Mutation Strategy Selection Based on Fitness Landscape Analysis: A Preliminary Study

Jing Liang[1](✉), Yaxin Li[1](✉), Boyang Qu[2], Kunjie Yu[1], and Yi Hu[1]

[1] School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China
{liangjing,yukunjie}@zzu.edu.cn, zzuliyaxin@163.com, eehuyi@163.com
[2] School of Electronic and Information Engineering,
Zhongyuan University of Technology, Zhengzhou 450007, China
qby1984@hotmail.com

Abstract. Different algorithms and strategies behave disparately for different types of problems. In practical problems, we cannot grasp the nature of the problem in advance, so it is difficult for the engineers to choose a proper method to solve the problem effectively. In this case, the strategy selection task based on fitness landscape analysis comes into being. This paper gives a preliminary study on mutation strategy selection on the basis of fitness landscape analysis for continuous real-parameter optimization based on differential evolution. Some fundamental features of the fitness landscape and the components of standard differential evolution algorithm are described in detail. A mutation strategy selection framework based on fitness landscape analysis is designed. Some different types of classifiers which are applied to the proposed framework are tested and compared.

Keywords: Mutation strategy selection · Fitness landscape analysis · Classifier · Differential evolution algorithm

1 Introduction

In the context of Evolutionary Algorithms (EA), the option of previous search strategies often relies on the experience of the algorithm designer, the prior knowledge or parameters of the problem to be optimized. However, based on the No Free Lunch theory, different algorithms may have advantages over different problems and researches cannot ask an algorithm to perform well for all types. It is difficult for the engineers to find a proper optimization algorithm to solve a certain problem. Therefore, we consider the fitness landscape analysis (FLA) and search for the suitable strategy based on the nature and characteristics of optimization problems. According to the definition of fitness landscape [1], it is a triple $L = (S, V, F)$, where S is a collection of all solutions, V: S → 2^s is a specified neighborhood function, for each $s \in S$, the set of its neighbors $V(s)$, and f: S → $\mathbb{R}$ is the fitness function. The role of the fitness landscape is to contrast

L. Pan et al. (Eds.): BIC-TA 2019, CCIS 1159, pp. 284–298, 2020.
https://doi.org/10.1007/978-981-15-3425-6_23

with the real landscape, so that we can understand the working principle of the algorithm and solve the practical problems better [2]. FLA refers to a collection of data technologies used to extract descriptive or numerical metrics related to fitness landscape attributes [3].

Several descriptive features have been proposed in the earlier researches of FLA. For instance, the concept closely related to multimodality is smoothness, which refers to the magnitude of change in fitness within the neighborhood. Besides, the rugged landscape has great fluctuations between the neighborhoods, showing steep ascents and descents. Furthermore, the neutral landscape has large flat areas or stepped surfaces whose input changes do not produce significant output changes. Of course, there are some other accessorial measures. For example, information landscape hardness (ILH) by Borenstein and Poli [4,5] with extensions [6] was focused on deception in terms of difference from a landscape with perfect information for search. The result of the evaluation is a value in the range of [0,1], where 0 indicates no misleading information and 1 indicates maximum misleading information. In addition, fitness cloud (FC) by Verel et al. [7] with extensions [8] was concentrated on evolvability. It uses a scatter plot to represent the relationship between parents and the child. What is more, the negative slope coefficient (NSC) has been defined to capture some of the characteristics of FC with a single number. It is known from the classification hypothesis [8]: If NSC $= 0$, the problem is easy to solve; if NSC < 0, its value quantifies this difficulty: the smaller the value, the harder the problem. Vanneschi et al. [8,9] discussed the pros and cons of this measure.

These measures can help us find characteristics of the fitness landscape. It is worth noting that the ultimate goal of FLA in this work is to find the correlation between the properties of the fitness landscape and the performance of algorithms. Some studies have made constructive progress in this regard.

(1) Discrete fitness landscape analysis and its application on practical industrial problems

Information on perfect landscapes through the means of discrete time fourier transform (DTFT) and dynamic time warping (DTW) distances was obtained by Lu et al. [10]. In order to analyze the fitness landscape deeply, the authors proposed five methods, including the stability of amplitude variation, keenness, periodicity, similarity and the degree of change in average fitness. In addition, the author applied these criteria to task scheduling issues to illustrate fairness and adaptability.

(2) Improvement and statistical study on the characteristics of fitness landscape

In the evolutionary computation community, the properties of dispersion metric were studied by Morgan et al. [11]. In order to improve the defects of dispersion metric, the author proposed three independent modifications to the basic methodology, namely, the standardization with the dispersive boundary, the LP norm of fraction p, and the random walk with fixed steps. Finally, the results demonstrated that these improvements can promote convergence and increase the separability of problems. A theorem was proved in this paper by deriving the formula: Given $t = 1$ and $S = [0, 1]^D$, the dispersion of solutions sampled uniform randomly from S will converge to $1/\sqrt{6}$ as $D \rightarrow \infty$. In the meantime, t is the number of subsamples, S is the value in the interval [0,1] and D is the dimension of space.

(3) Improving adaptive algorithms by using the characteristics of fitness landscape

Li et al. [12] proposed a new self-feedback DE algorithm (SFDE), which selected the optimal mutation strategy by extracting the features of local fitness landscape. The probability distributions of single mode and multimodality are calculated in each local fitness landscape. The advantage of the self-feedback control mechanism is that when the group falls into the local optimal solution, the inferior solution is more conducive to the population which can help to jump out of the local optima.

(4) Analysis and research on dynamic fitness landscape

Static FLA focuses on extracting the attributes of a problem without considering any information about the optimization algorithm. In contrast, dynamic FLA combines the behavior of the algorithm with the attributes of optimization problems to determine the effectiveness of a given algorithm to solve the problem. Wang et al. [3] used the concept of population evolvability as an important basis for dynamic FLA. The authors utilized the evolutionary of the population (evp) to represent the evolution of the entire population. Finally, the effectiveness of the proposed algorithm selection framework was proved by experiments.

The main difference between this work and the above researches is that we find that the spatial topography of the optimization problem can directly reflect characteristics of the problem. Since Differential Evolution (DE) algorithm is a very popular EA, we take it as an example and design mutation strategy selection based on DE to solve optimization problems. Therefore, the purpose of this work is to find the proper strategy of DE algorithm for each problem. Then, we establish a strategy selection model by learning the relationship between excellent strategies and features of the landscape to improve the intelligent solving ability of optimization problems.

The rest of this paper is organized as follows. Section 2 is devoted to explaining the measures of FLA and DE algorithm summarized in this paper. Section 3 discusses the mutation strategy selection task based on the black box optimization problem in detail. Further experimental analyses are presented in Sect. 4. Finally, Sect. 5 provides the conclusion.

2 Measures of Fitness Landscape Analysis

2.1 Global Sampling and Local Sampling

Sampling refers to the process of converting continuous variables in the spatial domain into discrete variables. Global sampling is to taking values over the entire spatial region while local sampling is to collecting data in a partial region [13]. We reduced the problem space to 2 dimensions and used function $F = x^2$ to better demonstrate both sampling methods. The schematic diagram is shown in Fig. 1. It should be mentioned that the two sampling methods are applied to the characteristics of different fitness landscapes according to the dissimilar situation.

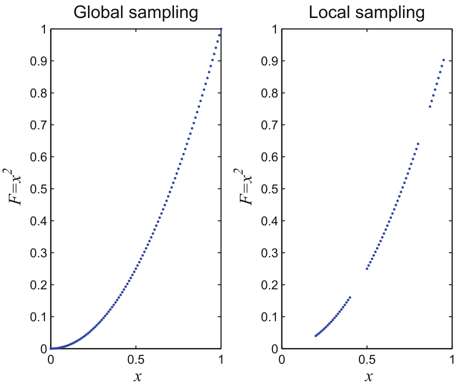

Fig. 1. The schematic diagram of global sampling and local sampling

2.2 Fitness Landscape Characteristics

The extraction of landform information helps to analyze characteristics of the problem to be optimized. As a preliminary study, this paper selects four characteristics to measure continuous problems with unknown optima based on sampling points of fitness landscape.

(1) The number of optimal values (NUM)

We have improved a simple method to measure NUM in FLA which is based on the achievements of Sheng et al. [14]. Its implementation is described as follows:

For a random sample $x_1, \ldots, x_u$.

1) Finding the best point of the sample and set it to x^*. Then, calculating the distance (d_i) between each $x_i (i = 1, \ldots, u)$ and x^* as Eq. (1):

$$d_i = \sum_{j=1}^{n} \left| x_{i,j} - x_j^* \right| . \tag{1}$$

The difference between the above distance and the Euclidean distance is that the former performs fast calculation by removing the square root.

2) Sorting the individuals based on the distance value from low to high, and denoting the order by $k_1, k_2, \ldots, k_u$.

3) Setting $c = 0$ initially. Then, the value of c will be increased by 1, if $x(k_m)(m = 1, \ldots, u)$ is better than $x(k_{m-1})(m = 1, \ldots, u)$ (if exists) and $x(k_{m+1})(m = 1, \ldots, u)$ (if exists). Finally, the c is taken as the number of optimal value in the fitness landscape observation. It should be emphasized that $x(k_m)$ is only the optimal value estimated from the sample to reflect attributes of the fitness landscape, which is not the true optimum.

Intuitively, the ruggedness can be estimated by the distribution of the optimal values in the current sample.

(2) Basin size ratio (BSR)

BSR is caculated by Eq. (2) and it pictures the existence of a dominant basin [15].

$$BSR = \frac{\max_x |B(x)|}{\min_x |B(x)|}. \tag{2}$$

where $\max_x |B(x)|$ is the maximum fitness value in the local sampling points, $\min_x |B(x)|$ is the minimum value. Due to the wide range in fitness values of various problems, normalized BSR is employed in our work.

(3) Keenness (KEE)

Lu et al. [16] proposed a method to describe keenness of the topography. It is computed by Eq. (3):

$$\begin{aligned} KEE = a_{\text{sum}} \times (-1) + b_{\text{sum}} \times (-0.6) + c_{\text{sum}} \\ \times (-0.2) + d_{\text{sum}} \times (-0.2) + e_{\text{sum}} \times (+1). \end{aligned} \tag{3}$$

where the coefficients for a_{sum}, b_{sum}, c_{sum}, d_{sum} and e_{sum} are allocated according to the contribution to the keenness degree. The larger the value of KEE, the sharper the solution space.

(4) Fitness distance correlation (FDC)

FDC was proposed by Jones et al. [17] to measure the relationship between the parameter space and the fitness values. It is designed to evaluate whether the landscape is unimodal or multimodal and whether it has a strong global structure. It is shown by Eq. (4):

$$FDC = \frac{1}{n-1} \sum_{i=1}^{n} \left(\frac{y_i - \overline{y}}{\hat{\varepsilon}_y} \right) \left(\frac{d_i - \overline{d}}{\hat{\varepsilon}_d} \right). \tag{4}$$

where $\overline{y}$ and $\overline{d}$ are the mean fitness and the mean distance between x_0 and x_i, $\widehat{\varepsilon_y}$ and $\widehat{\varepsilon_d}$ are the sample standard deviation of the fitness and the distance, respectively. It is worth mentioning that FDC is invariant to shifts and rotations on the parameter space and the fitness values, because they are global isometries of the Euclidean space.

2.3 Differential Evolution Algorithm

In EA, there always are many different offspring generating operators, such as different crossover operators, different mutation operators and so on. Usually, different operators have the different performance on different types of optimization problems. Since DE has a variety of mutation strategies [18,19], we take it as an example in this paper. Let D is the dimension of problems, x_j^{U} and x_j^{L} are the upper and lower bounds of the constraint range of individual x_j, respectively. Then the minimization problem can be described as: $f_{\min}(x_1, x_2, \ldots, x_D)$, where $x_j^{\mathrm{L}} \leq x_j \leq x_j^{\mathrm{U}}, j = 1, 2, \ldots, D$. DE is committed to continuously improving the ability of populations to adapt to the external environment through personal communication, competition and iteration to achieve the goal of getting the best solution [20]. The flow chart of the standard DE is shown in Fig. 2.

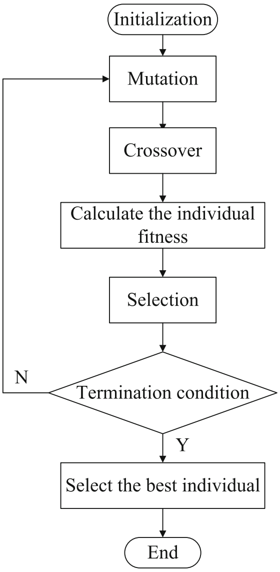

Fig. 2. The flow chart of the standard DE

(1) Initialization

We use Eq. (5) to generate initial individuals which is satisfying constraints in the D-dimensional space as the $0th$ generation population.

$$x_{i,j,0} = x_{i,j}^{\mathrm{L}} + rand \cdot \left(x_{i,j}^{\mathrm{U}} - x_{i,j}^{\mathrm{L}} \right). \tag{5}$$

where $i = 1, \ldots, NP; j = 1, \ldots, D$; $rand$ is a uniformly generated random number in [0,1]; NP is the size of the population.

(2) Mutation

The six most common used mutation strategies are listed from Eqs. (6) to (11).

DE/rand/1/bin:

$$v_{i,G} = x_{r1,G} + F \cdot (x_{r2,G} - x_{r3,G}). \tag{6}$$

DE/best/1/bin:

$$v_{i,G} = x_{best,G} + F \cdot (x_{r1,G} - x_{r2,G}). \tag{7}$$

DE/rand/2/bin:

$$v_{i,G} = x_{r1,G} + F \cdot [(x_{r2,G} - x_{r3,G}) + (x_{r4,G} - x_{r5,G})]. \tag{8}$$

DE/best/2/bin:

$$v_{i,G} = x_{best,G} + F \cdot [(x_{r1,G} - x_{r2,G}) + (x_{r3,G} - x_{r4,G})]. \tag{9}$$

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

$$v_{i,G} = x_{i,G} + F \cdot [(x_{r1,G} - x_{i,G}) + (x_{r2,G} - x_{r3,G})]. \tag{10}$$

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

$$v_{i,G} = x_{i,G} + F \cdot [(x_{best,G} - x_{i,G}) + (x_{r1,G} - x_{r2,G})]. \tag{11}$$

where $r1, r2, r3, r4, r5$ represent the random numbers between 1 and NP, which are different from each other and not equal to the number of target individual. $x_{best,G}$ symbolizes the optimal individual in the Gth generation population. $v_{i,G}$ stands for the individual after the mutation operation. F is the scaling factor that controls the amplification of the bias variable. The value of F is generally set to 0.5.

(3) Crossover

There are two common crossover means in the DE algorithm. The binomial crossover is usually preferred than exponential crossover [21,22], which is expressed as Eq. (12):

$$u_{i,G} = \begin{cases} v_{i,G} & \text{if } \ rand[0,1] \leq CR \ \text{ or } \ j = j_{rand} \\ x_{i,G} & \text{otherwise} \end{cases} . \tag{12}$$

where j_{rand} is a random number in $[1, 2, ..., D]$ to ensure that at least one dimension component of the intersecting individual will be different from the target individual. CR is called crossover probability which is generally recommended to 0.9.

(4) Selection

The selection is to determine if there are individuals in the parents who can become members of the next generation. The rules for selecting operation are as Eq. (13), where $f(\cdot)$ is the value of the objective function.

$$x_{i,G+1} = \begin{cases} u_{i,G} & \text{if } \ f(u_{i,G}) \leq f(x_{i,G}) \\ x_{i,G} & \text{otherwise} \end{cases} . \tag{13}$$

(5) Repeat steps (2)–(4), until the stopping criterion is satisfied.

3 Mutation Strategy Selection for Black-Box Optimization Problems Based on FLA

3.1 Overall Framework

Figure 3 depicts an overall framework of the mutation strategy selection task based on FLA, where the features of candidate function A are used as input portion, and the best strategy B is recommended as its output. This framework is composed of three related components.

Sampling: For the sake of generality, uniform sampling is used in this work to generate samples that can represent various states (ie, $P_1, \ldots, P_n$, which are presented in Fig. 3). For some features, global sampling (ie, BSR, KEE, etc.) is not applicable. Therefore, this paper performs local sampling and normalization to make these features more effective and persuasive.

Calculating the Features: It should be reminded that this paper is a preliminary study, so the benchmark functions are used for convenience. Since their structures are known, it is easy to get various characteristics of the problem to improve the performance of algorithms. What is more, the study of single objective optimization algorithms is the basis of more complex optimization algorithms [23–27].

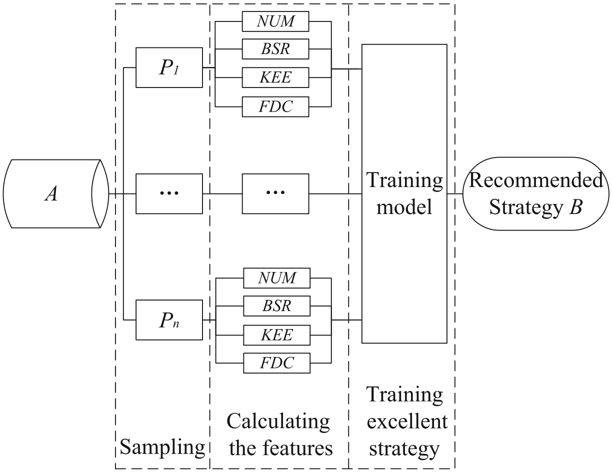

Fig. 3. Overall framework of the mutation strategy selection

In the CEC2005, the focus is on low and medium dimensional problems [23]. The benchmark set consists of 25 test functions which is divided in four groups: unimodal, basic multimodal, expanded multimodal, and hybrid composition. In the CEC2013 test set [24], the previously proposed composition functions are improved and additional test functions are included. In the CEC2014 [25], CEC2015 [26] and CEC2017 [27], Liang and Suganthan et al. developped benchmark problems with some novel features such as novel basic problems, composing test problems by extracting features dimension-wise from several problems, graded level of linkages, rotated trap problems, and so on.

It is worth noting that CEC2017, CEC2015 and CEC2014 with a total of 75 functions are treated as the black box optimization problems in this paper. The individual features computed for each function are used as the input of our model.

Training the Excellent Strategy: The benchmark functions are tentatived under the standard operating conditions in combination with six strategies of the standard DE algorithm, respectively. Furthermore, K-Nearest Neighbor (KNN) and Random Forest (RF) are used as learners for the training model of all functions with 4 features, respectively. The core idea of KNN [28] is that if most of the k nearest neighbors in the characteristical space belong to a certain category, the sample also belongs to this category and has the characteristics of the neighboring samples on this type. RF is a classifier with multiple decision trees whose output categories are determined by the mode of the multi-tree output class [29]. For each test function, we recommend the best mutation strategy based on the proposed strategy selection framework.

4 Experimental Analysis

4.1 Feature Values and Benchmark Functions

The population size of all benchmark functions is set to 100 while the sampling points of the feature is tune into 3000. Moreover, the search range is $[-100, 100]^D$ and the FEs budget is preset to $10^4 \times D$, where D is the problem dimension.

The mean and variance for each feature of 75 functions were obtained by running 20 times. The values are shown in Fig. 4. From the figure, we can see that each feature presents diversity among different functions, especially in complex functions. In the meanwhile, the value of variance for BSR and FDC are far less than others in the range of 0 to 0.4, so they are more stable than NUM and KEE.

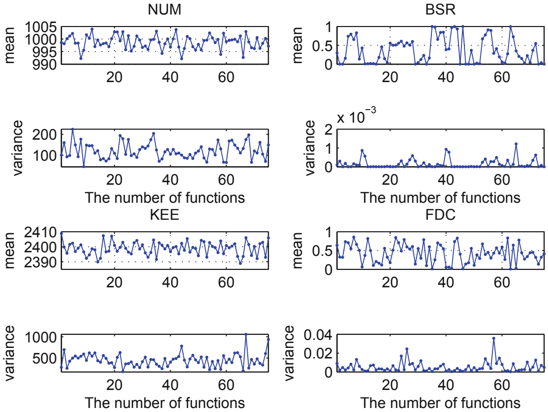

Fig. 4. Feature values for each function

In addition, in order to compare the significance of difference between various strategies, the assessment is performed using the Friedman non-parametric test method [30]. In that case, the parameters of standard DE are: $F = 0.5$, $CR = 0.9$. The standard DE with six strategies are run independently 51 times to find the standard deviation, respectively. When the average ranks obtained by each strategy in the Friedman test are the same small, we hold them to be equally good strategies. The judgments are concluded in Table 1, where the number in the even columns refers to the used mutation strategy corresponding to Sect. 2 and the number in parentheses refers to the location of the function in this benchmark problem.

As we can see from the table, for a given optimization problem, each function is not limited to an excellent strategy, such as f1, f2 in CEC2017. We put the same excellent strategies for the function as the different data in this situation (ie. F1–2, F3–4, etc.). We can also find that DE/rand/1/bin which is recorded as the first strategy has been proven to perform well on most functions.

4.2 Results and Discussion

The results of Table 1 which are obtained by performance evaluation can be considered as the gold standard for mutation strategy selection. We can evaluate the outcomes gained from the proposed framework at the same time. The evaluation criteria are as follows: For the test functions, if the excellent strategy selected based on the performance evaluation contains the strategy recommended by the

Table 1. The excellent strategy for benchmark functions

CEC2017	Excellent	CEC2014	Excellent	CEC2015	Excellent
F1–2 (f1)	1,4	F35–36 (f1)	1,4	F69–70(f1)	1,4
F3–4 (f2)	1,2	F37 (f2)	1	F71 (f2)	1
F5 (f3)	1	F38–39 (f3)	1,2	F72 (f3)	3
F6 (f4)	1	F40 (f4)	2	F73 (f4)	6
F7 (f5)	6	F41 (f5)	3	F74 (f5)	4
F8–9 (f6)	1,4	F42–43 (f6)	1,5	F75 (f6)	1
F10 (f7)	6	F44 (f7)	5	F76 (f7)	6
F11 (f8)	6	F45 (f8)	6	F77 (f8)	1
F12–13 (f9)	1,2	F46 (f9)	6	F78 (f9)	6
F14 (f10)	6	F47 (f10)	4	F79 (f10)	4
F15 (f11)	1	F48 (f11)	6	F80 (f11)	6
F16 (f12)	1	F49 (f12)	6	F81 (f12)	6
F17 (f13)	4	F50 (f13)	6	F82 (f13)	2
F18 (f14)	1	F51 (f14)	6	F83 (f14)	1
F19 (f15)	1	F52 (f15)	5	F84 (f15)	1
F20 (f16)	1	F53 (f16)	4		
F21 (f17)	1	F54 (f17)	1		
F22 (f18)	1	F55 (f18)	1		
F23 (f19)	1	F56 (f19)	6		
F24 (f20)	1	F57 (f20)	1		
F25 (f21)	6	F58 (f21)	1		
F26 (f22)	5	F59 (f22)	1		
F27 (f23)	5	F60–61 (f23)	1,2		
F28 (f24)	4	F62 (f24)	4		
F29 (f25)	1	F63 (f25)	5		
F30 (f26)	6	F64 (f26)	6		
F31 (f27)	2	F65 (f27)	6		
F32 (f28)	1	F66 (f28)	2		
F33 (f29)	1	F67 (f29)	1		
F34 (f30)	1	F68 (f30)	1		

proposed framework, the recommended strategy can be deemed as a correct answer; otherwise, it is a mistake.

On the basis of this evaluation method, the output accuracy is a statistical indicator in the proposed framework. It refers to the number of correct classifications divided by the total number of test functions. It is employed to KNN and RF to manifest which learner is more suitable for the suggested framework.

The Selection of the Number of Classifier Nodes: The proposed strategy selection framework is based on the appropriate number of classifier nodes and data set folds. Under the circumstance of CEC2015 as a testing set, we put up the nodes number from 1 to 10 for each classifier and then run the suggested framework in this research. The results are demonstrated in Fig. 5.

We show the impact of nodes number on the output accuracy of the classifier as a line graph. Experimental results prove that the value of RF is higher than KNN for each node from the figure. Moreover, $K = 4$ behaves better in the case of less consumption.

The Selection of the Number of Data Set Folds: After that, we set up M from 1 to 10 to verify the effect of folds number on the output accuracy on the basis of $K = 4$. The specific details are shown in Fig. 6.

It can be seen that, for RF, the classification accuracy is as high as 0.525 if $M = 3$. In the same time, the best value is up to 0.571 in KNN, which manifests that the KNN algorithm successfully satisfies the intrinsic mapping relationship of the proposed framework. However, it can also be seen that the values of KNN are more volatile than RF. Because the output accuracy of KNN is as low as 0 when M = 1, which is closely related to the working principle of KNN. At the same time, although the lowest value of RF at M = 9 is close to 0.1, it is still

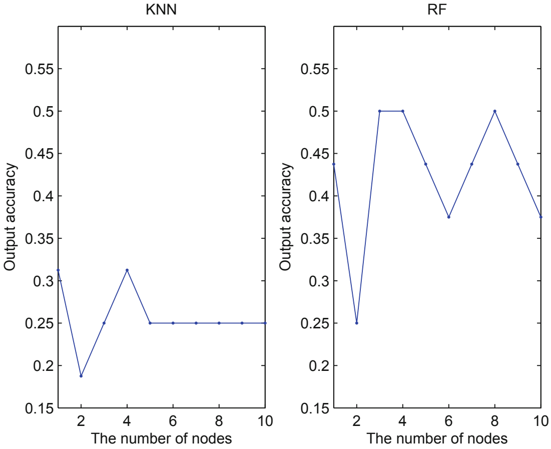

Fig. 5. The impact of nodes number on the output accuracy of each classifier

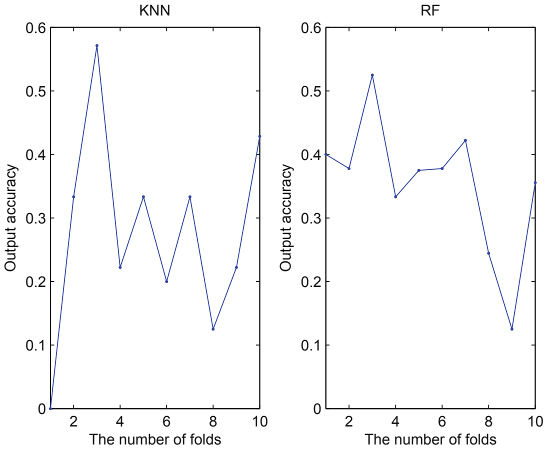

Fig. 6. The impact of folds number on the output accuracy of each classifier

Table 2. The effect of the classifier on the output accuracy

Classifier	Output accuracy
KNN	**0.571**
RF	0.525

stable overall. The final results are shown in Table 2. And the best value is bold. It must be admitted that the results are not as good as we expected, but this framework has proven to be effective.

5 Conclusion

As the complexity of optimization problems increases, there is an urgent need to develop learning-based methods to adaptively guide population evolution based on the multifaceted requirements and nature of practical problems. This paper proposes a mutation strategy selection based on fitness landscape analysis to meet the demand. The sample model are designed, where four features of the fitness landscape are used as inputs and the recommended strategy after training are applied as the output. Then, we use some classifiers to study the mapping relationship. Finally, experimental analyses show that the proposed framework can efficiently match excellent strategies to improve the intelligent solving ability of optimization problems.

In the future, we will conduct extended research based on the proposed framework. From the prospect of frame designers, further work can be focused on expanding the sampling size and adding verified sets to demonstrate effectiveness of the proposed work. From the perspective of these measures, the next step is to increase features of the fitness landscape to improve the classified accuracy.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (61922072, 61876169, 61673404, 61976237).

References

1. Stadler, P.F.: Fitness landscapes. In: Lässig, M., Valleriani, A. (eds.) Biological Evolution and Statistical Physics, pp. 183–204. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45692-9_10
2. Pitzer, E., Affenzeller, M.: A comprehensive survey on fitness landscape analysis. In: Fodor, J., Klempous, R., Suárez Araujo, C.P. (eds.) Recent Advances in Intelligent Engineering Systems, pp. 161–191. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-23229-9_8
3. Wang, M., Li, B., Zhang, G., et al.: Population evolvability: dynamic fitness landscape analysis for population-based metaheuristic algorithms. IEEE Trans. Evol. Comput. **22**, 550–563 (2018)
4. Borenstein, Y., Poli, R.: Information landscapes. In: GECCO 2005: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, pp. 1515–1522. ACM Press, New York (2005)
5. Borenstein, Y., Poli, R.: Information landscapes and problem hardness. In: GECCO 2005: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, pp. 1425–1431. ACM Press, New York (2005)
6. Borenstein, Y., Poli, R.: Decomposition of fitness functions in random heuristic search. In: Stephens, C.R., Toussaint, M., Whitley, D., Stadler, P.F. (eds.) Foundations of Genetic Algorithms, pp. 123–137. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73482-6_8
7. Verel, S., Collard, P., Clergue, M.: Where are bottlenecks in NK fitness landscapes? In: The 2003 Congress on Evolutionary Computation, pp. 273–280 (2003). https://doi.org/10.1109/CEC.2003.1299585
8. Vanneschi, L.: Theory and practice for efficient genetic programming. Ph.D. thesis, Faculty of Sciences, University of Lausanne, Switzerland (2004)
9. Vanneschi, L., Tomassini, M., Collard, P., Vérel, S.: Negative slope coefficient: a measure to characterize genetic programming fitness landscapes. In: Collet, P., Tomassini, M., Ebner, M., Gustafson, S., Ekárt, A. (eds.) EuroGP 2006. LNCS, vol. 3905, pp. 178–189. Springer, Heidelberg (2006). https://doi.org/10.1007/11729976_16
10. Lu, H., Shi, J., Fei, Z., et al.: Measures in the time and frequency domain for fitness landscape analysis of dynamic optimization problems. Soft Comput. **51**, 192–208 (2017)
11. Morgan, R., Gallagher, M.: Sampling techniques and distance metrics in high dimensional continuous landscape analysis: limitations and improvements. IEEE Trans. Evol. Comput. **18**(3), 456–461 (2013)
12. Li, W., Li, S., Chen, Z., et al.: Self-feedback differential evolution adapting to fitness landscape characteristics. Soft Comput. **23**(4), 1151–1163 (2019)
13. Morris, M.D., Mitchell, T.J.: Exploratory designs for computational experiments. J. Stat. Plann. Infer. **43**(3), 381–402 (1992)
14. Shen, L., He, J.: A mixed strategy for evolutionary programming based on local fitness landscape. In: 2010 IEEE Congress on Evolutionary Computation, Barcelona, Spain, pp. 1–8. IEEE (2010)

15. Munoz, M.A., Kirley, M., Halgamuge, S.K.: Landscape characterization of numerical optimization problems using biased scattered data. In: 2012 IEEE Congress on Evolutionary Computation, Brisbane, QLD, Australia, pp. 1–8. IEEE (2012)
16. Lu, H., Shi, J., Fei, Z., et al.: Analysis of the similarities and differences of job-based scheduling problems. Eur. J. Oper. Res. **270**(3), 809–825 (2018)
17. Munoz, M.A., Kirley, M., Smith-Miles, K.: Reliability of exploratory landscape analysis (2018). https://doi.org/10.13140/RG.2.2.23838.64327
18. Storn, R., Price, K.: Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical report, International Computer Sciences Institute, Berkeley, California, USA (1995)
19. Storn, R., Price, K.: Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341–359 (1997)
20. Otieno, F.A.O., Adeyemo, J.A., Abbass, H.A., et al.: Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces. Trends Appl. Sci. Res. **5**(1), 531–552 (2002)
21. Das, S., Suganthan, P.N.: Differential evolution: a survey of the state-of-the-art. IEEE Trans. Evol. Comput. **15**(1), 4–31 (2010)
22. Das, S., Mullick, S.S., Suganthan, P.N.: Recent advances in differential evolution - an updated survey. IEEE Trans. Evol. Comput. **27**, 1–30 (2016)
23. Suganthan, P.N., Hansen, N., Liang, J.J., et al.: Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Technical report, Nanyang Technological University, Singapore, KanGAL Report, IIT Kanpur, India (2005)
24. Liang, J.J., Qu, B.Y., Suganthan, P.N., et al.: Problem definitions and evaluation criteria for the CEC 2013 special session and competition on real-parameter optimization. Technical report, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore (2013)
25. Liang, J.J., Qu, B.Y., Suganthan, P.N.: Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Technical report, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore (2013)
26. Liang, J.J., Qu, B.Y., Suganthan, P.N., et al.: Problem definitions and evaluation criteria for the CEC 2015 competition on learning-based real-parameter single objective optimization. Technical report, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore (2014)
27. Wu, G., Mallipeddi, R., Suganthan, P.N.: Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization. Technical report, National University of Defense Technology, Changsha, China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore (2017)
28. Friedman, M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J. Am. Stat. Assoc. **32**(200), 675–701 (1937)
29. Altman, N.: An introduction to Kernel and nearest-neighbor nonparametric regression. Am. Stat. **46**(3), 175–185 (1992)
30. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–23 (2001)