be the best for the present problem.

Table 2. Parameters of the EA.

Mutation Probability	2/1000
Crossover probability	75/100
Population Size	200
Chromosome Length	24
Crossover	Two point

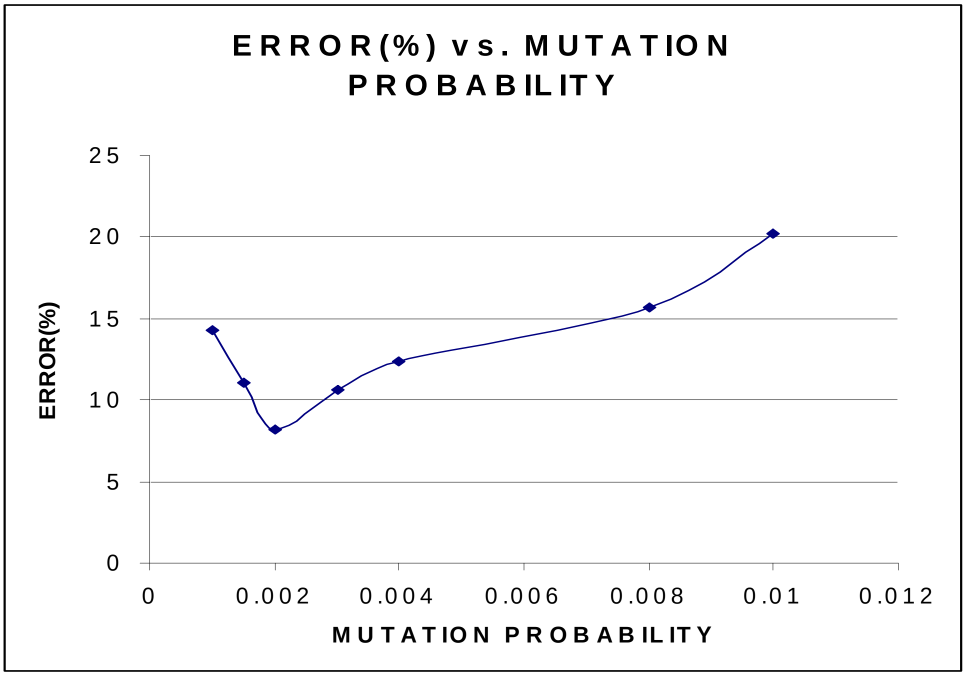

Figure 4. Error (%) vs. Mutation Probability.

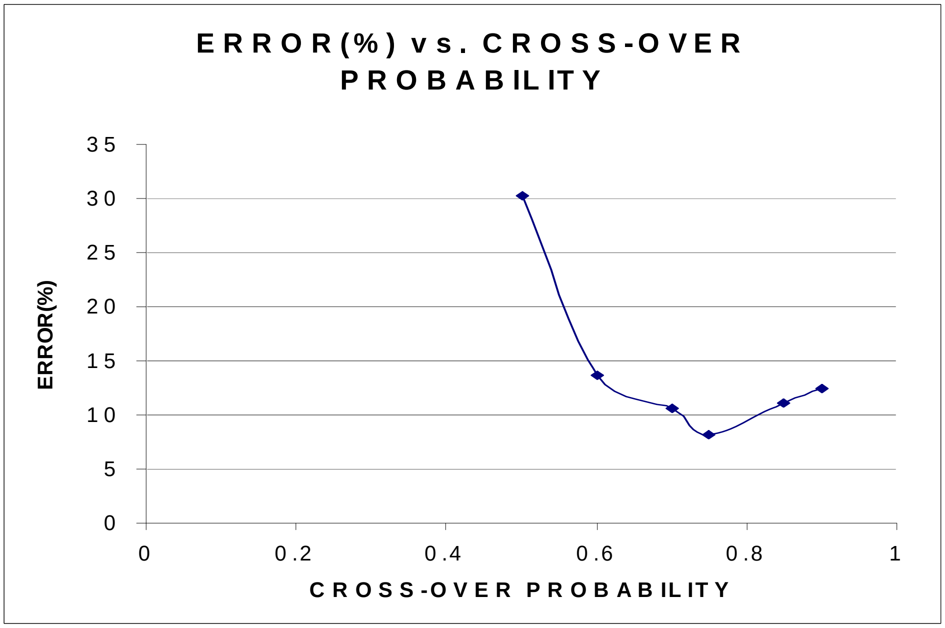

Figure 5. Error (%) vs. Crossover Probability.

Results

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

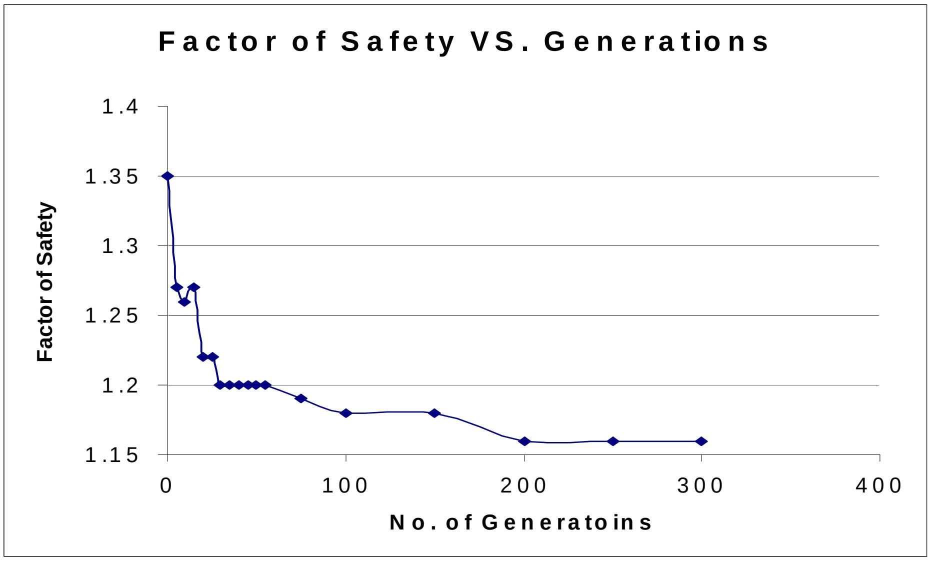

Figure 6. Factor of Safety vs. Number of Generations

As can be seen from above figure, 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 EA, upon increasing the number of generations, the factor of safety stabilizes finally at around 1.16 (at 300 generations) which represents the global minima. This ability of the evolutionary algorithm to zero in on the global minima is its central strength.

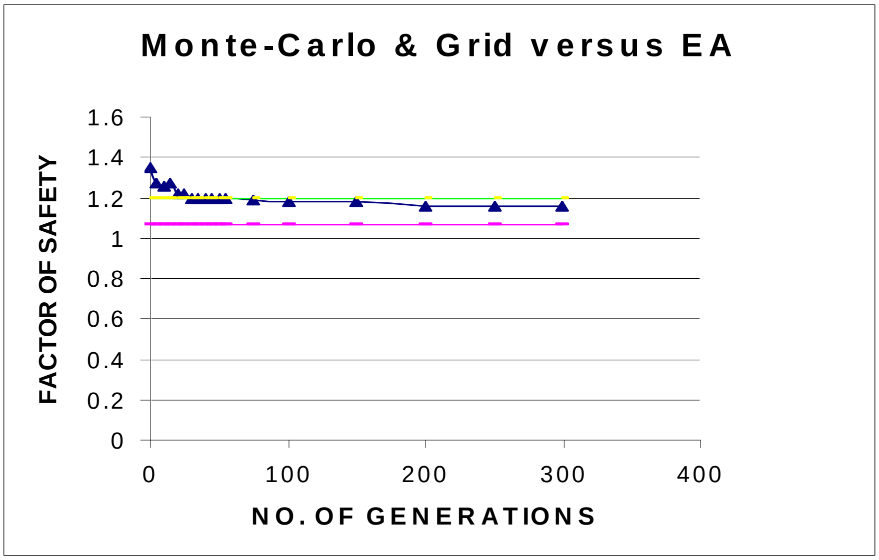

Figure 7. Comparison of Factor of Safety by EA, Monte-Carlo & Grid Methods.

In Figure 7, the data points represented by triangles show the variation of the factor of safety with the number of generations as determined by EA. 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 EA zeroes in on the global minima of 1.16 after 300 generations.

Comparison of CPU Memory Usage

The above three methods (EA, Monte-Carlo (MC) and Grid methods) were tested for CPU memory usage on a Pentium4 processor, 384 Mb RAM.

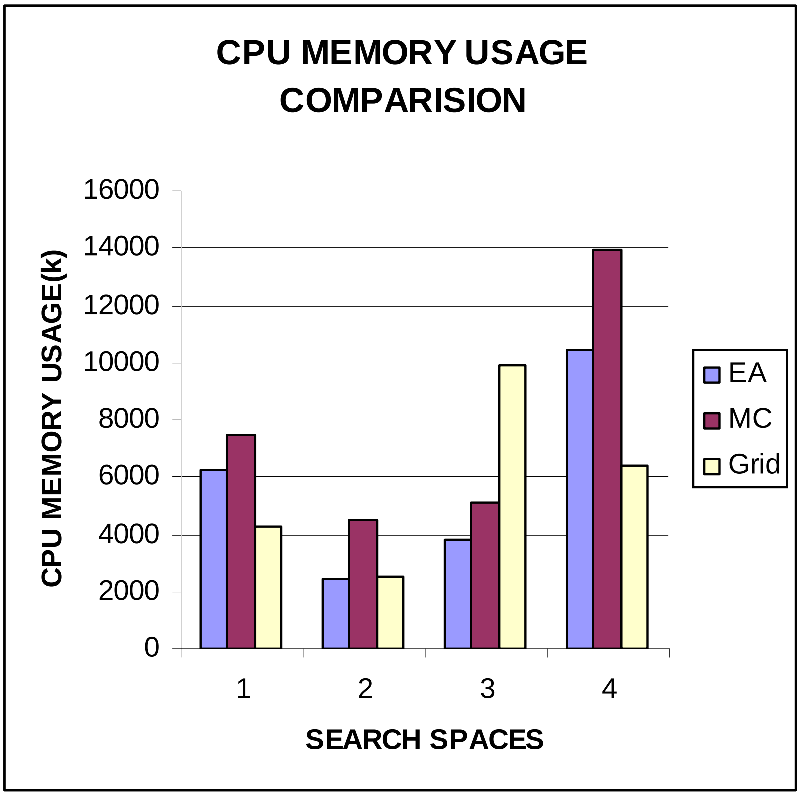

Figure 8. Comparison of CPU memory usage for EA, Monte-Carlo and Grid Method.

The dimensions of the search space were varied in each of the four search spaces. The evolutionary 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.

CONCLUSIONS

The results show that an evolutionary algorithm (EA) 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 EA 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 EA 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 evolutionary 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 minima is achieved early, the CPU memory consumption is lowest for the Grid method.

The EA should be tried for reinforced soil slopes as well. If an objective function is derived for non-linear slope failure and non-circular failure cases, it stands high chances of being successful in these cases also.

REFERENCES

1. Goldberg, D. E. (1989) "Genetic Algorithms in Search, Optimization, and Machine Learning," Addison-Wesley, New York.

2. Baker, R., and M. Garber (1978) "Theoretical Analysis of the Stability of Slopes," Geotechnique, 28, 341–95.

3. Boutrup, E., and C. W. Lovell (1980) "Searching Techniques in Slope Stability Analysis," Engineering Geology, 16, 51–61.

4. Chen, Z., and N. R. Morgenstern (1983) "Extension to the Generalized Method of Slices for Stability Analysis," Canadian Geotechnical Journal, 20(1), 104–109.

5. Chen Z. and C. Shao (1983) "Evaluation of Minimum Factor of Safety in Slope Stability Analysis," Canadian Geotechnical Journal, 25(4), 735–748.

6. Li, K.S., and W. White (1986) "Rapid Evaluation of the Critical Slip Surface in Slope Stability Analysis," Report No. 9, Australian Defence Force Army, University of New South Wales, Australia.

7. Revilla, J., and E. Castillo (1977) "The Calculus of Variations Applied to Stability of Slopes," Geotechnique, 27, 1–11.

8. Sarma, S.K. (1987) “A Note on the Stability of Slopes,” Geotechnique, 37(1), 107–111.

9. Husein Malkawi, A. I., W.F. Hassen, and S. K. Sarma (2002) “A Global Search Method for Locating General Slip Surface Using Monte Carlo Techniques,” J. of Geotechnical and Geoenvironmental Engg., ASCE, 127(8), 688-698.