Enhancement of backtracking search algorithm for identifying soil parameters

Yin-Fu JIN[1] and Zhen-Yu YIN[1,*]

Affiliation:

1 Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China

* Corresponding author: Dr Zhen-Yu YIN, Tel. +852 34008470, Fax +852 23346389, E-mail: zhenyu.yin@polyu.edu.hk; zhenyu.yin@gmail.com

Abstract: In this paper, an enhanced backtracking search algorithm (so-called MBSA-LS) for parameter identification is proposed with two modifications: (1) modifying the mutation of original BSA considering the contribution of current best individual for accelerating convergence speed and (2) novelly incorporating an efficient differential evolution (DE) as local search for improving the quality of population. The proposed MBSA-LS is first validated with better performance than the original BSA and some other typical state-of-the-art optimization algorithms on a benchmark of soil parameter identification in terms of effectiveness, efficiency and robustness. Then, the efficiency of the MBSA-LS is further illustrated by two representative cases: identifying soil parameters from both laboratory tests and field measurements. All comparisons demonstrate that the proposed MBSA-LS algorithm can give accurate results in a short time. Finally, to conveniently solve the problems of parameter identification, a practical tool ErosOpt for parameter identification is developed by integrating the proposed MBSA-LS and some other efficient algorithms for readers to conduct the parameter identification using optimisation algorithms.

Key words: Parameter identification; optimisation; pressuremeter; constitutive model; local search; graphical user interface

1 Introduction

An impressive variety of constitutive models have been developed for soils in geotechnical engineering. These range from linear-elastic, perfectly plastic models (such as the Mohr-Coulomb model), to nonlinear models (such as the nonlinear Mohr-Coulomb [1]), to critical-state-based advanced models (such as the modified cam-clay model [2]; the critical-state based nonlinear Mohr-Coulomb [1, 3, 4]; UH model [5-7]; the SANISAND model [8]; and the micromechanical models by Chang and Hicher [9] and Yin et al. [10-13]), to hypoplasticity models [14-17]. These have allowed for the garnering of increasingly accurate and reliable descriptions of the mechanical behaviours of soils, which has also resulted in complexities along with additional model parameters. Thus, more parameters are generally required to be determined before the model can be applied to the solving of engineering problems, which poses a considerable challenge for engineers. Therefore, an efficient procedure, in conjunction with a tool for parameter identification, would be extremely useful.

Yin et al [18] distinguished three approaches: analytical methods, empirical correlations, and optimisation methods to determine soil parameters based on experimental data. Among these techniques, the inverse analysis by optimisation has been successfully used in the geotechnical area [3, 18-25] because it produces a relatively objective determination of the parameters for an adopted soil model, even for those that have no direct physical meaning. The existing optimisation techniques can be divided into two categories: (1) deterministic optimisation techniques; and (2) stochastic optimisation techniques. Deterministic optimisation techniques, such as gradient-based algorithms and simplex [19, 26], work with a single solution and are local minimiser in nature because they begin the search procedure with a guess solution (often chosen randomly in the search space), and if this guess solution is not close enough to the global minimum solution, it is likely to be trapped in the local minimum solution. Most of the deterministic optimisation techniques are designed to solve a particular class of optimisation problem. On the other hand, stochastic optimisation techniques such as evolutionary algorithms [27], simulated annealing [28], particle swarm optimisation [29] can guarantee an optimal global solution, although it rely heavily on computational power. The speed and accuracy are two important aspects of performance for optimisation algorithm used in parameter identification. However, the development of computational techniques, such as distributed parallel computing, can reduce the computational time. Therefore, a fast and efficient optimisation algorithm for parameter identification is required in geotechnical engineering.

Although optimisation-based procedures are highly effective for solving parameter identification problems, the writing of a computer programme for implementing the sophisticated algorithms according to user ~~needs~~ requires certain programming expertise, not to mention considerable time and effort. Because of the tedious nature of this task, the use of an optimisation tool is more advantageous and attractive. To date, various kinds of optimisation-based tools have been developed. These offer an object-oriented design for evaluating fitness functions using a variety of optimisation algorithms, which are written against the framework of Java (e.g., Evolvica[30], JCLEC[31], jMetal[32]), C#(e.g., MOEAT[33]) or MATLAB (e.g., YALMIP[34]). However, among these tools, the identification of soil parameters has not received any attention. Therefore, a case can be made for the development of a tool offering a powerful environment for various kinds of parameter identification, so that engineers can apply it to solve a range of engineering problems without the need to reproduce its model of operation, which is also a challenge for them.

In this paper, the methodology of optimisation-based parameter identification is first briefly introduced. Then, an enhanced backtracking search algorithm (BSA) with DE-based local search (MBSA-LS) is proposed and examined. Subsequently, the carrying of two cases of parameter identification is recounted. Finally, the development of an optimisation-based parameter identification tool (ErosOpt) with efficient optimisation algorithms for geotechnical engineering is described.

2 Methodology of optimisation-based parameter identification

The mathematical procedure of optimisation essentially consists of three parts: (a) an identification procedure having a clear structure; (b) the formulation of an error function to measure the difference between model responses and experimental results; and (c) the selection of an optimisation strategy to enable a search for the minimum of this error function.

2.1 Procedure of parameter identification

To ensure successful parameter identification, a procedure that has a clear structure is both necessary and important. The aim of such a procedure is to define the error function as well as the search strategy. Therefore, the procedure should be defined before the optimisation is conducted. Fig. 1 shows the flowchart of the optimisation-based parameter identification.

2.2 Formulation of an error function

For the optimisation problem of identifying the parameters of constitutive models based on experimental or observed data, the parameters of the constitutive model in question play the role of the variables to be optimized. Theoretically, more reliable model parameters can be obtained if a range of qualitatively different experimental tests ~~from~~ form the database for the optimisation. To carry out an inverse analysis, a function that can evaluate the error between the experimental and numerical results, so-called error function or fitness function (see Fig. 2), must be defined before being minimised.

To render the error independent of the type of test and the number of measurement points, an advanced error function can be adopted with two modifications of 100 percentage and of adding weight to each calculation point (Levasseur et al. [20]). The average difference between the measured and simulated results is expressed in the form of the least square method,

$$\mathrm{Error}\left(x\right)=\sqrt{\frac{1}{N}\sum_{i=1}^{N}w_i\left(\frac{U_{\exp}^{i}-U_{num}^{i}}{U_{\exp}^{i}}\right)^2}\times 100 \qquad (1)$$

where w_i is the weight for the calculation at point i.

Generally, deformation and stress are two extremely important indicators of the mechanical behaviour of soils. For the identification of soil parameters, the error function should involve both. Therefore, the generalised individual error function can be expressed as follows:

$$\min\left[\mathrm{Error}(x)\right]=\min\left[\mathrm{Error(stress)},\ \mathrm{Error(deformation)}\right] \qquad (2)$$

For mono-objective problems, the total error function is expressed as:

$$\text{Error\_total}(x)=\sum_{\mathrm{i}=1}^{Num}\left(l_i \cdot \text{Error}(x)_i\right) \tag{3}$$

where *Num* is the number of the individual errors (e.g. *Num*=2 for selecting the void ratio and deviatoric stress); and $\text{Error}(x)_i$ is the value of the individual error corresponding to the objective *i*. l_i is the weight factor with $\sum(l_\mathrm{i}) = 1$. Generally, the l_i is suggested to 1/*Num* for a trade-off between different objective experiments. Finally, the set of parameters with the lowest error value can be selected as optimal.

2.3 Selection of the search strategy

The solution to an optimisation problem is a vector x_0 which, for any $x_l \leq x \leq x_\mathrm{u}$, satisfies the following condition, which is a global minimum:

$$f(x_0) \leq f(x) \tag{4}$$

where *f* is the error function, and in this study the *f*(*x*) is represented by $\text{Error\_total}(x)$

To obtain a more accurate choice, a highly efficient optimisation method, with the ability to search for a global minimum, should be adopted. Due the complexity of parameter identification for soil constitutive models, the original stochastic optimisation methods (such as genetic algorithm (GA) and differential evolution algorithm (DE)) cannot fully ~~applicable~~ apply so that some modifications on specific optimisation operators should be performed. Coincidentally, the authors have some experiences on the performance improvement of various optimisation methods in geotechnical engineering [1, 3, 18, 19, 21, 25, 35], which is the primary base for developing ~~an~~ a practical platform. To develop a more efficient algorithm, the basic backtracking search algorithm proposed by Civicioglu [36] was enhanced by modifying mutation and implementing ~~an~~ a novel local search.

3 Enhancements of backtracking search optimisation

3.1 Basic backtracking search optimisation

The backtracking search optimisation (BSA) is a population-based evolutionary algorithm (EA). It contains five processes: initialisation, selection-I, mutation, crossover, and selection-II. For any evolutionary algorithm, a population of candidate solutions (called individuals) to an optimization problem is evolved towards better solutions. Traditionally, solutions are represented in real number (such as model parameters in Jin et al. [35]), but other encodings are also possible. The evolution is an iterative process usually starting from a randomly generated population and this population in each iteration is also called a generation. In each generation, the fitness of each individual is evaluated, in which the fitness is usually the value of the error function. Better individuals are stochastically selected according to the value of fitness from the current population, and some of them are probably modified (e.g., recombined and mutated according to the operation probability) to form a new generation. The new generation is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a pre-defined maximum number of generation has been reached, or a satisfactory fitness level has been achieved.

A detailed description of the BSA can be found in [36]. The MATLAB code of basic BSA can be found in http://www.pinarcivicioglu.com/bsa.html. Here, we briefly describe the five steps of the basic BSA.

Step 1: Initialisation. The evolution population P_{ij} and the history population (old population, $oldP_{ij}$) are randomly initialised, as follows:

$$P_{ij} = low_j + rand \cdot \left(up_j - low_j\right) \tag{5}$$

$$oldP_{ij} = low_j + rand \cdot \left(up_j - low_j\right) \tag{6}$$

where *rand* is a uniform distribution function within [0, 1], i=1, 2 , · · · , N, and N is the size of population, j=1, 2, · · · , D, and D is the dimensionality of the space of variables. The quantities low_j and up_j are the lower and upper boundaries of variables.

Step 2: Selection-I. In the beginning of each iteration, the history population *oldP* is redefined through Eq.(7), and then the order of the individuals in *oldP* is randomly changed by Eq.(8).

$$oldP = \begin{cases} P, & \text{if}\left(a < b \middle| a,\ b:\ U\left(0,1\right)\right) \\ oldP, & \text{otherwise} \end{cases} \tag{7}$$

$$oldP = permuting\left(oldP\right) \tag{8}$$

where a and b are two uniformly distributed random numbers between 0 and 1, and *permuting* function is a random shuffling function used to a randomly selected previous generation as the historical population.

Steps 3-4: Mutation and crossover. The mutant individuals are generated by the historical individuals and current individuals as shown in Eq. (9). Then, the crossover operator is conducted by Eq. (10). A binary integer-valued matrix (*map*) of size N×D guides the crossover directions of BSA algorithm. The value of *map* is controlled by the mixrate parameter that is the only control parameter should be determined in BSA, the details can be found in [36].

$$M = P + F \cdot \left(oldP - P\right) \tag{9}$$

$$V_{ij} = \begin{cases} P_{ij}, & \text{if } map_{\mathrm{ij}}\text{=1} \\ M_{ij}, & \text{if } map_{\mathrm{ij}}\text{=0} \end{cases} \tag{10}$$

where F is a scale factor which controls the amplitude of the search direction matrix, and its value is commonly set to 3×randn, where randn~N (0, 1). V_{ij} is the value of the jth variable for the ith trial individual.

Step 5: Selection-II. At this step, the population of the next generation P_i^{new} is generated according to a greedy selection mechanism. As shown in Eq.(11), V_{i} is accepted if it provides a better function value than P_{i} considering the minimum problem.

$$P_i^{new} = \begin{cases} V_i, & \text{if } f\left(V_i\right) \le f\left(P_i\right) \\ P_i, & \text{otherwise} \end{cases} \tag{11}$$

Then, Steps 2–5 are continually performed until the termination criterion is satisfied. Fig. 3(a) shows the flowchart of basic BSA.

3.2 Motivation of improvement for optimisation algorithm

Exploitation and exploration are key search mechanisms in solving complex optimisation problems. Numerous optimisation cases [37-44] have demonstrated that basic BSA has a strong ability of exploration because of its mutation and crossover strategies, which the better or worse individuals have the same probability to be selected. However, the poor exploitation of basic BSA is also due to the strategies, leading to the problems of slow convergence speed and low search precision. Thus, the motivation of this study is to improve the convergence speed while ensuring the exploration ability of basic BSA for parameter identification in geotechnical engineering.

In the search process of basic BSA, the mutation operation plays a significant role in generating new individuals. However, only the historical population information is used to guide the search, and the information of the best individual of current population is ignored. Practice has shown that considering the best individual can improve the convergence speed and increase the exploitation capability of the algorithm [18, 45]. Furthermore, the local search can also accelerate the convergence speed to improve the ability of exploitation.

In this study, a modified backtracking search optimisation algorithm with a novel local search (MBSA-LS) is proposed. The proposed MBSA-LS algorithm considers the contribution of the current best individual in convergence speed and adopts a novel differential evolution (DE)-based local search to further refine the quality of current population. During the optimisation process, the local search could accelerate the convergence speed, but could easily be trapped in the local minimum. On the other hand, the basic BSA process can overcome the local minimum problem due to its strong exploration ability. Therefore, the advantages of each process are obtained through this hybrid strategy, resulting in better optimisation performance.

3.3 Modification 1: enhanced backtracking search algorithm

The mutation operation of basic BSA is modified by considering the current best individual as follows,

$$M = P + F \cdot (oldP - P) + F \cdot (pbest - P) \tag{12}$$

where the *pbest* is the current best individual; *F* is a uniformly distributed random number within [0.3, 1.0], which is different from the *F* using in basic BSA [36] and the *F* in other similar algorithms that consider the best individual in the mutation [38, 39, 43].

Furthermore, the mixrate of the *i*th individual proposed by Nama et al. [43] is adopted to improve the performance of crossover as follows,

$$\text{mixrate} = 0.5 \cdot \left(1 + rand\left(0,1\right)\right) \tag{13}$$

Through the proposed mutation, some individuals are devoted to improving the population diversity by learning from the historical population, while other individuals are focused on enhancing the convergence speed via learning from the best individual in current population. Consequently, the performance of algorithm is enhanced with both considerations of the exploration and exploitation abilities.

3.4 Modification 2: implementation of DE-based local search

To further refine the quality of current population, an elite method using the differential evolution as local search is introduced. The DE-based local search aims to find the better solution around the current best individual, which is performed after the BSA stage. The mutation strategy is presented as follows,

$$P_{DE,i} = P_{best,i}^{new} + 0.5 \cdot \left(P_{r1}^{new} - P_{r2}^{new}\right) + 0.5 \cdot \left(P_{r3}^{new} - P_{r4}^{new}\right) \tag{14}$$

where the indices r_1, r_2, r_3 and r_4 are distinct integers uniformly chosen from the set $\{1,\ 2,\ ...,\ N\}$; $\left(P_{r1}^{new} - P_{r2}^{new}\right)$ and $\left(P_{r3}^{new} - P_{r4}^{new}\right)$ are two difference vectors to mutate the corresponding parent $P_{best,i}^{new}$; $P_{best,i}^{new}$ is the best individual in the current generation *i*, which is randomly chosen as one of the top 10% individuals in the current population obtained by Selection-II of BSA. It is obviously seen that the DE-based local search presented in Eq.(14) is employed to generate new individuals $P_{DE,i}$ around the best individual.

After mutation, a binomial crossover operation forms the final trial/offspring vector

$$\left(P_{\mathrm{DE}}^{new}\right)_{i,j} = \begin{cases} \left(P_{DE}\right)_{i,j}, & \text{if } \mathrm{rand}\left(0,1\right) \le CR \text{ or } j = j_{rand} \\ \boldsymbol{x}_{i,j}, & \text{otherwise} \end{cases} \tag{15}$$

where ~~rand(*a*, *b*) is a uniform random number in the interval [*a*, *b*] and is independently generated for each *j* and each *i*;~~ j_{rand}=randint (1, *D*) is an integer randomly chosen from 1 to *D* and is newly generated for each i, D being the dimension of the problem; the crossover probability $CR\in[0, 1]$ corresponds roughly to the average fraction of the vector components that are inherited from the mutation vector; CR=0.9 was taken in this study.

In order to avoid a rapid loss of diversity, a competition between parents and children with elitism strategy is adopted to perform the selection III. In the selection, 10% of individuals with the highest fitness are selected from the parents and children to survive to the next generation. The remainders are chosen by tournament selection from the mating pool composed of parents and children apart from the 10% individuals.

3.5 Framework and pseudo code of the enhanced MBSA-LS

Based on aforementioned descriptions, the pseudo code of MBSA-LS is summarized in Algorithm 1 and the flowchart of MBSA-LS is shown in Fig. 3(b). It can be observed that the proposed MBSA-LS retains the simple structure as that of basic BSA apart from the implementation of local search and the selection III.

Algorithm 1. Pseudo code of MBSA-LS

```
1. Initialise population size and maximum number of iterations;
2. Initialise the current and historical populations using Eqs.(5) and  (6) for P and oldP;
3. Evaluate the errors of all individuals in the current population P;
4. iter=0;
5. For iter=1:Max_iter
6. Perform selection-I using Eqs.(7) and  (8) to form the historical population oldP;
7. Find the best individual from the current population;
8. Perform the new mutation using Eq.(12) to generate the trial population M;
9. Perform the crossover using Eq.(10) to generate the trial population V;
10. Check the Boundary for each trial individual
11. Evaluate the errors of trial population V;
12. Implement greedy selection-II using Eq.(11) to remain the better individual $P^{new}$;
13. Perform DE-based local search to generate new offspring $P_{\mathrm{DE}}^{new}$;
```

14. Perform the competition between P_{DE}^{new} *and P to generate the new P for next iteration;*

15. End For

3.6 Evaluation of performance

To test the performance of MBSA-LS for identifying parameters of constitutive models, several parameter identifications from synthetic test data were carried out. Note that the purpose of using synthetic test data as the objective is to facilitate a fair comparison due to the known of optima. To illustrate the effectiveness of MBSA-LS, five state-of-the-art algorithms (Genetic algorithm−GA [46], Differential evolution−DE [47], Particle Swarm Optimisation−PSO [48], Simulated Annealing−SA [49], Artificial Bee Colony−ABC [50]), the Whale Optimisation Algorithm−WOA [51], the basic BSA, the NMGA [21] and NMDE [18] were used as comparison algorithms. The MATLAB codes of the used state-of-the-art algorithms can be freely downloaded on www.yarpiz.com. The source codes of WOA are publicly available at http://www.alimirjalili.com/WOA.html. The performance of all adopted algorithms is tested on the following case: Identifying parameters of SIMSAND model (see Appendix II for a brief introduction) from synthetic test data.

All of the calculations were conducted on the same machine with an Intel (R) Core (TM) i7-6800K CPU @3.40 GHZ, RAM 16 GB, and Windows 10 64 bit operating system, with MATLAB 2017b. All calculations run in a parallel environment in MATLAB with SPMD technique, as shown in Appendix-I. The population size (popsize) should be multiple of the number of workers (n_workers=12 for this study, where two workers means one CPU core), and the population size was set to 48 for this case. The sensitivity of the number of population size was carried out and is shown in details in the following section. The maximal number of iterations was used as the stopping criterion for all algorithms. In this experiment, the maximal number of iterations was 200 for all calculations. The settings of all used algorithms are summarized in Table 1.

To generate the synthetic tests, the SIMSAND model proposed by Jin et al. [1, 3, 4, 21, 24, 52, 53] and a set of typical parameters of were adopted, as summarized in Table 1. Fig. 4 shows the illustration of synthetic test data generated by SIMSAND. Using these values, three synthetic drained triaxial tests (e_0=0.6 and p'_0=800 kPa, e_0=0.7 and p'_0=200 kPa, e_0=0.8 and p'_0=25 kPa) were computed. According to Jin et al. [1], tests with wider range of confining stress involved in parameter identification would be helpful on finding the accurate critical state line (CSL). The results of three synthetic drained tests are shown in Fig. 5.

The bounds of all parameters are also summarized in Table 2. In this case, the Poisson's ratio υ, bulk modulus K_0 and nonlinear elasticity parameter n were fixed. The remaining parameters of SIMSAND were identified using the parameter identification, adopting MBSA-LS and other state-of-the-art algorithms, respectively.

The optimal parameters obtained by different algorithms are summarized in Table 2. It can be seen that the MBSA-LS can accurately find the pre-set optimal parameters while other algorithms can't find the pre-set optimal solution, which demonstrate a high effectiveness of MBSA-LS. With respect to the convergence speed, Fig. 6 shows the comparison of evolution for different algorithms. Obviously, the MBSA-LS has the fastest convergence speed compared to other algorithms, which indicates a high efficiency of MBSA-LS. Table 3 summarises the calculation time for different optimisation algorithms. The calculation time for other algorithms is normalised by that of the MBSA-LS algorithm, as shown in Fig. 7. Apart from DE, BSA and WOA, all other algorithms take longer calculation time than the MBSA-LS. Note that the MBSA-LS combines the BSA and DE in a series connection, the longer calculation time of MBSA-LS than DE and BSA is reasonable, but the performance of MBSA-LS is much better than DE and BSA and also WOA. Therefore, the proposed MBSA-LS outperformed the other nine algorithms in terms of accuracy (effectiveness) and convergence speed (efficiency) on parameter identification.

3.7 Sensitivity of size of population

It is a general consensus that the convergence of optimization becomes faster with the increase of the number of population size. For a given optimization algorithm, the amount of population size evaluates its efficiency. A good optimization algorithm can give an accurate solution with a small population within a few generations. Therefore, to highlight the performance of MBSA-LS, a sensitivity analysis for the size of population for different optimizations was carried out and compared. According to the preliminary results, the SA, ABC, GA, NMDE and NMGA with poor performance of effectiveness and efficiency were discarded in the following comparison.

For the MBSA-LS, the population size decreases from 48 to 12 at an interval 12 due to the n_works=12. For BSA, DE and WOA, the population size increases from 48 to 72 at an interval 12. The number of generation for all calculations is 200. Other settings for all involved algorithms are also presented in Table 1. As shown in Fig. 8, it can be found that a relatively more accurate result is obtained by MBSA-LS with a smaller population (popsize=24 and 36) than others. In contrast, the results are still very poor for DE, original BSA and WOA regardless of increasing the population. All comparisons demonstrate that the proposed MBSA-LS can give an accurate solution even if the population is small with saving computational cost, which apparently indicates the robustness of MBSA-LS.

Overall, the proposed MBSA-LS has not only the outstanding performance in terms of effectiveness and efficiency but also the robustness for parameter identification in geotechnical engineering.

4 Representative cases

In this section, two representative cases of parameter identification were selected to examine the performance of optimisation algorithms. For each case, the MBSA-LS is adopted and compared to results by nine typical algorithms (GA, DE, PSO, SA, ABC, WOA, BSA, NMGA and NMDE). The settings of all compared algorithms are summarized in Table 1.

4.1 Case 1: Identifying parameters of sand from laboratory tests

The tests selected for this case were triaxial tests conducted on Hostun sand by Liu et al. [54] and Li et al. [55]. In the optimisation, three triaxial tests (two drained tests p'_0=100 kPa, e_0=0.849; p'_0=200 kPa, e_0=0.832 and one undrained test p'_0=200 kPa, e_0=0.73;) were selected as the objective in the optimisation. All tests were isotropically consolidated to the corresponding consolidation pressure before shearing.

The adopted soil model was SIMSAND. A typical value for Poisson's ratio, υ=0.2, can be assumed. Apart from the elasticity parameters (K_0=45 and ζ=0.6), which can be easily obtained from the compression curves [3, 4], the other parameters (ϕ', e_{ref}, λ, ξ, A_{d}, n_{d}, k_{p} and n_{p}) were funnelled into the optimisation procedure without considering the grain breakage. The intervals of the parameters given in Table 4 were much larger than those corresponding to the typical values.

For all algorithms, the initial population size is set to 48 and the number of generations was set to 100. Fig. 9 shows the evolution of the minimum objective error with the increasing number of generations for all used algorithms. This indicates that the MBSA-LS as well as NMGA and NMDE perform well in parameter identification, with rapid convergence and low objective error, compared to other state-of-the-art algorithms. All obtained parameters for said sand are summarised in Table 5. Fig. 10 displays the comparison between the objectives and simulations. Good agreement between the two demonstrates that the developed MBSA-LS is efficient in parameter identification. Note that the ~~obtained~~ parameters obtained by NMDE and NMGA are almost the same ~~to~~ as that obtained by MBSA-LS, which also indicates the high efficiency of NMDE and NMGA. However, according to the computational time shown in Fig. 11, the proposed MBSA-LS is superior to NMDE and NMGA. Furthermore, the simulations demonstrate that SIMSAND is able to capture the mechanical behaviours of sand (including critical state, contraction and dilatancy). Therefore, the proposed MBSA-LS has been proven effective in identifying soil parameters from laboratory tests.

4.2 Case 2: Identifying parameters of soft clay from field tests

Two self-boring pressuremeter (SBP) tests performed in the Burswood Peninsula site at a depth of 5.25 m by Lee and Fahey [56] were selected as the objectives. A 2D model, with boundaries in an axisymmetric condition to simulate the pressuremeter, was created in ABAQUS, as shown in Fig. 12. This was ~~consistently~~ consistent with the field tests. The upper and bottom sides were only fixed for vertical displacement, while the right side was fixed for horizontal displacement. The loading could then be generated by applying horizontal displacement at the left side. Therefore, the horizontal displacement had its biggest value at the left side and gradually decreased to zero as it reached the right side. A total of 240, 4-node reduced-integration elements (CAX4R), were used to simulate the soil. The same displacement as seen in a typical field test was applied, and at each step, the same displacement increment was applied.

For the two tests, the expansion phase was conducted at two different rates up to around 10 % of the cavity strain: 0.167 %/s and 0.0185 %/s respectively. The effective vertical stress σ'_{v0} was 31 kPa and the lateral earth coefficient K_0 was 0.9. The water pore pressure was 38.8 kPa. In this case, the elasto-viscoplastic model "ANICREEP" was adopted to simulate the rate-dependency behaviour of soft clay. To focus on identifying the rate-dependent parameters, the initial void ratio, e_0, the swelling index, κ, and the compression index, λ, with the permeability (k=3.3×10^{-9} m/s) measured from oedometer tests by Lee and Fahey [56] were given in the optimisation procedure. The Poisson's ratio υ was assumed to be 0.3, which can be considered as a typical value assuming ideally elastic-perfectly plastic Mohr–Coulomb (MC) model. The other parameters (M, σ'_{p0} and $C_{\alpha e}$) were determined by the proposed algorithm. The intervals of parameters given in Table 6are much larger than those corresponding to typical values.

For all algorithms, the population size is set to 30 and the number of generations was set at 50. Other settings for all optimization algorithms were kept as default values shown in Table 1. Fig. 13 shows the evolution of minimum objective error with the increasing number of generations for all used algorithms. Compared to the state-of-the-art algorithms, rapid convergence with low objective error for MBSA-LS as well as NMGA and NMDE is found, indicating a high efficiency of such algorithms. Fig. 14 shows the consumed calculation time for three outstanding algorithms. Among them, the MBSA-LS takes the shortest time, which highlights a high efficiency of MBSA-LS.

The obtained optimal parameters of ANICREEP are summarized in Table 7. Fig. 15 compares the measured and predicted results using the parameters by MBSA-LS for the SBP tests. The identified M (≈1.70) and σ'_{p0} (ranging from 57.0 to 57.6 kPa) by MBSA-LS were close to values measured by Low[57]. The identified parameter $C_{\alpha e}$ ranging from 0.0199 to 0.021 was close to experimentally estimated $C_{\alpha e}$=0.020 gained by Low[57] based on the strain-rate dependency parameter β = 0.055 of Burswood clay using the unified formulation of Yin et al. [58, 59]. Thus, the proposed MBSA-LS has proven its effectiveness for identifying soil parameters from field tests.

5 Development of ErosOpt tool for parameter identification in practice

In order to conveniently solve the problems of parameter identification in geotechnical engineering, ErosOpt, a practically useful tool is developed. ErosOpt makes it easy to specify and solve geotechnical optimization problems without expert knowledge, and at the same time it allows experts to implement advanced algorithmic techniques or advanced constitutive models. The distinguishing features of ErosOpt are presented in the following sections.

5.1 Mixed-language programming

Fig. 16 shows a schematic overview of the mixed-language programming for ErosOpt. The tool was programmed using an admixture of Microsoft Visual C$^{\#}$, MATLAB and FORTRAN. The graphical user interface of ErosOpt was programmed in C$^{\#}$, the post-processing (for plotting figures, exporting the results and reading the help documentation) was realised using MATLAB, and the constitutive models were programmed in FORTRAN. All MATLAB files were built as dynamic library files (*.dll) under the .NET Framework 4.0. The version used was MATLAB 2016b.

5.2 Parallelization

The parameter identification using optimisation method always requires numerous simulation runs. For large-order computational models, the computational demands involved in the fitness calculation are excessive, sometimes to the point of being unacceptable when tracking large geotechnical engineering. ErosOpt is developed for large number of parameter identification with a parallelized implementation with the SPMD (Single Program/Multiple Data) technique using MATLAB. The SPMD technique is computationally efficient when the computational time for a fitness evaluation is the same independent of the location of sample in the parameter space. The details of the implementation of parallel computing using SPMD technique are presented in Appendix-I.

5.3 Architecture of ErosOpt

The general structure of ErosOpt is shown in Fig. 17 and the three main features are summarised in this section.

5.3.1 Dealing with various parameter identification problems

Three common types of parameter identification problem are covered by ErosOpt: (1) identification of parameters based on the results of laboratory tests; (2) identification of parameters based on the results of in-situ tests; (3) identification of parameters based on field measurements. All test or monitoring results can be imported into ErosOpt from the Excel file (*.xlxs) at a fixed format. The user should prepare the measured data according to the stated format before saving them as an Excel file.

5.3.2 Provision of a variety of constitutive models of soils

In ErosOpt, a total of five soil constitutive models (NLMC, MCC, SIMSAND, ASCM and ANICREEP) are provided, which covers most commonly adopted mechanical models of soils. Other advanced soil models will be available in the next version of ErosOpt. The presented soil models are briefly described in Appendix-II. To improve the extensibility of the proposed tool, the user-defined material (UMAT) is supported, which allows the user to implement other soil models in ErosOpt.

5.3.3 Provision of various efficient optimisation algorithms

Apart from the proposed MBSA-LS, the ErosOpt also provides three existing efficient optimisers: (1) the Nelder-Mead simplex (NMS) proposed by [26]; (2) the genetic algorithm enhanced by NMS, named “NMGA” for simplicity proposed by the authors [21]; and (3) the differential evolution (DE) algorithm enhanced by NMS, and named “NMDE” for simplicity also proposed by the authors [18]. All these algorithms can be found in Appendix-III and can be used to effectively solve the parameter identification problems. The performances of the adopted algorithms have been already validated [18, 21]. Although the NMS algorithm has drawback in guaranteeing the global minima, the convergence speed of NMS algorithm is fast so that it can be very effective for low-dimensional problems [18].

5.4 Graphical user interface and usage instructions

Fig. 18 shows the main interface of ErosOpt, which is divided into three zones: the *Navigation* pane, the *Command* pane and the *Manipulation* pane. The *Navigation* pane features a simple procedure with five modules, *Problem*, *Soil model*, *Algorithm*, *Results* and *Report*, which are aimed at guiding the user to solve problems. The *Command* pane boasts three commands that can help the user to run and exit the ErosOpt or get help on how to use the programme. The usage of instructions for ErosOpt is shown in Fig. 19.

5.5 Advice on the choice of parameters for optimisation based identification

As demonstrated in previous two cases, the parameter identification from conventional tests (e.g. one oedometer or isotropic compression test and three drained or undrained triaxial tests) or from field tests (e.g. one oedometer test with PMT tests) is effective and efficient. To some extent, for a specific problem such as excavation, tunnelling, etc., it is suggested to conduct a sensitivity analysis first through which the most important parameters relating to key responses of the given problem can be determined [18]. This function has been integrated in the tool (see Fig.II-1).

6 Conclusions

An enhanced backtracking search optimisation with implementing DE-based local search (MBSA-LS) was proposed to improve the performance of parameter identification. The new MBSA-LS modifies the mutation of basic BSA by considering the contribution of the current best individual for accelerating the convergence speed, and integrates a different evolution (DE) based local search to further improve the quality of population. The outstanding performance of MBSA-LS was evaluated by conducting a typical case of parameter identification of an advanced soil model from synthetic data using different algorithms including seven additional state-of-the-art algorithms. All comparisons demonstrate that the enhanced MBSA-LS outperforms the other seven algorithms for parameter identification in terms of accuracy (effectiveness) and convergence speed (efficiency).

Then, two selected cases on typical problems surrounding the identification of soil parameters from both laboratory tests and field measurements were carried out, the outcome demonstrating that the proposed MBSA-LS is a highly efficient and ease of use algorithm in engineering practice for parameter identification.

Finally, to make the optimization-based parameter identification practically useful, an optimisation-based parameter identification tool (ErosOpt) for geotechnical engineering was developed. The tool provides support for both research and teaching regarding the practice of optimisation methods in the fields of geomechanics and geotechnics. Simple and clear interfaces enable great ease of use for engineers, while the friendly graphical interface help users to view and analyse results. The presented cases in this paper can be reproduced by the developed platform ErosOpt.

Acknowledgments

This research was financially supported by the Research Grants Council (RGC) of Hong Kong Special Administrative Region Government (HKSARG) of China (Grant No.: 15209119, PolyU R5037-18F) and the National Natural Science Foundation of China (Grant No. 51579179).

Appendix-I: Parallel computing using SPMD technique

Fig. I-1 shows the parallel computing strategy in parameter identification using the SPMD technique, where "Error_function" represents the user-defined cost function for computing the fitness; N_workers is the total number of computer workers (e.g., 8 for a 4-cores CPU) and "labindex" is the index of computer workers, automatically identified in MATLAB. Note that each worker can operate on one different data set or different portion of distributed data, and can communicate with other participating workers while performing the parallel computations. The entire calculation will be distributed among different computer workers according to the value of labindex. Note that the size of population N should be an integer multiple of the number of computer workers, N_workers, in parallel computation.

Appendix-II: Brief introduction of constitutive models in ErosOpt

Fig. II-1 shows the GUI window of selecting the soil model in ErosOpt. The Nonlinear Mohr-Coulomb model (NLMC) was developed against the framework of Mohr-Coulomb, by implementing nonlinear elasticity, nonlinear plastic hardening, and a simplified 3D strength criterion (Jin et al.[1]). This model is similar to the shearing part of the Hardening Soil model (HS). Meanwhile, the Modified Cam-Clay model (MCC) was developed by researchers at the University of Cambridge based on the mechanical behaviour of remoulded clay (Roscoe & Burland [2]) and is widely used in geotechnical analysis. The critical-state-based SIMple SAND model (SIMSAND) was devised on the basis of the NLMC by implementing the critical state concept and the cap mechanism (Jin et al.[1, 3]). Elsewhere, the Anisotropic Structured Clay Model (ASCM) was developed using the MCC as its foundation and takes into account the behaviour of intact clays because of its natural structure (Yang et al.[60]). This model can be used to predict the mechanical behaviour of soft structured clay, stiff clay and artificial reinforced clay. The ANIsotropic CREEP model (ANICREEP) for natural soft clays was also based on the MCC, the overstress theory and the different time-dependent behaviours of natural soft clays (Yin et al. [59, 61, 62]). The ANICREEP can be applied to various natural soft clays, stiff clays and artificial soils.

The details for above constitutive models can be found in section 6.4~6.9 of ErosOpt manual which can be downloaded from this link: https://www.researchgate.net/publication/338233794_EROSOPT_User's_Guide_by_GeoInvention_Studio?_sg=started_experiment_milestone

Appendix-III: Optimisation algorithms in ErosOpt

Fig. III-1 shows the GUI window of selecting the optimisation algorithm and assigning settings. Apart from the MBSA-LS, three other algorithms will be described following.

(1) Nelder-Mead Simplex

The simplex algorithm is a nonlinear optimisation algorithm developed by Nelder and Mead [26] for minimising an objective function in a poly-dimensional space, which enables the adoption of a direct search strategy. The method uses the concept of a simplex, which is a polytope of N+1vertices in N dimensions, to find a locally optimal solution to a problem with N variables when the objective function varies monotonically. The NMS can change in five different ways during iteration in two dimensions, as shown in Fig. III-2 and Fig. III-3. Said simplex can affect the best solution using a limited number of calculations. However, most direct search strategies, such as the gradient-based and simplex methods described herein, are only capable of searching for a local minimum. A possible solution to this problem is to start the search from different initial positions; if the local minimum remains the same, this is most likely also the global minimum.

(2) Nelder-Mead GA (NMGA)

The NMGA was developed by Jin et al.[21]; the flow chart is shown in Fig. III-4. In NMGA, the NMS is used to accelerate the convergence speed. Before performing the crossover of the adopted RCGA, all the individuals are sorted based on their fitness values, and the best n+1 (n is the number of variables) individuals are selected to perform the NMS. Based on the results of the NMS, the best individual is updated and then recombined with the N-(n+1) remaining individuals to perform the RCGA crossover. The crossover used in the NMGA is the simulated binary crossover (SBX). To prevent the optimisation from converging on a local solution, a dynamic random mutation (DRM), developed by Chuang et al. [63], is adopted. In the surviving process, the elitism strategy developed by Deb et al. [64] is adopted in the NMGA. The 10% of individuals with the highest fitness level are selected from the parents and children to survive into the next generation. The remainders are chosen by tournament selection from the mating pool composed of parents and children. The completion mechanism can help the NMGA to find better solutions.

(3) Nelder-Mead DE (NMDE)

The NMDE was developed by Yin et al.[18], its flow chart is shown in Fig. III-5. In the NMDE, to accelerate the convergence speed, the NMS is used. Before performing the DE mutation, all individuals are sorted based on their fitness value, and the best n+1 (n is the number of variables) is selected to perform the NMS. Based on the results of the NMS, the best individual is updated and then recombined with the N-(n+1) remaining individuals to perform the DE mutation. This process will be executed N times, resulting in a new population of N individuals. Then, the obtained population is applied to the crossover operation. To avoid a rapid loss of diversity, an elitism strategy is adopted when performing the selection, in which the 10% of individuals with the highest fitness are selected from the parents and children to survive to the next generation. The remainder are chosen by tournament selection from the mating pool composed of parents and children. The completion mechanism can help the NMDE to identify better solutions.

References

[1] Jin Y-F, Yin Z-Y, Shen S-L, Hicher P-Y. Selection of sand models and identification of parameters using an enhanced genetic algorithm. Int J Numer Anal Methods Geomech 2016; 40(8): 1219-40.

[2] Roscoe KH, Burland J. On the generalized stress-strain behaviour of wet clay. Engineering Plasticity. Cambridge, UK: Cambridge University Press, 1968. p. 535-609.

[3] Jin Y-F, Yin Z-Y, Shen S-L, Hicher P-Y. Investigation into MOGA for identifying parameters of a critical-state-based sand model and parameters correlation by factor analysis. Acta Geotech 2016; 11(5): 1131-45.

[4] Jin Y-F, Wu Z-X, Yin Z-Y, Shen JS. Estimation of critical state-related formula in advanced constitutive modeling of granular material. Acta Geotech 2017; 12(6): 1329-51.

[5] Yao Y, Hou W, Zhou A. UH model: three-dimensional unified hardening model for overconsolidated clays. Geotechnique 2009; 59(5): 451-69.

[6] Yao Y, Sun D, Luo T. A critical state model for sands dependent on stress and density. Int J Numer Anal Methods Geomech 2004; 28(4): 323-37.

[7] Yao Y, Sun D, Matsuoka H. A unified constitutive model for both clay and sand with hardening parameter independent on stress path. Computers and Geotechnics 2008; 35(2): 210-22.

[8] Taiebat M, Dafalias YF. SANISAND: Simple anisotropic sand plasticity model. Int J Numer Anal Methods Geomech 2008; 32(8): 915-48.

[9] Chang CS, Hicher PY. An elasto-plastic model for granular materials with microstructural consideration. Int J Solids Struct 2005; 42(14): 4258-77.

[10] Chang CS, Yin ZY. Micromechanical Modeling for Inherent Anisotropy in Granular Materials. Journal of Engineering Mechanics-Asce 2010; 136(7): 830-9.

[11] Yin ZY, Chang CS, Hicher PY. Micromechanical modelling for effect of inherent anisotropy on cyclic behaviour of sand. Int J Solids Struct 2010; 47(14-15): 1933-51.

[12] Yin ZY, Chang CS. Stress–dilatancy behavior for sand under loading and unloading conditions. Int J Numer Anal Methods Geomech 2013; 37(8): 855-70.

[13] Yin Z-Y, Zhao J, Hicher P-Y. A micromechanics-based model for sand-silt mixtures. Int J Solids Struct 2014; 51(6): 1350-63.

[14] Kolymbas D. An outline of hypoplasticity. Archive of applied mechanics 1991; 61(3): 143-51.

[15] Mašín D. A hypoplastic constitutive model for clays. Int J Numer Anal Methods Geomech 2005; 29(4): 311-36.

[16] Wu W, Bauer E, Kolymbas D. Hypoplastic constitutive model with critical state for granular materials. Mech Mater 1996; 23(1): 45-69.

[17] Wu W, Kolymbas D. Hypoplasticity then and now. Constitutive modelling of granular materials 2000: 57-105.

[18] Yin Z-Y, Jin Y-F, Shen JS, Hicher P-Y. Optimization techniques for identifying soil parameters in geotechnical engineering: Comparative study and enhancement. Int J Numer Anal Methods Geomech 2018; 42(1): 70-94.

[19] Yin ZY, Hicher PY. Identifying parameters controlling soil delayed behaviour from laboratory and in situ pressuremeter testing. Int J Numer Anal Methods Geomech 2008; 32(12): 1515-35.

[20] Levasseur S, Malécot Y, Boulon M, Flavigny E. Soil parameter identification using a genetic algorithm. Int J Numer Anal Methods Geomech 2008; 32(2): 189-213.

[21] Jin Y-F, Yin Z-Y, Wu Z-X, Zhou W-H. Identifying parameters of easily crushable sand and application to offshore pile driving. Ocean Eng 2018; 154(416-29.

[22] Jin Y-F, Yin Z-Y, Zhou W-H, Shao J-F. Bayesian model selection for sand with generalization ability evaluation. Int J Numer Anal Methods Geomech 2019; 0(0).

[23] Jin Y-F, Yin Z-Y, Zhou W-H, Huang H-W. Multi-objective optimization-based updating of predictions during excavation. Eng Appl Artif Intell 2019; 78(102-23.

[24] Jin Y-F, Yin Z-Y, Zhou W-H, Horpibulsuk S. Identifying parameters of advanced soil models using an enhanced transitional Markov chain Monte Carlo method. Acta Geotech 2019.

[25] Jin Y-F, Yin Z-Y, Riou Y, Hicher P-Y. Identifying creep and destructuration related soil parameters by optimization methods. KSCE Journal of Civil Engineering 2017; 21(4): 1123-34.

[26] Nelder JA, Mead R. A simplex method for function minimization. The computer journal 1965; 7(4): 308-13.

[27] Bäck T. Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. Oxford, UK: Oxford university press, 1996.

[28] Van Laarhoven PJ, Aarts EH. Simulated annealing: theory and applications. Reidel, Dordrecht, Netherlands: Springer Science & Business Media, 1987.

[29] Kennedy J. Particle swarm optimization. Encyclopedia of Machine Learning: Springer, 2010. p. 760-6.

[30] Rummler A. Evolvica: a Java framework for evolutionary algorithms. 2007.

[31] Ventura S, Romero C, Zafra A, Delgado JA, Hervás C. JCLEC: a Java framework for evolutionary computation. Soft Computing-A Fusion of Foundations, Methodologies and Applications 2008; 12(4): 381-92.

[32] Durillo JJ, Nebro AJ. jMetal: A Java framework for multi-objective optimization. Advances in Engineering Software 2011; 42(10): 760-71.

[33] Sağ T, Çunkaş M. A tool for multiobjective evolutionary algorithms. Advances in Engineering Software 2009; 40(9): 902-12.

[34] Lofberg J. YALMIP: A toolbox for modeling and optimization in MATLAB. Computer Aided Control Systems Design, 2004 IEEE International Symposium on: IEEE, 2004. p. 284-9.

[35] Jin Y-F, Yin Z-Y, Shen S-L, Zhang D-M. A new hybrid real-coded genetic algorithm and its application to parameters identification of soils. Inverse Problems in Science and Engineering 2017; 25(9): 1343-66.

[36] Civicioglu P. Backtracking search optimization algorithm for numerical optimization problems. Applied Mathematics and Computation 2013; 219(15): 8121-44.

[37] Bhattacharjee K, Bhattacharya A, nee Dey SH. Backtracking search optimization based economic environmental power dispatch problems. Int J Electr Power Energy Syst 2015; 73(830-42.

[38] Chen D, Lu R, Zou F, Li S, Wang P. A learning and niching based backtracking search optimisation algorithm and its applications in global optimisation and ANN training. Neurocomputing 2017; 266(579-94.

[39] Chen D, Zou F, Lu R, Wang P. Learning backtracking search optimisation algorithm and its application. Information Sciences 2017; 376(71-94.

[40] Das S, Mandal D, Kar R, Prasad Ghoshal S. A new hybridized backtracking search optimization algorithm with differential evolution for sidelobe suppression of uniformly excited concentric circular antenna arrays. International Journal of RF and Microwave Computer ‐ Aided Engineering 2015; 25(3): 262-8.

[41] Lin J. Oppositional backtracking search optimization algorithm for parameter identification of hyperchaotic systems. Nonlinear Dyn 2015; 80(1-2): 209-19.

[42] Nama S, Saha A, Ghosh S. A new ensemble algorithm of differential evolution and backtracking search optimization algorithm with adaptive control parameter for function optimization. International Journal of Industrial Engineering Computations 2016; 7(2): 323-38.

[43] Nama S, Saha AK, Ghosh S. Improved backtracking search algorithm for pseudo dynamic active earth pressure on retaining wall supporting c-Φ backfill. Applied Soft Computing 2017; 52(885-97.

[44] Su Z, Wang H, Yao P. A hybrid backtracking search optimization algorithm for nonlinear optimal control problems with complex dynamic constraints. Neurocomputing 2016; 186(182-94.

[45] Zhang J, Sanderson AC. JADE: adaptive differential evolution with optional external archive. Evolutionary Computation, IEEE Transactions on 2009; 13(5): 945-58.

[46] Goldberg DE. Real-coded genetic algorithms, virtual alphabets, and blocking. Complex Systems 1991; 5(2): 139-68.

[47] Price K, Storn R. Differential evolution: a simple evolution strategy for fast optimization. Dr Dobb's journal 1997; 22(4): 18-24.

[48] Eberhart RC, Kennedy J. A new optimizer using particle swarm theory. Proceedings of the sixth international symposium on micro machine and human science: New York, NY, 1995. p. 39-43.

[49] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. science 1983; 220(4598): 671-80.

[50] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of global optimization 2007; 39(3): 459-71.

[51] Mirjalili S, Lewis A. The whale optimization algorithm. Advances in Engineering Software 2016; 95(51-67.

[52] Jin Y-F, Yin Z-Y, Wu Z-X, Daouadji A. Numerical modeling of pile penetration in silica sands considering the effect of grain breakage. Finite Elem Anal Des 2018; 144(15-29.

[53] Jin Y-F, Yin Z-Y, Zhou W-H, Shao J-F. Bayesian model selection for sand with generalization ability evaluation. Int J Numer Anal Methods Geomech 2019; 43(14): 2305-27.

[54] Liu Y-J, Li G, Yin Z-Y, Dano C, Hicher P-Y, Xia X-H, et al. Influence of grading on the undrained behavior of granular materials. CR Mec 2014; 342(2): 85-95.

[55] Li G, Liu Y-J, Dano C, Hicher P-Y. Grading-Dependent Behavior of Granular Materials: From Discrete to Continuous Modeling. Journal of engineering mechanics 2014.

[56] Lee Goh A, Fahey M. Application of a 1-dimensional cavity expansion model to pressuremeter and piezocone tests in clay. Proceeding of the Seventh International Conference on Computer Methods and Advances in Geomechanics, Cairns1991. p. 255-60.

[57] Low HE. Performance of penetrometers in deepwater soft soil characterisation: University of Western Australia, 2009.

[58] Yin Z-Y, Zhu Q-Y, Yin J-H, Ni Q. Stress relaxation coefficient and formulation for soft soils. Géotechnique Letters 2014; 4(January-March): 45-51.

[59] Yin Z-Y, Yin J-H, Huang H-W. Rate-Dependent and Long-Term Yield Stress and Strength of Soft Wenzhou Marine Clay: Experiments and Modeling. Marine Georesources & Geotechnology 2015; 33(1): 79-91.

[60] Yang J, Yin Z-Y, Huang H-W, Jin Y-F, Zhang D-M. A bounding surface plasticity model of structured clays using disturbed state concept based hardening variables. Chinese Journal of Geotechnical Engineering 2017; 39(554-61.

[61] Yin ZY, Chang CS, Karstunen M, Hicher PY. An anisotropic elastic-viscoplastic model for soft clays. Int J Solids Struct 2010; 47(5): 665-77.

[62] Yin ZY, Karstunen M, Chang CS, Koskinen M, Lojander M. Modeling Time-Dependent Behavior of Soft Sensitive Clay. Journal of geotechnical and geoenvironmental engineering 2011; 137(11): 1103-13.

[63] Chuang Y-C, Chen C-T, Hwang C. A real-coded genetic algorithm with a direction-based crossover operator. Information Sciences 2015; 305(1): 320-48.

[64] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. Evolutionary Computation, IEEE Transactions on 2002; 6(2): 182-97.

Tables

Table 1 Parameter settings of the compared algorithms

Algorithms	Settings
GA	p_c=0.7, p_M=0.1, mu=0.1, Tournament selection, tour size=2
DE	F=rand(0.4, 0.95), CR=0.9;
PSO	w=1.0, damp ratio of w is 0.99, c_1=1.5; c_2=2.0
SA	T_0=0.1, α=0.99, n_{move}=5, mu=0.5
ABC	Onlooker number=N, L=round(0.6*D*N); a=1;
WOA	No hyper parameters
BSA	mixrate=0.9
NMDE	F=rand(0.4, 0.95), CR=0.9;
NMGA	p_c=0.7, p_M=0.05, mu=0.1, Tournament selection, tour size=2
MBSA-LS	F=rand(0.3, 1.0), mixrate=0.5*(1+rand), CR=0.9

Note: p_c: crossover probability in GA; p_M: mutate probability in GA; mu: mutate rate; n_{move}: number of neighbours per Individual; L: abandonment limit parameter; a: upper bound of acceleration coefficient; w: inertia weight; c_1: personal learning coefficient; c_2: global learning coefficient

Table 2 Identification results of SIMSAND from synthetic data and the bounds used in the calculation

Parameters	Fixed parameters			Optimized parameters								Error
	υ	K_0	n	e_{ref}	λ	ξ	ϕ_μ	k_p	A_d	n_p	n_d	-
Upper bound	-	-	-	1.5	0.1	1.0	50	0.01	5.0	0.1	0.1	-
Lower bound				0.5	0.001	0.01	20	0.0001	0.5	10.0	10.0	-
Pre-set value	0.2	100	0.6	**0.750**	**0.03**	**0.60**	**30.0**	**0.003**	**1.0**	**2.0**	**2.0**	**-**
GA				0.786	0.0739	0.351	30.25	0.00406	0.766	2.379	3.927	5.61
DE				0.749	0.0280	0.624	30.29	0.00379	0.855	1.708	2.829	1.68
PSO				0.759	0.0390	0.533	30.02	0.00323	0.972	2.146	2.175	0.81
SA				0.782	0.0625	0.418	30.03	0.00348	0.944	2.324	2.268	2.42
ABC				0.778	0.0580	0.460	29.80	0.00441	1.000	3.400	2.700	3.77
WOA				0.817	0.0989	0.363	30.99	0.01069	1.942	4.921	7.585	15.7
BSA				0.713	0.0019	1.000	30.10	0.00215	0.673	0.665	1.809	7.38
NMDE				0.754	0.0034	0.570	30.00	0.0031	1.000	2.100	2.100	0.45
NMGA				0.738	0.0019	0.720	29.80	0.0017	1.400	1.400	1.000	2.30
MBSA-LS				***0.750***	***0.0300***	***0.600***	***30.00***	***0.00300***	***1.000***	***2.000***	***2.000***	***0.00***

Table 3 Comparison of time cost for different optimisation algorithms (popsize=48)

Calculation time / s	Algorithms									
	GA	DE	PSO	SA	ABC	WOA	BSA	NMDE	NMGA	**MBSA-LS**
	2215.9	809.8	2187.6	10583.3	6495.7	799.6	761.9	2898.7	2465.9	1537.4

Table 4 Search domain for different parameters of SIMSAND

Soil model	SIMSAND							
Parameters	e_{ref}	λ	ξ	ϕ_{μ}	k_{p}	A_{d}	n_{p}	n_{d}
Lower bound	0.1	0.001	0.01	20	10^{-4}	0.1	0.1	0.1
Upper bound	1	0.1	1.0	50	0.1	5	10	10

Table 5 Parameters of SIMSAND for Hostun sand

Parameters	Fixed parameters			Optimized parameters							
	K_0	υ	n	e_{ref}	λ	ξ	ϕ_{μ}	k_{p}	A_{d}	n_{p}	n_{d}
GA	45	0.2	0.6	0.817	0.0980	0.610	29.56	0.0115	0.637	5.31	7.79
DE	45	0.2	0.6	0.785	0.0385	0.912	28.54	0.0063	0.929	1.22	7.07
PSO	45	0.2	0.6	0.795	0.0549	0.588	28.07	0.0030	2.010	3.56	0.31
SA	45	0.2	0.6	0.817	0.0820	0.422	28.22	0.0031	1.899	3.56	0.50
ABC	45	0.2	0.6	0.790	0.0370	0.700	29.30	0.0093	0.600	1.20	10.0
WOA	45	0.2	0.6	0.811	0.0851	0.010	29.13	0.0097	0.580	0.64	7.97
BSA	45	0.2	0.6	0.787	0.0430	0.690	29.21	0.0066	1.020	5.84	2.58
NMGA	**45**	**0.2**	**0.6**	**0.829**	**0.0970**	**0.360**	**28.10**	**0.0026**	**2.200**	**3.60**	**0.0**
NMDE	**45**	**0.2**	**0.6**	**0.824**	**0.0900**	**0.390**	**28.10**	**0.0027**	**2.200**	**3.50**	**0.0**
MBSA-LS	**45**	**0.2**	**0.6**	**0.828**	**0.0941**	**0.390**	**28.04**	**0.0026**	**2.260**	**3.42**	**0.0**

Table 6 Search domain for different parameters of ANICREEP

Parameters	M	σ'_{p0} / kPa	$C_{\alpha\mathrm{e}}$
Lower bound	0.5	10	0.001
Upper bound	2.0	200	0.05

Table 7 Values of ANICREEP model for Burswood clay

Parameters	Fixed parameters				Optimized parameters		
	υ	e_0	κ	λ	M	σ'_{p0} / kPa	$C_{\alpha e}$
GA	0.3	2.8	0.036	0.40	1.631	65.4	0.0113
DE	0.3	2.8	0.036	0.40	1.628	63.0	0.0141
PSO	0.3	2.8	0.036	0.40	1.629	57.7	0.0197
SA	0.3	2.8	0.036	0.40	1.653	53.3	0.0255
ABC	0.3	2.8	0.036	0.40	1.620	60.0	0.0177
WOA	0.3	2.8	0.036	0.40	0.922	114.0	0.0042
BSA	0.3	2.8	0.036	0.40	1.595	60.3	0.0169
NMGA	**0.3**	**2.8**	**0.036**	**0.40**	**1.700**	**57.5**	**0.0200**
NMDE	**0.3**	**2.8**	**0.036**	**0.40**	**1.710**	**57.0**	**0.0210**
MBSA-LS	**0.3**	**2.8**	**0.036**	**0.40**	**1.703**	**57.6**	**0.0199**

Figure captions

Figure 1

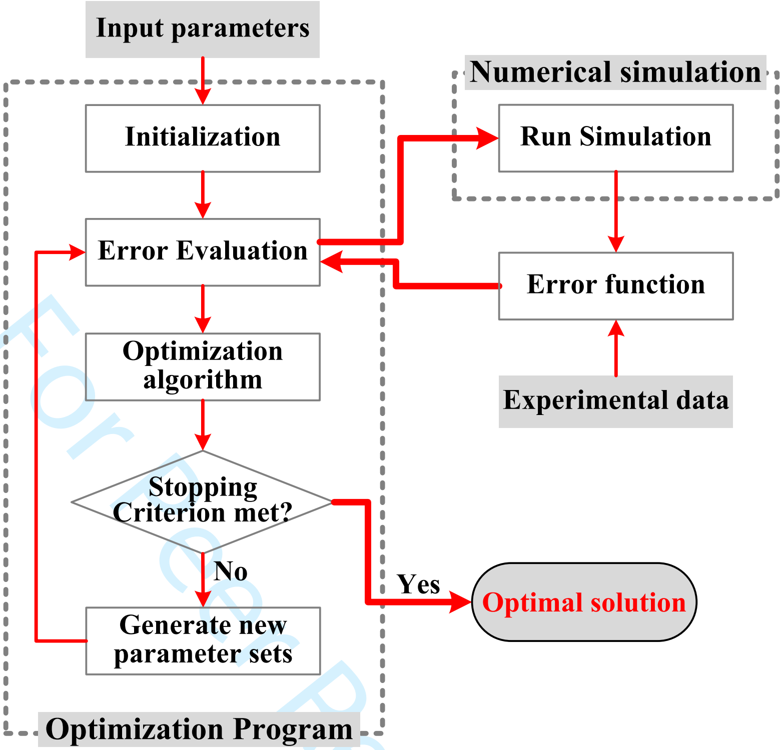

Input parameters
Numerical simulation
Initialization
Run Simulation
Error Evaluation
Error function
Optimization algorithm
Experimental data
Stopping Criterion met?
No
Yes
Optimal solution
Generate new parameter sets
Optimization Program

Figure 2

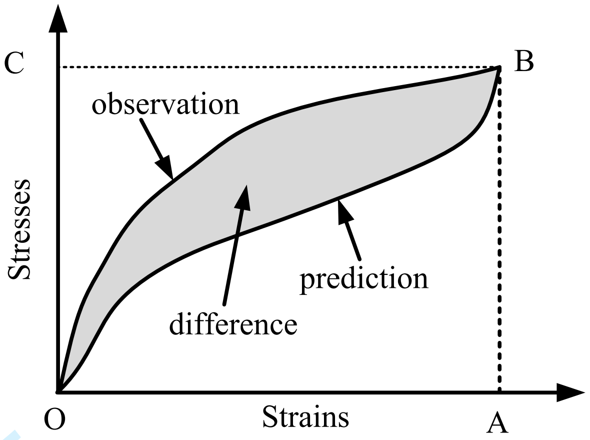

C
B
observation
Stresses
prediction
difference
O
Strains
A

Figure 3

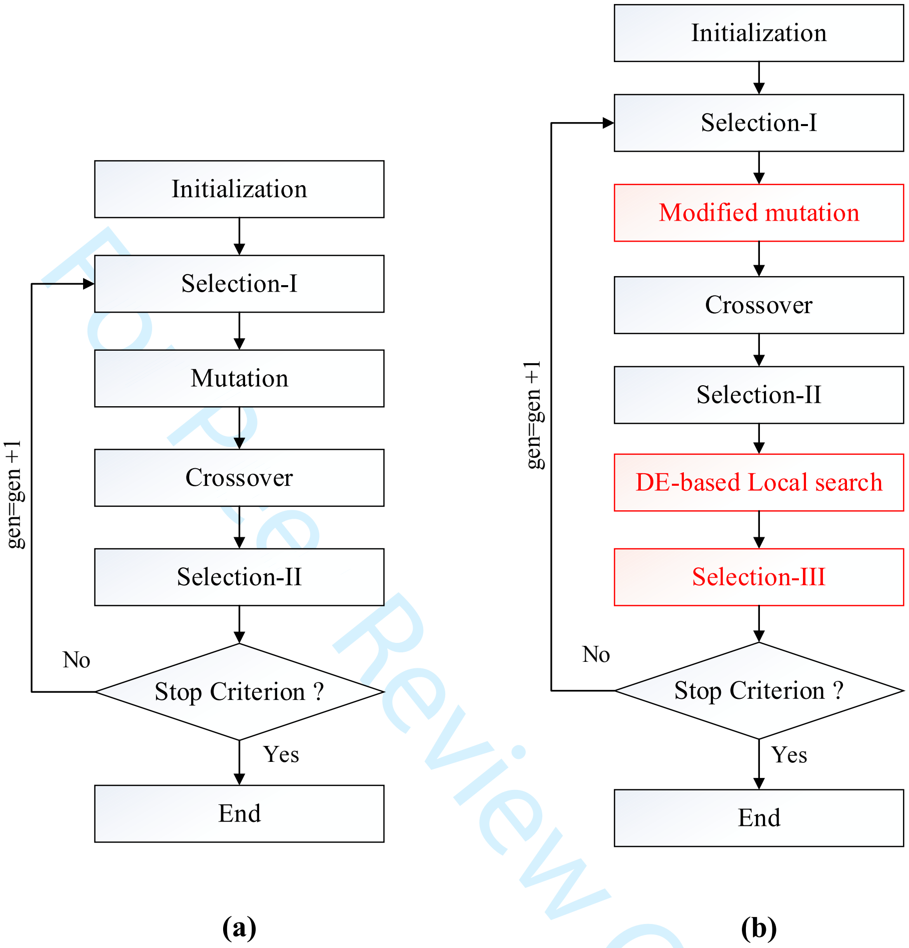

Initialization
Selection-I
Mutation
Crossover
Selection-II
Stop Criterion ?
No
Yes
gen=gen +1
End
(a)
Initialization
Selection-I
Modified mutation
Crossover
Selection-II
DE-based Local search
Selection-III
Stop Criterion ?
No
Yes
gen=gen +1
End
(b)

Figure 4

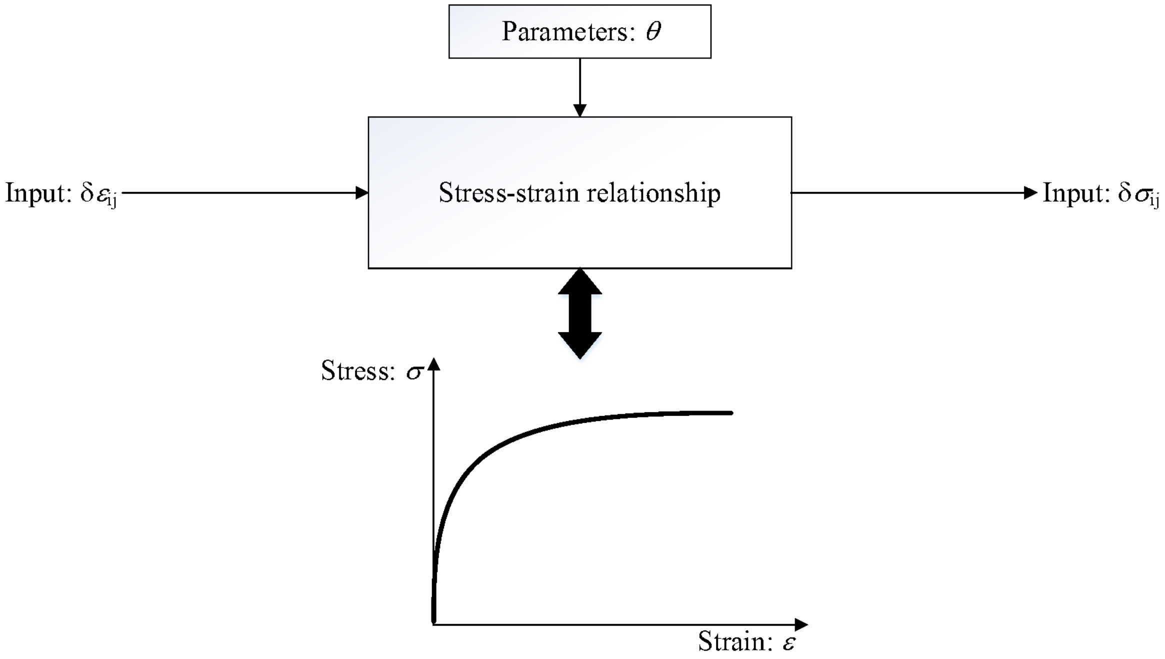

Parameters: θ
Input: $\delta\varepsilon_{ij}$
Stress-strain relationship
Input: $\delta\sigma_{ij}$
Stress: σ
Strain: ε

Figure 5

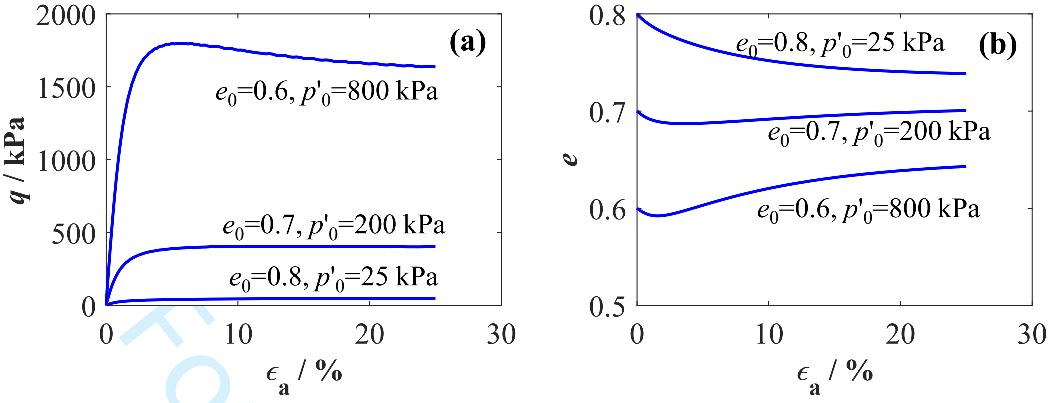

2000
1500
1000
500
0
q / kPa
0
10
20
30
ϵ_a / %
(a)
e_0=0.6, p'_0=800 kPa
e_0=0.7, p'_0=200 kPa
e_0=0.8, p'_0=25 kPa
0.8
0.7
0.6
0.5
e
0
10
20
30
ϵ_a / %
(b)
e_0=0.8, p'_0=25 kPa
e_0=0.7, p'_0=200 kPa
e_0=0.6, p'_0=800 kPa

Figure 6

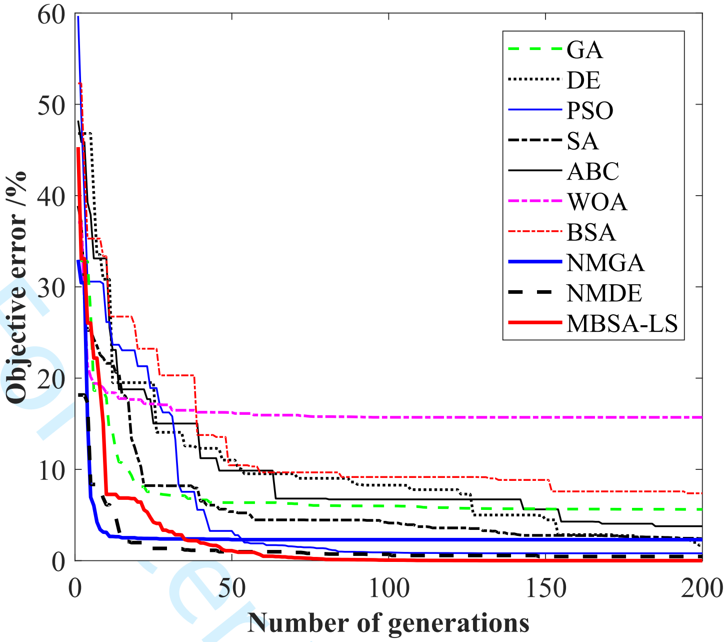

GA
DE
PSO
SA
ABC
WOA
BSA
NMGA
NMDE
MBSA-LS
Objective error /%
0
10
20
30
40
50
60
Number of generations
0
50
100
150
200

Figure 7

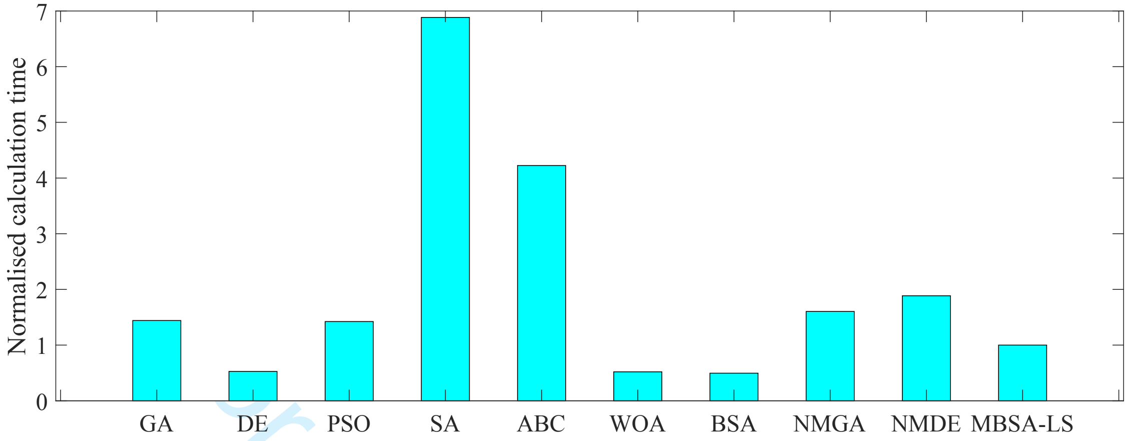

Normalised calculation time
0
1
2
3
4
5
6
7
GA
DE
PSO
SA
ABC
WOA
BSA
NMGA
NMDE
MBSA-LS

Figure 8

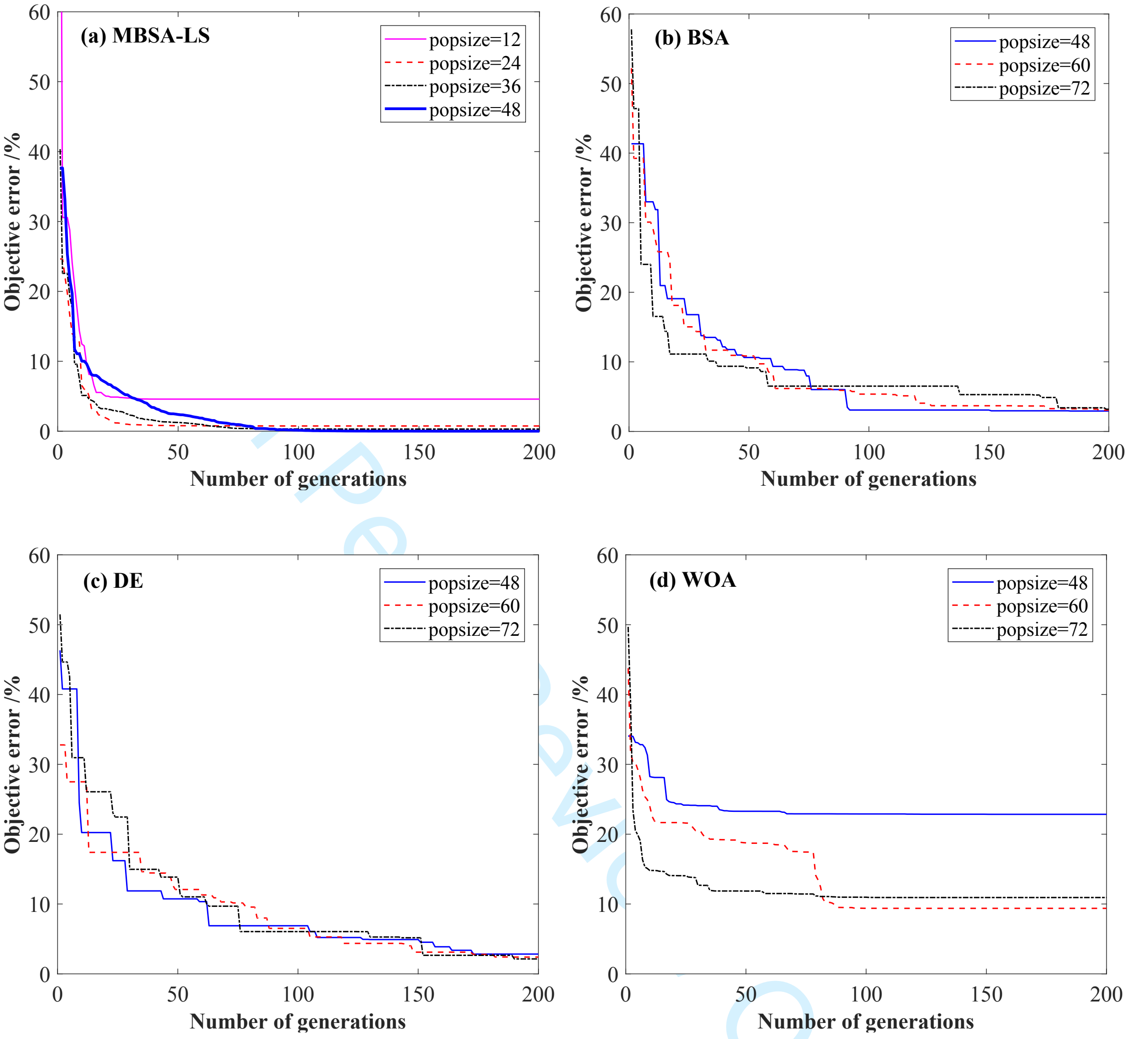

(a) MBSA-LS
popsize=12
popsize=24
popsize=36
popsize=48
(b) BSA
popsize=48
popsize=60
popsize=72
(c) DE
popsize=48
popsize=60
popsize=72
(d) WOA
popsize=48
popsize=60
popsize=72
Objective error /%
0
10
20
30
40
50
60
Number of generations
0
50
100
150
200

Figure 9

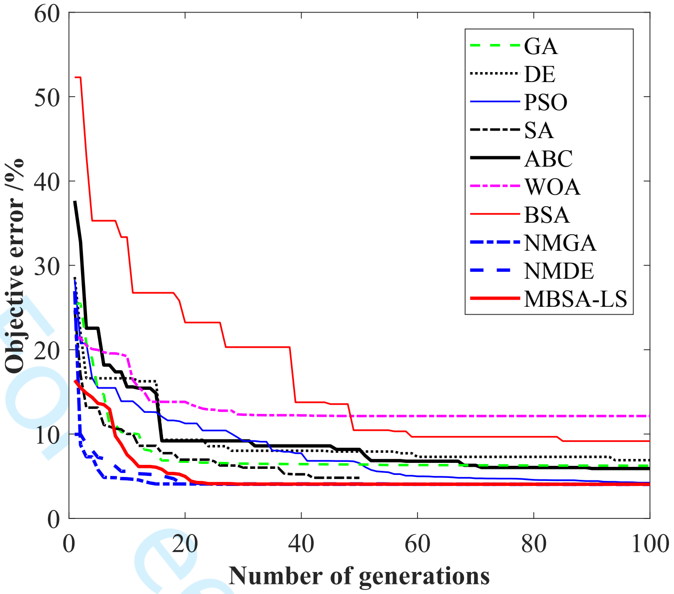

Objective error /%
60
50
40
30
20
10
0
0
20
40
60
80
100
Number of generations
GA
DE
PSO
SA
ABC
WOA
BSA
NMGA
NMDE
MBSA-LS

Figure 10

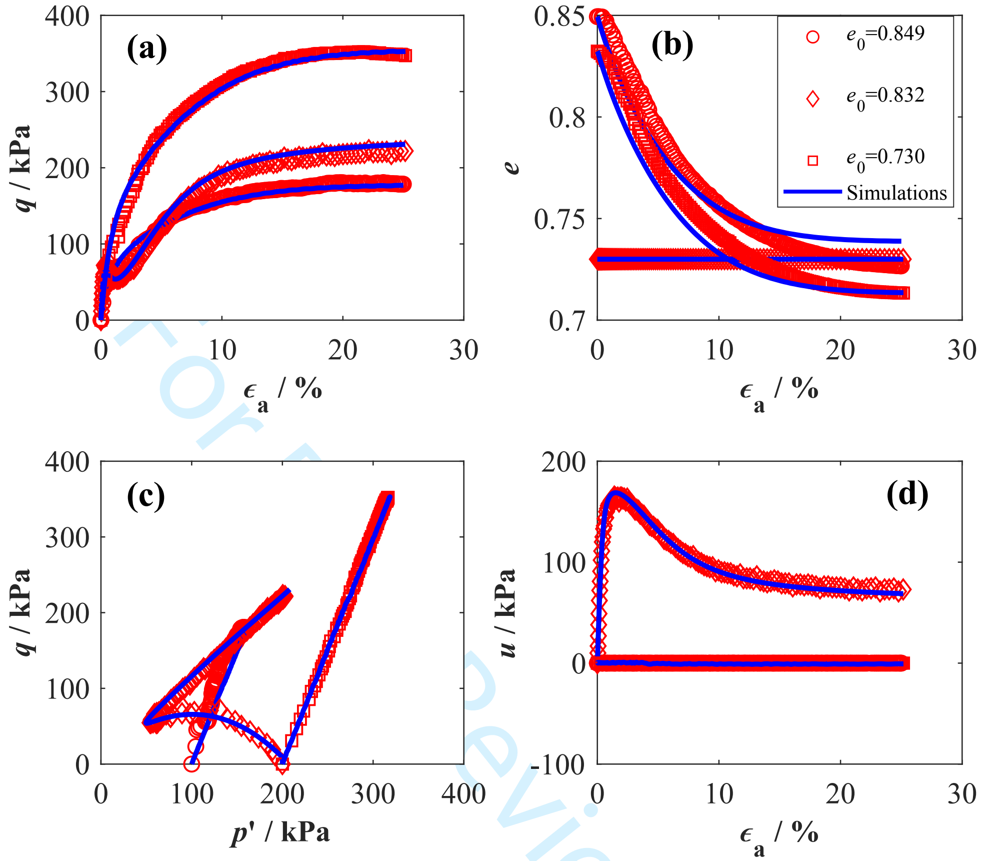

(a)
q / kPa
ϵa / %
0
100
200
300
400
0
10
20
30
(b)
e
0.7
0.75
0.8
0.85
○ e0=0.849
◇ e0=0.832
□ e0=0.730
Simulations
(c)
p' / kPa
(d)
u / kPa
-100
0
100
200

Figure 11

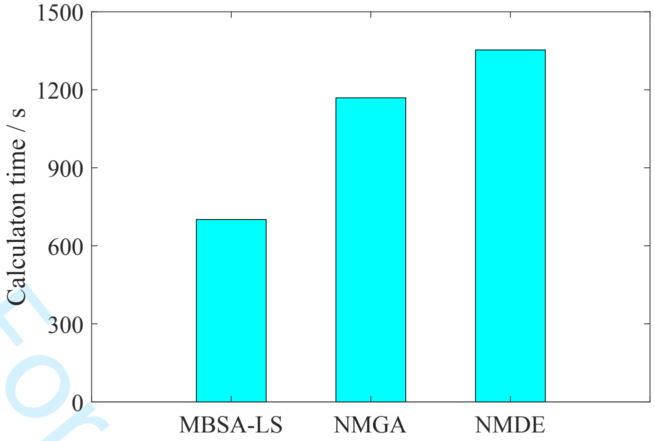


Figure 12

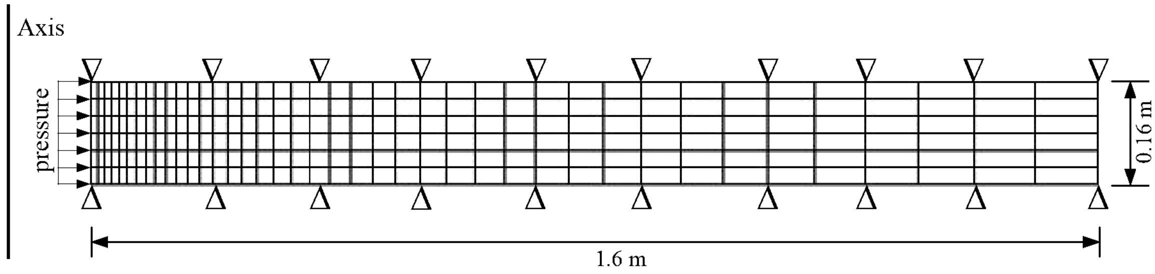


Figure 13

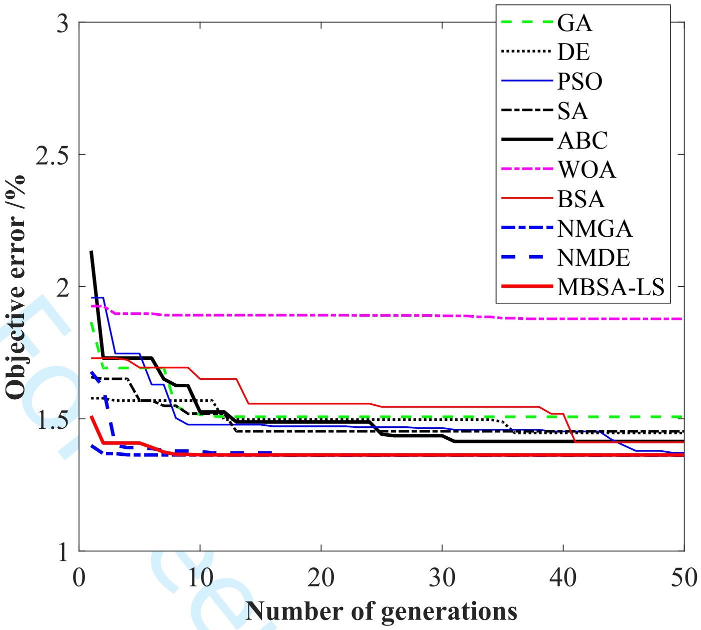

GA
DE
PSO
SA
ABC
WOA
BSA
NMGA
NMDE
MBSA-LS
Objective error /%
3
2.5
2
1.5
1
0
10
20
30
40
50
Number of generations

Figure 14

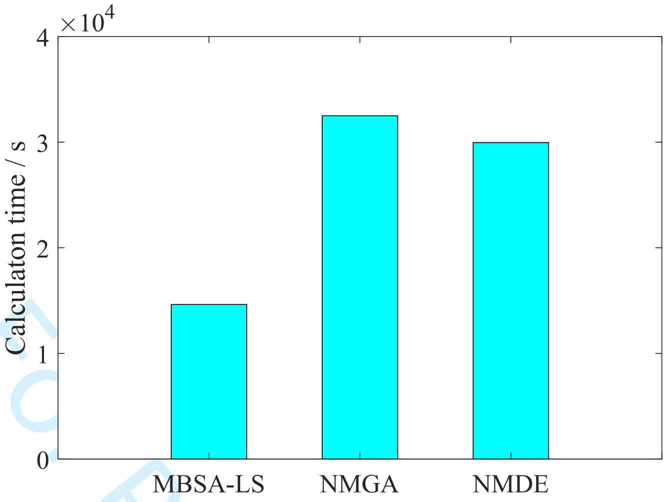


Figure 15

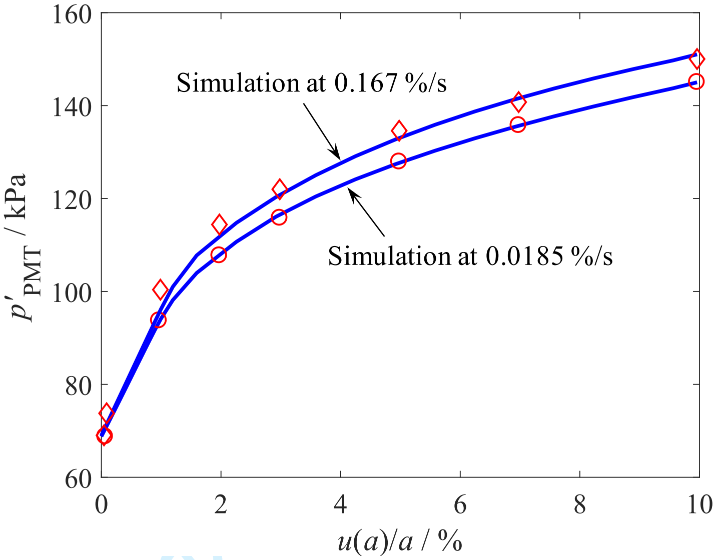

160
140
120
100
80
60
0
2
4
6
8
10
p'_{PMT} / kPa
$u(a)/a$ / %
Simulation at 0.167 %/s
Simulation at 0.0185 %/s

Figure 16

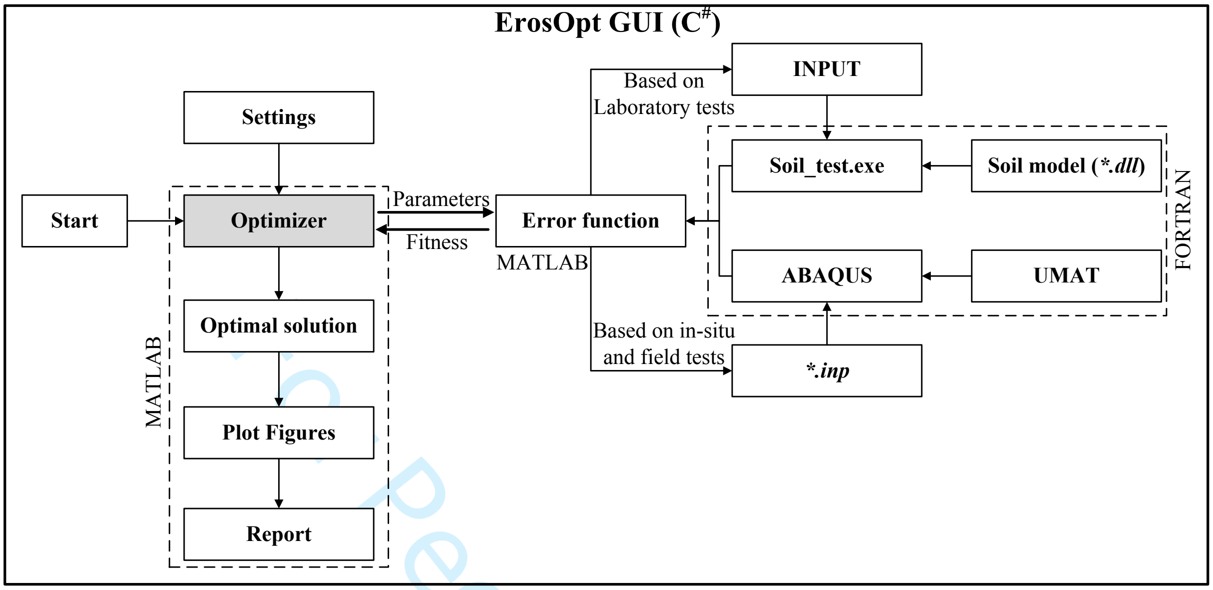

ErosOpt GUI (C#)
Start
Settings
Optimizer
Optimal solution
Plot Figures
Report
MATLAB
Parameters
Fitness
Error function
MATLAB
Based on
Laboratory tests
INPUT
Soil_test.exe
Soil model (*.dll)
ABAQUS
UMAT
FORTRAN
Based on in-situ
and field tests
*.inp

Figure 17

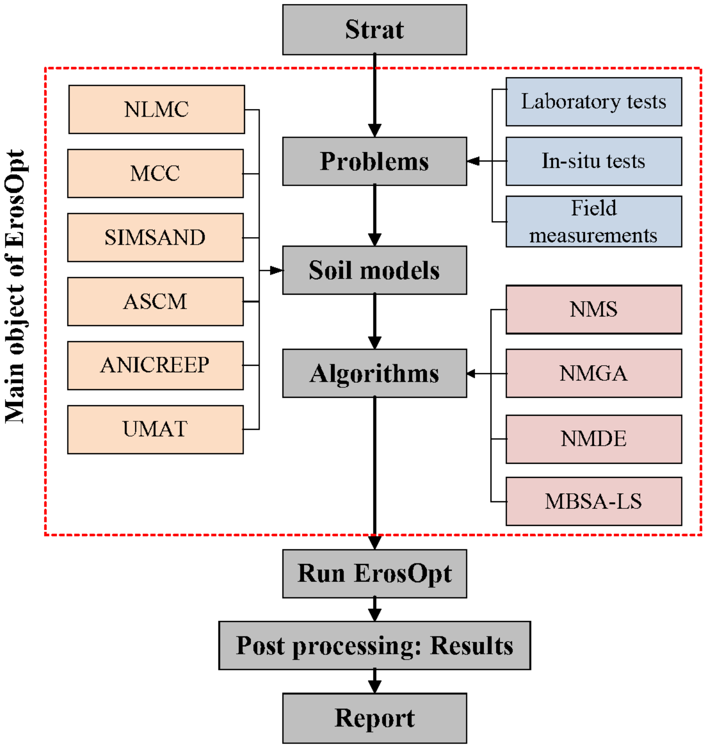
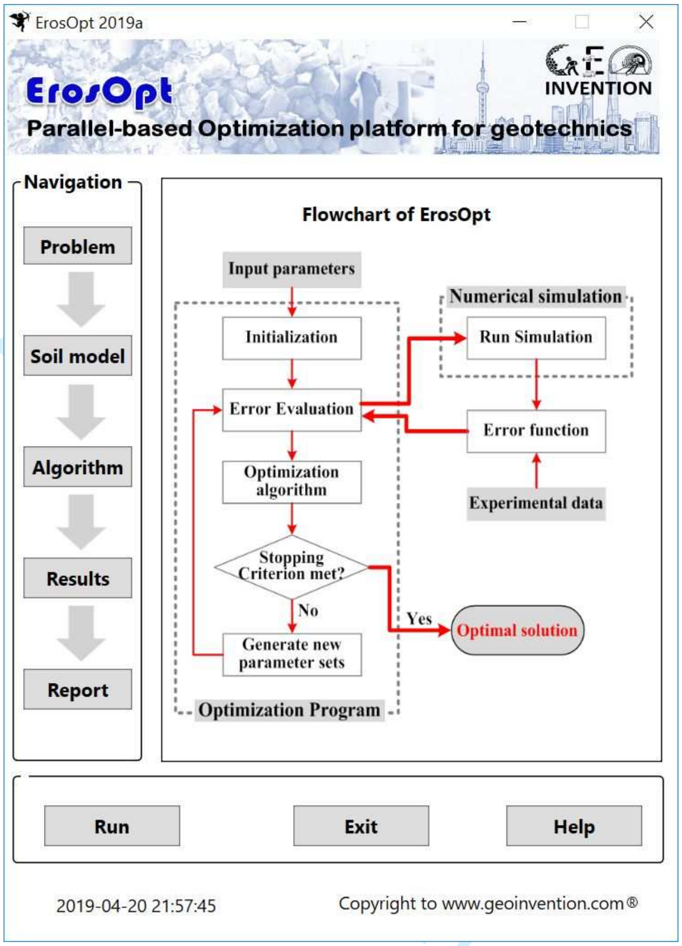

Strat
Main object of ErosOpt
NLMC
MCC
SIMSAND
ASCM
ANICREEP
UMAT
Problems
Soil models
Algorithms
Laboratory tests
In-situ tests
Field measurements
NMS
NMGA
NMDE
MBSA-LS
Run ErosOpt
Post processing: Results
Report

Figure 18



2019-04-20 21:57:45

Copyright to www.geoinvention.com®

Figure 19

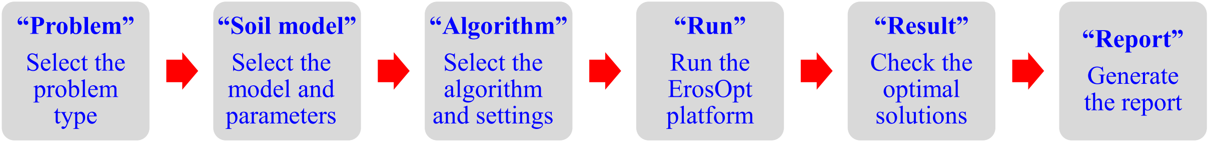
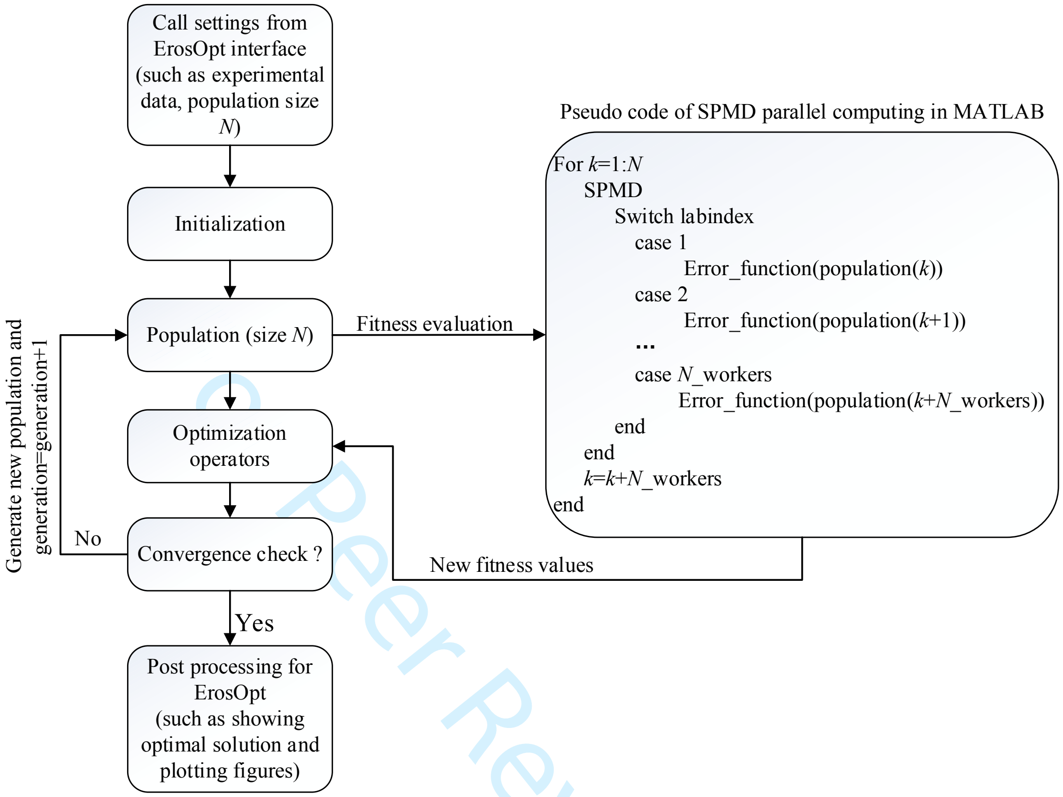

"Problem"
Select the problem type
"Soil model"
Select the model and parameters
"Algorithm"
Select the algorithm and settings
"Run"
Run the ErosOpt platform
"Result"
Check the optimal solutions
"Report"
Generate the report

Figure I-1


Call settings from ErosOpt interface (such as experimental data, population size N)
Initialization
Population (size N)
Optimization operators
Convergence check ?
Post processing for ErosOpt (such as showing optimal solution and plotting figures)
Fitness evaluation
New fitness values
No
Yes
Generate new population and generation=generation+1
Pseudo code of SPMD parallel computing in MATLAB

```
For k=1:N
    SPMD
        Switch labindex
           case 1
                  Error_function(population(k))
           case 2
                  Error_function(population(k+1))
           ...
           case N_workers
                 Error_function(population(k+N_workers))
        end
    end
    k=k+N_workers
end
```

Figure II-1

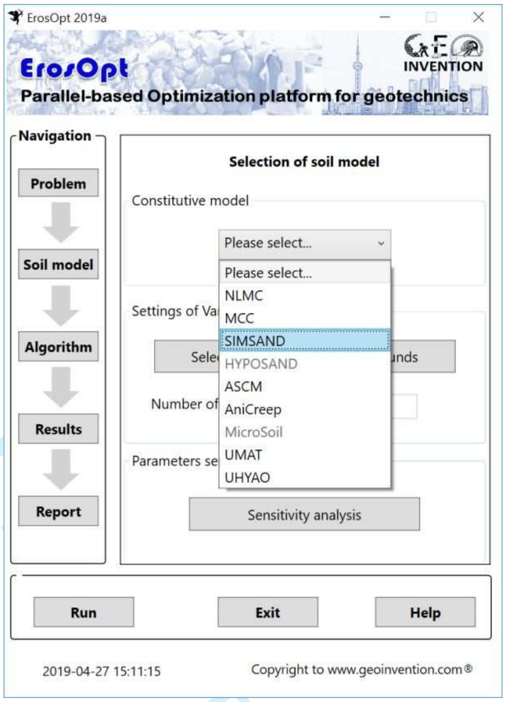
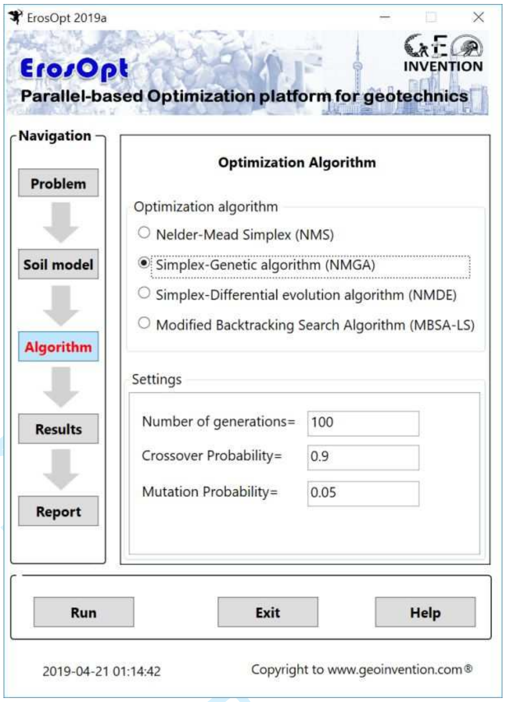

ErosOpt 2019a
ErosOpt
GEO INVENTION
Parallel-based Optimization platform for geotechnics
Navigation
Problem
Soil model
Algorithm
Results
Report
Selection of soil model
Constitutive model
Please select...
Please select...
NLMC
MCC
SIMSAND
HYPOSAND
ASCM
AniCreep
MicroSoil
UMAT
UHYAO
Settings of Va
Sele
nds
Number of
Parameters se
Sensitivity analysis
Run
Exit
Help
2019-04-27 15:11:15
Copyright to www.geoinvention.com®

Figure III-1



Optimization Algorithm

Optimization algorithm

- ○ Nelder-Mead Simplex (NMS)
- ● Simplex-Genetic algorithm (NMGA)
- ○ Simplex-Differential evolution algorithm (NMDE)
- ○ Modified Backtracking Search Algorithm (MBSA-LS)

Settings

Number of generations= 100

Crossover Probability= 0.9

Mutation Probability= 0.05

2019-04-21 01:14:42

Copyright to www.geoinvention.com®

Figure III-2

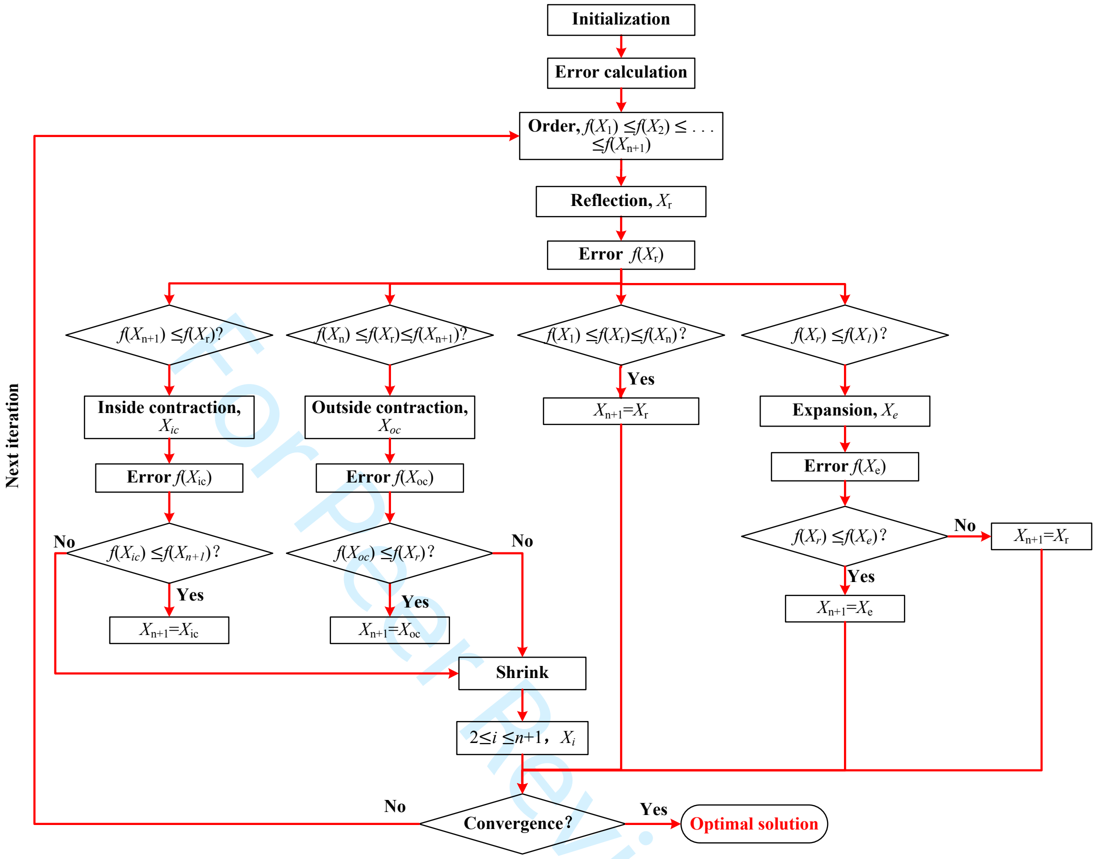

Initialization
Error calculation
Order, $f(X_1) \leq f(X_2) \leq . . . \leq f(X_{n+1})$
Reflection, X_r
Error $f(X_r)$
$f(X_{n+1}) \leq f(X_r)$?
$f(X_n) \leq f(X_r) \leq f(X_{n+1})$?
$f(X_1) \leq f(X_r) \leq f(X_n)$?
$f(X_r) \leq f(X_1)$?
Yes
Inside contraction, X_{ic}
Outside contraction, X_{oc}
$X_{n+1}=X_r$
Expansion, X_e
Error $f(X_{ic})$
Error $f(X_{oc})$
Error $f(X_e)$
No
$f(X_{ic}) \leq f(X_{n+1})$?
$f(X_{oc}) \leq f(X_r)$?
No
$f(X_r) \leq f(X_e)$?
No
$X_{n+1}=X_r$
Yes
Yes
Yes
$X_{n+1}=X_{ic}$
$X_{n+1}=X_{oc}$
$X_{n+1}=X_e$
Shrink
$2 \leq i \leq n+1$, X_i
No
Convergence?
Yes
Optimal solution
Next iteration

Figure III-3

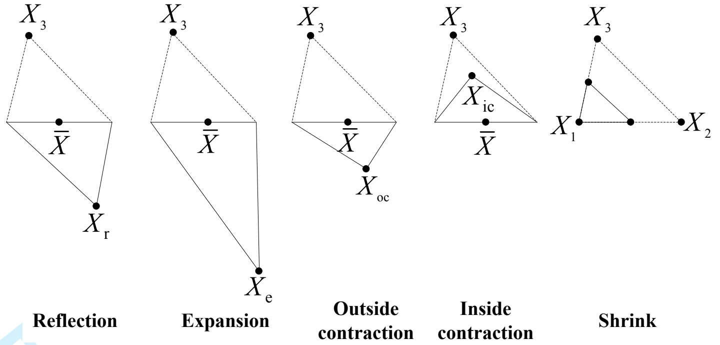

X_3
$\bar{X}$
X_r
X_3
$\bar{X}$
X_e
X_3
$\bar{X}$
X_{oc}
X_3
X_{ic}
$\bar{X}$
X_3
X_1
X_2
Reflection
Expansion
Outside contraction
Inside contraction
Shrink

Figure III-4

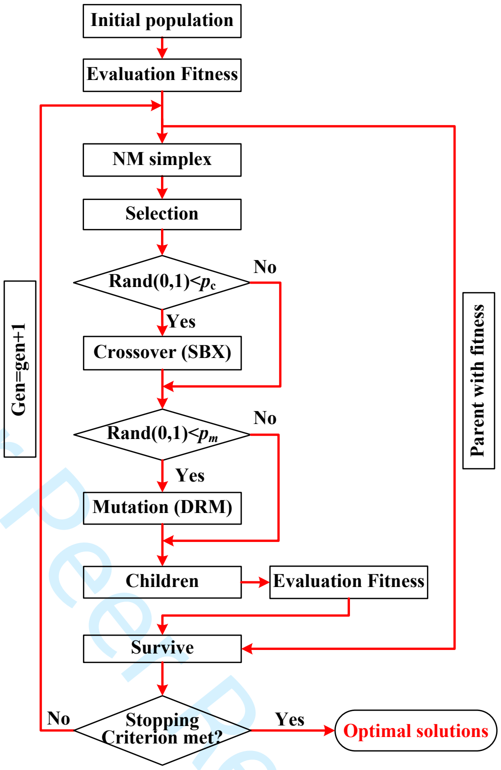
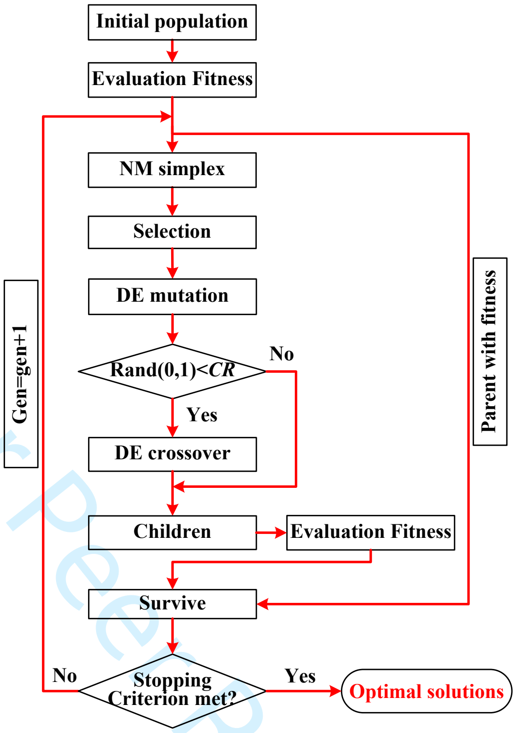

Initial population
Evaluation Fitness
NM simplex
Selection
Rand(0,1)<p_c
No
Yes
Crossover (SBX)
Rand(0,1)<p_m
No
Yes
Mutation (DRM)
Children
Evaluation Fitness
Survive
Stopping Criterion met?
No
Yes
Optimal solutions
Gen=gen+1
Parent with fitness

Figure III-5


Initial population
Evaluation Fitness
NM simplex
Selection
DE mutation
Rand(0,1)<CR
No
Yes
DE crossover
Children
Evaluation Fitness
Survive
Stopping Criterion met?
No
Yes
Optimal solutions
Gen=gen+1
Parent with fitness