Soft Computing (2018) 22:7773–7785
https://doi.org/10.1007/s00500-018-3448-7

FOCUS

A self-feedback strategy differential evolution with fitness landscape analysis

Ying Huang[1,2] · Wei Li[3] · Chengtian Ouyang[3] · Yan Chen[4]

Published online: 16 August 2018

Abstract

Differential evolution (DE) has been widely applied to complex global optimization problems. Different search strategies have been designed to find the optimum conditions in a fitness landscape. However, none of these strategies works well over all possible fitness landscapes. Since the fitness landscape associated with a complex global optimization problem usually consists of various local landscapes, each search strategy is efficient in a particular type of fitness landscape. A reasonable approach is to combine several search strategies and integrate their advantages to solve global optimization problems. This paper presents a new self-feedback strategy differential evolution (SFSDE) algorithm based on fitness landscape analysis of single-objective optimization problem. In the SFSDE algorithm, in the analysis of the fitness landscape features of fitness-distance correlation, a self-feedback operation is used to iteratively select and evaluate the mutation operators of the new SFSDE algorithm. Moreover, mixed strategies and self-feedback transfer are combined to design a more efficient DE algorithm and enhance the search range, convergence rate and solution accuracy. Finally, the proposed SFSDE algorithm is implemented to optimize soil water textures, and the experimental results show that the proposed SFSDE algorithm reduces the difficulty in estimating parameters, simplifies the solution process and provides a novel approach to calculate the parameters of the Van Genuchten equation. In addition, the proposed algorithm exhibits high accuracy and rapid convergence and can be widely used in the parameter estimation of such nonlinear optimization models.

Keywords Differential evolution · Self-feedback · Fitness landscape

1 Introduction

The fitness landscape is a new theory proposed by the theoretical biologist Sewall Wright that regards evolution as a process of movement or adaptive migration in a three-dimensional landscape with basins and valleys (Wright 1932). The concept of the fitness landscape was initially used for biological evolution optimization dynamics, which has a very important role in the analysis and understanding of evolutionary algorithms. The fitness landscape predicts the performance of the algorithm, and the study of the fitness landscape will aid in the design of improved evolutionary algorithms. When the evolutionary algorithm is used to solve complex optimization problems, the corresponding landscape of fitness is usually complex. For instance, a complex three-dimensional fitness landscape may include ridges, valleys and basins. Each point in the landscape represents a possible gene combination, and the height of each point denotes the fitness. Peaks are points at which the gene combination has a higher fitness, whereas valleys are points at

Communicated by B. B. Gupta.

✉ Wei Li
nhwslw@gmail.com

Ying Huang
nhwshy@whu.edu.cn

Chengtian Ouyang
oyct@163.com

Yan Chen
cheny@scau.edu.cn

[1] Key Laboratory of Jiangxi Province for Numerical Simulation and Emulation Techniques, Institute of Mathematical and Computer Sciences, Gannan Normal University, Ganzhou, China

[2] Department of Electrical and Computer Engineering, University of Calgary, Calgary, Canada

[3] School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, China

[4] College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China

which the gene combination has a lower fitness. Due to differences in the fitness of different gene combinations, these features present a rugged landscape of mountains and valleys, which we call a fitness landscape. The fitness landscape is a commonly used metaphor for expressing the features of evolutionary algorithms in solving complex problems. The fitness landscape originates in evolutionary biology, and scholars have begun to study the fitness landscape in the field of evolutionary computation. We discuss the concept of a fitness landscape and analyze and extract the features of such a landscape, which will help scholars design and optimize the performance of evolutionary algorithms.

Different search strategies have been employed to improve the performance of evolutionary algorithms (Eskandarpour et al. 2014). For example, Gaussian, Cauchy and Levy mutation operations are used in evolutionary programming (Yao et al. 1999; Lee and Yao 2004). The Gaussian mutation operator is effective for certain uni-modal and multi-modal functions with only a few local optimal points. The Cauchy mutation operator works well only on multi-modal functions with many local optimal points (Yao et al. 1999). However, according to the no free lunches theorem (Wolpert and Macready 1997, 2005), none of these mutation operators is efficient in solving all optimization problems; each is efficient only for specific fitness landscapes. From the perspective of fitness landscapes, the efficiency of a search strategy is closely associated with the features of the fitness landscape. Following the terms of game theory, this mixed search strategy is the probability that each individual or population chooses a strategy from a strategy pool (Li et al. 2018). Therefore, a reasonable approach is to combine multiple search strategies together and dynamically select these strategies to search complex fitness landscapes (Li et al. 2017a, b).

Many complicated phenomena such as weather changes, stock market fluctuations and soil water-saving irrigation may seem stochastic. However, these unpredictable and unrelated phenomena in fact follow a certain order (Cao et al. 2018). We examine these phenomena to find similar concepts with unique properties: randomness, which is often presented as the performance of random variables; ergodcity, which is the nonrepetition of all states over a range; regularity, which is produced by deterministic iteration (Yang et al. 2018). The system of these phenomena possesses the property of self-feedback. Through the iteration of initial values, the distribution of ergodicity and randomness is formed (Zhou et al. 2018). Self-feedback search is carried out in the iterative process. The concept of self-feedback is introduced into the evolutionary algorithm to balance the global exploration and local exploitation ability.

At present, there is little research that directly links the features of self-feedback evolutionary algorithms and fitness landscapes. Merz examined the combination of fitness landscape theory and the memetic algorithm (Merz and Freisleben 1999, 2000; Merz 2004). Shen initially designed a mixed-strategy evolutionary programming using the fitness landscape (Shen and He 2010). Relevant experimental results show that the use of fitness landscape analysis helps design highly efficient evolutionary algorithms to solve combinatorial optimization problems (Rohlfshagen and Yao 2013). Therefore, combining the features of fitness landscapes is a feasible approach to designing effective evolutionary algorithms.

Since the first appearance of precision agriculture in the early 1980s (McBratney et al. 2005), its development has become a research hot spot. Precision agriculture is defined in an assessment report formed by the experts of the National Research Council of the USA in 1997 as a mechanical strategy to obtain values from a variety of sources through information technology (Zhang and Kovacs 2012). The key goal of precision agriculture technology is to determine the actual existence of spatial and temporal differences in the crop yield and crop growth in the field of environmental factors, for example, soil water textures, geology, fertilizer nutrition, crop diseases. The factors affecting the differences in crop yields are identified, and the technical feasibility of economic control strategies is evaluated to implement the most effective controls. The ultimate goal of precision agriculture is to help farmers conserve environmental quality while reducing waste, improving efficiency and increasing returns. Precise agriculture requires precise control, and many precision agriculture problems can be characterized as optimization problems (Bongiovanni and Lowenberg-DeBoer 2004).

Precision agriculture can be applied to soil textures, one of the most important factors influencing crop growth (Vachaud et al. 1985). An accurate and timely analysis of the status of soil nutrients will be of great value to guiding water-saving agricultural production. Soil nutrients are also a major technical issue in the sustainable development of agriculture. The soil water characteristic curve is an important indicator of the basic hydraulic properties of soils and plays a very important role in studying the retention and transport of soil water (Van Genuchten 1980). Scholars have proposed mathematical models and computing methods for these problems. The Van Genuchten equation has a wide range of applications because the linear curve is very similar to the measured data (Wang et al. 2000). However, there are many parameters in the Van Genuchten equation, and conventional linear fitting methods are difficult to realize. The parameter fitting of this equation is a nonlinear fitting problem.

In recent years, evolutionary algorithms have become a hot spot for solving nonlinear problems. These algorithms have the characteristics of initial independence, delineating feasibility regions and global optimization (Gupta et al. 2016). These optimization problems are increasingly characterized by high-dimensional, discontinuous, multi-peak, big-data and other complex features (Hossain et al. 2018). Recent studies show that traditional algorithms are not suitable for solving these optimization problems. However, differential evolution (DE) (Storn and Price 1997) algorithms possess characteristics such as strong search space, fast convergence speed and exact solving ability, which is why these algorithms have become the research direction of many researchers (Wang et al. 2018a, b). In recent years, research on DE algorithms has focused on the aspects of mutation, parameter adjustment and selection strategy. In particular, research on population topology, population diversity and the representation of individuals in probabilistic space has been developed (Li et al. 2017a, b; Al-Smadi et al. 2017). This paper continues the line of inquiry into evolutionary computation. Based on the fitness landscape, self-feedback operation and mixed strategies described above, we design a new DE algorithm to solve the optimization problem in practical applications.

The next section describes related work on the subject of fitness landscapes, specifically completely summarized methods employing fitness landscape feature analysis. In Sects. 3, 4 and 5, we define self-feedback operations, the Van Genuchten model, their associated functions, as well as self-feedback search and transfer operations based on the mixed-strategies DE algorithm. We then propose our approach, the self-feedback mixed-strategies differential evolution (SFSDE) algorithm, in Sect. 6. Finally, we apply the proposed new DE algorithm to solve six types of Van Genuchten equation parameters. We close the paper with our conclusions and suggestions for future research directions.

2 Fitness landscape features analysis

Analyzing the features of fitness landscapes plays an important role in optimizing and designing effective search algorithms (Li et al. 2017a, b). However, in general, it is difficult to accurately represent a fitness landscape in polynomial time. Instead, a prior observation of the fitness landscape is often obtained. In evolutionary computation, a population contains finite sample points in the fitness landscape, and each population can be used as an observation of the fitness landscape. For the statistical sampling of a local fitness landscape, we study its features; for the fitness landscape in three-dimensional space, we intuitively use ridges, valleys and basins to describe the features of the local fitness landscape. However, for a fitness landscape in high-dimensional space, we cannot employ the same visualization. To approximate the real fitness landscape, new features and analysis methods must be developed.

A fitness landscape Λ_S can be denoted by the following (Stadler and Stephens 2003; Kallel et al. 2001):

$$\Lambda_S = (X, n, f) \tag{1}$$

where X is the solution space; $n(x)$ denotes a neighborhood structure, used to assign to all neighbors $x \in X$ (far or near); and $f(x) : X \to \mathbb{R}$ is a fitness function. In other words, the solution space is a linked network of a neighborhood structure, and this neighborhood structure can be expressed as a local function. Fitness is represented as a local quadrature mapping in the form of altitude or height (Richter and Engelbrecht 2014).

We aim to describe the features of observed local landscapes. In certain low-dimensional spaces, the features of local landscapes might be easy to understand using intuitive expressions such as valleys, ridges and basins. However, as the problem dimensions increase, it may be very difficult to describe the features of local landscapes Wu et al. (2016). Therefore, we need to use new approaches to analyze the fitness landscape.

2.1 Local analysis method

The local features of a fitness landscape are defined based on certain statistical measures (Reeves and Eremeev 2004). The local analysis method for the local landscapes is the local optima in an observed local landscape. This approach is used to measure the roughness of a local landscape. We summarize the following methods for analyzing the local fitness landscape obtained from statistical sampling.

Roughness indicates changes in the neighborhood of fitness points. Each optimization problem corresponds to its own fitness landscape features; they may contain peaks, valleys and basins. Therefore, in the analysis of optimization problems, the analysis of roughness is also very important. The roughness of the fitness landscape increases the difficulty in searching for the optimal solution with evolutionary algorithms. Vassilev et al. (2003) presented the first entropic measure (FEM) as a measure of ruggedness. The FEM was adapted for real-valued and discrete problems (Malan and Engelbrecht 2009). The FEM is generated by a random walk adapting to a statistical sample, and it can generally be used as metric of information entropy for a sequence of symbols, indicating changes in fitness (Malan and Engelbrecht 2014).

The dispersion metric (Lunacek and Whitley 2006) is defined by the average pairwise distance between the best points.

$$DM = \frac{1}{p(p-1)} \sum_{i=1}^{p} \sum_{j=1, j \neq i}^{p} d\left(x_{rk_i}, x_{rk_j}\right) \tag{2}$$

where p is the number of best points and x_{rk_i} is the i-th point among λ samples. The dispersion metric is generally used in continuous optimization problems (Muñoz et al. 2012; Baumgartner and Sbalzarini 2009; Morgan and Gallagher 2014). In the application of this metric, candidate solutions are normalized to $[0,\ 1]^n$ for comparison, where n is the dimension of problem. Morgan and Gallagher (2014) clearly showed that the dispersion metric will converge to $1\big/\sqrt{6}$ as the number of dimensions goes to infinity.

2.2 Global analysis method

Another means of increasing the complexity of the global optimization problem is to consider the following global analysis fitness landscapes method.

In the initial definition of fitness-distance correlation (Jones and Forrest 1995), setting the fitness-distance correlation is used to obtain more global optimal solutions, in which the global optimal solution must be predicted. However, in practice, the species is not known, which is a shortcoming of this method. The main idea of fitness-distance correlation is to determine how close the relationship between fitness and distance is, which is used to find the optimum in the search space.

The fitness-distance correlation (FDC) (Müller and Sbalzarini 2011) is a global metric that measures the correlation between distance and fitness in adapting to the global optimum. The FDC can be defined as follows:

$$FDC = \frac{1}{\sigma_{\mathrm{F}}\sigma_{\mathrm{D}}\lambda}\sum_{i=1}^{\lambda}\left(f(x_i) - \overline{f}\right)\left(d_i - \overline{d}\right) \tag{3}$$

where λ is the number of candidate solutions $\chi = \{x_1, \ldots, x_\lambda\}$, $d_i = d\left(x_i, \_x_g\right)$ is the i-th distance function, $\overline{d}$ is the distance values, $\overline{f}$ denotes the average fitness, σ_{F} is the standard deviation of fitness, and σ_{D} denotes the standard deviation of distance. In general, the FDC employs the best candidate solution $x_g = \arg\min_{x\in\chi} f(x)$ instead of the global optimum. The result of the FDC measure is a useful fitness landscape feature and can be applied to combinatorial and continuous problems (Huang et al. 2018).

To improve the population space, evolvability is a commonly used method for landscape fitness analysis for evolutionary computation.

Smith et al. (2002) utilized the probability distribution to analyze the neighbor and solution space and proposed the evolvability metric for the candidate solution. The evolvability metric can be defined as follows:

$$EVO = \frac{1}{\lambda}\sum_{i=1}^{\lambda}\left|N^*(x_i)\right|\big/\left|N(x_i)\right| \tag{4}$$

where $N(x)$ denotes the set of neighborhood solutions, $|N(x)|$ is the number of $N(x)$, $N^*(x) = \{y|y \in N(x),\ f(y) \le f(x)\}$ is a new set of neighborhood solutions, whose value is better than the evaluation value. Note that evolvability adapts to each candidate solution. The evolvability metric representing the global landscape feature can be given in various ways using the values of $\left|N^*(x_i)\big/N(x_i)\right|$, $i = 1, \ldots, \lambda$. We can simply obtain the evolvability metric by averaging the evolvability values of λ candidate solutions. The evolvability metric can provide important information and reveal new fitness landscape features. However, previous experimental results show that the evolvability metric alone does not provide complete information (Pitzer and Affenzeller 2012); it is best used a complementary addition for fitness landscape analysis.

These new features might lead to a better understanding of the roughness of local landscapes along different axes. As other statistical measures of features of landscapes exist in the literature, such as autocorrelation (Weinberger 1990), epistasis variance (Davidor 1991), fitness-distance correlation and fitness variance (Radcliffe and Surry 1995), an intuitive research question is whether the combination of several features provides a better understanding of local landscapes. To test this idea, mixed-strategies differential evolution algorithms using multiple features were designed and evaluated in the present study.

3 Self-feedback operation

All of the optimization problems were given goals, and all obtained some responses in relation to their goals. These responses can be expressed as feedback. In general, goal presetting alone enhances the performance of optimization (Huang et al. 2017). The idea of self-feedback operations comes from the above description of feedback, which is self-adaptation to the goal, and can also increase the likelihood of achieving the goal and improving the optimization performance. The self-feedback operation designed in this paper is implemented with a self-feedback mapping function, self-feedback search and self-feedback transfer.

3.1 Self-feedback mapping function

We utilize the following formula as a self-feedback mapping function:

$$\lambda_{n+1} = \begin{cases} \lambda_n/0.4 & 0 < \lambda_n \le 0.4 \\ (1-\lambda_n)/0.6 & 0.4 < \lambda_n \le 1 \end{cases} \tag{5}$$

This self-feedback mapping function has the characteristics of randomness, ergodicity and regularity. The initial conditions meet the requirements of uniform dispersion and homogeneity. The self-feedback mapping function does not change the randomness of the initial conditions but increases the diversity of the population and individual search ergodicity. Figure 1 illustrates a self-feedback mapping function of the iterative point, which reflects the self-feedback mapping function with randomness, ergodicity and regularity.

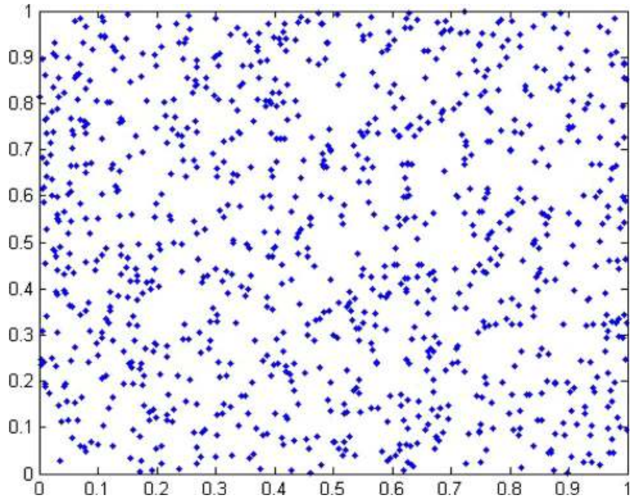

Fig. 1 A self-feedback mapping function of the iterative point

3.2 Self-feedback search

In the iterative process of self-feedback, we assume that one of the better individuals is $P = \left[x_1, x_2, \ldots, x_j, \ldots, x_n\right]$, and the sequence λ_j^k is generated by the self-feedback mapping function. In each iteration, $p_c = x_{j,\min} + \lambda_j^k(x_{j,\max} - x_{j,\min})$ and the new location of x_j is $x_j^{k+1} = \left(1 - \beta_g\right)x_j^k + \beta_g p_c$, $\beta_g = 1-((g-1)/g)^m$ where g is the evolution generation, and m is used to control the speed of compression. We regularly calculate the fitness of P. If the fitness is better than the original, this fitness replaces the original; otherwise, it is left unchanged. By controlling the self-feedback compression factor, the algorithm reduces the search space around the better individual gradually with the increase in evolution generations. The mutation space is large in the early stages of evolution, which helps the algorithm search for the global optimal solution in a wide space. In the late stages of evolution, as the mutation space becomes smaller, searching closely around the local optimal solution will help improve the accuracy of the solution in a small space.

3.3 Self-feedback transfer

In the traditional evolutionary algorithm, the convergence of the algorithm will be relatively fast at the beginning, but after a certain number of iterations, the similarity of the individuals in the population is higher and the diversity smaller. Despite the crossover operations, the impact on individual diversity is not obvious. To maintain the population diversity and prevent premature convergence, it is necessary to test the degree of population aggregation, which is the diversity measure of the population (Wang et al. 2017). When population diversity is low, a self-feedback transfer operation is performed, which generates a number of new individuals in each individual dimension according to a self-feedback iterative formula and substitutes those individuals for closer individuals. The equation for the diversity measure is as follows:

$$avgsum = [avgs_1, avgs_2, \ldots, avgs_D] = \left[\sum_{i=1}^{NP} x_{i1}, \sum_{i=1}^{NP} x_{i2}, \ldots, \sum_{i=1}^{NP} x_{iD}]/NP\right] \tag{6}$$

$$ts = [ts_1, ts_2, \ldots, ts_D] = \left[\frac{\sum\limits_{i=1}^{NP}(x_{i1} - avgs_1)^2}{(x_{1,\max} - x_{1,\min})}, \frac{\sum\limits_{i=1}^{NP}(x_{i2} - avgs_2)^2}{(x_{2,\max} - x_{2,\min})}, \ldots, \frac{\sum\limits_{i=1}^{NP}(x_{iD} - avgs_D)^2}{(x_{D,\max} - x_{D,\min})}\right] \tag{7}$$

$$\text{diversity} = \left(\sum_{i=1}^{D} ts_i\right)/(D * NP) \tag{8}$$

When the diversity measure value is less than the set threshold, the following self-feedback transfer operation is performed.

The self-feedback mapping function is used to generate an array $t\lambda$ of length D, and a self-feedback coefficient $s\lambda = t\lambda * 2 - 1 \in (-1, 1)$ is obtained. For an individual population $P = \left[x_1, x_2, \ldots, x_j, \ldots, x_D\right]$, the individual SP is computed through the following self-feedback transfer operation, and the original individual p is replaced by the individual SP:

$$SP(j) = \begin{cases} Pbest(j) + s\lambda(j) * (Pbest(j) - x_{j,\min}) & s\lambda(j) < 0 \\ Pbest(j) + s\lambda(j) * (x_{j,\min} - Pbest(j)) & s\lambda(j) \geq 0 \end{cases} \tag{9}$$

4 Van Genuchten model

Van Genuchten proposed a soil water characteristic curve, which is a key curve reflecting the movement of soil water and the movement of chemical substances in unsaturated soils.

Through the analysis of the curve, researchers can predict the movement of water in the soil. The Van Genuchten equation is defined as follows:

$$\theta(h) = \theta_r + \frac{\theta_s - \theta_r}{\left[1 + |\alpha h|^n\right]^{1-\frac{1}{n}}} \tag{10}$$

where θ is the soil water texture (cm^3/cm^{-3}), h is the pressure head (cm), θ_s denotes the equilibration of water texture, and θ_r is the surplus of water texture (cm^3/cm^{-3}). α is an empirical parameter and is often referred to as the inverse of the air entry point cm^{-1}, and n is an empirical constant affecting the shape of the curve (Van Genuchten 1980).

For the problem of fitting curves and optimization parameters, the least squares method is usually used to estimate the parameters as follows:

$$\min(f) = \sum_{i=1}^{n} \left|x_i - \hat{x}\right|^2 \tag{11}$$

where x_i is the measured data and $\hat{x}$ is the result of fitting curves. From the above description, we can set the fitness function of the Van Genuchten model as follows:

$$\min(f) = \sum_{l=1}^{n} \left|\theta(h_l) - \hat{\theta}(h_l)\right|^2 \tag{12}$$

However, the Van Genuchten equation has the following drawbacks: (1) The equation is a nonlinear complex equation, which means it is complicated to use with traditional mathematical methods. (2) The parameter values in the equation are difficult to handle, and it is not easy to obtain the global optimal solution. (3) The equation has poor versatility. (4) Human factors introduce uncertainties in adapting to the different experimental operations and environments.

Because the Van Genuchten model has the above shortcomings, scholars have used the method of fitting solutions to solve it. Wang et al. (2017) combined the nonlinear simplex method with the least squares method to solve the problem of retention characteristics in the soil water holding curve. Ma et al. (2005) used the damped least squares method to fit the parameters of the Van Genuchten equation. Although these methods reduce the difficulty in using the Van Genuchten equation to a certain extent, they are somewhat sensitive to the selection of initial values. With the development of the Van Genuchten equation, scholars have proposed solutions that are independent of the initial values. Chen and Ma (2006) proposed a random particle swarm optimization method to optimize parameters of the Van Genuchten equation. Xu et al. (2008) used the genetic algorithm to solve for the parameters of the Van Genuchten equation. Guo et al. (2009) used the hybrid genetic algorithm to estimate the parameters of the Van Genuchten equation. Their experimental results show that the use of a genetic algorithm to solve the Van Genuchten equation parameters not only speeds up the solution but also simplifies the solution process. In addition, the parameters are more accurate and the rate of convergence is improved.

Therefore, the present paper follows in the footsteps of these scholars and applies the SFSDE algorithm to the practical problems of soil water retention. Through its application to solving for the parameters of Van Genuchten equation and comparing the solution with the results of other algorithms, the feasibility of the SFSDE algorithm in practical applications is verified. To compare with other algorithms, this paper uses the mean square error (MSE) to evaluate the performance as follows:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \widehat{x}_i)^2 \tag{13}$$

5 Differential evolution

Storn and Price (1997) first proposed the differential evolution (DE) algorithm, which was originally used to solve the Chebyshev polynomial problem. As the DE algorithm has won many international algorithm competitions, an increasing number of scholars have studied it, expanded it and applied it successfully to all fields (Wu et al. 2017). The DE algorithm has been favored by many scholars as an emerging evolutionary computation because of its following advantages: (1) A global population-based search strategy is retained. (2) Real-world encoding is enabled. (3) Based on the differential mutation operation, the complexity of the algorithm is reduced. (4) The principle is simple and the algorithm is easy to apply. A problem to be optimized is expressed as follows:

$$\min f(\boldsymbol{x}),\ \boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in S \tag{14}$$

$\left[L_j, U_j\right]^n$ provides the initial vector randomly. The equation employs a random variable $rand$ to generate an initial individual.

$$x_i = rand\left(U_j - L_j\right) + L_j,\ i = 1, 2, \ldots, n \tag{15}$$

where $0 \le rand \le 1$ and $1 \le i, j \le n$.

Compared with other evolutionary algorithms, the basic framework of the DE algorithm is basically the same, including all three basic operations of selection, mutation and crossover. However, the major difference in the DE is the differential mutation operation. The following are six commonly used mutation strategies (Iorio and Li 2004; Qin et al. 2009; Storn 1996):

DE/rand/1/bin:

$$\mathbf{v}_{i,g} = \mathbf{x}_{r1,g} + F \cdot \left(\mathbf{x}_{r2,g} - \mathbf{x}_{r3,g}\right) \tag{16}$$

DE/rand/2/bin:

$$\mathbf{v}_{i,g} = \boldsymbol{x}_{r1,g} + F \cdot \left(\mathbf{x}_{r2,g} - \mathbf{x}_{r3,g}\right) + F \cdot \left(\mathbf{x}_{r4,g} - \mathbf{x}_{r5,g}\right) \tag{17}$$

DE/best/1/bin:

$$\mathbf{v}_{i,g} = \mathbf{x}_{best,g} + F \cdot \left(\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g}\right) \tag{18}$$

DE/best/2/bin:

$$\mathbf{v}_{i,g} = \mathbf{x}_{best,g} + F \cdot \left(\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g}\right) + F \cdot \left(\mathbf{x}_{r3,g} - \mathbf{x}_{r4,g}\right) \tag{19}$$

DE/current-to-rand/1/bin:

$$\mathbf{v}_{i,g} = \mathbf{x}_{i,g} + F \cdot \left(\mathbf{x}_{r1,g} - \mathbf{x}_{i,g}\right) + F \cdot \left(\mathbf{x}_{r2,g} - \mathbf{x}_{r3,g}\right) \tag{20}$$

DE/current-to-best/1/bin:

$$\mathbf{v}_{i,g} = \mathbf{x}_{i,g} + F \cdot \left(\mathbf{x}_{best,g} - \mathbf{x}_{i,g}\right) + F \cdot \left(\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g}\right) \tag{21}$$

The large differences among individuals in the initialization improve the initial search space of the algorithm. With the continuous iteration of the DE algorithm, the individual differences gradually reduce, causing the search range of the differential mutation operation to gradually shrink and the accuracy of local search to gradually increase, thus improving the convergence of the algorithm.

Based on the features of the mutation operator and fitness landscape, we propose a new mixed strategy of the DE/rand/1/bin, DE/rand/2/bin and DE/current-to-best/1/bin operators. This strategy is a dynamic and adaptive selection of a mixed mutation strategy based on fitness landscape feature distributions. The following equations are the new mixed mutation operators using the Gaussian and Cauchy distributions.

- s_{Gaussian}:

$$\mathbf{v}_i = \mathbf{x}_i + F_{\text{Gaussian}} \cdot (\mathbf{x}_{best} - \mathbf{x}_i) + F_{\text{Gaussian}} \cdot (\mathbf{x}_{r1} - \mathbf{x}_{r2}) \tag{22}$$

- s_{Cauchy}:

$$\mathbf{v}_i = \boldsymbol{x}_{r1} + F_{\text{Cauchy}} \cdot (\mathbf{x}_{r2} - \mathbf{x}_{r3}) + F_{\text{Cauchy}} \cdot (\mathbf{x}_{r4} - \mathbf{x}_{r5}) \tag{23}$$

where F_{Gaussian} and F_{Cauchy} are control parameters that refer to the Gaussian and Cauchy distribution, respectively.

$$\begin{cases} F_{\text{Gaussian}} = randgi(F, 0.1) \\ F_{\text{Cauchy}} = randci(F, 0.1) \end{cases} \tag{24}$$

where $randgi(F, 0.1)$ is a random value transfer from the Gaussian distance and $randci(F, 0.1)$ denotes a random value transfer from the Cauchy distance.

6 The proposed approach

In this paper, we propose a new self-feedback mixed-strategies differential evolution (SFSDE) based on fitness landscape features for the problem of soil water texture optimization. In the SFSDE algorithm, through the analysis of local fitness landscape features in a population space, a self-feedback operation is used to iteratively select and evaluate mutation operators. Moreover, mixed strategies and self-feedback transfer are combined to design a more efficient DE algorithm, enhance the search space and improve the convergence and accuracy (He et al. 2016; Lin et al. 2017). The framework of SFSDE is described in Algorithm 1.

```
Algorithm 1: The Proposed SFSDE Algorithm
Input: training function set $f = \{f_1, f_2, \cdots, f_r\}$ and control
       parameters $F = \{F_1, F_2, \cdots, F_L\}$
Initialize population based on the extract operation;
While $t \leq MaxGen$ do
For each training function $f_i$ do
Count the number of $Peaks_i$;
For $j = 1,2,\cdots,$ n do
Count control parameter $F_j$, denote by <$Peaks_i$, $F_j$>;
    For each control parameter $F_j$ do
    Add the components on all axes $X'^{(t)} = \sum_{j=1}^{n} X_j'^{(t)}$;
    Choose $F_j$ for mutation $X_j^{(t)} \to X_j'^{(t)}$ by equation
      $d_i = \sum_{j=1}^{D} | x_{ij} - x_{bestj} |$;
    Select better individuals to form a new population by
      self-feedback transfer Equations (7), (8) and (9);
$t = t + 1$;
End for
End while
Output: The highest fitness solution
```

7 Experimental results and analysis

In this section, we apply the SFSDE algorithm to the practical application problem of solving six types of Van Genuchten equation parameters. The following detailed experiments will explore the algorithm performance of the SFSDE from many aspects, such as the estimated value, the measured value, the accuracy, the convergence rate (Wang et al. 2018a, b).

7.1 Experiment setup

In the experiments, our approach, the SFSDE algorithm, is applied to six different types of Van Genuchten equation parameters. The SFSDE algorithm is compared with methods in three other references and three standard DE methods. The involved references are listed as below.

In Xu et al. (2008), the algorithm used to acquire 4 unknown parameters is developed by means of coupling the least squares method and the nonlinear simplex method, and the application of the hysteresis curve to numerical simulation is discussed (LSNS).

In Wang et al. (2000), parameters are estimated by solving the Van Genuchten equation using the damped least squares method (DLS).

In Ma et al. (2005), the optimized algorithm for estimating parameters is used to solve the Van Genuchten equation based on stochastic particle swarm optimization (SPSO).

Our approach is the SFSDE algorithm.

For the above five algorithms, the parameters θ_s, θ_r, α and m fall within the region [0,1], the control parameter F and crossover rate CR are equal to 0.5, the population size NP is equal to 30, and the maximum iteration number $g_{\max}$ is equal to 2000. The weight of selection P is equal to 0.05, and the weight of self-feedback c is equal to 0.5.

7.2 Experiment 1

In order to evaluate the SFSDE algorithm in solving the Van Genuchten characteristic curve equation, experiment 1 will be carried out from the following aspects: The SFSDE algorithm is applied to solve the Van Genuchten characteristic curve equation relevant parameters; then, the actual values and the calculated values are compared with the existing research methods in different references. The performance evaluation of the SFSDE algorithm in this part of the experiment is mainly focused on the optimization accuracy. The six types of Van Genuchten equation parameter values in Table 1 are the result of Eq. (10).

Tables 2 and 3 are for different methods of comparing the simulated value and actual value; the actual values in Tables 2 and 3 were derived from Xu et al. (2008) and Wang et al. (2000), respectively. Better results from SFSDE algorithm optimization are marked in bold. The experimental results show that the SFSDE algorithm is more accurate and computationally more efficient than the other three algorithms in calculating the parameters of the Van Genuchten equation.

The comparison results of the four methods used to calculate the parameters of the Van Genuchten equation are detailed in Table 4. The results of the three parameters θ_s, α and n calculated by the SFSDE algorithm are superior to those of the Xu et al. (2008), Wang et al. (2000) and Ma et al. (2005), and the parameters θ_r and m are also close to each other. For the objective function of the Van Genuchten equation, the experimental results show that the SFSDE algorithm not only has obvious advantages in the accuracy of the solution compared with the other three literature, but also overcomes the phenomenon of falling into a local optimum.

Table 5 shows the comparison of MSE between the simulated and actual values for different algorithms. The bold values represent the best results in this table. The values are from Table 5 in Ma et al. (2005). Compared with the SPSO method in Ma et al. (2005), we can conclude that the MSEs of the SFSDE algorithm for silt loam dehumidification and moisture absorption and sand loam moisture absorption and gravel dehumidification are low. Therefore, the SFSDE algorithm produces results closer to the actual values. The SFSDE algorithm produces smaller MSE values than the LSNS val-

Table 1 The six types of Van Genuchten equation parameters

Types	$\theta_s/\mathrm{cm}^3\cdot\mathrm{cm}^{-3}$	$\theta_r/\mathrm{cm}^3\cdot\mathrm{cm}^{-3}$	α	m	n
Silt loam dehumidification	3.19E−01	6.59E−02	1.27E−02	7.53E−01	4.01E+00
Silt loam moisture absorption	3.52E−01	6.25E−02	1.28E−02	8.13E−01	5.34E+00
Sand loam dehumidification	3.78E−01	8.47E−02	1.50E−02	7.75E−01	4.43E+00
Sand loam moisture absorption	3.74E−01	8.17E−02	1.52E−02	7.98E−01	4.96E+00
Gravel dehumidification	2.74E−01	6.05E−02	2.34E−02	6.71E−01	3.04E+00
Gravel moisture absorption	2.69E−01	5.65E − 02	2.27E−02	7.08E−01	3.43E+00

Table 2 The results of h calculated by different methods for silt loam

Types	Solution methods		Matric potential (h/cm)								
			5	30	70	100	200	300	500	1000	1300
Silt loam	Dehumidification	Actual data	3.63E−01	3.47E−01	2.79E−01	1.65E−01	9.50E−02	8.10E−02	6.90E−02	5.90E−02	5.40E−02
		LSNS	3.63E−01	3.61E−01	3.04E−01	2.10E−01	**9.70E−02**	6.00E−02	5.40E−02	5.30E−02	5.30E−02
		SPSO	3.59E−01	3.54E−01	2.69E−01	1.77E−01	8.30E−02	7.10E−02	6.70E−02	6.60E−02	6.60E−02
		SFSDE	3.59E−01	**3.44E−01**	**2.70E−01**	**1.67E−01**	8.27E−02	**8.00E−02**	**6.80E−02**	**5.80E−02**	**5.38E−02**
	Moisture absorption	Actual data	3.62E−01	3.40E−01	2.69E−01	1.41E−01	8.20E−02	6.90E−02	6.10E−02	5.60E−02	5.30E−02
		LSNS	3.63E−01	3.58E−01	2.58E−01	1.56E−01	6.60E−02	**6.70E−02**	5.40E−02	5.30E−02	**5.30E−02**
		SPSO	3.52E−01	3.51E−01	2.66E−01	1.45E−01	6.70E−02	6.30E−02	6.30E−02	6.30E−02	6.30E−02
		SFSDE	**3.62E−01**	**3.41E−01**	**2.66E−01**	**1.40E−01**	**7.89E−02**	6.34E−02	**6.26E−02**	**5.55E−02**	5.65E−02

ues in Xu et al. (2008) for the six types being compared, and the result shows that the SFSDE algorithm is significantly better than the LSNS method.

7.3 Experiment 2

To further analyze the SFSDE algorithm's convergence rate and accuracy, this section compares the standard DE algorithm with the proposed SFSDE algorithm on the experimental results of the parameters from the dehumidification and moisture absorption of silt loam.

In this experiment, the standard DE algorithm uses the mixed strategy to carry out mutation operations, where F is set to 0.5 and CR is 0.9. The SFSDE algorithm has the same F at 0.5, but the CR is 0.5, the weight of selection p is 0.05, and the weight of self-feedback C is 0.5. The population size NP is 30, and the maximum number of iterations g_{max} is 2000.

Table 6 compares the experimental data of the SFSDE algorithm and the standard DE algorithm using three different mutation operators. The experimental data show that the SFSDE algorithm has the highest accuracy among the four different algorithms, followed by the standard DE algorithm with the DE/rand/2/bin mutation operator. The accuracies of min(f) calculated with the DE/current-to-best/1/bin and DE/rand/1/bin mutation operators are lower than with the LSNS and SPSO methods.

Figures 2 and 3 are iterative curves of the SFSDE algorithm and the standard DE algorithm using three different mutation operators to calculate the parameters of the Van Genuchten equation for silt loam dehumidification and moisture absorption, respectively. The two figures reflect the relationship between the number of iterations and fitness. Each figure takes the first 500 generations of each algorithm as curve data. The star label line is an iterative curve of the SFSDE algorithm. The circle, plus and rhombus label are iteration curves using the DE/rand1/bin, DE/rand/2/bin and DE/current-to-best/1/bin mutation operators, respectively. In the iterative curve figures, if the curve descends faster, the algorithm converges faster. The closer the curve is to 0, the faster the algorithm converges and the more accurate is the algorithm. We use Eq. (12) as a fitness function to calculate the parameters of the Van Genuchten equation for silt loam dehumidification and moisture absorption with a minimum value of 0. The following conclusions can be drawn from these two iteration curve figures.

When the standard DE algorithm adopts the DE/rand/2/bin and DE/current-to-best/1/bin mutation operators, the performance of the algorithm in convergence rate and accuracy is not ideal. When compared with the SFSDE algorithm, the DE/rand/2/bin operator standard DE algorithm has a final convergence rate that deviates relatively far from the minimum value of 0. The SFSDE algorithm converges fastest, as indicated by the steepest descent of the star label curve in the

Table 3 The results of *h* calculated by different methods for sand loam

Types		Solution methods	Matric potential (h/cm)							
			10	50	100	200	400	800	1000	1300
Sand loam	Dehumidification	Actual data	3.77E−01	3.30E−01	1.46E−01	1.06E−01	9.20E−02	8.30E−02	7.90E−02	7.60E−02
		LSNS	3.80E−01	3.18E−01	1.60E−01	8.70E−02	7.30E−02	**7.00E−02**	7.00E−02	**7.00E−02**
		SPSO	3.78E−01	3.28E−01	1.50E−01	9.20E−02	8.60E−02	8.50E−02	8.50E−02	8.50E−02
		SFSDE	**3.78E−01**	**3.18E−01**	**1.50E−01**	**8.15E−02**	**7.24E−02**	8.48E−02	**8.08E−02**	7.48E−02
	Moisture absorption	Actual data	3.74E−01	3.26E−01	1.33E−01	9.20E−02	8.90E−02	8.10E−02	7.80E−02	7.40E−02
		LSNS	3.80E−01	3.25E−01	1.36E−01	7.70E−02	7.10E−02	8.20E−02	7.00E−02	7.00E−02
		SPSO	3.74E−01	3.25E−01	1.32E−01	8.50E−02	8.20E−02	8.20E−02	8.20E−02	8.20E−02
		SFSDE	**3.74E−01**	**3.25E−01**	**1.32E−01**	**8.93E−02**	**8.81E−02**	**8.11E−02**	**7.70E−02**	**7.20E−02**
Gravel	Dehumidification	Actual data	2.72E−01	1.73E−01	9.30E−02	7.40E−02	6.60E−02	6.10E−02	5.90E−02	5.70E−02
		LSNS	2.71E−01	1.91E−01	8.02E−01	6.50E−02	5.50E−02	5.30E−02	5.30E−02	5.20E−02
		SPSO	2.72E−01	1.72E−01	9.60E−02	**6.90E−02**	6.30E−02	6.10E−02	6.10E−02	6.10E−02
		SFSDE	**2.72E−01**	**1.72E−01**	**9.21E−02**	6.95E−02	**6.57E−02**	**6.10E−02**	**5.78E−02**	**5.67E−02**
	Moisture absorption	Actual data	2.68E−01	1.67E−01	8.10E−02	6.90E−02	6.20E−02	5.60E−02	5.40E−02	5.20E−02
		LSNS	2.68E−01	1.65E−01	8.80E−02	6.10E−02	5.40E−02	5.30E−02	5.20E−02	5.20E−02
		SPSO	2.68E−01	1.66E−01	**8.40E−02**	6.20E−02	5.80E−02	5.70E−02	5.70E−02	5.70E−02
		SFSDE	**2.68E−01**	**1.66E−01**	8.44E−02	**6.11E−02**	**5.35E−02**	**5.65E−02**	**5.47E−02**	**5.16E−02**

Table 4 The results of Van Genuchten parameters calculated by different methods

Parameters	LSNS	DLS	SPSO	SFSDE
$\theta_s/cm^3 \cdot cm^{-3}$	3.63E−01	3.63E−01	3.59E−01	**3.19E−01**
$\theta_r/cm^3 \cdot cm^{-3}$	5.30E−02	5.30E−02	6.60E−02	6.59E−02
α	1.30E−02	1.40E−02	1.30E−02	**1.27E−02**
m	7.64E−01	7.62E−01	7.53E−01	7.53E−01
n	4.24E+00	4.20E+00	4.04E+00	**4.01E+00**
$\min(f)$	2.09E−03	3.82E−03	8.17E−04	**7.54E−04**

Table 5 Comparison of the MSE values for different methods

Types	LSNS	SPSO	SFSDE
Silt loam dehumidification	430.8	83.8	**81.37**
Silt loam moisture absorption	125.4	69.0	**65.19**
Sand loam dehumidification	169.6	48.6	51.26
Sand loam moisture absorption	101.5	21.4	**20.49**
Gravel dehumidification	91.6	7.4	**7.18**
Gravel moisture absorption	24.3	13.8	14.72

figure, which is the fastest point in the four curves to reach its optimum.

8 Conclusions

In this paper, the SFSDE algorithm is proposed to analyze the local fitness landscape features in each population space, and a self-feedback operation is used to iteratively select and evaluate mutation operators. Mixed strategies and self-feedback transfer are combined to design a more efficient DE algorithm and then enhance the search space, convergence rate and accuracy. Finally, the proposed SFSDE algorithm is applied to the optimization of soil water textures. The scope of the experiments covers more comprehensive territory, including a total of six texture types: silt loam dehumidification, silt loam moisture absorption, sand loam dehumidification, sand loam moisture absorption, gravel dehumidification and gravel moisture absorption. According to the comparison of the six types' experimental and actual values (Table 3, 4 and 5), the $\hat{\theta}$ value obtained by the SFSDE algorithm is closer to the actual $\hat{\theta}$ value, demonstrating that the SFSDE algorithm is more effective.

To analyze the performance of the SFSDE algorithm in parameter optimization of nonlinear complex equations more comprehensively, this paper also compares the SFSDE algorithm with the three mutation operators of the standard DE. The experimental results in Table 6 show that the SFSDE algorithm has the lowest fitness value among the four algorithms, which indicates that the $\hat{\theta}$ value obtained by the SFSDE algorithm is the closest to the actual value among the four algorithms. Figures 2 and 3 illustrate that the SFSDE algorithm has faster convergence and higher accuracy than the DE algorithm using the other three mutation operators. This result confirms once again that the SFSDE algorithm can be used to calculate nonlinear complex equation parameters with good performance.

By experimenting with the Van Genuchten equation to verify the performance of the algorithm, we prove that the proposed SFSDE algorithm can simplify the solution process and provides a new way to calculate the parameters of the

Table 6 Comparison between the SFSDE algorithm and three standard DE algorithms by calculating Van Genuchten equation parameters for silt loam

Types	Solution methods	θ_s	θ_r	α	m	n	$\min(f)$
Silt loam							
Dehumidification	DE/rand1/bin	0.366	0.054	0.014	0.643	4.197	1.266E−03
	DE/rand/2/bin	0.359	0.066	0.013	0.751	4.011	7.542E−04
	DE/current-to-best/1/bin	0.416	0.036	0.027	0.466	1.873	6.002E−03
	SFSDE	0.359	0.066	0.013	0.753	4.053	**7.540E−04**
Moisture absorption	DE/rand1/bin	0.361	0.053	0.014	0.683	3.151	1.551E−03
	DE/rand/2/bin	0.356	0.059	0.013	0.758	4.130	8.569E−04
	DE/current-to-best/1/bin	0.413	0.032	0.032	0.457	1.847	8.476E−03
	SFSDE	0.352	0.063	0.013	0.813	5.336	**6.205E−04**

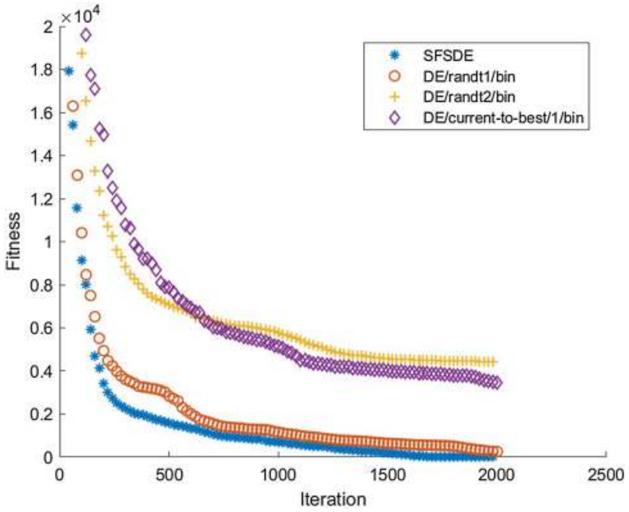

Fig. 2 Iteration curves of the SFSDE algorithm for silt loam dehumidification

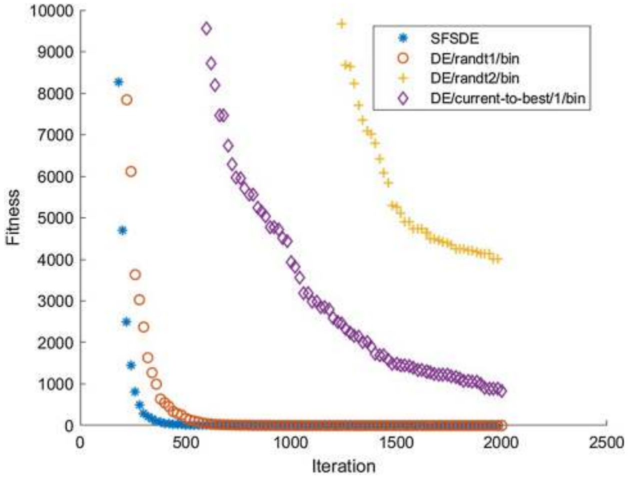

Fig. 3 Iteration curves of the SFSDE algorithm for silt loam moisture absorption

Van Genuchten equation. Our next work will be to optimize the SFSDE algorithm for all aspects, and we believe the algorithm can be generalized to solve a variety of other complex optimization problems.

Acknowledgments This work was supported by the National Natural Science Foundation of China under Grant Nos. 61561024 and 61703170, and the Science and Technology Research Project of Jiangxi Province under Grant Nos. GJJ160930 and GJJ160631.

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this paper.

References

Al-Smadi M, Qawasmeh O, Al-Ayyoub M, Jararweh Y, Gupta B (2017) Deep recurrent neural network versus support vector machine for aspect-based sentiment analysis of Arabic hotels' reviews. J Comput Sci

Baumgartner B, Sbalzarini IF (2009) Particle swarm CMA evolution strategy for the optimization of multi-funnel landscapes. In: IEEE congress on evolutionary computation, 2009. CEC'09. IEEE, pp 2685–2692

Bongiovanni R, Lowenberg-DeBoer J (2004) Precision agriculture and sustainability. Precis Agric 5(4):359–387

Cao Y, Zhou Z, Sun X, Gao C (2018) Coverless information hiding based on the molecular structure images of material. Comput Mater Contin 54(2):197–207

Chen DC, Ma YJ (2006) Optimized algorithm for estimating parameters by solving Van Genuchten equation based on stochastic particle swarm optimization. Trans Chin Soc Agric Eng 22(12):82–85

Davidor Y (1991) Epistasis variance: a viewpoint on GA-hardness. In: Proceedings of FOGA I, Morgan Kaufmann, pp 23–35

Malan KM, Engelbrecht, AP (2014) A progressive random walk algorithm for sampling continuous fitness landscapes. In 2014 IEEE congress on evolutionary computation (CEC). IEEE, pp 2507–2514

Eskandarpour M, Nikbakhsh E, Zegordi SH (2014) Variable neighborhood search for the bi-objective post-sales network design problem: a fitness landscape analysis approach. Comput Oper Res 52:300–314

Guo XH, Sun XH, Ma JJ (2009) Parametric estimation of the van Genuchten's equation based on hybrid genetic algorithm. Adv Water Sci 20(5):677–682

Gupta B, Agrawal DP, Yamaguchi S (2016) Handbook of research on modern cryptographic solutions for computer and cyber security. IGI Global, Hershey, PA, pp 1–589. https://doi.org/10.4018/978-1-5225-0105-3

He P, Deng Z, Wang H, Liu Z (2016) Model approach to grammatical evolution: theory and case study. Soft Comput 20(9):3537–3548

Hossain MS, Muhammad G, Abdul W, Song B, Gupta BB (2018) Cloud-assisted secure video transmission and sharing framework for smart cities. Future Gener Comput Syst 83:596–606

Huang Z, Liu S, Mao X, Chen K, Li J (2017) Insight of the protection for data security under selective opening attacks. Inf Sci 412:223–241

Huang Y, Li W, Liang Z, Xue Y, Wang X (2018) Efficient business process consolidation: combining topic features with structure matching. Soft Comput 22(2):645–657

Iorio AW, Li X (2004) Solving rotated multi-objective optimization problems using differential evolution. In: Australasian joint conference on artificial intelligence. Springer, Berlin, Heidelberg, pp 861–872

Jones T, Forrest S (1995) Fitness distance correlation as a measure of problem difficulty for genetic algorithms. Santa Fe Institute, Working Paper 95-02-022

Kallel L, Naudts B, Reeves CR (2001) Properties of fitness functions and search landscapes. In: Theoretical aspects of evolutionary computing, Springer, Berlin Heidelberg, pp 175–206

Lee CY, Yao X (2004) Evolutionary programming using mutations based on the Lévy probability distribution. IEEE Trans Evol Comput 8(1):1–13

Li W, Li K, Huang Y, Yang S, Yang L (2017a) A EA-and ACA-based QoS multicast routing algorithm with multiple constraints for ad hoc networks. Soft Comput 21(19):5717–5727

Li W, Li S, Chen Z, Zhong L, Ouyang C (2017b) Self-feedback differential evolution adapting to fitness landscape characteristics. Soft Comput. https://doi.org/10.1007/s00500-017-2833-y

Li T, Gupta BB, Metere R (2018) Socially-conforming cooperative computation in cloud networks. J Parallel Distrib Comput 117:274–280

Lin W, Xu S, Li J, Xu L, Peng Z (2017) Design and theoretical analysis of virtual machine placement algorithm based on peak workload characteristics. Soft Comput 21(5):1301–1314

Lunacek, M., & Whitley, D. (2006). The dispersion metric and the CMA evolution strategy. In Proceedings of the 8th annual conference on Genetic and evolutionary computation. ACM, pp 477–484

Ma YS, Hu D, Shen B (2005) Estimating parameters by solving Van Genuchten equation using the damper least square method. Trans Chin Soc Agric Eng 21(8):179–180

Malan KM, Engelbrecht AP (2009) Quantifying ruggedness of continuous landscapes using entropy. In IEEE congress on evolutionary computation, 2009. CEC'09. IEEE, pp 1440–1447

McBratney A, Whelan B, Ancev T, Bouma J (2005) Future directions of precision agriculture. Precis Agric 6(1):7–23

Merz P (2004) Advanced fitness landscape analysis and the performance of memetic algorithms. Evol Comput 12(3):303–325

Merz P, Freisleben B (1999) Fitness landscapes and memetic algorithm design. In: Corne D, Dorigo M, Glover F (eds) New ideas in optimization. McGraw-Hill, pp 245–260

Merz P, Freisleben B (2000) Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. IEEE Trans Evol Comput 4(4):337–352

Morgan R, Gallagher M (2014) Sampling techniques and distance metrics in high dimensional continuous landscape analysis: limitations and improvements. IEEE Trans Evol Comput 18(3):456–461

Müller CL, Sbalzarini IF (2011) Global characterization of the CEC 2005 fitness landscapes using fitness-distance analysis. In: European conference on the applications of evolutionary computation, Springer, Berlin, Heidelberg, pp 294–303

Muñoz MA, Kirley M, Halgamuge SK (2012) A meta-learning prediction model of algorithm performance for continuous optimization problems. In: International conference on parallel problem solving from nature, Springer, Berlin, Heidelberg, pp 226–235

Pitzer E, Affenzeller M (2012) A comprehensive survey on fitness landscape analysis. Recent Adv Intell Eng Syst 378:161–191

Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evol Comput 13(2):398–417

Radcliffe NJ, Surry PD (1995) Fitness variance of formulae and performance prediction. In: Prof. of FOGA III, Morgan Kaufmann, pp 51–72

Reeves CR, Eremeev AV (2004) Statistical analysis of local search landscapes. J Oper Res Soc 55(7):687–693

Richter H, Engelbrecht A (eds) (2014) Recent advances in the theory and application of fitness landscapes. Springer, New York

Rohlfshagen P, Yao X (2013) Dynamic combinatorial optimization problems: a fitness landscape analysis. Metaheuristics Dyn Optim 433:79–97

Shen L, He J (2010) A mixed strategy for evolutionary programming based on local fitness landscape. In: 2010 IEEE congress on evolutionary computation (CEC). IEEE, pp 1–8

Smith T, Husbands P, Layzell P, O'Shea M (2002) Fitness landscapes and evolvability. Evol Comput 10(1):1–34

Stadler PF, Stephens CR (2003) Landscapes and effective fitness. Comments® Theor Biol 8(4):389–431

Storn R (1996) On the usage of differential evolution for function optimization. In: 1996 Biennial Conference of the North American Fuzzy Information Processing Society, 1996, NAFIPS. IEEE, pp 519–523

Storn R, Price K (1997) Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11(4):341–359

Vachaud G, Passerat de Silans A, Balabanis P, Vauclin M (1985) Temporal stability of spatially measured soil water probability density function. Soil Sci Soc Am J 49(4):822–828

Van Genuchten MT (1980) A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. Soil Sci Soc Am J 44(5):892–898

Vassilev VK, Fogarty TC, Miller JF (2003) Smoothness, ruggedness and neutrality of fitness landscapes: from theory to application. In: Ghosh A, Tsutsui S (eds) Advances in evolutionary computing. Natural computing series, Springer, Berlin, Heidelberg, pp 3–44

Wang JS, Yang ZF, Chen JJ, Wang ZM (2000) Study on water hysteresis in aerated soil. J Hydraul Eng 31(2):1–6

Wang F, Zhang Y, Rao Q, Li K, Zhang H (2017) Exploring mutual information-based sentimental analysis with kernel-based extreme learning machine for stock prediction. Soft Comput 21(12):3193–3205

Wang F, Zhang H, Li K, Lin Z, Yang J, Shen XL (2018a) A hybrid particle swarm optimization algorithm using adaptive learning strategy. Inf Sci 436:162–177

Wang H, Wang W, Cui Z, Zhou X, Zhao J, Li Y (2018b) A new dynamic firefly algorithm for demand estimation of water resources. Inf Sci 438:95–106

Weinberger E (1990) Correlated and uncorrelated fitness landscapes and how to tell the difference. Biol Cybern 63(5):325–336

Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82

Wolpert DH, Macready WG (2005) Coevolutionary free lunches. IEEE Trans Evol Comput 9(6):721–735

Wright S (1932) The roles of mutation, inbreeding, crossbreeding, and selection in evolution (Vol 1, No 8)

Wu Z, Tian L, Li P, Wu T, Jiang M, Wu C (2016) Generating stable biometric keys for flexible cloud computing authentication using finger vein. Inf Sci 433:431–447

Wu H, Kuang L, Wang F, Rao Q, Gong M, Li Y (2017) A multiobjective box-covering algorithm for fractal modularity on complex networks. Appl Soft Comput 61:294–313

Xu XJ, Huang XP, Zhang JL (2008) Using genetic algorithm to estimate parameters of Van Genuchten equation. Geotech Eng Tech 22(2):75–78

Yang L, Han Z, Huang Z, Ma J (2018) A remotely keyed file encryption scheme under mobile cloud computing. J Netw Comput Appl 106:90–99

Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. IEEE Trans Evol Comput 3(2):82–102

Zhang C, Kovacs JM (2012) The application of small unmanned aerial systems for precision agriculture: a review. Precision Agric 13(6):693–712

Zhou J, Wang F, Xu J, Yan Y, Zhu H (2018) A novel character segmentation method for serial number on banknotes with complex background. J Ambient Intell Hum Comput 1–15

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.