Kochi Chapter

Indian Geotechnical Conference
IGC 2022
15th – 17th December, 2022, Kochi

A novel swarm-intelligence optimization: Learning sparrow search algorithm applied to slope-stability in tracing the critical slip-surface

Abhijit Saha[1]

[1] Consulting Engineer, Surface Water Hydrology & Geotechnical Specialist, Kolkata

Abstract. The popularity of nature-inspired meta-heuristics in solving complex optimization problems is on a steady rise within a rapidly evolving world. Swarm intelligence (SI) optimization inspired by the behavior of social organisms in flocks of bird, schools of fish, colonies of ants and bees perform the search through agents whose trajectories are primarily adjusted stochastically and sporadically deterministically, in accordance with golden rules drawn from Mother Nature. Each entity within the swarm is influenced by its own 'best' and group 'best' position, while moving randomly to converge to optimal through competition and cooperation. The sparrow search algorithm (SSA), developed by Xuea and Shen (2020) and slightly improvised (LSSA) by Ouyang, Zhu and Wangis (2021) is a very recent SI approach which adopts the sparrow producer–scrounger model metaphorically for designing optimum searching strategies, inspired by the group wisdom, foraging and anti-predation behavior of sparrows. LSSA is experimented on some hard benchmark test functions to test its effectiveness and thereafter, applied in searching the critical failure surface in slope-stability problem. The objective function is the factor of safety against failure. The stability analysis is performed integrating the present tool with Bishop's simplified method (1955). Results show LSSA is a strong contender to methods like genetic-algorithm, simulated-annealing, big-bang big-crunch and artificial bee colony algorithms. The study illustrates the flexibility, efficiency and robustness of the methodology in function optimization.

Keywords: nature inspired meta-heuristics; swarm optimization; learning sparrow search algorithm; slope-stability; critical surface.

1. Introduction

Nature is a principal source of inspiration in devising optimization models for solving high degree of complex problems. Swarm intelligence within nature is defined as "*any attempt to de-sign algorithms or distributed problem-solving devices inspired by the collective behaviour of social insect colonies and other animal societies*" (Bonabeau et al, 1999). It is a paradigm that considers collective intelligence as a behaviour that emerges through the interaction and cooperation of large numbers of homogeneous lesser intelligent agents (like fish, birds, ants, bees etc.) in the environment. "*Two fundamental concepts, self-organization and division of labour, are necessary and sufficient properties to obtain swarm intelligent behaviour such as distributed problem-*

solving systems that self-organize and adapt to the given environment in recent years" (Karaboga, 2005). The information is typically stored in the participating homogeneous agents, or is stored or communicated in the environment through the use of pheromones in ants, dancing in bees, proximity in fish and birds and immune system in cells. Like evolutionary computation, swarm intelligence 'algorithms' are adaptive strategies that are typically ap-plied to search and optimization domains. They are characterized by decentralized control, self-organization and adaptation.

In the present study, LSSA algorithm is initially tested on benchmark test functions like the Rosenbrock function (1960), Rastrigin function (1974), Ackley function(1987), and then applied to single objective optimisation of a 3D real-world problem- *finding the critical failure surface in slope-stability problem*.

2. Nature Inspired Optimization: Thrust of Artificial Intelligence

Artificial intelligence (AI) is the study and design of 'intelligent agents'- a system that perceives its environment and takes action accordingly to maximize its chances of success. To solve the most difficult problems, AI has developed a large set of tools, the main amongst these being the search algorithms and optimization methods encompassing evolutionary computation. Simple exhaustive searches are rarely sufficient for most real-world problems of varied dimensions, as the number of points in the search spacerapidly grows to astronomical numbers. The result is a search that is too slow or virtu-ally incomplete. The solution is to use "heuristics" or "rules of thumb" that eliminate exploration of futile spaces or redundant points and supplies the program with the "best guess" to the optimal global solution track. An optimization method that employs multiple candidate solutions (a population or swarm) and in-effect are evolutionary; typically require some kind of selection or survival of the fittest rule for combining and generating new candidate solutions. It begins with a random population of organ-isms (the 'wild' guesses) within the limits of the search domain of each problem dimensions, and then allow them to recombine and mutate in accordance with a mixture of stochastic (random) and deterministic strategies for exploration (wide search) and exploitation (neighbourhood search) respectively, with the sole aim of the fittest to survive in each generation (refining the guesses). In the process, the stochastic element is generally higher than the deterministic element in a pure effort to significantly maintain diversity in the swarm as they converge to the global optima. In AI, an evolutionary algorithm (EA) is a generic population-based metaheuristic optimization algorithm that is inspired by the mechanics of biological evolution: re-production, mutation, recombination, and selection. Candidate solutions play the role of individuals in a population, and the fitness function determines the environment within which the solutions "live". EAs often perform well with approximating solutions to all problems, as they prima-facie do not make any assumption about the underlying fitness landscape. This generality has led to successes in fields of genetics, physics, chemistry, biology, engineering, art, eco-nomics, marketing, operations research, robotics, social sciences and politics.

Nature inspired algorithms covers the wealth of modern metaheuristic algorithms, developed with an aim to carry out global search, typical examples are Genetic Algorithms (GA) (Holland, 1975; Goldberg, 1989), inspired by Darwin's natural selection and survival of the fittest rule and Particle Swarm Optimisation (PSO) (Kennedy &

Eberhart, 1995), inspired by movements of flocks of birds and shoals of fish. Ant colony optimization (ACO) (Dorigo et al, 1991; Colomi et al, 1991) takes inspiration from the foraging behavior of ants. Another metaheuristic algorithm is Simulated Annealing (SA) and its variants (Kirkpatrick et al 1982, 1983) that mimics the slow cooling of molten metals to achieve a crystalline absolute minimum energy state. Recently, two new metaheuristic algorithms, called Cuckoo Search (CS)-(Yang & Deb 2009) based on the obligate brood parasitic behaviour of some cuckoo species in combination with the Lévy flight behaviour of some birds and Firefly Algorithm (FA) (Yang, 2009, 2010) -inspired by the flashing pattern of tropical fireflies have been used successfully to solve hard multimodal problems. Sparrow Search Algorithm (SSA) (Xue J. & Shen B.2020) is the latest add-in to this wealth of knowledge inspired by the foraging behav- ioral strategies of house sparrows.

3. Sparrow -Their Biological Characteristics

The sparrow, a sexually dimorphic, highly gregarious passerine bird displays a natural instinct of curiosity about anything and everything around and are always vigilant of the surrounding. A typical group constitutes three member types, the producers, scroungers and rangers (or scouts). In contrast with other small bird species, sparrows are exceptionally intelligent with strong memory. Sparrows are opportunistic, intelligent feeders and often use a variety of feeding techniques, adapting their methods to best suit the current conditions of their habitat and prey. The producers actively search for the food sources, while scroungers obtain food by the producers. Evidence has shown that sparrows usually use behavioural strategies flexibly switching between producing and scrounging. In other words, the sparrows usually use the strategy of both the producer and scrounger to find their food (Barnard & Sibly, 1981). Spar-rows foraging in groups can obtain food by searching or by social interaction with other group members. This is the producer-scrounger (PS) model, where producers actively search for food and scroungers (sparrows of lower dominance rank) profit from producers' efforts through joining or stealing. The discoverer (producer) searches for food, and provides direction to other individuals in the population. Studies have shown that the individuals monitor the behaviour of others in the group. Meanwhile, the attackers in the bird flock, which want to increase their own predation rate, competes with food resources of the companions with high intakes. The energy reserves of the individuals play an important role and the sparrows with low energy re-serves scrounge more. The birds located on the periphery of the population, are more likely to be attacked by predators and constantly try to get a better position. The birds which are located near the center, may move closer to their neighbours in order to minimize their domain of danger. When a bird detects a predator, one or more individuals give a chirp and the entire group flies away.

4. Sparrow Search Algorithm-A Novel Optimizing Tool

The sparrow search algorithm (SSA) is an effective swarm intelligence optimization technique, which simulates the group wisdom foraging and anti-predation behaviors of sparrows. In accordance to the description of sparrows narrated above, a mathematical

$$x = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & \cdots & x_{2,d} \\ \vdots & \vdots & & \vdots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & \cdots & x_{n,d} \end{bmatrix} \text{-(1)} \qquad F_x = \begin{bmatrix} f([x_{1,1} & x_{1,2} & \cdots & \cdots & x_{1,d}]) \\ f([x_{2,1} & x_{2,2} & \cdots & \cdots & x_{2,d}]) \\ \vdots & \vdots & & \vdots & \vdots \\ f([x_{n,1} & x_{n,2} & \cdots & \cdots & x_{n,d}]) \end{bmatrix} \text{-(2)}$$

model is constructed for the SSA with simplified idealized behavioural pattern of sparrows with the consistent rules (A to F) as mentioned hereunder that has been put forward by Xue & Shen (2020). The SSA is divided into three phases: discoverer, follower, and investigator (scout or ranger).

A. The producers typically have high energy levels and provide foraging areas or directions (guides) for scroungers. They have wide search range in identifying areas of rich food sources. The level of energy reserves depends on the assessment of the fitness values of the individuals.
B. As the sparrow detects a predator, the individuals begin to chirp alarming signals. When the alarm value is greater than the safety threshold, the producers lead all scroungers to a safe area.
C. Each sparrow can become a producer as long as it searches for better food sources, but the proportion of producers and scroungers is unchanged in the population.
D. The sparrows with higher energy level act as producers. Starving scroungers are more likely to fly to other places for food in order to gain more energy.
E. The scroungers always follow the producers who can provide the best food source location. Meanwhile, some scroungers may constantly monitor the producers and compete for food in order to increase their own predation rate.
F. The sparrows at the edge of the group quickly move toward the safe area to get a better position when aware of danger, while the sparrows in the middle of the group randomly walk in order to be close to others.

In the simulation experiment, virtual sparrows are randomly generated in the search space to find potential food sources. The position of sparrows is represented by the matrix shown in Eqn.-1, where *n* is the number of sparrows and *d* is the dimension of variables to be optimized. Upon random generation, the fitness value of individual sparrows is expressed by the vector F_x given in Eqn.-2. The value of each row in F_x represents the fitness value of individual species of each dimension. In this context, the location of a potential food source is used synonymously and interchangeably with the sparrow position and so is its fitness. Producers with better fitness naturally have priority to obtain food in the search process. Further, the producers with the highest energy level being responsible for searching food and guiding the movement of the entire population, has the ability to search food within the entire search domain effectively than the scroungers. According to rules (A) and (B), during each iteration, the location of the producer is updated as per Eqn.-3, where *t* indicates the current iteration, *j* the problem dimension = 1, 2, . . . , *d* and $x^t_{i,j}$ represents the value of the j^{th} dimension of the i^{th} sparrow at iteration *t*. The current number of iterations is represented by *i* and $iter_{max}$ is the maximum number of iterations (which needs some trials initially to find atwhat value this should be fixed such that the objective function converges to global optimum, since higher the $iter_{max}$ value, higher is the algorithm run-time). $\alpha \in [0, 1]$ is a random number. $R_2 \in [0, 1]$ and $ST \in [0.5, 1.0]$ represent the alarm (or alert) value and the safety threshold respectively. R_2, ST are both random numbers. Q is a randomnumber which obeys normal distribution. L shows a matrix of $1 \times d$ for which each - 4element inside is 1. At $R_2 < ST$, implies there are no predators around and the producer(discoverer) enters a wide search space exploration mode. At $R_2 \geq ST$, some sparrows have discovered the predator and issued an alert, and all sparrows need to quickly fly

$$x = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & \cdots & x_{2,d} \\ \vdots & \vdots & & \vdots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & \cdots & x_{n,d} \end{bmatrix} \text{-(1)} \qquad F_x = \begin{bmatrix} f([x_{1,1} & x_{1,2} & \cdots & \cdots & x_{1,d}]) \\ f([x_{2,1} & x_{2,2} & \cdots & \cdots & x_{2,d}]) \\ \vdots & \vdots & & \vdots & \vdots \\ f([x_{n,1} & x_{n,2} & \cdots & \cdots & x_{n,d}]) \end{bmatrix} \text{-(2)}$$

to other safe areas and that the discoverer will lead the follower to a safe location.

Followers perform food searches after the discoverer and does neighborhood searches around the discoverer's location. Scrounger's (follower) position update is as per Eqn.-4, where x_p is the optimal position currently occupied by the producer (discoverer). x_{worst} represents the worst location currently. 'A' is a matrix of 1 × d where the value of each element is randomly assigned +1 or -1, and $A^+ = A^T (AA^T)^{-1}$. When i > n/2, it indicates that the sparrow population is aware of danger, at which point they make antipredation behavior. As for the scroungers, they need to enforce the rules (D) and (E). As some scroungers monitor the producers more frequently, as soon as they find that a producer has found good food, they immediately leave their current position to compete for that food. If they win, they get the food of the producer immediately (swapping its role as producer, while the producer which loses in the tussle becomes a scrounger), otherwise they continue to execute rule (E). In the simulation experiment, it is assumed that sparrows which are aware of danger, account for 10% to 20% of the total population. According to rule (F), the mathematical model can be expressed as Eqn.-5, where x_{best} is the current global optimal location. β, the step-size control parameter is a random number obeying normal distribution with mean 0 and variance 1. K ∈ [−1, 1] is another random number and represents the direction in which the sparrow moves while also controlling the moving step-size. Here f_i is the fitness value of the current sparrow. f_g and f_w are the optimal and worst fitness values within the current search scope, respectively. ε is a minutest real number so as to avoid zero-division-error. At $f_i \neq f_g$, the current sparrow is at the boundary of the population and vulnerable to predator attack and the location needs to be adjusted. And $f_i = f_g$ indicates

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t \cdot \exp\left(\dfrac{-i}{\alpha \cdot iter_{max}}\right), & if\ R_2 < ST \\ x_{i,j}^t + Q \cdot L, & if\ R_2 \geq ST \end{cases} \text{-(3)}$$

$$x_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\dfrac{x_{worst}^t - x_{i,j}^t}{i^2}\right), & if\ i > \dfrac{n}{2} \\ x_p^{t+1} + \left|x_{i,j}^t - x_p^{t+1}\right| \cdot A^+ \cdot L, & otherwise \end{cases} \text{-(4)}$$

$$x_{i,j}^{t+1} = \begin{cases} x_{best}^t + \beta \cdot \left|x_{i,j}^t - x_{best}^t\right|, & if\ f_i \neq f_g \\ x_{i,j}^t + K \cdot \left(\dfrac{\left|x_{i,j}^t - x_{worst}^t\right|}{(f_i - f_w) + \varepsilon}\right), & if\ f_i = f_g \end{cases} \text{-(5)}$$

that individual sparrow in the interior of the population is aware of the danger and need to be close to other sparrows to avoid the danger (search space exploitation or local search). Based on the idealization and feasibility of the above model, the basic steps of the SSA model can be summarized as the pseudo-code shown in Algorithm-1.

The idea of adaptive weight is brought in Eqn.-3, but this adaptive weight even has defects in the face of high-dimensional complex functions and may not open up a global vision. "*Therefore, it is necessary to make use of lens reverse learning and random reverse learning to dig out more hidden positions, and to increase the diversity of the population"* and supplement the optimization runs in the later stage. Eqn.-4 has the defect of near-zero points; hence, *"nonlinear sine-cosine guidance is used to balance the local and global search. From the overall formula, the update distance between the front and back position of SSA is far, so the blind area between them becomes more. The local search based on the difference can improve the search precision and reduce the scope of the blind areas"* (Ouyang C. et al, 2021).

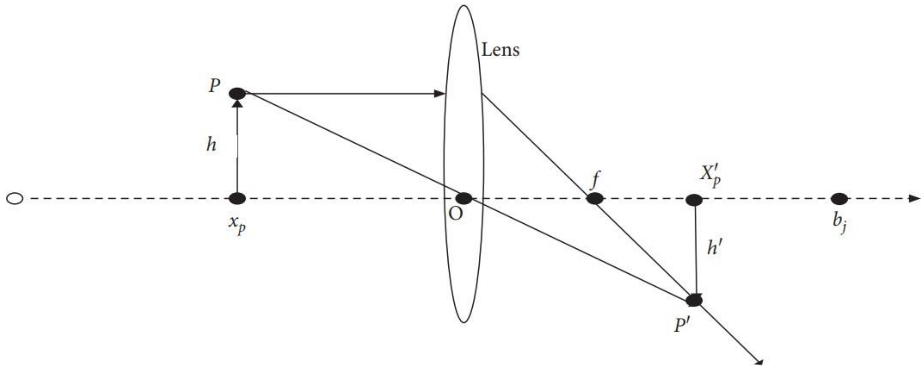


Figure 1. Lens schematic diagram (Ouyang C. et al, 2021).

4.1. Learning Sparrow Search Algorithm

Opposition-Based Learning Strategy Based on Convex Lens Principle. For mining new solutions in unknown areas and increase the population diversity and to avoid monotonicity and risk of local optimization, the lens learning principle is mimicked. In a certain search space, suppose an individual *P* of height *h*, x_p is the projection of P onto the x-axis. A lens of focal length *f* is placed on the base point position O, O being the midpoint of $[a_j, b_j]$, where a_j and b_j represent the lower and upper bounds of the j^{th} dimension of the current solution. An image *P′* of height *h′* is obtained by the lens imaging, and its projection is x_p' (reverse point). The schematic diagram is shown in Figure 1. The reverse point x_p' of individual x_p is obtained by taking O as the base point, by the lens imaging principle: $\frac{a+b/2-x_p}{x'_p-a+b/2}=\frac{h}{h'}=k$ (scaling factor). After transformation, the reverse point for the j^{th} dimension can be obtained as $x'^j_p=\frac{a_j+b_j}{2}+\frac{a_j+b_j}{2k}-\frac{x^j_p}{k}$ - (6). Different values of scaling factor, *k* from 0.7 to 1.3 has been used in the present study simulation runs.

Opposition-Based Learning of the Worst Position. After the producer (discoverer) has searched, the worst position they get may not be necessarily reliable. From Eqn.-3 and Eqn.-4, it is known that the worst solution will affect the later stage of optimization, and the minimum value will give scroungers (followers) a better search range. This implies updating at the worst location is also necessarily important, which is generally ignored, only tracking the optimal location and ignoring the overall integrity of the algorithm. A random opposition-based mechanism to update the worst position is used in accordance with: $x_{worst}'(t) = a_j + rand * b_j - x_{worst}$ –(7).

<u>ALGORITHM-1:</u> <u>SSA Framework</u>

<u>Input:</u>
G: the max. generations or iterations
PD: the number of producers (discoverers)
SD: the number of scroungers who perceive danger (investigators/scouts)
R_2: the alarm or alert value ∈ [0, 1]
ST: Safety Threshold ∈ [0.5, 1.0]
$\alpha \in [0, 1]$, $\beta = N(\mu=0, \sigma^2=1)$,
$K \in [-1, 1]$
(All Random Numbers)
$\varepsilon \approx 0$ (= 0.0001, say)
n: number of sparrows in the swarm

<u>Output:</u> x_{best}, f_g.
Initialize a population of n sparrows randomly within the feasible search domain (*In this study for doing so, the RAND() function in MS-Excel is used that returns an evenly distributed random real number ≥0 & <1, that uses the Mersenne Twister algorithm to generate random numbers*). $x^1_{ij}=a_i+RAND()*(b_i-a_i)$ [a_i, b_i are the lower & upper search bounds of each decision variable x_i of dimension *j*.

1: t (iteration number) = 1;
2: **While** (t < G)
3 : Rank the fitness values and find the current best & current worst individual.
4 : R_2 = rand(1)
5 : **For** i = 1 : PD
6 : Use Eqn.-3 & Eqn.-6 to update sparrow's location;
7 : **End for**
8 : Update the worst location found by the discoverer according to Eqn.-7;
9 : **For** i = (PD + 1) : n
10 : Use Eqn.-4 & Eqn.-8 to update sparrow's location;
11 : **End for**
12 : **For** l = 1 : SD
13 : Use Eqn.-5 to update the individual position of a sparrow that is aware of danger;
14 : **End for**
15 : Get the location of new optimal sparrow;
16 : If new location is better than previous, update it, else retain existing (greedy selection / elitist strategy);
17 : t = t + 1
18 : **End while**
19 : **Return** x_{best}, f_g.

Guidance Strategy Based on Improved Sine-Cosine Algorithm. In the scrounger (follower) location update there are a few dynamic parameters, so it is easy to limit the search range of sparrow population and blindness, which limits the searchability of the algorithm. To improvise and expand the search scope, dynamic update of follower sparrow's individual position by the sine-cosine characteristics is used:

$$x_i^{t+1} = \begin{cases} x_i^t + r_1 \cdot sin(r_2) \cdot \left| r_3 \cdot x_p^t - x_i^t \right|, & r_4 \le 0.5, \\ x_i^t + r_1 \cdot \cos(r_2) \cdot \left| r_3 \cdot x_p^t - x_i^t \right|, & r_4 > 0.5, \end{cases} \quad \text{-(8), where } r_1^j = a_j - i \cdot \left(\frac{a_j}{iter_{max.}} \right) \text{-(9).}$$

Here, r_1^j is a parameter, determined by the number of iterations, and it is the key to determine the individual search range. As the number of iteration increases, r_1^j gradually shrinks, and the sparrow search range tends smaller. a_j is the lower search bound of the j^{th} dimensional component of the concerned decision variable x_i. r_2 is a random number in the range [0, 2π], which determines the individual movement distance; r_3 and r_4 are random numbers in [0, 2] and [0, 1], respectively. As per Ouyang C. et al (2021), r_1 uses linear decline to balance the search scope, there are chances to get trapped into local optimal when optimizing high-dimensional complex functions. A nonlinear decline of r_1 to balance local and global search was suggested by them: $r_1 = c + \frac{b}{\exp\left\{4\times\left(t / iter_{\max}\right)^4\right\}+1}$ -(10), where, b is fixed at 0.1 and c is a regulating factor suggested as $c = 0.9$ and they finally summarizes that "*the introduction of improved sine-cosine guidance strategy reduces the blindness of sparrow searches, accelerates information exchange between individuals in the population and those in the best and worst positions, and makes followers more purposeful in their searches…The nonlinear decreasing parameters make the search more detailed and improve the convergence accuracy of the algorithm*". In the present study, r_1 as per Eqn.-9 has been used.

In the investigation reported herein, a modest population size of 50 (=n) sparrows has been assumed in the simulation runs, out of which 30% (PD=15) are assumed as producers (discoverers), 60% (S=30) as scroungers (followers) and 10% (SD=5) are assumed as scroungers perceiving danger (rangers, investigators or scouts).

The program is written in Visual Basic for Applications at back-end with MS-Excel at front-end. Customised *functions* of the hard Benchmark Test Functions, Bishop's FoS, LSSA functions (Eqn.-3 to Eqn.-9) and optimization subroutine of LSSA algorithm framework is written in VBA Code Editor modules. The optimization subroutine is called from Developer Tab of MS-Excel through macros and LSSA is run repeatedly, and all input and output data (variables) of each generation (cycle or iteration) are stored in multiple cells of a worksheet of any specific generation (G_i) and in consecutive worksheets in chronological order (G_1,G_2,G_3….,G_{max}) for multiple iterations. Finally, the results are plotted graphically for a lucid understanding and critical comparison.

5. Slope-Stability Problem Definition & the Search Procedure

By the advent of computers, the use of optimization techniques in locating the critical slip surface has been a major topic for the researchers. Non-traditional optimization algorithms which are population based and stochastic in nature that discretizes the search space has proved to be efficient in locating the global optima. A problem cited by Spencer (1967) is chosen for analysis. The problem parameters, soil-data and search boundaries are spelt in Table-1 & depicted in Figure-1.

In the search process, the three decision variables are the abscissa (CX) and ordinate (CY) of the circle center and the depth factor (N_d) of the circular failure arc. Based on a few trials, the *feasible* bounds of the decision or design variables, has been identified as: -0.10B ≤ CX ≤B, 1.05H ≤ CY ≤ 3x1.05H, 0.80 ≤ N_d ≤ 1.25. The FoS (F), the objective function to be optimized is a function of slip circle center and the radius of failure arc (R), where, R=*f*{CX, CY, N_dH}. F is related to the total height of the slope H, the effective subsoil parameters $c^{/}$, $\varphi^{/}$ and γ, the pore pressure ratio r_u (= u/γh), the individual slices of width b_i, height h_i and α_i-the inclination of slice on the failure arc with the horizontal is given by Eqn.-11 (Bishop, 1955). Total number of slices considered in the analysis is 15. To employ the novel LSSA algorithm, the objective function to be minimized is the FoS (F) and is given by:

$$F = \frac{\sum_{i=1}^{n}\left[\left\{\left(\frac{c'}{\gamma H}\right)\left(\frac{b_i}{H}\right)+\left(\frac{b_i}{H}\right)\left(\frac{h_i}{H}\right)(1-r_u)\tan\varphi'\right\}\frac{\sec\alpha_i}{1+\frac{\tan\alpha_i\ \tan\varphi'}{F}}\right]}{\sum_{i=1}^{n}\left(\frac{b_i}{H}\right)\left(\frac{h_i}{H}\right)\sin\alpha_i} \quad (11)$$

Table - 1. Slope geometry & soil properties

B	**H**	**β**	**$c'/\gamma H$**	**r_u**	**φ**
(m)	(m)	(deg)			(deg)
60	30	26.565	0.02	0.50	40

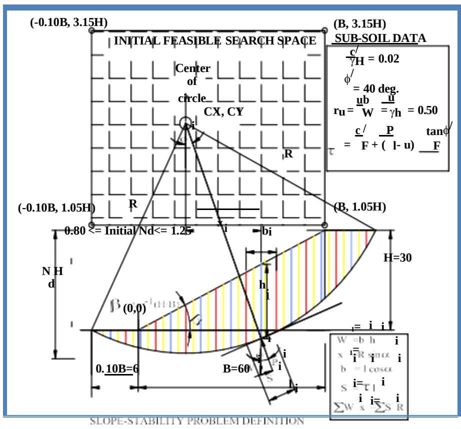


Figure 2. Slope-stability problem definition.

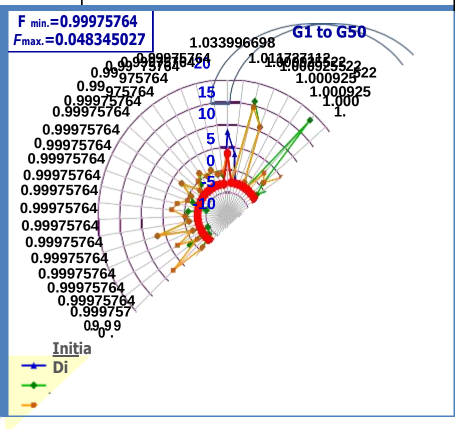


Figure 3. Obj. Function$_{min.}$ [FoS$_{min}$] Vs. a design variable (abscissa of circle centre, CX).

6. Imitating Sparrow Search: The Producer-Scrounger Model

Figure-3 shows the typical variation of a design variables (CX) values against minimum objective function $f(x)_i^{min.}$ (F_{min}), in a LSSA search, captured in 50 successive cycles of a typical simulation run. The variation is plotted in a concentrically radiating circular scale, clockwise, with the $f(x)_i^{min.}$ corresponding to cycle G_1 placed at the apex of the outermost circle. The process clearly demonstrates the stochastic nature of the algorithm. The global minimum FoS ($F_{min.}$), the corresponding maximum fitness ($F_{max.}$) in 'windowing technique' is shown on top left of the figure.

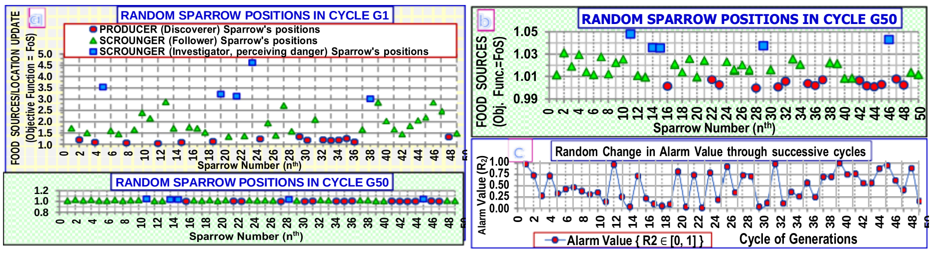


Figure 4 (a). Fast convergence of candidate solutions (Initial random G_1 converges within G_{50}).
(b). Sparrow positions at global optima (*Randomness maintained*).
(c). Random alarm value

Figure-4(a) shows two plots in the same scale to compare the fast convergence of candidate solutions. Initial random generation (G_1) of candidate solutions (FoS), yields the maximum at 4.617 and minimum at 1.034, that converges to 1.048 and 0.9997 respectively after 50 iterations (G_{50}), maintaining the randomness in population (Figure-4b).

Figure-4(c) shows the typical change of alarm (alert) value randomly during successive iterations. The nature of Safety Threshold (ST) and α graphs being very similar, are not reproduced here.

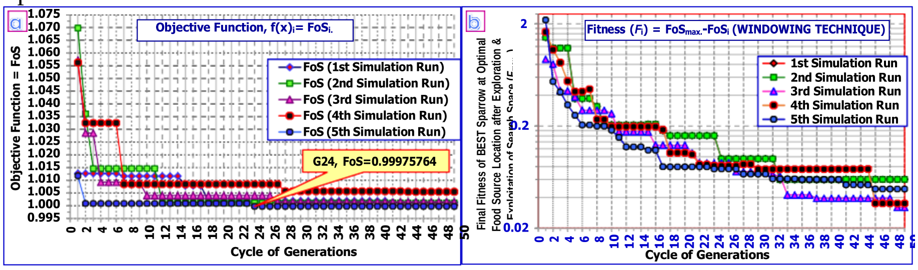
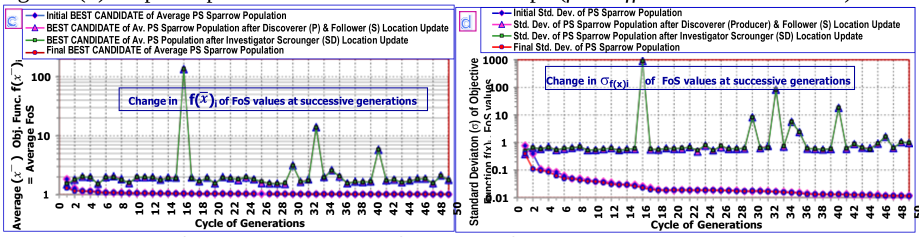


Figure 5(a). Fast convergence of FoS with increasing generations (*5 different simulation runs*).

Figure 5(b). Rapid improvement in final fitness landscape (*five different simulation runs*).



Statistics of change depiction of 'Factor of Safety' in the search procedure

Figure 5(c). Change in Av. (x̄) FoS with increasing generations [*random initial, exploration & exploitation (global & local search*), *wide random* (*investigator phase) & final optimal*].

Figure 5(d). Change in Std. Dev. (σ) of FoS with increasing generations [*random initial*, *exploration & exploitation (global & local search*), *wide random* (*investigator phase) & final optimal*].

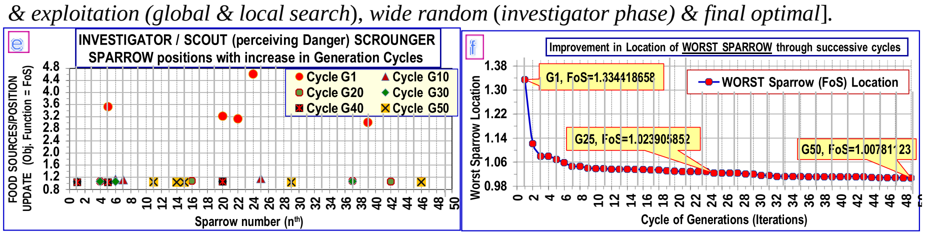


Figure 5(e). Change in the value of *Worst* candidate solution (FoS) with increasing generations. (*Quick convergence within 10 iterations*).

Figure 5(f). Rapid relocation of *Worst* sparrow to the best part of the search space.

Figure-5(a) & Figure-5(b) portrays the fast convergence of FoS and rapid increase of fitness of final solution (the optimal sparrow location and improvement in fitness landscape) with increasing generations. The $FoS_{min.}$ value is captured at G_{24} of the 5th simulation run. Figure-5(c) & Figure-5(d) shows the change in average ($F_{av.}=\Sigma f(x)_i/\Sigma n$) and standard deviation [$\sigma_{f(x)i}$] of FoS values respectively with increasing generations from initiation to final convergence through the global and local search procedure displaying the stochastic nature of the algorithm. Figure-5(e) depicts the change in the value of worst candidate solution with increasing generations. The power of the algorithm drives the worst objective function (FoS_{worst}) to near optimal values within 10 iterations, preserving its heuristic character. Figure-5(f) portrays the rapid relocation of the worst sparrow, to the best part of the search space with potential food source with increasing generations. Figure-6 portrays the fast random movement of candidate solutions to the best part of the search space with increasing generations. It is observed that initial deterministic search space turns out heuristic at the immediate next; thereby taking a quick shift towards the best part of search space. In the present case, modest convergence could be achieved in G_{30} and fine convergence at G_{50}.

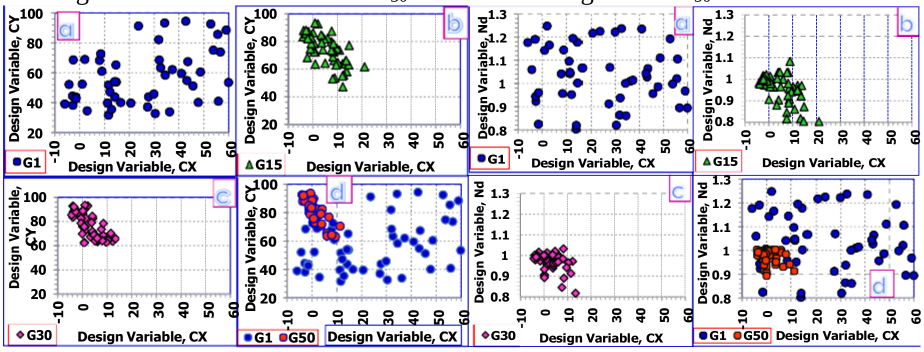


Figure 6. 2-D plots of the 3 design variables (CX, CY, N_d). Plots of CX Vs. CY & CX Vs. N_d: ***The artificial intelligent character of the algorithm is clearly evident***. Fast movement of solutions to best part of the search space as generation cycle increases [a-G_1, b-G_{15}, c-G_{30}, d-G_{50}].

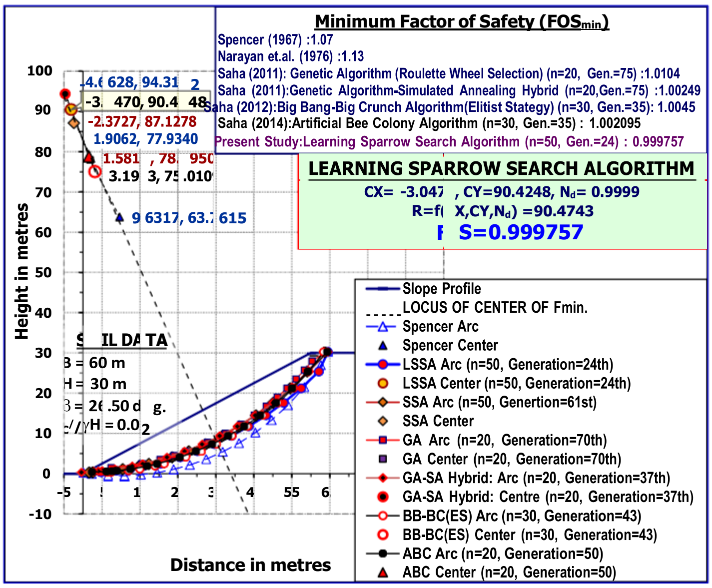


Figure 7. Comparison of results

Figure-7 gives a comparison of results. The LSSA result is superimposed on the traditional directed grid search result of Spencer (1967) and the results of author's experiments with other bio-inspired optimization algorithms (2003, 2008, 2011a, 2011b, 2014) for validation.

7. Conclusions

Diverse aspects of LSSA, an artificial intelligent population-based algorithm is presented. From the simulation results it emerged that LSSA has the ability to get out of the local minimum and can efficiently be used for multivariable, multimodal function optimization. The spectrum of application area of LSSA is widespread since it has the inherent potential of optimizing any multi-dimensional and multi-modal function. Multivariable functions; both continuous and discontinuous can be programmed. Function value evaluations at discrete points only enable it to handle non-differentiable functions at ease. Typical trials with 50 cycle run produced fine convergence.

A typical soil-slope investigated has revealed that minimum FoS to be about 7% less than that of a directed grid search (Spencer, 1967) and slightly less than that of GA (Saha, 2011a), hybrid GA-SA (Saha, 2011b), Big Bang-Big Crunch (Saha, 2012), Artificial Bee Colony (Saha, 2014) algorithm. Moreover, it is 13% less than that obtained by variational method (Narayan et al, 1976).

8. Appendix

In mathematical optimization, the Rosenbrock function (Howard H. Rosenbrock, 1960) is a non-convex function and a classic optimization problem (also known as the second function of De Jong). It is also called Rosenbrock valley or Rosenbrock banana function. The global minimum is inside a long, narrow, parabolic shaped flat valley. The function has the following definition:

$f(x)=\sum_{i=1}^{n-1}\left[100\left(x_{i+1}-x_1^2\right)^2+\left(1-x_i\right)^2\right]$. To find the valley is trivial, however

convergence to the global optimum is difficult and hence this problem is preferred frequently by researchers to test the performance of optimization algorithms. The test area is usually restricted to hypercube: $-2.048 \le x_i \le +2.048$; where, i=1,2,3,….n. It has a global minimum of f(x) = 0, obtainable at $x_i = 1$ (i=1,2,3,…..n).

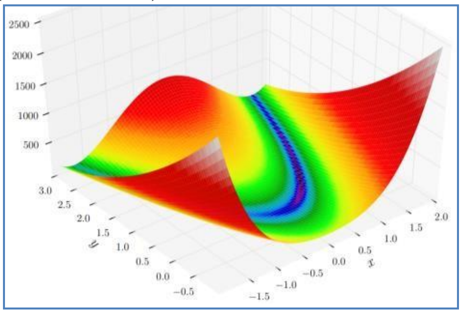


Figure 8. 3-D graph of Rosenbrock function.

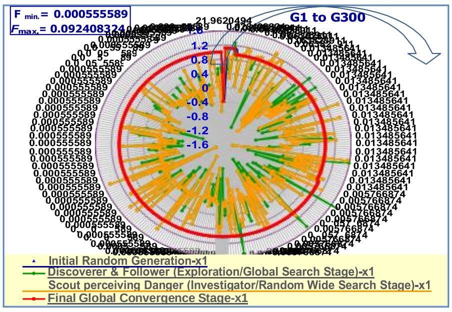


Figure 9. $f(x)i_{min}$.= Rosenbrock $Function_{min}$ Vs. a design variable, x_1.

In the current study, the 3D version of Rosenbrock function (Figure-8) is explored, as the slope-stability problem is typified by 3 variables: CX, CY, N_d. Figure-9 to Figure-14 unfold the powerful features of the learning sparrow search algorithm, as captured in the path to optimization and fast yet robust convergence of this difficult benchmark test function. The graphs are self-explanatory, and an in-depth scrutiny of the graphs reveals the powerful features of the algorithm. The convergence is steady and fast and it never gets trapped into local optima. In all simulation runs it is noticed that the powerful algorithm adjusts the fitness landscape to the best part of the search space within about 30 iterations and by and by refines the candidate solution swarm as the iteration cycle increases.

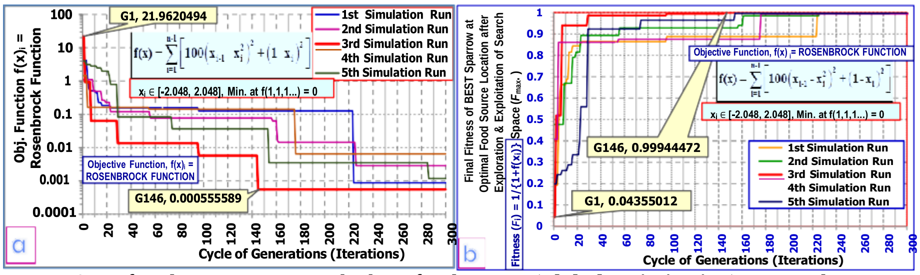


Figure 10. Defined convergence to the best food source (***global optimization***) captured in 5 simulation runs of LSSA (a) Objective Function (Rosenbrock Function) & (b) Its Fitness $\{F_i = 1/\{1+f(x_i)\}$ in traditional context [At Opt., $f(x_i)\approx 0$, $F_i\approx 1$]. [*Steady & fast global convergence*].

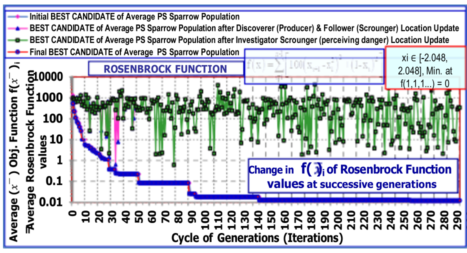


Figure 11. Exponential decrease of Average value ($\bar{x}$) of Rosenbrock function

[*random initial, exploration & exploitation (discoverer & follower phase), wide random (investigator phase) & final optimal*].
[*Steady & defined convergence*].

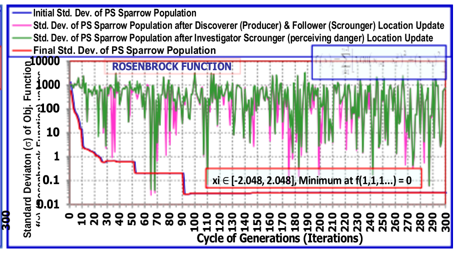


Figure 12. Exponential decrease of Standard Deviation, $\sigma_{f(x)}$ value of Rosenbrock function during the entire process of LSSA.

[Steady and defined convergence].

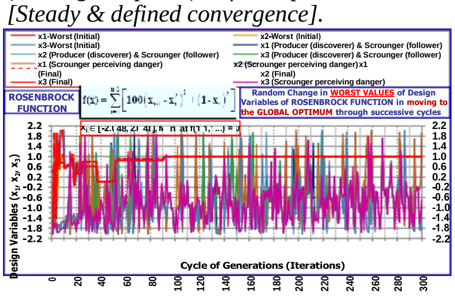


Figure 13. Random change in Worst values of the decision variables of Rosenbrock function through the entire process of LSSA.

[The WORST values of the 3 decision variables converges to optimal ($x_i\approx 1$) after about 80 runs of search space exploration & exploitation (global & local search)]

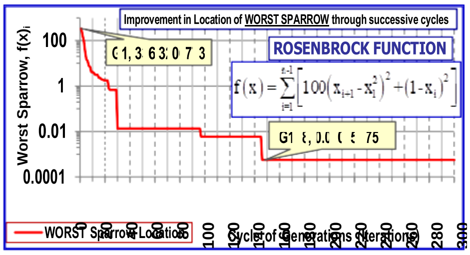


Figure 14. Rapid exponential relocation of Worst Objective Function (Rosenbrock) value to best part of search domain with increasing generations.

[Within 148 runs, worst Rosenbrock function value steadily converges].

References

1. Ackley, D. H.: A connectionist machine for genetic hill climbing, Kluwer Academic Publishers, Boston MA (1987).
2. Barnard, C. J., & Sibly, R. M. Producers and scroungers: A general model and its application to captive flocks of house sparrows. Animal Behaviour, 29, pp. 543–550 (1981).
3. Bishop A.W.: The use of slip circles in the stability analysis of slopes. Geotechnique. Vol.5, pp. 7-17 (1955).

4. Bonabeau E., Dorigo M., Theraulaz G.: Swarm Intelligence: From Natural to Artificial Intelligence, NY: Oxford University Press, New York (1999).
5. Goldberg, D. E.: Genetic Algorithms in Search, Optimisation and Machine Learning, Reading, Mass.: Addison Wesley (1989).
6. Holland J. H.: Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor (1975).
7. Karaboga Dervis: An idea based on honey bee swarm for numerical optimization. Technical Re-port TR06, Erciyes University, Engineering Faculty, Computer Engg. Dept. (2005).
8. Kennedy J. and Eberhart R. C.: Particle swarm optimization. Proc. of IEEE International Conference on Neural Networks, Piscataway, NJ. pp. 1942-1948. (1995).
9. Kirkpatrick S., Gerlatt Jr.C.D., and Vecchi M.P.: Optimization by Simulated Annealing, IBM Re-search Report RC 9355 (1982).
10. Kirkpatrick S., Gelatt Jr. C.D., and Vecchi M.P. Optimization by Simulated Annealing, Science, Vol. 220, pp. 671-680 (1983).
11. Narayan C.G.P., Bhatkar V.P. & Ramamurthy T.: Slope stability analysis by variational method. Indian Geotechnical Journal, Vol. 6, pp. 68-90 (1976).
12. Ouyang C., Zhu D. & Wang F.: A Learning Sparrow Search Algorithm. Hindawi. Computational Intelligence & Neuroscience, Vol.-2021, https://doi.org/10.1155/2021/3946958 (2021).
13. Rastrigin, L. A.: "Systems of external control" (1974).
14. Rosenbrock, H. H.: "An automatic method for finding the greatest or least value of a function", The Computer Journal, 3: 175–184 (1960).
15. Saha Abhijit: Genetic algorithm-based search in slope stability analysis. Proceedings of 12th Asian Regional Conf. in Soil Mechanics and Geotechnical Engineering, Singapore, Vol.-I, pp. 981-984 (2003).
16. Saha Abhijit: Genetic algorithm as a function optimizer in solving the tri-variate slope-stability problem. National Seminar on 'Geotechnique in Present Development Scenario', Kolkata, India (2008).
17. Saha Abhijit: Genetic algorithm-based search coupled with Boltzmann selection in locating the critical surface in slope stability. Proc. of 14th ARC, SMFE, Hong Kong, China, Paper No.115, pp. 172-177 (2011a).
18. Saha Abhijit: Adaptive elitist-population based 'Big-Bang Big-Crunch' optimization algorithm applied to slope-stability. Proc. of National Seminar of Kolkata Chapter of IGS, Paper No.7, Sept., 2011 (2011b).
19. Saha Abhijit: 'Big-Bang Big-Crunch' optimization in locating the critical surface in slope-stability. Proc. of Indian Geotech. Conference-2013, ISBN: 978-81-925548-1-5, Paper No.-82, Roorkee, India (2013).
20. Saha Abhijit: Artificial Bee Colony (ABC) algorithm applied to slope-stability in searching the critical surface. Proc. of National Seminar of Kolkata Chapter of IGS (2014).
21. Spencer E.: A Method of Analysis of the Stability of Embankments assuming Parallel Inter-Slice Forces. Geotechnique, London, Vol.17, pp. 11-26 (1967).
22. Xue J. & Shen B.: A novel swarm intelligence optimization approach: sparrow search algorithm. Systems Science & Control Engineering, Vol. 8, No. 1, pp. 22–34 (2020).
23. Yang X. S.: Firefly algorithms for multimodal optimization. Stochastic Algorithms: Foundations and Applications (Eds. O. Watanabe and T. Zeugmann), SAGA 2009, Lecture Notes in Computer Science, 5792, Springer-Verlag, Berlin, pp. 169-178 (2009).
24. Yang X. S.: Firefly Algorithm, Stochastic Test Functions and Design Optimisation. Int. J. Bio-Inspired Computation, Vol. 2, No. 2, pp. 78–84 (2010).