University of Waterloo

Faculty of Engineering

Global Optimization of General Failure Surfaces in Slope Analysis by Hybrid Simulated Annealing

Rocscience, Inc.

Toronto, Ontario

Prepared by

Su, Xiao

ID 20251755

2B Chemical Engineering

May 1st , 2009

4105 Credit Pointe Dr,
Mississauga, Ontario
L5M 3K7

May 1st, 2009

Dr Thomas Duever,
Chair of Chemical Engineering
University of Waterloo
Waterloo, Ontario
N2L 3G1

Dear Dr Duever,

I am submitting this document, entitled "**Global Optimization of General Failure Surfaces in Slope Analysis by Hybrid Simulated Annealing"** as my 2B Work Report for Rocscience, Inc. My project investigated the efficiency of a hybrid simulated annealing (HSA) method for finding the minima (critical failure surfaces and factors of safety) of slope stability equations. The HSA combines a Very Fast Simulated Annealing (VFSA) and a Local Monte-Carlo (LMC) search, the latter being an optimization algorithm designed specifically for solving slope stability problems.

This work is an extension as well as a generalization of the project from my summer work-term (2008), also undertaken at Rocscience. During both terms, I worked under the supervision of Dr. Brent Corkum, software development manager at Rocscience. I implemented the simulated annealing algorithm as a new optimization method in SLIDE 5.0, a slope stability analysis program developed by Rocscience. In addition to learning professional C++ and principles of object oriented programming, I had the opportunity to study the theoretical and practical aspects of non-convex optimization, especially the performance of stochastic methods when used on multi-modal functions. I also learned soil mechanics and slope stability principles.

This report was written entirely by me and has not received any previous academic credit at Rocscience or the University of Waterloo. The current work will be published in an engineering journal. I would like to thank Dr Brent Corkum for his excellent supervision and Dr Reginald Hammah for invaluable references and discussion. I also thank Dr Jim Hazzard, Damir Valiulin and Jeremy Smith for their programming tips and continued help during the program development. I would also like to thank Igor Pashutinski for carefully revising the manuscript. Finally, I would like to thank Dr John Curran, Dr Thamer Yacoub, and Ms Lesley-Ann Foulds for helpful discussions.

Sincerely,

Xiao Su
20251755

Contents

List of Figures

List of Tables

Summary

This work proposes hybrid simulated annealing for the global optimization of the critical failure surface in slope stability analysis. The aim of slope stability analysis is to find the critical surface along which failure of a soil or rock mass is likely to occur. This is achieved by minimizing a cost function known as the factor of safety $\mathbf{F(v)}$, for a position parameter $\mathbf{v}$. This position parameter can be either a circular surface (a problem with three control variables) or a general surface represented by vertices (a problem with n control variables). In the latter case, n can be very high, and the presence of multi-minima makes standard optimization highly inefficient.

Simulated annealing (SA) is a method of global optimization, and was first developed for solving combinatorial systems such as the traveling salesman problem and chip placement. The original SA has been applied recently to slope stability with varying degrees of success. A major issue with SA is the large number of iterations required and asymptotic convergence.

The current work proposes a hybrid simulated annealing (HSA) algorithm for the search of general failure surfaces. The method couples a Very Fast Simulated Annealing algorithm (VFSA) with an efficient searching technique which we will refer to as Local Monte-Carlo (LMC). The method was implemented in C++ and incorporated into the SLIDE 5 software developed by Rocscience. The VFSA is a state-of-art SA algorithm, relying on a probabilistic random walk and an exponentially decreasing schedule. The LMC relies on local exploration of each vertex, with a step-reduction mechanism as the search approaches the global minimum. Dynamic bounds for each control variable were applied.

HSA was found to be far superior to path-search for all verification and customer cases tested. A comparison with other global optimization methods demonstrates that HSA has a higher precision and employs significantly less iterations to find the global minimum. The hybrid algorithm couples the robustness of a global optimization algorithm with the speed and refinement of a local optimizer. Recommendations for future work include the use of non-linear interpolation between vertices, and parallel processing to increase computation speed.

1 Introduction

The analysis of the stability of slopes is essential to modern society, as it examines the safety of major human constructions such as dams, embankments and structures built along the sides of slope. At the core of slope analysis lies the challenge of finding the geological surface with the greatest instability or representing the most probable region in which the mass of soil or rock will slide. The potential instability is estimated by the factor of safety F, and critical failure is the surface with the lowest F.

The problem then becomes finding the surface with position vector $v(x_1, x_2, x_3, \ldots, x_n)$ which minimizes the safety function F. For examples, the parameters x_1 to x_n can be the centre and radius of a circular surface or the vertices of a non-circular surface. The challenge lies in that the cost function is often highly multi-modal, and the function domain can be strongly discontinuous. Most optimization methods, especially gradient-based, tend to get trapped in local minima. Simulated annealing (SA), on the other hand, is a global optimization method that allows up-hill as well as down-hill moves, being capable of escaping from local minima.

In the current section, some basic slope stability concepts are briefly reviewed. The original simulated annealing is then presented, its major drawbacks are discussed, as well as factors that led to the development of faster annealing algorithms. Finally, the inherent weaknesses of global search are discussed and the use of hybrid optimization is proposed.

1.1 Slope Stability Concepts

The factor of safety F is defined as the ratio of the total shear strength (S) available along a slip surface to the shear stress (S_m) that is actually mobilized along the surface due to actions of the weight of the soil mass and possible external loads:

$$F = \frac{S}{S_m}. \tag{1}$$

Many methods were developed to compute factor of safety, either from moment or force equilibrium, most of them pertaining to the well established class of limit equilibrium methods (LEM). Even though LEMs do not consider stress-strain relationships, they provide

an accurate estimate of the factor of safety without the need for extensive knowledge of initial conditions (Cheng et al., 2007a).

The LEMs generally result in a problem formulation which is statically indeterminate. To obtain a solution, different assumptions can be made on the distribution of internal forces. Thus, the quality of the solution is closely tied with the reality of these approximations. Other techniques exist for computing factors of safety such as strength reduction methods (SRM). These methods do not require assumptions on inter-slice shear force distributions and employ finite element techniques (Abramson et al., 2002). However, the SRM have disadvantages which include long solution time and the need for detailed knowledge of boundary conditions, the latter being unknown in many cases.

The flexibility and speed of LEMs lie in the method of slices. The earth mass within the trial surface is divided into vertical slices and at each slice, force and moment equilibrium resultants must be zero for static equilibrium (Abramson et al., 2002). The factor of safety at each slice is taken as the factor of safety F of the slope. It results in a non-linear equation with F as a root, usually found through iterations. Often, the procedures for solving the factor of safety equation do not converge (the solution "blows up"), due to inappropriate force assumptions or negative friction values (Duncan and Wright, 2005).

1.2 The Method of Slices

The method of slices is used by most computer programs, as it can readily accommodate complex slope geometries, variable soil conditions and the influence of external boundary loads (Abramson et al., 2002). The method divides the slope geometry into n vertical slices, creating $6n - 2$ unknowns (Abramson et al., 2002), as summarized in Table 1.

A common assumption is that the normal force on the base of the slice acts at the midpoint thus reducing the number of unknowns to $5n - 2$. From moment, force and Mohr-Coulomb relationships, only $n - 2$ assumptions are left to make the problem determinate (Table 1). How these last degrees of freedom are handled differentiate the available methods of analysis (Abramson et al., 2002). The most popular methods are reviewed below:

Table 1: Required variables for safety factor analysis

Equations	Condition
n	Moment Equilibrium for each Slice
$2n$	Force equilibrium in two directions for each slice
n	Mohr-Coulomb relationship between shear strength and normal effective stress
$4n$	Total Number of equations
Unknowns	**Variable**
1	Factor of safety
n	Normal force at base of each slice
n	Location of normal force
n	Shear force at base of each slice
$n-1$	Interslice force
$n-1$	Inclination of interslice force
$n-1$	Location of interslice force (line of thrust)
$6n-2$	Total number of unknowns

- *Ordinary Method of Slices (OMS)* — This method neglects all inter-slice forces, therefore relying on moment equilibrium alone. It is one of the simplest analysis procedures.

- *Bishop's Simplified Method* — assumes that all inter-slice shear forces are zero, reducing the number of unknowns by $n-1$. This leaves $4n-1$ degrees of freedom, leaving the solution overdetermined as horizontal force equilibrium will not be satisfied for one slice.

- *Janbu's Simplified Method* — Janbu also assumes zero inter-slice shear forces. Similar to Bishop's method, the solution is over-determined and does not satisfy moment equilibrium completely. Janbu presents a correction factor f_0 to account for this inadequacy.

- *Spencer's Method* — Spencer proposes a method that rigorously satisfies static equilibrium by assuming that resultant inter-slice forces have a constant, but unknown

inclination. It requires an iterative method for both the factor of safety and the inter-slice force inclination between the slices, matching the required $4n$ equations. It is considered a complete procedure (i.e. force and moment equilibrium are satisfied for all slices).

1.3 The Need for Global Optimization

For all LEMs presented above, the critical failure surface is generally found by trial and error. Many search methods were developed (Greco, 1996, Malkawi et al., 2001), but in most cases a good initial guess is needed, and the quality of results depends on the engineers' experience. The factor of safety function is often highly multimodal, and it is certainly non-smooth, being regarded as an $N-P$ complete problem (Cheng, 2007). This multi-modality generally results from multi-layering of the soil or mineral lenses.

Therefore, finding the critical slip surface pertains to the field of global optimization. Well-known deterministic methods such as the gradient or Newton's method inherently perform local optimization (Polyak, 1987). Further, Newton's method does not even distinguish minima (or maxima) from saddle points (Polyak, 1987). A class of techniques that has been relatively successful in global search is that of the stochastic methods. These methods are particularly useful for slope stability, as they are generally robust and relatively insensitive to problem type.

A promising start-up is simulated annealing (SA), a powerful stochastic method for global optimization.

1.4 Simulated Annealing (SA)

Simulated annealing (SA) was developed in the early 80s (Kirkpatrick et al., 1983) for combinatorial optimization. It differs from other search methods as it allows downhill as well as uphill moves, based on the underlying similarity between the physico-chemical process of annealing and the minimization of a function (Kirkpatrick, 1984). The method soon became as popular as the well-known genetic algorithms (GA), being used in various fields for image and signal processing, biology, geophysical inversion (Pei, Louie, & Pullammanappallil,

2007) and finance (Van Laarhoven & Aarts, 1987). Similarities between the two methods were soon drawn, due to the "natural" basis of the algorithms (Davis, 1987).

In annealing, a material (i.e. metal) is heated until its molecules acquire sufficient mobility (a melted state). Then, by decreasing the temperature slowly, the molecules undergo various configuration changes, always seeking for a lower energy state. If the decrease is sufficiently slow, a perfect crystalline solid will form, and the system will be at its minimum energy state. If the temperature is decreased fast, as in quenching, the molecules will collapse into an amorphous solid and have poor physical properties (its energy state will be at a "local minimum").

In statistical mechanics, the probability P of the atom existing at an energy state E can be modeled by a Boltzmann distribution $P(E) = \exp(-E/T)$. It is clear that the higher the temperature T the greater the chance of an atom existing at a higher energy state. This concept is implemented as a transition probability for the optimization through the well-known Metropolis criterion (Metropolis et al., 1953).

Similarly for SA, let E_1 be the energy of the system at configuration 1. To model the probability of the system changing to state 2 (at an Energy E_2), the Metropolis criterion consists in the acceptance/rejection probability $P(E) = \exp(-E/T)$. Simulated annealing uses this criterion as the acceptance probability for a random move - let the function $F(x)$ be defined. For the move to be accepted from x_1 to x_2, a random number r with uniform distribution between $(0, 1)$ is generated and compared with $P(dF) = \exp(-dF/T)$. If $r < P(dF)$, then the move is accepted. If higher, the move is rejected. It can be seen that if $dF < 0$ ($F_2 < F_1$), the move will always be accepted as the move is clearly downhill.

It can be seen that at high temperatures, more uphill moves will be accepted. It is the goal of simulated annealing to start the temperature T high enough so the search method can sample every valley of the parameter space. A "melting criterion" was initially proposed as a 0.8 ratio between accepted and rejected moves (Kirkpatrick, 1984). Then, by reducing the temperature the search becomes more selective and less up-hill moves are accepted, until at $T = 0$ only down-hill moves are accepted.

The algorithm was soon extended to continuous functions (Corana et al., 1987). The position vector becomes then analogous to the molecular arrangement of a metal, and the cost

function to the energy of the arrangement. The goal becomes to find the perfect arrangement of atoms which will minimize the overall energy of a system, by performing random-walks along the function domain.

In a more rigorous formulation, simulated annealing depends on three functions (Van Laarhoven and Aarts, 1987, Yao, 1995). All of these three functions are dependent on T_k, which denotes the temperature at the $k-th$ annealing iteration:

1. A generation function $G_{XY}(T_k)$, which generates a state Y based on a state X at the annealing temperature T_k.

2. An acceptance function $A_{XY}(T_k)$, which is the transition probability from one state to another. This is determined by the Metropolis criterion.

3. An annealing schedule $S(T_k)$, which determines the temperature reduction rule.

During the past two decades, intensive research has been invested in improving the performance of the method by proposing more sophisticated functions for $G_{XY}(T_k)$ and $S(T_k)$, such that global convergence can still be guaranteed at fewer iterations and a higher speed.

1.5 Improvements on Simulated Annealing

The original annealing has many issues, and is now considered in fact a simulated quenching (SQ) algorithm as its convergence is not guaranteed for every case (Ingber, 1993). These mainly refer to algorithms which use an exponentially decreasing schedule without an adequate generation functon, such as:

$$T_{k+1} = aT_k, \tag{2}$$

which fails to comply with the necessary convergence conditions. The constant a is a temperature decreasing parameter between 0 and 1. However, this is not to say that quenching methods are inefficient. In fact, for many specific problems in which the behavior of the system is well-known, simulated quenching can be far superior and should therefore be used.

Adaptive step-size for the random-walk has been proven to increase performance (Corana et al., 1987). By increasing the step-size if the search is going well (and vice-versa if badly),

the ratio between accepted and rejected moves is kept close to 1:1. Even though sufficiency conditions have not been proved for this algorithm, it was shown to be efficient in finding global minima of very hard test functions, such as Rosenbrock valleys of up to 10^{20} local minima.

The concept of using an adaptive step-size has been essential to the development of better annealing algorithms. The Classical Simulated Annealing (CSA) was the first annealing algorithm with a rigorous mathematical proof for its global convergence (Geman and Geman, 1984). It was proven to converge if a Gaussian distribution is used for $G_{XY}(T_K)$, coupled with an annealing schedule $S(T_k)$ that decreases no faster than $T = T_o/(\log k)$. The integer k is a counter for the external annealing loop, as will be detailed in the next section. However, a logarithmic decreasing schedule is considered to be too slow, and for many problems the number of iterations required is considered as "overkill" (Ingber, 1993).

An exponentially faster variant known as fast simulated annealing (FSA) was proposed by Szu and Hartley (1987) which uses a Cauchy distribution as $G_{XY}(T_K)$. It was heuristically proven that if coupled with an annealing schedule $T = T_0/k$, the method will converge rapidly, due to its capabilities of performing longer jumps in the parameter space. Ingber (1989), using a similar proof, proved the convergence of a schedule exponentially faster than the fast annealing, namely $T = T_0/\exp k$. This algorithm is commonly referred to as very fast simulated annealing (VFSA).

Ingber (1989) also proposed a different generation temperature for each dimension, and the re-annealing of temperature every couple of iterations. The combination of the VFSA with a re-annealing is often referred to as adaptive simulated annealing (ASA) (Chen and Luk, 1999, Ingber, 1993). It has been shown successful in fields such as signal processing (Chen and Luk, 1999) as well as wavelet estimation in geophysics (Velis and Ulrych, 1996). However, for highly dimensional problems such as slope stability analysis, the assumption of different generation temperatures is theoretically valid but its proper control is practically impossible due to frustration problems.

1.6 The Need for Hybrid Optimization

The original simulated annealing, as proposed by Kirkpatrick et al. (1983), was applied by Cheng (2007) to slope stability analysis with a fair amount of success. However, in a recent comparison (Cheng et al., 2007b) simulated annealing was found to require a large amount of iterations, and its accuracy is unsatisfactory, especially for multi-layer problems. Also, parameter tuning was also found to be an issue for the original SA algorithm. Due to the number of different slope geometries, the optimization method is required to be relatively insensitive to change in function characteristics. Similar problems were encountered with other global optimization methods when applied to slope stability problems. Currently, all optimization methods applied to slope problems have either accuracy, speed or function representation issues. In the work completed in the summer work-term of 2008, the ASA was successfully applied to the optimization of a circular surface. However, with the increase in the dimensionality of the problem, even ASA becomes inefficient. The curse of dimensionality is expressed in the $1/n$ exponent in the function for temperature annealing (see Section 2), in which n is the number of dimensions in the problem (Ingber, 1993). Further, the identifying aspect of ASA, parameter re-annealing, was implemented in the start of the current project and found to be inefficient for non-circular surfaces. The reasons and implications are detailed in Section 5, Discussion.

At high-dimensionality, convergence issues for stochastic methods are not related to any specific method, but instead are inherent to heuristic methods of global optimization. This has been noted not only for SA, but also for many other global search methods (Davis, 1987, Haupt and Haupt, 2004, Hedar and Fukushima, 2002). A practical approach to increase the efficiency, and to truly "nail down" the global minimum, is to combine two or more optimization techniques, resulting in the now popular field of **hybrid optimization**.

Hybrid methods have been used in a variety of applications, such as the optimization of alumina production (Song et al., 2008), flow-shop scheduling (Murata et al., 1996), as well as the optimization of molecular cluster (Moyano et al., 2002). The goal of hybrid optimization is to combine the power of the global optimizer in gravitating towards the region of the global minimum, with the descent speed of the local optimizer. The following modes of

hybridization were suggested for a genetic algorithm (Haupt and Haupt, 2004), however, the statements can be easily generalized for any global optimization method.

1. Running the global optimizer until it slows down, and letting the local optimizer take over. This requires the assumption that the first method is already in a valley close to the global minimum, which is not always the case as early convergence can occur (Haupt and Haupt, 2004).

2. Seeding the global optimizer with points found from a random-generation function. This approach was partly adopted in the hybrid simulated annealing algorithm, as a random initialization function makes the method far more robust.

3. After every set of iterations from the global optimizer, a local optimizer can be used, and the best solution can be updated into the scheme of the global search.

4. A fourth mode of hybridization can be applied specifically for functions with a clearly encapsulated generation function, such as SA. The use of an intelligent, local generation function can be used instead of a purely random-walk. This mode of hybridization is more intrinsic than the previous three.

Some modes of hybridization may be more effective than others depending on the model and the optimization algorithms involved. Hybrid optimization has been applied extensively with SA and GA, due to their global search capabilities. SA has been coupled with conjugate gradient for the simulation of mercury clusters (Moyano et al., 2002), as a type 1 hybridization technique. Alternatively, Hedar and Fukushima (2002) have proposed the use of SA in conjunction with the direct search method.

A popular hybrid method is combining SA with the simplex method by Nelder and Mead (1965). Kvasnicka and Pospichal (1997) propose the simplex as the state-generation function for the simulated annealing, that is, after every move made by the simplex, its update is decided by the Metropolis criterion. Hedar and Fukushima (2002) extend the hybridization: they combine the simplex-simulated annealing with a direct search method, and finalize the search with a simplex method, thus being both type 1 and 4.

A similarity shared between the different local optimizers, such as conjugate-gradient or simplex, is that they all possess some capability of reducing the search radius as the method closes on the global minimum. This can be understood intuitively: as the method applies the global minimum, it is desired for the accuracy of the optimization to increase. This is especially important in slope stability analysis, as weak layers act as very narrow valleys in the energy landscape. The similarity between these narrow regions and Dirac functions has been noted by Cheng (2007).

In the current work, we apply hybrid optimization by combining the very fast simulated annealing (VFSA) with an efficient local-Monte-Carlo (LMC) search, and can be classified under mode 1 mentioned above. The LMC shares similarities between the conjugate-gradient and the simplex in the fact that it descends towards the minimum and adjusts its step-size. It does not need gradient computation, and is very easily programmed. The major advantage is that it has been developed specifically for critical surface, and has been tested successfully for various slopes (Greco, 1996).

2 Methods

The **H**ybrid **S**imulated **A**nnealing (HSA) was implemented in the Slide 5 engine using C++. Slide is a 2D slope stability program designed for rock or soil slopes and contains a variety of modeling capabilities, including material shear strength, external loading, groundwater and support. The software already employs several search methods for non-circular surfaces, an important one being the Path-Search algorithm. The path-search resembles a pattern-search in the fact that it generates a multitude of non-circular surfaces and then computes the factor of safety for all of them. The Path-Search when combined with a Local Monte-Carlo (LMC) optimizer is fairly powerful, and is currently the most efficient technique for non-circular surfaces in SLIDE.

The HSA is composed of two parts: a **V**ery **F**ast *S*imulated **A**nnealing (VFSA), followed by a **L**ocal **M**onte-**C**arlo (LMC). Simulated annealing is a stochastic method, therefore requiring the use of random-number generators. For all purposes, pseudo-random number generation uses the `CRandomNumber` class developed by Rocscience. The class allows a variety of capabilities and carries a variety of statistical distributions. Finally, `CRandomNumber` allows the use of seeds for pseudo-random number generation, therefore making the optimization results reproducible.

Spencer's method is used for all cases as it considers both moment and force equilibrium, and is considered the simplest of the complete equilibrium procedures (Duncan and Wright, 2005). As the HSA is a combination of the VFSA and the **LMC**, the current section will describe each of these two methods separately.

2.1 Very Fast Simulated Annealing (VFSA)

The VFSA was implemented in the `Aslide.cpp` file under the function `NC_ASA_XY.cpp`. The algorithm is derived from a standard Boltzmann simulated annealing with a more sophisticated generation function and an exponential annealing schedule. The control mechanism of VFSA is ruled by two parameters: a generation temperature T_{gen} which decides on the radius of search, and an acceptance temperature T_{accept} that decides on the probability of visiting a state. The very fast simulated annealing relies mainly on two loops: (1) an internal

loop in which the algorithm performs N_{gen} random moves, with each move having a chance of being accepted according to a probability P_{accept}; (2) an external loop decreases T_{accept} and P_{accept}, which is a direct function of T_{accept}. A flowchart for the VFSA is shown in Figure 1, and the details for its implementation are given below.

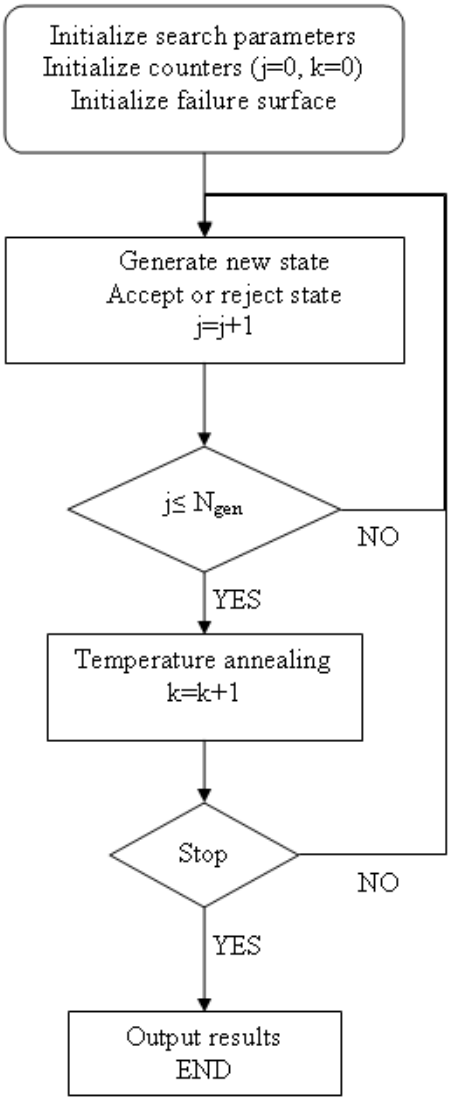


Figure 1: Flowchart for VFSA

We will give a quick description of the notation which will be used throughout the rest of this work. The index i refers to the control variable being optimized ($1 \leq i \leq n$). The index j refers to the internal iteration of the annealing algorithm, and the index k refers to the external annealing loop. At the start of every k-th annealing iteration, j is set to 0, and N_{gen} iterations are performed. Variables are subscripted with according indices, that is, on whether they are a function of its respective index. As an example, the Cauchy random variable $r_{i,j,k}$ refers to the random number generated for the $i-th$ control variable at the $j-th$ internal iteration and $k-th$ annealing iteration. Other variables such as the generation temperature $T_{gen,k}$ depend only on the annealing iteration, at least for the current VFSA formulation. It must be noted that for ASA, $T_{gen,k}$ is also a function of the control variable (therefore requiring the index i).

However, before the implementation of the algorithm can be discussed, the control variables must be clearly defined. As discussed before, the aim of an optimization algorithm is to minimize a function $F(v)$, in which v is a vector containing the control variables. For the case of a non-circular surface, $v = (x_1, x_2, ..., x_N, y_1, y_2, ..., y_N)$, where (x_m, y_m) are the coordinates for the m-th vertex, and N equals the number of vertices ($1 \leq m \leq N$). In the current investigation, a linear interpolation is assumed between the control vertices on the failure surface. The distinction between the number n of control variables and the number N of vertices must be emphasised. In a general geometric problem, $n = 2N$. For the specific case of slope stability problems, the number of degrees of freedom is reduced by 2 ($n = 2N - 2$) as the two extremity points of the failure surface can only move along the slope line. In the implementation of the original annealing (Cheng, 2007), it was suggested that equi-distant vertices should be used in order to reduce the degrees of freedom of the problem, thus making it easier to optimize . The equi-distant implementation assumes all x coordinates to be equi-distant, and only the slope points and the y-coordinates are allowed to move ($n = N - 1$ as opposed to $2N - 2$).

The equi-distant approach was initially adopted, and it was found out that the ability of the vertices to represent an arbitrary surface is greatly reduced. To compensate for this, more vertices must be used, which again increases the dimensionality. Therefore, the increase in dimension due to adding vertices offsets the decrease in dimension due to the equi-

distant assumption, and the balance is often negative. The approach taken in the current investigation is different from both the previously discussed methods: the HSA starts with a VFSA that allows both the x-coordinates as well as the y-coordinates to move, although with relatively few vertices (around nine to ten). This allows flexibility to the surface and a reasonable dimensionality to the optimization, although at a slightly decreased precision. The local optimizer then takes over and further optimizes the surface by adaptively inserting vertices as the search proceeds.

Finally, it was noted that the optimization is much more efficient when the domain of each control variable is properly bounded. This was found in the work with circular surfaces, and the results found in the previous work-term motivated the implementation of bounds for each control variable. The restriction of the domain for the variables contributes to the algorithm by allowing only kinematically feasible surfaces to be generated, as well as making the problem one of constrained optimization as opposed to unconstrained. We will discuss the use of bounds for the x and y coordinates.

2.1.1 Bounds for x-coordinates

The bounds for the horizontal coordinates (which we will denote as x) are set to be the equidistant divisions between the two slope points. Let x_1 be the first/left slope point, and x_N be the right slope point. Let D_m be the x-location of the m-th equidivision given by:

$$D_m = x_1 + (m-1)\frac{(x_N - x_1)}{N-2} \tag{3}$$

In which $D_1 = x_1$, and $D_{N-1} = x_N$. Note that there are $N-1$ divisions as opposed to N vertices. Mathematically:

$$D_{m-1} < x_m < D_m \tag{4}$$

The actual x-coordinates x_m for the vertices between the slope points are parameterized with respect to the width W of each division as:

$$PX_m = \frac{x_m - D_{m-1}}{W} \tag{5}$$

in which

$$W = \frac{[x_N - x_1]}{N-2}$$

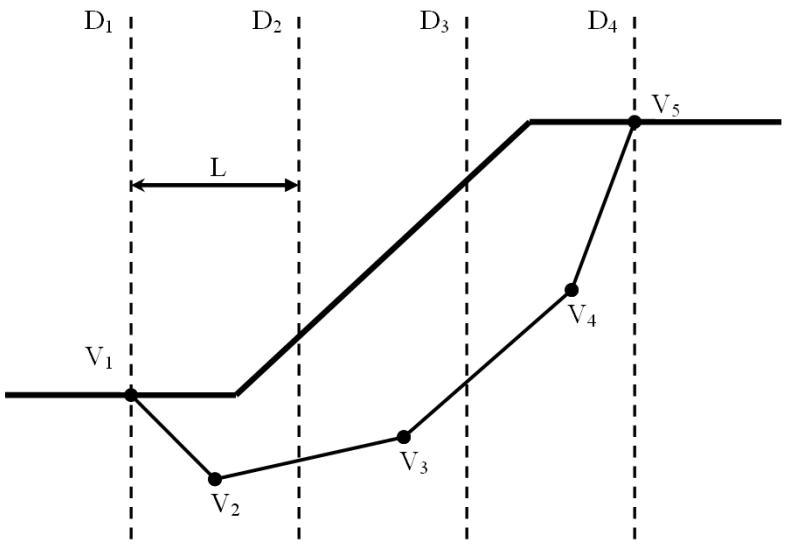


Figure 2: Bounds for the x-coordinates

The need for bounds is clear if we consider a random walk, especially for simulated annealing, for which at high temperatures the control variables often walk the span of the search space. Two vertices might occasionally overlap each other, creating difficulties in sorting, especially as the x- and y-coordinates are intrinsically related. A representation of the equi-divisions and the respective location of five vertices is shown in figure 2. It can be seen that there are four equidivisions for five vertices.

2.1.2 Bounds for y-coordinates

The use of a dynamic range for the y-coordinates has been proposed by Cheng (2007), as to increase the number of kinematically acceptable surfaces. Let us consider the surface in Figure 3 (Cheng, 2007), with six vertices ($m = 6$) represented by $V_1, V_2, \ldots, V_6$.

1. First, the two slope points V_1 and V_6 are defined. The x-coordinates for the various vertices are known previous to the calculation of the dynamic ranges for the y-coordinates.

2. The y-coordinate for V_2 is generated between the upper slope limit $y_{2,max}$ and the lower slope limit $y_{2,min}$.

3. With V_1 and V_2 generated, the bound for V_3 becomes the lines connecting V_1V_2 and

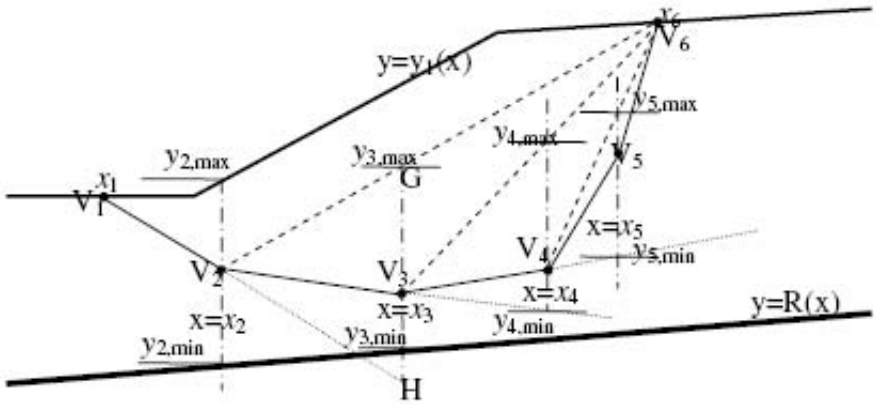


Figure 3: Representation of Dynamic Bounds by Cheng (2007)

V_2V_6. Let the function denoting the slope line be represented by $R(x)$, therefore the minimum bound for y_3 becomes the maximum of y_H for $R(x_3)$, and the minimum of y_G and $y_1(x)$.

4. Each control variable y_m is parameterized to PY_m as follows:

$$PY_m = \frac{y_m - y_{m,min}}{y_{m,max} - y_{m,min}} \tag{6}$$

The dynamic bounds were initially proposed by (Cheng, 2007) in order to generate kinematically feasible surfaces by allowing smaller domain ranges for each control variable, as well as preventing the generation of concave failure surfaces. In HSA, concave surfaces can still be produced (as in the physical world convex surfaces can indeed exist), although only at the local optimization phase.

2.1.3 Initialization of Surfaces

A random number between 0 and 1 using the `CRandomNumber` class is generated and assigned to each control variable. This is feasible and contains all surface possibilities due to the parameterization of the control variables. The initial acceptance temperature $T_{accept,in}$ is set to the initial factor of safety value. The initial value of T_{gen} is set to 1.0.

2.1.4 State-generation Function

In the very fast simulated annealing, the generation function is based on the product of a D-dimensional Cauchy distribution. The state of the system v is updated according to this rule from v_j to v_{j+1} in the j-th iteration. It must be noted that v is a vector containing the control variables, and is n values long. The index j is set to 0 at the start of the internal loop, and incremented until it reaches N_{gen} (see Figure 1) . The generation is based on a random walk for each control variable i, ruled by the following functional relationship:

$$v_{i,j+1} = v_{i,j} + r_{i,k} L_i \tag{7}$$

where L_i denotes the characteristic length of the problem for the i-th control variable. In the original formulation by Ingber (1989), $L_i = U_i - B_i$ in which U_i represents the upper bound of the i-th control variable and B_i represents the lower bound. In the current algorithm L equals unity as each control variable is normalized with respect to its domain, according to PX_m and PY_m in equations 5 and 6.

The random variable $r_{i,k} \in [0, 1]$ follows a Cauchy distribution, and is mapped from a uniform random variable $u_i \in [0, 1]$ using the following function:

$$r_{i,k} = \operatorname{sgn}(u_i - 0.5) T_{gen,k} \left[\left(1 + \frac{1}{T_{gen,k}} \right)^{|2u_i - 1|} - 1 \right] \tag{8}$$

The subscripts i and k show that a different r is generated for each control variable, all of them a function of the current annealing iteration k. This relationship between r and k can be seen in the previous equation through the control parameter $T_{gen,k}$. In the C++ program, the generation of the Cauchy variable and the random number is performed by the function `void GetStep()`. The walk is performed by the function `void PerformWalk()`. The VFSA presented here does not perform a re-annealing, therefore a single generation temperature T_{gen} is equal for all control variables ($T_{gen,i,k} = T_{gen,k}$ for $1 < i < n$).

2.1.5 Acceptance Function

The acceptance function is dependent on the acceptance temperature $T_{accept,k}$ as well as v_{j+1} and v_j. The annealing algorithm uses the difference between the values of the objective

function $F(v_{j+1})$ and $F(v_j)$ to decide on whether to accept or reject the new move. Let $dE = F(v_{j+1}) - F(v_j)$. If $dE \leq 0$, the acceptance probability $P_{accept} = 1$. Else, if $dE \geq 0$, the acceptance probability is given by the following equation (Chen and Luk, 1999):

$$P_{accept} = \frac{1}{1 + exp(dE/T_{accept,k})} \tag{9}$$

A uniform random number u is generated and compared with P_{accept}. If $u \leq P_{accept}$, the move is accepted. If $u > P_{accept}$, the move is rejected. It can be seen that down-hill moves are always accepted, whereas up-hill moves are only sometimes accepted. The acceptance criterion stated above is a modification of the original criterion. The current criterion is more selective than the original one, however, it preserves the main features of the Metropolis criterion.

2.1.6 Scheduling Function

After N_{gen} states are generated, the temperature annealing is performed in the function `void ReduceTemperature`, according to an exponentially decreasing schedule (Ingber, 1989). The schedule corresponds to the function $S(T_k)$ discussed in the introduction. The acceptance temperature $T_{accept,k}$ is decreased according to:

$$T_{accept,k} = T_{accept,in} \exp\left(-ck^{1/n}\right) \tag{10}$$

Similarly, the generation temperature $T_{gen,k}$ is decreased according to:

$$T_{gen,k} = T_{gen,in} \exp\left(-ck^{1/n}\right) \tag{11}$$

It must be noted that the index k is 0 at the start of the search, and is incremented once N_{gen} states are generated. This way, at the first temperature annealing, $k = 1$ and the temperature is decreased according to equation 11. The control parameter c is a user-defined control parameter (Chen and Luk, 1999). This value is assumed to be constant throughout the search, and the same for all dimensions. Its optimal value was found by tuning, although a rigorous sensitivity analysis is recommended. The value of $c = 8.0$ was adopted, and it has been noted by (Chen and Luk, 1999) and confirmed in the current study that values

in the range of 1.0 to 10.0 are adequate. In the VFSA, the generation temperature and the acceptance temperature are decreased according to the same rule. However, they do not always have the same value, due to the control mechanisms placed on the search. In ASA, the value of the generation temperature is often updated through re-annealing (Chen and Luk, 1999), in the current algorithm, the value of the acceptance temperature is often re-scaled.

2.1.7 Stopping Criterion

The stopping criterion consists in comparing the last few safety factors found at each N_{gen} iterations (Cheng et al., 2007b). The values of $f_{opt}[k]$ are compared with $f_{opt}[k-j]$, where j is an integer between 0 and n_ϵ. If the difference $f_{opt}[k] - f_{opt}[k-j] < f_{tol}$ for all j, the search is stopped, where f_{tol} is a pre-defined tolerance level. In other words, if there has not been any visible improvement for the global optimum in the previous n_ϵ consecutive runs, the algorithm is to be stopped. An alternative criterion often used in standard annealing (Kirkpatrick, 1984) is to stop the search once the acceptance temperature has almost reached zero. However, the placement of control mechanisms often renders the latter stopping criterion meaningless. For example, in ASA, the acceptance temperature is frequently reset to the latest value of the cost function, therefore preventing the acceptance temperature from decreasing (Chen and Luk, 1999, Ingber, 1989).

2.1.8 Control Mechanisms for VFSA

Corana et al. (1987) proposed a control mechanism that maintains the ratio r between accepted states and rejected states close to 1.0. This mechanism shortens or elongates the walk in order to achieve this stationary state.

In the current work, we use the acceptance temperature T_{accept} to achieve a similar effect. Instead of forcing r to be 1.0, we allow it to be within a range from 0.5 to 2.0, and adjust T_{accept} according to r itself:

- If $r > 2.0$, $T_{accept} = 0.5T_{accept}$;
- If $r < 0.5$, $T_{accept} = 2T_{accept}$.

where the parameters were chosen from experience.

Another control mechanism was implemented so the algorithm can perform more iterations at higher temperatures. In the current annealing algorithm, the number of generated states N_{gen} is halved for consecutive annealing iterations. This approach was adopted as it is clear that annealing requires more iterations in the start of the search to sample the function domain completely. For the models tested, the parameter N_{gen} was initially set to be $1000n$ for which n is the number of dimensions. In the three consecutive annealing iterations, N_{gen} is consecutively reduced to $500n$, $250n$ and finally $125n$. These control mechanisms are an attempt to maintain a trade-off between the semi-global and local search necessary for the optimization (Szu and Hartley, 1987).

2.1.9 Error Handling and Bounds Check

A crucial issue is the validity of the failure surfaces generated, as well as whether each control variable is within its domain. The parameters PX_m and PY_m must be within 0 and 1.0. It was initially proposed that if a parameter is generated outside its domain, reset it to its state. This was found to be inefficient, as the point would remain stationary for the given j-th iteration.

An alternative solution for bounds check is to re-generate the surface by using a Cauchy distribution properly truncated with respect to the bounds. Let us say the parameter PX_m has a value of 0.2. The `void CheckBounds()` function will now re-generate a new random variable $r_{i,k}$ using equation 8. However, the function will also truncate $r_{i,k}$ with respect to the position of PX_m and its bounds. As PX_m has a domain within [0,1], the bounds check will truncate $r_{i,k}$ between the values of -0.2 and 0.8. That is, values of $r_{i,k} < -0.2$ or $r_{i,k} > 0.8$ are discarded. In general terms, for a parameter P_i, the bounds check truncates the Cauchy random number according to $-P_i < r_{i,k} < 1 - P_i$. This way, we avoid wasting iterations and as such, keep the algorithm constantly moving.

As for error-handling, the following procedure is adopted: if an invalid surface is generated, the optimal surface so far is retrieved and a new random-walk is performed using this surface. If this procedure fails for a certain number of trials, the algorithm is reset to the current optimum location.

2.2 Local Monte-Carlo (LMC)

The Local Monte-Carlo (LMC) search was developed by Greco (1996), with the aim of motivating the use of stochastic algorithms as opposed to the then more popular deterministic methods. The LMC can be considered a local explorer, exhaustively trying all nearby move-sets and adjusting its step-size during the search.

The LMC uses non-parameterized x and y coordinates of the slip surface as its control variables. For notation, let the n vertices of the surface be represented by $V_1, V_2, \ldots, V_i, \ldots, V_N$, with the respective coordinates $(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i), \ldots, (x_N, y_N)$.

The random-walk for the LMC has two procedures: an exploration phase and an extrapolation phase. In the extrapolation phase, one vertex of the current trial surface is shifted. The safety factor associated with the new surface is then computed and compared with the value of the original surface. If the function was optimized, then the new surface is accepted, otherwise, the surface is discarded.

In the extrapolation phase, the total displacement obtained in the exploration phase is repeated, and the surface is updated if the corresponding safety factor is smaller.

Exploration Phase: In the exploration phase, each vertex of the current slip surface is randomly moved in an attempt to reduce the safety factor. Thus, at iteration j, vertex i is shifted from point $< x_i^j, y_i^j >$ to point $< x_i^{j+1}, y_i^{j+1} >$ where

$$x_i^{j+1} = x_i^j + \xi_i \tag{12}$$

in which ξ_i is a random displacement in the i-th vertex given by:

$$\xi_i = N_x R_x D x_i^j \tag{13}$$

where R_x is a random number extracted from a uniform distribution in the range (-0.5, 0.5) and Dx_i is the width of the search step in the x-direction. N_x is a directional parameter with a value of -1, 0 or 1. An identical move-set applies for the y-coordinate, in which a displacement η_i is made with a random number R_y, a step-size Dy_i and a directional parameter N_y. For each vertex, N_x and N_y form eight combinations for the same R_x and R_y (see Figure 3), allowing the surface to exhaustively search its close neighborhood.

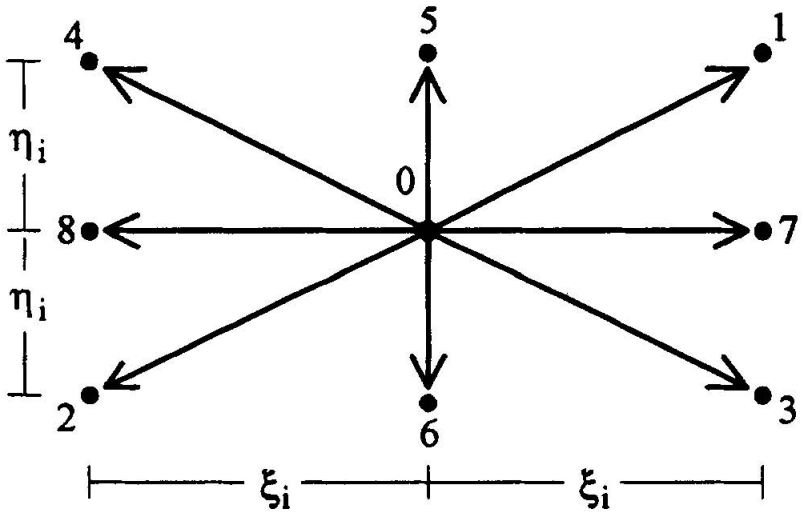


Figure 4: Possible Displacements for Exploration Phase Greco (1996)

Step Adjustment: If the displacement is successful, the following update is made to the width of each search step. Else, if no trial is successful for vertex i, a step reduction for the successive step $j+1$ occurs:

$$Dx_i^{j+1} = Dx_i^j(1-\epsilon) \tag{14}$$

$$Dy_i^{j+1} = Dx_i^j(1-\epsilon) \tag{15}$$

where ϵ is a number bettween 0 and 1. The value of ϵ must be carefully chosen, as a small value leads to a long computation time whereas a high value may lead to early convergence. The default ϵ in SLIDE is 0.75.

Extrapolation Phase: Once the exploration is performed, the extrapolation phase ensues. The movement of the vertices that occurred in the previous phase is repeated, and the new slip surface is then generated with vertices as given. The slip surface is checked with respect to the boundaries, and they are re-updated if necessary. If the safety factor is now at a minimum, the new vertices are updated. A new extrapolation is performed.

Stopping criterion: At the end of each search step, for both exploration and extrapolation phases, it is necessary to check if the current slip surface can be assumed as the critical one and if the algorithm can be stopped.

In the proposed method, the iterative procedure is stopped, and the current point $\mathbf{S}^{j+1}$ is assumed as minimum when the following two criteria are simultaneously verified:

$$Dx_i^{j+1} < \Delta \quad \text{and} Dy_i^{j+1} < \Delta \quad \text{for every} \quad i = 1 \quad \text{to} \quad n.$$

$$\left|F(S^{j+1}) - F(S^j)\right| < \delta$$

where Δ is the lowest admissible width for the search range; and δ=tolerable difference between the values of safety factors in subsequent iterations.

2.3 Path-Search

The path search in SLIDE is a pattern-search technique based on the STABL program (Boutrup et al., 1979). Trial surfaces are generated from a number of initiation points with equal horizontal spacing along the slope line.

The direction θ of the first segment is calculated according to the following formula:

$$\theta = \alpha_2 + (\alpha_1 - \alpha_2)R^2 \tag{16}$$

Where α_1 and α_2 are the counterclockwise and clockwise direction limits (see Figure 5), and R is a random function `Ranf(x)` (Boutrup et al., 1979). The use of R squared introduces a bias so that angles closer to the clockwise direction limit are more likely. This bias is necessary to obtain a good distribution of completed surfaces. The rest of the segments are generated with the equation:

$$\theta = \theta_2 + (\theta_1 - \theta_2)R^{1+R} \tag{17}$$

In which θ_1 and θ_2 are the new limits as defined by the Path-Search algorithm (Boutrup et al., 1979). The path-search bears similarities with the circular grid-search in SLIDE: it generates the surfaces without requiring feedback from the model.

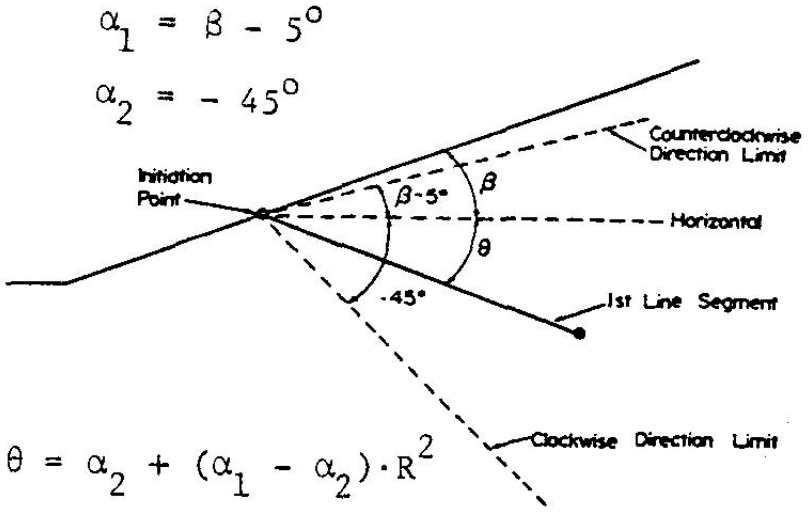


Figure 5: Segment Initialization and Bounds for Path-Search (Boutrup et al., 1979)

3 Results

The optimization methods were tested on fifty verification and twenty-six customer examples. The verification cases are actual engineering cases including a variety of slopes, embankments, dams, as well as multiple material layers and support properties. These test-cases can be found in the Examples folder of every commercial SLIDE version. The customer cases on the other hand consist of models sent by SLIDE users for individual engineering projects, and often present more complex geological features. The numbering for the original cases is preserved for the purposes of this report.

3.1 Verification Cases

The verification cases from SLIDE are a compilation of published examples in engineering journals and conference proceedings, and include a set of five slope cases as part of a survey sponsored by ACADS (Association for Computer Aided Design), in 1988 (Rocscience, 2006). Fifty examples were selected, comprising a variety of geological cases, with different drainage conditions and material properties for each of them. A summary of the optimization results is given in Table 2. The success rate refers to the percentage of verification cases for which the given method found the global minimum. The global minimum is assumed to be the lowest safety factor found by the methods.

Table 2: Summary of Results for Verification Cases

	VFSA	HSA	PATH-SEARCH
Success rate within 0.1(%)	100%	100%	72%
Success rate within 0.01(%)	88%	100%	56%
Initialization rate (%)	100%	100%	90%
Average time (s)	<497	497	72
Average No. of iterations	57,835	62,835*	9,000*

* The maximum LMC iteration number of 4000 is assumed, as well as the maximum 5000 iterations from the Path-Search. The 4000 from the LMC was added to both the HSA and Path-Search. There were 5 uninitialized examples from the path search.

Detailed results are presented in Tables 4 and 5. The performances are compared with respect to success in locating the global minimum, speed and the capability of generating a valid initial surface (initialization rate). Due to the fact that many cases show tension zones, a check for tensile normal stresses was performed in SLIDE for all verification cases in order to obtain critical failure surfaces that are kinematically feasible.

The success of finding the global minimum is measured in terms of accuracy in finding the lowest safety factor, with the results for both 0.1 and 0.01 tolerance levels being presented. From Table 2, it can be noticed that the HSA successfully finds all global minima within 0.01 of the cost function value. The verification cases were the basis for the development of the final version of the algorithm, including parameter-tuning for simulated annealing. The parameters chosen for the VFSA when applied to the verification cases are:

VFSA: $N_{gen} = 1000n$ and then decreased as mentioned in Section 2, $c = 8$, $T_{in} = 1.0$, with stopping criterion $N_\epsilon = 5$, where n=number of dimensions and n_{pts}=number of points. For all cases, n_{pts}=10, therefore n=18 (an 18-dimensional problem). A safety factor tolerance of 1e-06 was used, and a stopping criterion tolerance of 1e-04 was chosen.

LMC: N_{max}=4000, $f_{tol} = 1 \times 10^{-9}$, $\epsilon = 0.75$ (see Section 2 for details). The number of vertices is optimized as the optimization progresses.

Path-search: Number of surfaces generated = 5000.

It was noticed that all methods were usually slower for slopes using material properties that are not the standard Mohr-Coloumb, such as Hoek-Brown and Barton-Bandis.

3.2 Customer Cases

The customer cases are part of the Rocscience library, and they consist in challenging models sent by SLIDE users. Many of these models have dozens of material layers, often arranged in checkerboard or layered forms. Some of these user cases have input problems to the models, creating ill-conditioned slopes. A set of twenty-six user cases was selected, and they represent both the complexity and variety of the slopes encountered in every-day rock engineering. Table 3 summarizes the results for the twenty-six user cases. Detailed results are presented in Table 6.

Table 3: Summary of Results for Customer Cases

	HSA	PATH-SEARCH
Success rate within 0.1(%)	100%	55%
Success rate within 0.01(%)	92.3%	44%
Initialization rate (%)	100%	93%
Average time (s)	1207	200
Average No. of iterations	60,396*	9,000*

Table 4: Performance of optimization methods for all verification cases tested. Safety factors labeled in red indicate cases in which HSA had found a significantly lower factor of safety than the Path-Search — Part 1

Verification #	FOS PATH-SEARCH	FOS VFSA	FOS HSA	Time HSA	Number of Iterations
1	0.981	0.983	0.982	116	52500
2	1.579	1.576	1.553	159	70000
3	1.358	1.363	1.360	124	55000
4	0.979	0.981	0.977	172	50000
5	Uninitialized	1.948	1.948	76	50000
6	Uninitialized	1.948	1.948	69	50000
8	1.221	1.266	1.222	205	50000
9	0.725	0.733	0.708	150	50000
10	1.489	1.491	1.489	195	62500
11	1.108	0.812	0.812	173	50000
12	1.918	1.031	1.031	2541	71500
14	1.388	1.393	1.385	277	49500
15	0.413	0.418	0.414	196	50000
16	1.098	1.103	1.101	156	45000
19	1.400	1.406	1.400	238	72500
20	1.086	1.091	1.088	158	50000
21	1.982	1.990	1.983	199	58500
22	1.292	1.297	1.292	129	54000
23	4.029	2.494	2.494	997	63000
24	1.744	1.410	1.395	136	52500
25	0.943	0.943	0.943	439	55000
26	Not Initialized	0.787	0.787	183	50000
27	0.135	0.109	0.108	100	45000
28	0.008	0.007	0.004	254	55000
29	0.354	0.060	0.060	925	77500

Table 5: Performance of optimization methods for all verification cases tested. Safety factors labeled in red indicate cases in which HSA had found a significantly lower factor of safety than the Path-Search — Part 2

Verification #	FOS PATH-SEARCH	FOS VFSA	FOS HSA	Time HSA	Number of Iterations
30	Not Initialized	1.050	1.050	190	50000
31	Not Initialized	0.861	0.861	220	50000
32	0.861	0.799	0.799	742	50000
41	1.666	1.672	1.671	1082	47250
42	1.840	1.846	1.814	199	82500
43	1.472	1.310	1.310	2870	79750
44	0.976	0.977	0.976	243	50000
45	2.779	2.787	2.778	79	50000
46	2.500	2.499	2.499	101	45000
47	1.440	1.053	1.053	1963	88000
48	0.983	0.915	0.915	1794	92500
49	1.423	1.422	1.365	489	50000
50	1.088	0.361	0.361	274	47250
51	0.984	0.987	0.981	683	50000
52	2.006	2.012	2.006	181	52500
53	0.761	0.759	0.758	633	62500
54	1.154	1.156	1.152	477	55000
55	1.292	1.298	1.294	209	80000
56	1.288	1.290	1.287	198	87000
57	1.368	1.375	1.371	130	67500
58	0.355	0.227	0.227	520	50000
59	0.347	0.227	0.227	379	50000
60	1.060	1.007	1.007	1896	50000
61	1.363	1.362	1.360	461	66000
62	0.999	1.001	0.999	510	50000

Table 6: Performance of optimization methods for all customer cases tested. Safety factors labeled in red indicate cases in which HSA had found a significantly lower factor of safety

Case #	FOS PATH-SEARCH	FOS HSA	Time HSA	Number of Iterations
1	0.751	0.754	2043	50000
2	1.100	1.156	179	15000
3	1.214	1.216	893	57500
4	1.023	0.168	2437	50000
6	1.458	1.317	1326	65000
7	1.122	1.122	1244	57750
8	1.896	1.635	450	50000
9	1.329	1.217	1319	57500
10	Uninitialized	1.406	18	21000
11	Uninitialized	2.003	124	30800
12	1.451	1.071	228	80000
13	1.385	0.819	500	55000
14	0.871	0.868	398	50000
15	1.681	0.894	2375	67500
16	0.663	0.446	684	60000
17	1.075	0.982	179	57750
20	3.993	4.005	390	87000
21	1.456	1.457	232	102500
22	1.306	1.306	414	41400
27	1.401	1.391	375	27600
28	0.748	0.748	786	21000
32	0.780	0.535	263	21000
33	2.718	2.704	10667	32000
34	1.023	0.811	1454	62000
36	0.889	0.877	2157	21000
37	1.010	1.011	260	28000

4 Discussion and Explanatory Examples

4.1 Method Comparison — Verification Cases

The results in Table 2 demonstrate that for the verification cases the HSA is extremely efficient, finding the global minimum in 100% of the cases. This can be attributed primarily to the power of the VFSA. The VFSA algorithm by itself finds all surfaces within 0.1 of the global minimum, and achieves an accuracy of 0.01 for 88% of the verification cases. We can conclude that the VFSA by itself is highly successful in locating the neighborhood of the global minimum in all cases, with the 0.01 refinement being effectively carried out by the LMC.

4.2 Verification #15 - An Improved Function Representation

The advantages of using hybrid optimization can not be fully appreciated from a simple numerical point of view. Often-times the improvement between a non-smoothened and a smooth surface is very small in terms of safety factors. However, even for these cases, the insertion of vertices offers the advantage of being able to physically model the failure more accurately than otherwise.

This gain may be purely conceptual, but it definitely contributes insight to the physical behavior of the rock or soil mass. Verification #45 illustrates these two major advantages of using a local optimizer. Figure 6 shows the surface found through a stand-alone VFSA, initialized with 10 vertices. The safety factor of 2.787 found is extremely near the global minimum of 2.778.

A magnification (Figure 7) of the surface cutting the slope illustrates a lack of accuracy in the surface representation for the solution from VFSA.

With the hybrid algorithm, the surface (Figure 8) representation is much improved, and we can clearly see that the failure surface is actually composite, with both linear and approximately circular parts. The representation of the circular section is much smoother with the extra vertices inserted by the LMC (Figure 8).

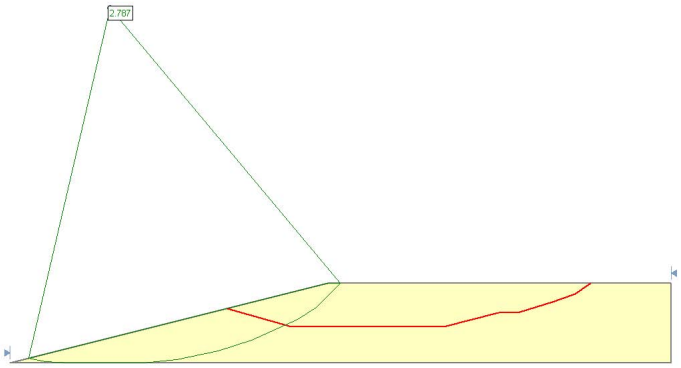


Figure 6: Very Fast Simulated Annealing for Verification Case #45

Figure 7: Rough Surface Generated by VFSA for Verification Case #15

4.2.1 Verification #24 - Composite Surface

Verification #24 is also a case in which the Path-search failed to find the global minimum. The model consists of a slope with three layers with different un-drained shear strengths (Rocscience, 2006). The material properties are given in Table 7.

Table 7: Material properties for Verification Case #24

	C_μ (kN/m^2)	γ (kN/m^3)
Upper Layer	30	18
Middle Layer	20	18
Bottom Layer	150	18

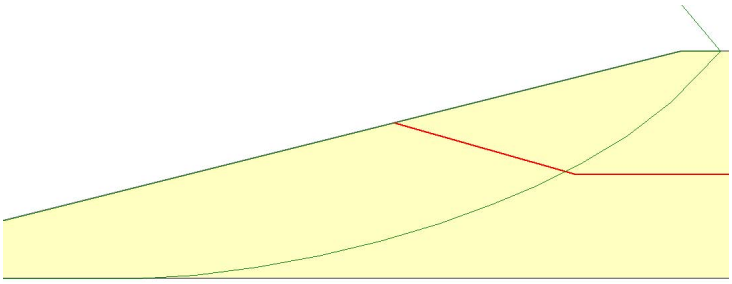


Figure 8: Smooth Surface Generated by HSA for Verification #15

The middle layer is the weakest layer, with a high-cohesion layer on the bottom. The minimum of 1.394 was found to be a log-spiral surface that flattens into a linear slip plane as it touches the hard-layer (Figure 9). Path-search fails to locate the minimum (FOS=1.744), even when the number of surfaces generated was set to 60000 (Figure 10). Verification #24 offers interesting insight into the HSA - the method is flexible enough to represent both circular and polygonal characteristics in the same failure surface.

4.2.2 Verification #48 - Slope Stability with Reinforcements

A final verification case was selected to represent slopes containing reinforcements, in this case, soil nailing. Soil nailing is a method of in situ reinforcement and uses passive inclusions that will be mobilized if movement occurs. The design of nailed excavations and slopes are generally based on limit equilibrium analyses in which critical potential failure surfaces must be assumed (Abramson et al., 2002, Duncan and Wright, 2005). Slope analysis with reinforcements are an important class of problems that global optimization must be able to solve.

Table 8: Material properties for Fontainebleau Sand Verification Case #48

Material	C_μ (kN/m^2)	f' (degrees)	γ (kN/m^3)
Fontainebleau Sand	3	38	20

Verification #48 examines the Clouterre Test Wall (Rocscience, 2006), constructed in Fontainebleau sand and failed by backfill saturation. The material properties of the sand are given in Table 8.

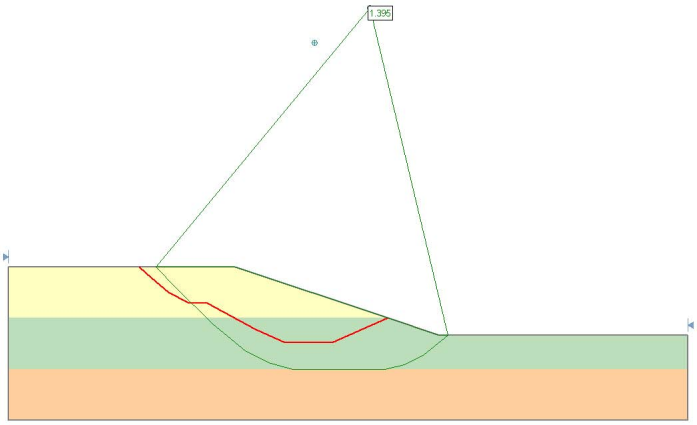


Figure 9: Hybrid Simulated Annealing for Verification #24

This test was carried out as part of the French national project on soil nailing. The test wall is reinforced by seven rows of soil nails with a shotcrete plate weighing 13.2 kN/m, and is modeled as a point load acting on the wall face. The properties for the support are given in Table 9.

Table 9: Soil Nail Properties for Verification #48

Type	Out-of-plane Spacing (m)	Tensile Strength (kN)	Plate Strength	Bond Strength (kN)
Fontainebleau Sand	1.5	1.5	59	7.5

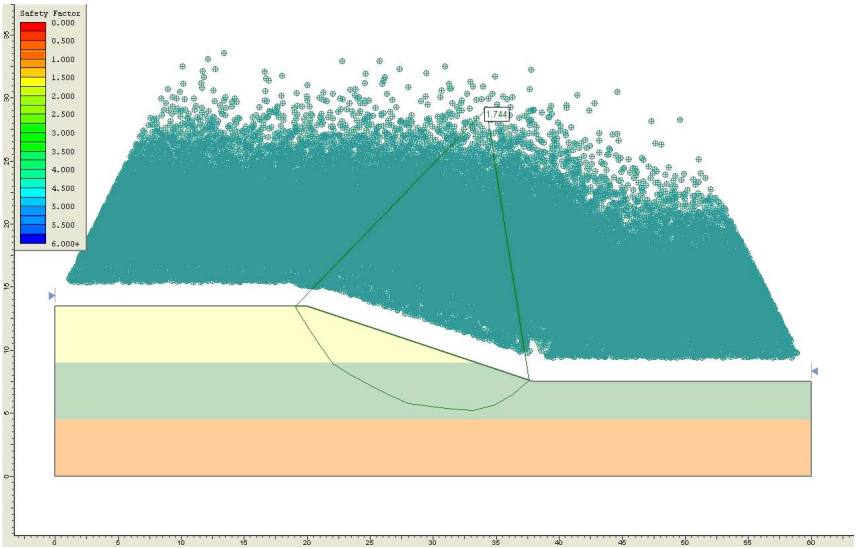


Figure 10: Path-Search with 60,000 surfaces for Verification #24

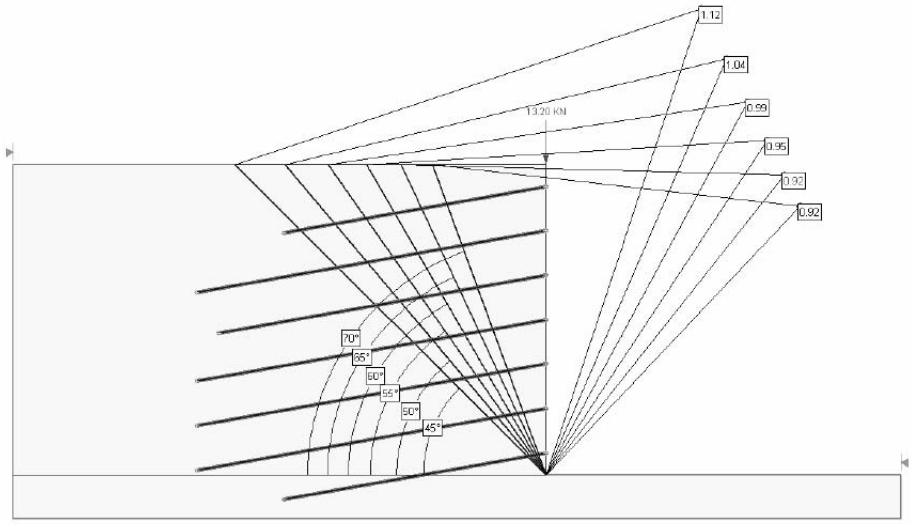


Figure 11: Plane angles for Clouterre Earth Wall - Verification #48

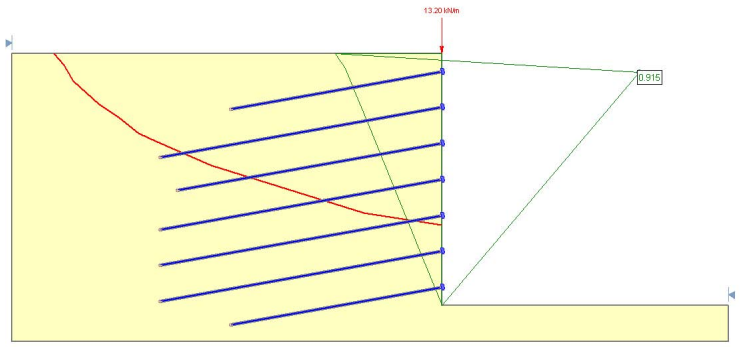


Figure 12: Hybrid Simulated Annealing for Verification #48

Verification #48 was designed to examine the relationship between the failure plane angle and factor of safety (Figure 11) in which the primary resistance against failure is friction generated by soil weight. The global minimum from the original analysis was 0.92 for an angle between 65^o and 70^o, by Spencer's method. The HSA successfully finds the global minimum within 22000 iterations (see Table 5). The angle of the plane slip found by HSA matches that of the original analysis (Figure 12).

From the point of view of a stochastic optimization, locating the global minimum for Verification #48 is a challenging task. A single, linear slip surface implies aligning all vertices, a task that is made extremely difficult by the random-nature of the generation function. The success of Verification #48 can be attributed to VFSA alone, as the global optimizer found the safety factor of 0.917 (see Table 5). This is due to the capability of the method to reduce the neighborhood of search to an almost point-like radius (see the discussion on T_{gen} in Section 2). The Path-Search, on the other hand, fails to align all vertices of its surface, as seen in Figure 13, finding a global minimum of 0.983.

4.3 Method Comparison — Customer Cases

4.3.1 Customer #04 - Extreme Multi-layering

From Table 3, it can be seen that the customer cases are significantly more challenging than the verification cases, both due to the longer time required to optimize (1207s as compared to 200) as well as the success rate of the path-search - only a 55% rate of finding the surface

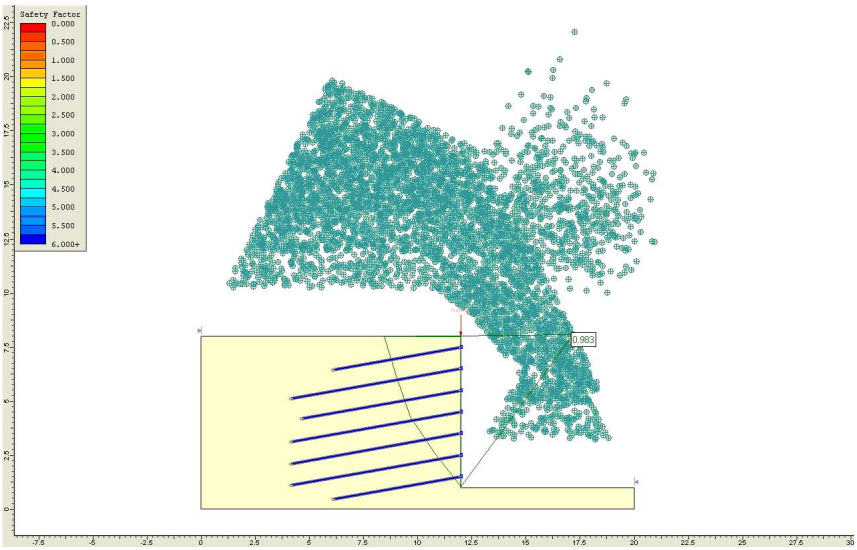


Figure 13: Path-Search for Verification #48

within 0.1 of the lowest safety factor. The large speed difference between the HSA and the Path-Search are due to the correspondingly higher number of iterations. However, the success rate achieved is almost double, therefore signifying that Path-Search faces serious early convergence problems with more complex models. From an engineering perspective, an average time of twenty-minutes for finding the global minimum should be considered negligible if compared to the resources and implications associated with correctly finding the most probable failure surface.

Customer case #4 stands for a challenging slope line, with dozens of material properties and over twenty layers, some of them relatively thin. The results from the HSA (FOS=0.168) can be seen in Figure 14. The minimum found by HSA is very challenging, as it is almost point-wise due if compared to the size of the entire slope. The path-search was able to find a failure surface, but clearly local (Figure 15). This same failure surface was also a challenging example for the circular search method tested in the work from the summer of 2008, in which only Adaptive Simulated Annealing was able to find the minimum.

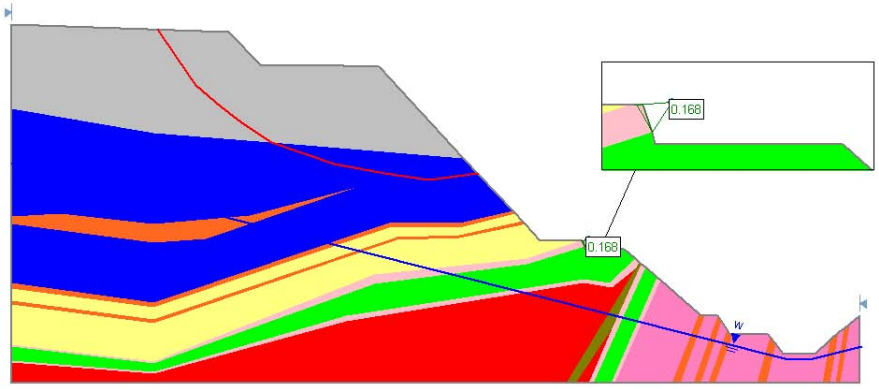


Figure 14: Hybrid Simulated Annealing for Customer Case #4

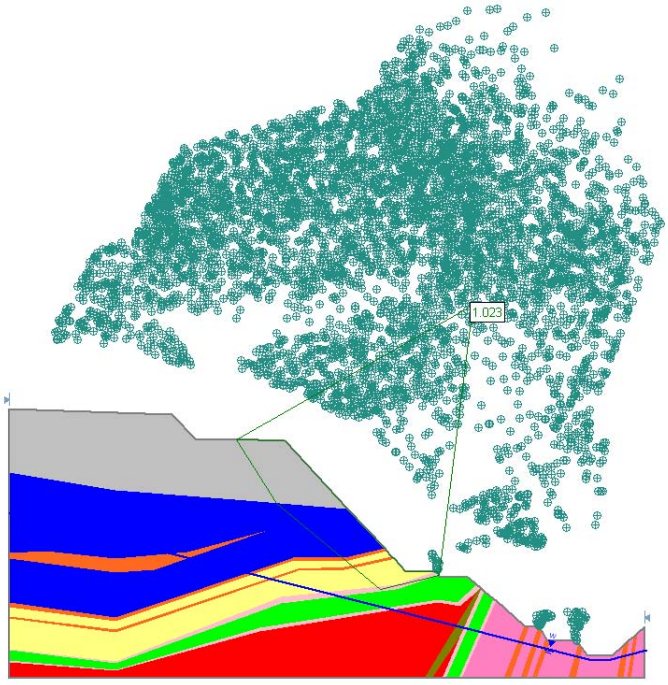


Figure 15: Path-Search for Customer Case #4

4.3.2 Customer #34 - Physical Modeling and LEMs

Verification #34 offers interesting insight into the nature of limit equilibrium methods. Before the tension-check was added to the model, the HSA successfully found the global minimum of the problem (Figure 16), which surprisingly displays a safety factor of 0.003. This is too low to be a realistic number, and the problem in this case lies not with the optimization, but rather with the modeling of the slope problem and the nature of LEM itself.

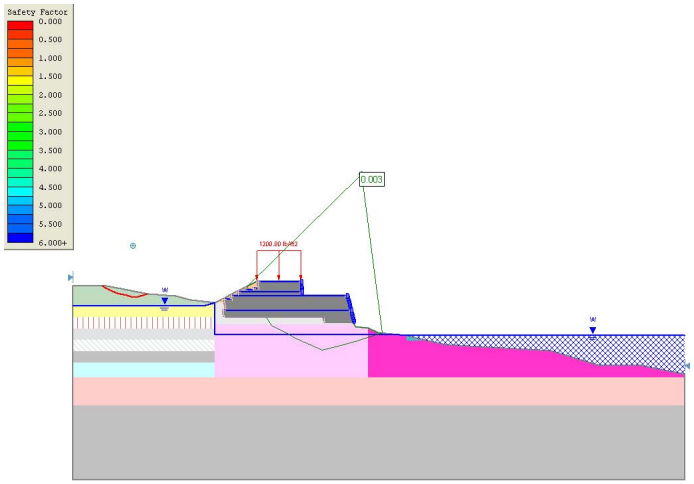


Figure 16: Hybrid Simulated Annealing for Verification Case #34

When significant tension develops, it can cause numerical problems in slope stability calculations (Duncan and Wright, 2005), and the engineer should employ techniques to reduce the effects of tension, including introducing a tension crack or adjusting the Mohr failure envelope. The conclusion from this and similar cases found (Verifications #27 - 29) is that HSA is an optimization tool that is powerful enough to find any global minimum, however, the validity of the model and thus its function is left to the responsibility of the engineer. Once the tension-check was added, the results from both HSA and Path-Search more consistency.

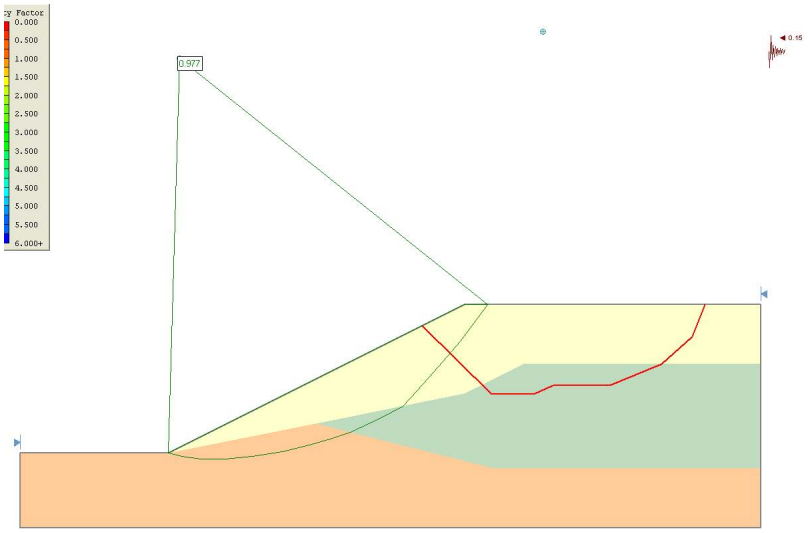


Figure 17: Hybrid Simulated Annealing for Verification #4

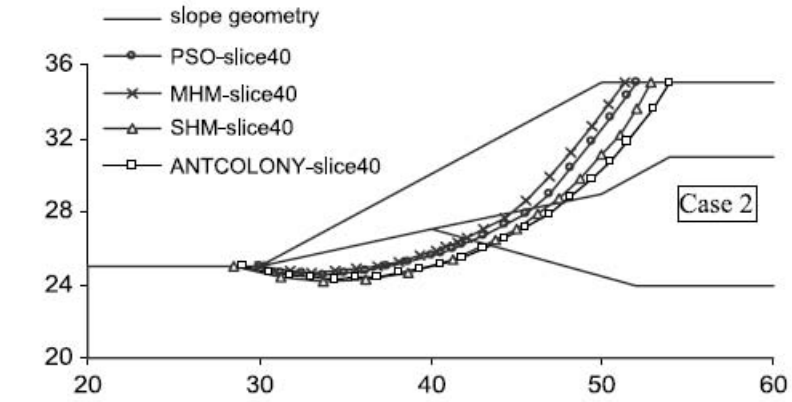
4.4 Method Comparison — HSA and Literature

4.4.1 Verification #3 & #4 - Seismic Loading Conditions

Verification #3 is a study from ACADS of a non-homogeneous slope with three layers (Rocscience, 2006). The same case was tested by Cheng et al. (2007a) as example 2 in his comparison of global optimization methods. The recommended solution from ACADS (1989) is 1.359.

Verification #4 has the same geometry and soil parameters as verification #3. The slope is subjected to a pseudo-static earthquake coefficient equal to 0.15. The lowest safety factor found by HSA was 0.977 (Figure 17) for 40 slices. The results found by Cheng et al. (2007a) from different optimization methods are shown in Figure 18. He states that failure surfaces found by SA and Particle Swarm Optimization (PSO) are virtually the same.

It can be noted that the surface found by most of the global optimization techniques (Ant-colony, SHM) reflect a circular surface, whereas the SA method actually finds a composite surface. A comparison between the results from HSA and the standard SA by Cheng et al. (2007a) shows that hybrid optimization is significantly faster when there are more slices (see Tables 10 and 11).



(failure surfaces by SA,PSO are virtually the same, failure surfaces by TABU and ANTCOLONY are virtually the same)

Figure 18: Results from Cheng et al. (2007a) for Verification #4

Table 10: Comparison of Safety Factors between HSA and SA (Cheng et al., 2007a) for Verification Case #3

Methods	Number of Slices	Minimum Safety Factor	Number of Iterations
HSA	20	1.360	55,000
HSA	30	1.358	55,000
HSA	40	1.358	55,000
SA	20	1.359	52,310
SA	30	1.357	74,120
SA	40	1.368	119,546

It can be seen that HSA achieves the same precision as SA within less than half the number of iterations from the standard SA algorithm. It is also seen that the standard SA (Cheng et al., 2007a) loses precision as the number of slices increases, and the number of iterations required doubles. The opposite effect is seen with the HSA: the more slices used the lower the safety factor found (in general - Table 11 shows an exception, attributable to the randomness of the stochastic method).

Table 11: Comparison of Safety Factors between HSA and SA (Cheng et al., 2007a) for Verification Case #4

Methods	Number of Slices	Minimum Safety Factor	Number of Iterations
HSA	20	0.979	62,500
HSA	30	0.977	60,000
HSA	40	0.993	60,000
SA	20	0.9815	52,040
SA	30	0.9769	77,366
SA	40	0.9857	108,278

Table 12: Comparison of Safety Factors between HSA and SA (Cheng et al., 2007a) for Verification Case #8

Methods	Number of Slices	Minimum Safety Factor	Number of Iterations
HSA	20	1.222	60,000
HSA	30	1.221	52,500
HSA	40	1.221	50,000
SA	20	1.2411	51,770
SA	30	1.2689	77,096
SA	40	1.3238	190,664

4.4.2 Verification #8 - ACADS (1989) Study of Thin Horizontal Layer

Verification #8 (Rocscience, 2006) corresponds to example #3 by Cheng et al. (2007). The water table is assumed to coincide with the base of the weak layer. The effect of negative pore water pressure above the water table is to be ignored (i.e. u=0 above water table). The effect of the tension crack is also to be ignored in this problem.

A comparison between the safety factors found by HSA and the standard annealing are shown in Table 12. In the current case with the thin layer, the improvement of HSA over other global optimization algorithms can be clearly appreciated. As with verification #3 and #4, it can be seen that the higher the number of slices in HSA, the more accurate

the solution becomes.

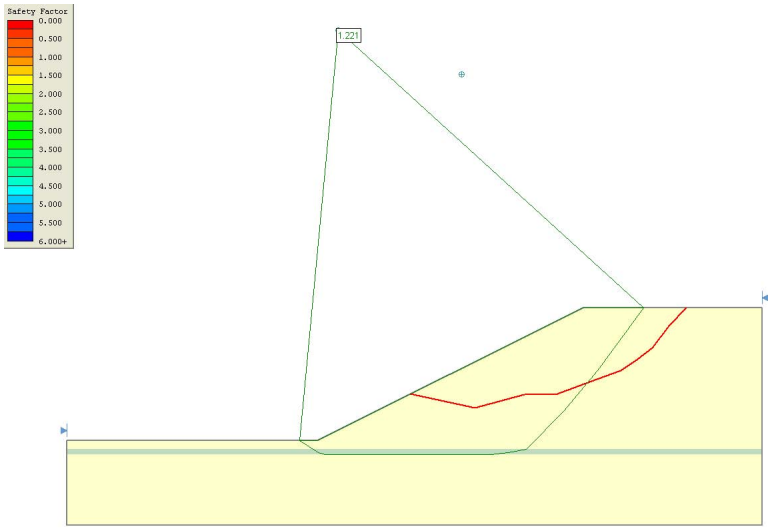


Figure 19: Hybrid Simulated Annealing for Verification #8

The main issue with the formulation by Cheng et al. (2007a) is that the division of slices is required to match the vertices. In SLIDE, the number of vertices is not dependent on the number of slices, or vice-versa. Therefore, the HSA can carry out an optimization of a surface containing 40 slices with simply ten vertices, and during the optimization process adaptively insert new vertices without affecting the number of slices (compare Figures 20 and 21). On the other hand, the standard SA proposed by Cheng et al. (2007a) requires 40 vertices to represent 40 slices. The increase in imprecision arises from the increase in problem dimensionality.

4.4.3 Verification #9 - ACADS (1989) Study of Diagonal Weak Layer

Verification Case #9 is a case of a diagonal weak layer, with loading conditions and a piezometric surface (Rocscience, 2006). The effect of a tension crack is to be ignored. The list of co-ordinates in the piezometric line can be either found in the original ACADS study or the Slide Manual. It can be seen from a comparison between Figures 21 and 22 that HSA successfully determines the safety factor accurately, whereas the the remaining global optimization methods demonstrate poor results, with as much as 0.1 difference in terms of safety factor.

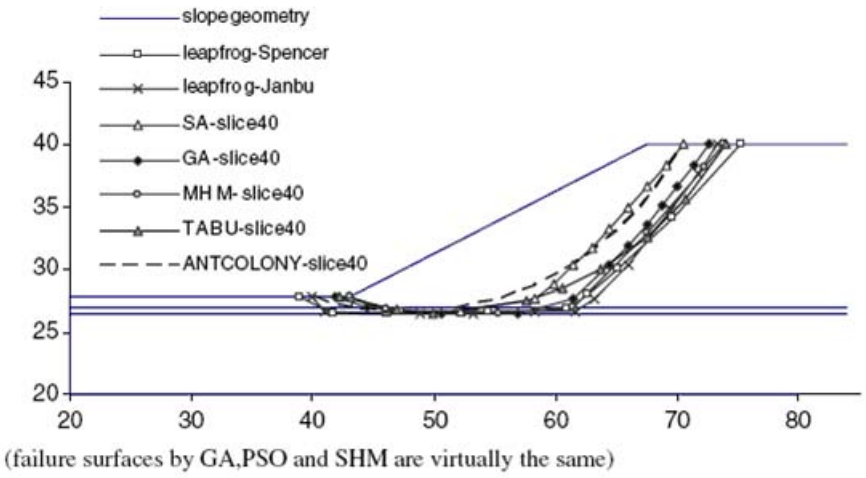


Figure 20: Results from Cheng et al. (2007a) for Verification #8

Table 13: Comparison of Safety Factors between HSA and SA (Cheng et al., 2007a) for Verification Case #9

Methods	Number of Slices	Minimum Safety Factor	Number of Iterations
HSA	20	0.709	50,000
HSA	30	0.709	72,000
HSA	40	0.708	70,000
SA	20	0.8122	183,341
SA	30	0.9369	85,937
SA	40	1.1657	119,800

4.5 Annealing — Theoretical Considerations

4.5.1 On Re-annealing and its Effectiveness

The choice of using the VFSA as the preferred global optimizer as opposed to the later created Adaptive Simulated Annealing (ASA) merits some discussion. During my work-term at Rocscience in the summer of 2008 ASA was found to be the most efficient method for the optimization of circular surfaces, due to its ability to re-scale the generation temperature at each dimension. However, when applied to non-circular search, frustration problems arise due to the high-dimensionality and ASA becomes inefficient. The details behind this will be

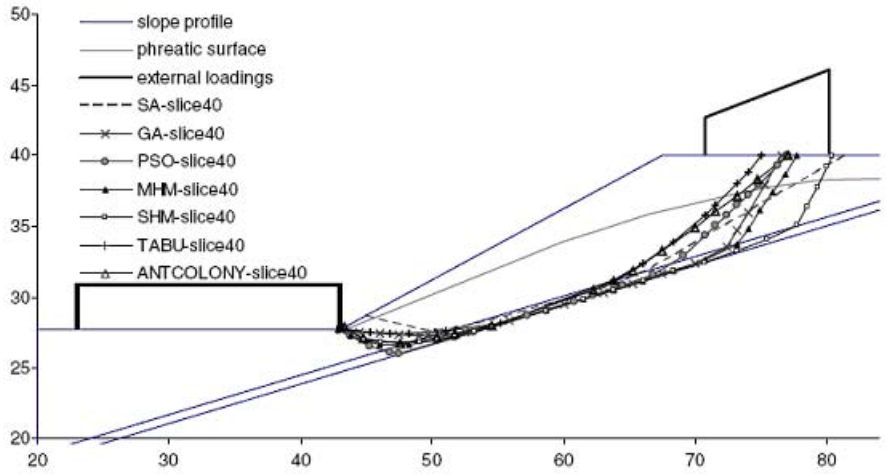


Figure 21: Results from Cheng et al. (2007a) for Verification #9

discussed therefore, first with an overview of the advantages of ASA and why they disappear when dealing with non-circular search.

ASA can be seen as a VFSA with an additional step, the temperature **re-annealing**. After every N_{accept} iterations, the generation temperatures for each dimension are re-scaled. The sensitivity s_i of each parameter is given by the partial derivative of the function with relation to the i-th dimension of the position vector:

$$s_i = \frac{\partial F}{\partial X_i} \tag{18}$$

The temperature $T_{i,gen}$ is linearly re-scaled proportional to the ratio of the maximum sensitivity s_{max} of the i-th dimensions:

$$T_{i,gen} = \frac{s_{max}}{s_i} T_{i,gen} \tag{19}$$

The optimization performed by ASA can be understood intuitively. If the search is going well, decrease the temperature rapidly and if the search is going badly, decrease the temperature slowly. For ASA, higher generation temperatures signify a larger search range and therefore faster convergence to a minimum. For a problem with few variables such as the circular search, the rescaling contributes considerably to the efficiency of the method by allowing longer jumps if the function landscape is relatively flat (insensitive). This works especially for the circular search because the variables (centre point and radius) are relatively independent of each other. Allowing one dimension, say the radius r, to increase its

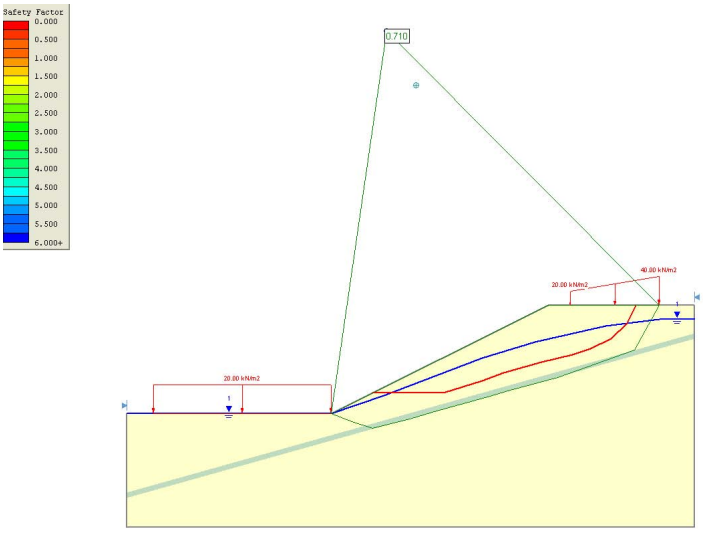


Figure 22: Hybrid Simulated Annealing for Verification #9

generation temperature does not affect the surface as much as allowing one vertex of a 20 vertex non-circular surface to walk wildly.

However, due to the high-dimensionality of the non-circular, the different sensitivities often provoke the algorithm to perform "wild walks". As there is no communication or pairing between the different parameters, one vertex will suddenly acquire a very high generation temperature, and therefore it would start random-walking.

In order to use sensitivities as feedback, a careful analysis has to be made on the nature of the safety factor function and its most common characteristics. Such a pursuit demands a fair amount of mathematical analysis and is beyond the scope of the current work. Possible approaches include assuming a single global sensitivity for the non-circular surface, which can be with respect to a circle, or pairing the sensitivities between two adjacent vertices. These suggestions will be left vague, and the reader may ponder on them at his or her leisure.

A further problem with the ASA is that it assumes the cost-function to be aligned with the co-ordinate axis of each parameter. This is especially inefficient when the major-axis of the problem is unaligned with the co-ordinate axis - as is often the case with a weak layer. A solution has been proposed by Rosen (Rotationally parameterized very fast simulated reannealing - RPVFSA, unpublished manuscript). The alternative suggests rotating the function landscape every time the algorithm performs a rescaling. However, improvements

are inconsiderable (around 10%), whereas the costs of calculating co-variance matrices in order to perform the rotation are fairly high. The generation function for VFSA is aligned along the co-ordinate axis, therefore the accuracy and speed of the method is sensitive to the local orientation and shape of the function. If a small step is taken, this effect is minimized.

4.5.2 On the Robustness of Adaptive Schedules

Despite the fact that re-annealing did not prove effective, the need for an alternative adaptive schedule is clear based on the difficulties of slope stability analysis. Faced with a multitude of geometries and properties, the function landscape may range from relatively quadratic to extremely discontinuous and multi-modal, and the optimization algorithm must have some feed-back that allows it to suit the different problems. Very fast simulated annealing, with its adaptive schedule, has been proven to be more efficient than the popular genetic algorithms (Ingber and Rosen, 1992). This may be due to the fact that as opposed to standard SA, the added re-annealing allows the search method to incorporate some measure of feed-back from the function through the calculation of sensitivities.

Simulated annealing can be considered a dynamical system whose control system is provided by the annealing schedule (Romeo and Sangiovannivincentelli, 1991). Static schedules are easier to implement, however, there is a need to fine-tune the annealing parameters to a specific problem. On the other hand, dynamic schedules are proven to increase robustness (Romeo and Sangiovannivincentelli, 1991). Adaptation as a route to better optimization was a core concept in the development of genetic algorithms (GA) (Holland, 1992). The power of GAs fundamentally relies on the robustness of their adaptive plans, and more sophisticated schemes are needed according to the complexity of the system.

The choice of using the acceptance ratio as a feedback for a dynamic schedule was initially provided by the work carried by Corana et al. (1987), and can be understood in view of the principles that simulated annealing apply in performing its optimization.

The success of a simulated annealing algorithm depends on visiting the search space extensively so that the region with the global minimum is sampled, yet retaining the ability to focus on that region towards the later stages of optimization. In standard annealing, this is the identifying feature of simulated annealing itself, and the annealing temperature is the

controlling mechanism behind it. The Metropolis criterion, whose input is the acceptance temperature, decides on number of accepted walks and rejected walks, and namely the *ratio* of accepted to rejected walks offers important information on the performance of the method. If the number of accepted states is very high, the algorithm undergoes a stage of "wild" walk, that is, it randomly searches the entire span of the domain. This is efficient at the initial stages of the algorithm. If the number of rejected states is high, the annealing is at a "down-hill" quenching stage, in which it only accepts strictly decreases in safety factor.

The optimization of annealing can be viewed as a hierarchical descent — at the start, the search is high-level, the large valleys and general landscape of the function are probed. As the annealing temperature decreases, the level of the search decreases and local, fine-grained exploration occurs. At any level, the algorithm must have both up-hill and down-hill steps so *at the current level* the algorithm can make a complete search. It was found in the current work that when the ratio between accepted and rejected points reaches equilibrium (see Figure 23), the efficiency of SA is maximized. Figure 23 displays the convergence for behavior of r as the program progresses. The same behavior of exponential decrease and periodic oscillation around 0.5 was noticed with many cases. If the adaptive schedule was not placed, this ratio would decrease to zero and the method would freeze prematurely. An equilibrium ratio of 0.5 was found by experience (a wide range of ratios was tested, ranging from 0.0 to 2.0).

Corana et al. (1987) achieve this ratio through controlling the step-size of the walk. In the current work, this is achieved by controlling the acceptance temperature as described in Section 2. This can be considered a "direct" approach, as the acceptance temperature directly influences the ratio.

4.5.3 On the Effectiveness of Hybrid Optimization

The results presented in the current work show that HSA is extremely successful in slope stability analysis. In addition to finding the global minimum, the HSA method possesses the capability of improving the surface representation by inserting vertices. The extra improvement brought from combining the LMC to the VFSA was seen to be crucial in many problems.

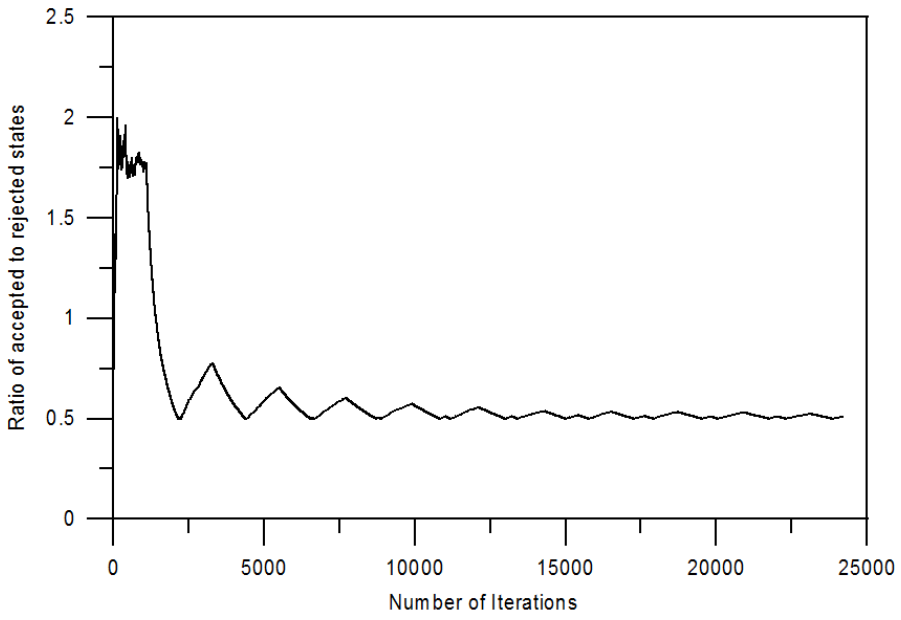


Figure 23: Ratio of Accepted and Rejected States for Verification #3

The success of the LMC can be attributed to the fact that it was specifically designed for solving slope stability problems. A primary consideration in its original development was to properly detect and explore a thin-layer (Greco, 1996). The exploration step, which shifts every vertex, allows the algorithm to perform a complete search in the neighborhood of the surface, and often allows the algorithm to tunnel to the global minimum despite the fact that the method is conceptually local. This global robustness from a local optimization method has been noted also for the simplex (Corana et al., 1987).

The LMC, due to its effectiveness, allows the VFSA method to run with more liberal settings. Lower N_{gen} and n_ϵ values can be used. The greatest contribution to the speed of the method, as discussed above, is the fact that the HSA can start the search with fewer vertices and thus allow the global optimizer to solve a lower-dimensional problem.

This advantage can be said to be specific to slope stability analysis - the underlying assumption is that the influence of all dimensions (of each vertex) is relatively small by itself, and solving a lower dimensional problem merely decreases the accuracy, and does not change the nature of the problem. For other problems this is often not possible, as every dimension inherently defines the nature of the model. The latter might be the case for molecular

simulations, in which one extra molecule changes the entire configuration of the problem, or even the circular search in slope stability, for which all three dimensions are necessary. The advantages seen in the current work closely mirror the conclusions made by Hedar and Fukushima (2002):

> The power of meta-heuristic methods comes from the fact that they are robust and can deal successfully with a wide range of problem areas. However, these methods, especially when they are applied to complex problems, suffer from the high computational cost due to their slow convergence. The main reason for this slow convergence is that these methods explore the global search space by creating random movements without using much local information about promising search direction. In contrast, local search methods have faster convergence due to their using local information to determine the most promising search direction by creating logical movements. However, local search methods can easily be entrapped in local minima.

4.5.4 No Free Lunch

Finally, we note the existence of the "No Free Lunch" theorem (Haupt and Haupt, 2004), which states that the averaged performance of all search algorithms over all problems is equal. Similarly, Ackley (Davis, 1987) stated in his review on bit vector function optimization that "There is simply no best strategy for solving all possible problems". This may be disheartening; regardless of how intelligent we build our optimization package, it is proven to have an average performance equal to a random-search.

However, the pessimism of this theorem is purely deceptive. It is not required of us to solve all functions in the physical world, and we have proven that at least for the important problem of slope stability analysis, simulated annealing seems to be a promising answer.

5 Conclusions

Based on the analysis of the global optimization methods, the following conclusions are made:

1. Very fast simulated annealing (VFSA) is a very powerful algorithm for the global optimization of slope stability problems. With the proper bounds, VFSA was successful and relatively fast for all verification and customer cases.

2. The dynamic bounds placed on the y-coordinates and the parameterization of the control variables greatly increase the speed of the method by reducing the number of invalid surfaces generated.

3. The hybrid optimization approach (HSA) was demonstrated to be a very successful to finding the lowest factor of safety for slope stability problems. The precision of the optimization was greatly improved with the use of the local Monte-Carlo search, both in terms of locally adjusting the vertices as well as inserting new vertices. This was found to be particularly useful for composite surfaces, with both circular and non-circular sections.

4. For simple geometries, the path-search when coupled with the LMC was found to be moderately successful, and twice as fast as HSA. It fails to optimize more complex slopes, even when the number of trial surfaces is increased.

5. In a comparison with classical slopes from geotechnical literature, the HSA is proven to be twice or three times as fast as regular global optimization methods, and determines critical safety factors with considerable more accuracy. It was demonstrated in the current work that HSA is more effective than the standard SA, and that the method does not lose precision as more slices are added.

6. It can be concluded that adaptive schedules are essential for guaranteeing the robustness of global optimization methods, and it was found that the re-annealing originally proposed with the VFSA is inefficient. An effective alternative is to maintain the ratio

of accepted and rejected points close to equilibrium through adjusting the acceptance temperature.

6 Recommendations

Based on the analysis of the global optimization methods, the following recommendations are made:

1. Improve interpolation between vertices by using B-Splines

   The current non-circular surface is represented through a linear interpolation between adjacent vertices. To improve the accuracy for composite surfaces, the current method inserts new vertices. This makes the optimization more difficult, as the number of dimensions is increased. It is recommended that a non-linear interpolation should be used, in special B-spline representation. B-splines are a powerful tool often used in graphics, and it allows the generation of flexible and diverse curves with fewer control variables.

2. Optimize the Optimizer

   Some of the parameters of the HSA are currently chosen based on experience. These parameters include c, n_ϵ, and N_{gen}. As opposed to the standard simulated annealing, these parameters do not need to be tuned for each slope model, but the optimal values were chosen by trials. It is recommended that an optimization method should be performed on a level higher than the HSA to tune these parameters. As the computational efforts of optimizing an optimizer are very high, a simpler method should be used to search for the optimal parameters.

3. Implementation of Parallel Processing

   Various researchers (Ingber and Rosen, 1992, Romeo and Sangiovannivincentelli, 1991) have commented on the speed and power gain of simulated annealing when implemented with parallel computing. The optimization efficiency can be increased by running parallel walks in different processors, or even generating the random numbers in parallel, and then sequentially checking all points that satisfy boundary conditions.

List of Symbols

a	coefficient for geometrical temperature decrease
$A_{XY}(T_k)$	acceptance function
ASA	Adaptive Simulated Annealing
B_i	lower domain bound for dimension i
c	control parameter for VFSA and ASA or cohesion of soil
Dx_i^j	length of horizontal displacement for the i-th vertex and k-th surface
Dy_i^j	length of vertical displacement for the i-th vertex and k-th surface
E	energy function analogous to objective function
F	objective function to be minimized
f_{opt}	optimal function value
f_{tol}	tolerance for stopping criterion
FSA	fast simulated annealing
GA	Genetic Algorithms
$G_{XY}(T_k)$	state-generation function
i	counter for dimension (VFSA and LMC)
j	counter for internal walk loop (VFSA and LMC) and stopping criterion (VFSA)
k	counter for each annealing loop
k_i	counter for each generation temperature at i-th dimension (VFSA)
LMC	Local Monte-Carlo method
L_i	characteristic length for dimension i
n	degrees of freedom
n_ϵ	iterations required for stopping criterion i
N	number of vertices for failure surface
N_{max}	maximum number of iterations for LMC
N_x	directional unit for LMC in the horizontal
N_y	directional unit for LMC in the vertical
P_{accept}	acceptance probability for new state
r	random number generated from probability distribution

R	random number for Path-Search
SA	Simulated Annealing
s_i	sensitivity of the function at the i-th dimension
s_{max}	maximum sensitivity of the function
$S(T_k)$	annealing schedule
T	annealing temperature for original SA
$T_{accept,in}$	initial acceptance temperature for VFSA
$T_{accept,k}$	acceptance temperature for VFSA
$T_{gen,in}$	initial generation temperature for VFSA
$T_{gen,k}$	generation temperature for VFSA
$VFSA$	very fast simulated annealing
V_m	vertex m for LMC representation
x_m	horizontal co-ordinate of vertex m
xd_m	the m-th division of surface
X_{opt}	optimal state
X	initial position vector
Y	Final position vector
y_i	vertical co-ordinate of vertex i
u	uniformly distributed random number
W	width of equi-division
U_i	upper domain bound for dimension i

Greek Letters

α	direction limit for Path-Search
δ	tolerance for factor of safety equation
Δ	tolerance length for step-size
ϵ	tolerance level or step reduction size for LMC
η_i	random displacement in y of the i-th vertex for LMC
θ	angle of line segment for Path-Search
ξ_i	random displacement in x of the i-th vertex for LMC
ϕ	Angle of Friction Envelope

Appendix I Simple VFSA Calculation

In the current section we will perform a simple simulated annealing, with only two internal iterations and two external annealing iterations (referring to the notation presented in section 2, $j = 0, 1, 2$ and $k = 0, 1$). The starting $j = 0$ is used to denote the initial state, no generation step occurs at $j = 0$.

Complete details for calculations can be seen in the tables presented in subsection 6. The problem to be solved is a one control variable ($i = 1$), circular slope stability case. The slope is from verification case #1. Note that the random numbers were chosen arbitrarily, and none of the annealing parameters are optimized or properly bounded. No control mechanism is placed. The annealing parameters are set as $N_{gen} = 2, c = 8, n_\epsilon = 2$. N_{gen} is the number of states generated at each annealing iteration, c is a temperature decreasing parameter and n_ϵ is the number of safety factors compared before stopping the method.

Initial Conditions

The factor of safety is calculated using Bishop's method. The aim of this presentation is to optimize the radius of a circular surface. The centre of the surface is fixed at co-ordinates (40, 40). The initial radius (which is the control variable ν_1) is chosen to be 15. The material properties as well as slope geometry are shown in Figure 24.

The initial generation temperature $T_{gen,in}$ is set to 1.0, and the initial acceptance temperature $T_{accept,in}$ is set to the initial safety factor, which is 1.292. The step-size for the search, L, is set to 5.

The Walk

Next, we perform the walk. In the first iteration $j = 1$, a uniform random number is generated ($u_1 = 0.2$). A Cauchy random number is mapped from u_1 based on the following equation:

$$r_{i,k} = \text{sgn}(u_i - 0.5) T_{gen,k} \left[\left(1 + \frac{1}{T_{gen,k}} \right)^{|2u_i - 1|} - 1 \right] \tag{20}$$

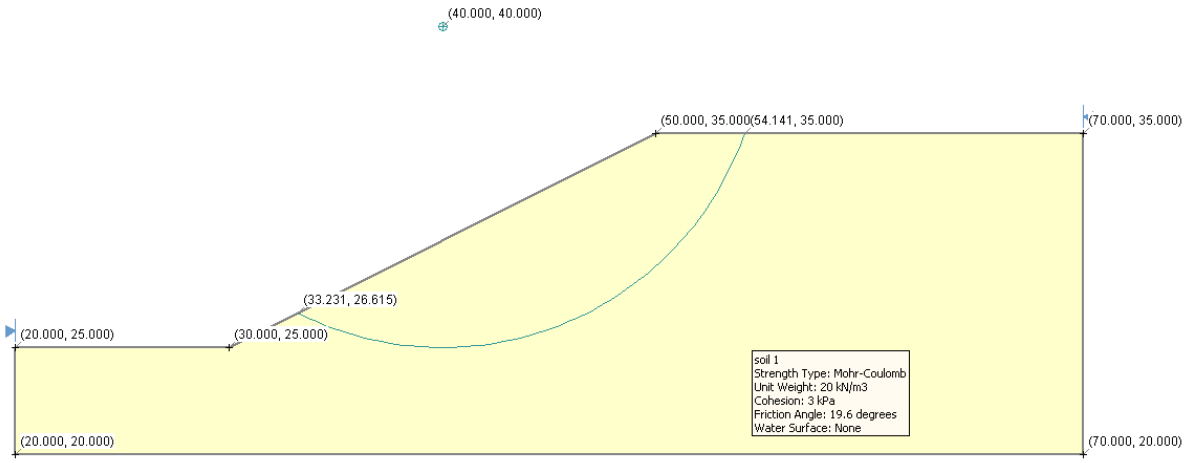


Figure 24: Slope geometry for Verification #1

Using the specific indices ($i = 1$ as the problem has one control variable, the radius. $k = 0$ as we are in the first annealing iteration):

$$r_{1,0} = \text{sgn}(u_1 - 0.5) T_{gen,1} \left[\left(1 + \frac{1}{T_{gen,1}} \right)^{|2u_1 - 1|} - 1 \right] \tag{21}$$

Substituting the values in the previous equations, we find that:

$$r_{1,0} = -0.52$$

For the walk, we use the following equation:

$$\nu_{i,j+1} = \nu_{i,j} + r_{i,k} L_i \tag{22}$$

Substituting the indices into the previous equation:

$$\nu_{1,1} = \nu_{1,0} + r_{1,1} L_1 \tag{23}$$

Substituting the values into the previous equation we find that:

$$\nu_{1,1} = 12.4$$

All other states are generated accordingly.

To Accept or Not to Accept

To accept or not to accept the new state, the factor of safety is calculated. If the walk is downhill ($F(v_{j+1}) < F(v_j)$), the state is updated and $v_{j+1} = v_j$. If the walk is up-hill, the Metropolis criterion is applied. In $j = 2$, at $k = 0$, the move was uphill ($dE = 0.036 > 0$). In this case, the modified Metropolis criterion (equation 9) is used.

As $T_{accept,in} = 1.292$, substituting all values into the previous equation we obtain a P_{accept} equal to 0.493. Therefore, a uniform random number u_{accept} was generated (0.73) and compared with P_{accept}. As $u_{accept} > P_{accept}$, the move was rejected. The previous radius was reset.

Temperature annealing

After $j = 3$, the temperature annealing must be carried out. The acceptance temperature and the generation temperature are decreased according to equations 26 and 27, respectively.

$$T_{accept,k} = T_{accept,in} \exp\left(-ck^{1/n}\right) \tag{24}$$

$$T_{gen,k} = T_{gen,in} \exp\left(-ck^{1/n}\right) \tag{25}$$

As mentioned in the introduction, $c = 8$, $k = 1$ and $n = 1$.

$$T_{accept,1} = 1.292 \times \exp\left(-8 \times 1\right) \tag{26}$$

$$T_{gen,1} = 1.0 \times \exp\left(-8 \times 1\right) \tag{27}$$

Stopping Criterion

A simple stopping criterion for the annealing algorithm is to compare the best safety factors found at n_ϵ consecutive *annealing* iterations (k-iterations). As $n_\epsilon = 2$, we compare the previous 2 annealing iterations and see whether they are within a tolerance level f_{tol}. Assuming

$f_{tol} = 0.1$, we compare between 1.144 (f_{min} at $k = 0$) and 1.13 (f_{min} at $k = 1$).

$$|f_{opt}(1) - f_{opt}(0)| < f_{tol}$$

$$|1.13 - 1.144| < 0.1$$

Therefore the method is stopped.

Solution

The global minimum f_{opt} found is 1.13, for a surface with radius=11.395 and center at coordinates (40,40). The failure surface is illustrated in Figure 25.

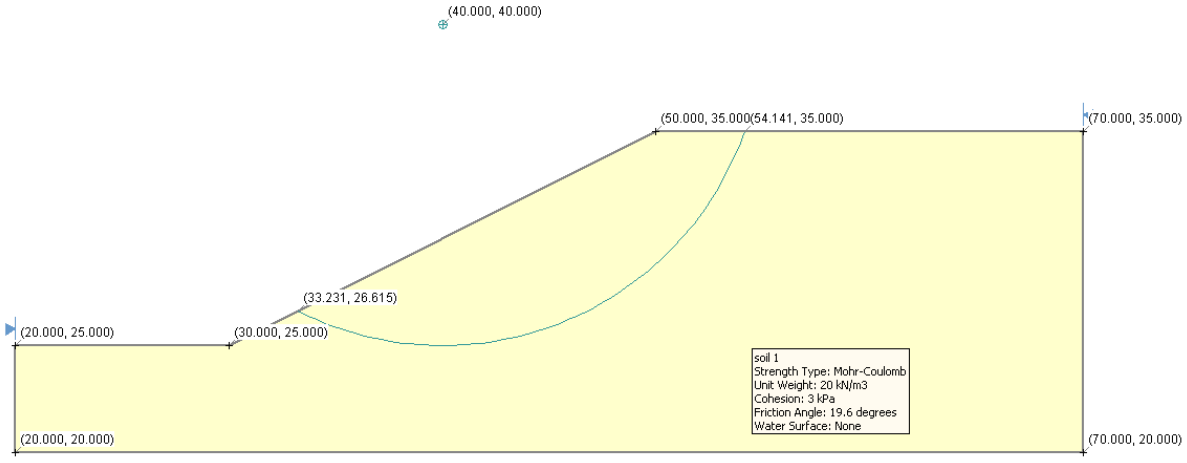


Figure 25: Solution for simple slope problem

Tables of Search Parameters

Table 14: VFSA at annealing iteration $k = 0$

	Initial$j = 0$	$j = 1$	$j = 2$
u_1	-	0.2	0.6
$r_{1,1}$	-	-0.52	0.148
Radius	15	12.4	13.14
FOS	1.292	1.144	1.180
$dE = FOS(j) - FOS(j-1)$	-	-0.148	0.036
P_{accept}	-	-	0.493
u_{accept}	-	-	0.73
Accepted?	-	YES	NO
$T_{gen,1}$	1.0	1.0	1.0
$T_{accept,1}$	1.292	1.292	1.292

Table 15: VFSA at annealing iteration $k = 1$

	Initial$j = 0$	$j = 1$	$j = 2$
u_1	-	0.1	0.36
$r_{1,1}$	-	-0.201	-2.81558×10^{-3}
Radius	12.4	11.395	11.38
FOS	1.144	1.13	1.131
$dE = FOS(j) - FOS(j-1)$	-	-0.014	0.001
P_{accept}	-	-	0.0905
u_{accept}	-	-	-.7
Accepted?	-	YES	NO
$T_{gen,1}$	3.354×10^{-4}	3.354×10^{-4}	3.354×10^{-4}
$T_{accept,1}$	4.334×10^{-4}	4.334×10^{-4}	4.334×10^{-4}

References

Abramson, L. W., Lee, T. S., Sharma, S., and Boyce, G. M. (2002). *Slope Stability and Stabilization Methods*. John Wiley & Sons, New York.

Boutrup, E., Lovell, C. W., and Siegel, R. (1979). Stabl 2 - a computer program for general slope stability analysis.

Chen, S. and Luk, B. L. (1999). Adaptive simulated annealing for optimization in signal processing applications. *Signal Processing*, 79(1):117–128.

Cheng, Y. M. (2007). Global optimization analysis of slope stability by simulated annealing with dynamic bounds and dirac function. *Engineering Optimization*, 39(1):17–32.

Cheng, Y. M., Lansivaara, T., and Wei, W. B. (2007a). Two-dimensional slope stability analysis by limit equilibrium and strength reduction methods. *Computers and Geotechnics*, 34(3):137–150.

Cheng, Y. M., Li, L., and Chi, S. C. (2007b). Performance studies on six heuristic global optimization methods in the location of critical slip surface. *Computers and Geotechnics*, 34(6):462–484.

Corana, A., Marchesi, M., Martini, C., and Ridella, S. (1987). Minimizing multimodal functions of continuous-variables with the simulated annealing algorithm. *Acm Transactions on Mathematical Software*, 13(3):262–280.

Davis, L., editor (1987). *Genetic Algorithms and Simulated Annealing*. Research Notes in Artificial Intelligence. Pitman Publishing, London, Great Britain.

Duncan, J. and Wright, S. (2005). *Soil Strength and Slope Stability*. John Wiley & Sons, Inc., New Jersey.

Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Ieee Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741.

Greco, V. R. (1996). Efficient monte carlo technique for locating critical slip surface. *Journal of Geotechnical Engineering-Asce*, 122(7):517–525.

Haupt, R. L. and Haupt, S. E. (2004). *Practical Genetic Algorithms*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2nd edition.

Hedar, A. R. and Fukushima, M. (2002). Hybrid simulated annealing and direct search method for nonlinear unconstrained global optimization. *Optimization Methods & Software*, 17(5):891–912.

Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems*. The MIT Press, Cambridge, Massachusetts.

Ingber, L. (1989). Very fast simulated re-annealing. *Mathematical and Computer Modelling*, 12(8):967–973.

Ingber, L. (1993). Simulated annealing - practice versus theory. *Mathematical and Computer Modelling*, 18(11):29–57.

Ingber, L. and Rosen, B. (1992). Genetic algorithms and very fast simulated reannealing - a comparison. *Mathematical and Computer Modelling*, 16(11):87–100.

Kirkpatrick, S. (1984). Optimization by simulated annealing — quantitative studies. *Journal of Statistical Physics*, 34(5-6):975–986.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

Kvasnicka, V. and Pospichal, J. (1997). A hybrid of simplex method and simulated annealing. *Chemometrics and Intelligent Laboratory Systems*, 39(2):161–173.

Malkawi, A. I. H., Hassan, W. F., and Sarma, S. K. (2001). Global search method for locating general slip surface using monte carlo techniques. *Journal of Geotechnical and Geoenvironmental Engineering*, 127(8):688–698.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092.

Moyano, G. E., Wesendrup, R., Sohnel, T., and Schwerdtfeger, P. (2002). Properties of small- to medium-sized mercury clusters from a combined ab initio, density-functional, and simulated-annealing study. *Physical Review Letters*, 89(10):4.

Murata, T., Ishibuchi, H., and Tanaka, H. (1996). *Genetic algorithms for flowshop scheduling problems*. Pergamon-Elsevier Science Ltd.

Nelder, J. A. and Mead, R. (1965). A simplex-method for function minimization. *Computer Journal*, 7(4):308–313.

Polyak, B. T. (1987). *Introduction to Optimization*. Translation Series in Mathematics and Engineering. Optimization Software, Inc., New York.

Romeo, F. and Sangiovannivincentelli, A. (1991). A theoretical framework for simulated annealing. *Algorithmica*, 6(3):302–345.

Song, S. L., Kong, L., Gan, Y., and Su, R. J. (2008). Hybrid particle swarm cooperative optimization algorithm and its application to mbc in alumina production. *Progress in Natural Science*, 18(11):1423–1428.

Szu, H. and Hartley, R. (1987). Fast simulated annealing. *Physics Letters A*, 122(3-4):157–162.

Van Laarhoven, P. J. M. and Aarts, E. H. L. (1987). *Simulated Annealing: Theory and Applications*. Mathematics and Its Applications. D. Reidel Publishing Company, Dordretch, Holland.

Velis, D. R. and Ulrych, T. J. (1996). Simulated annealing wavelet estimation via fourth-order cumulant matching. *Geophysics*, 61(6):1939–1948.

Yao, X. (1995). A new simulated annealing algorithm. *International Journal of Computer Mathematics*, 56(3-4):161–168.