A Ladder Spherical Evolution Search Algorithm

Haichuan YANG†, Nonmember, Shangge GAO††, Member, Rong-Long WANG†††, and Yuki TODO††††, Nonmembers

SUMMARY In 2019, a completely new algorithm, spherical evolution (SE), was proposed. The brand new search style in SE has been proved to have a strong search capability. In order to take advantage of SE, we propose a novel method called the ladder descent (LD) method to improve the SE population update strategy and thereby propose a ladder spherical evolution search (LSE) algorithm. With the number of iterations increasing, the range of parent individuals eligible to produce offspring gradually changes from the entire population to the current optimal individual, thereby enhancing the convergence ability of the algorithm. Experiment results on IEEE CEC2017 benchmark functions indicate the effectiveness of LSE.

key words: spherical evolution, ladder descent, population update

1. Introduction

In the last two decades, complex optimization problems in the real world have become the focus of research. To solve these problems, various nature-inspired meta-heuristic algorithms have been constantly proposed [1]. These algorithms can be classified into three main classes: evolutionary, physics-based and swarm intelligence (SI) [2] algorithms.

Evolutionary algorithms (EA) are usually inspired by biological evolution, such as reproduction, mutation, recombination and selection. The most popular type of EA is the genetic algorithm (GA) [3], [4]. Proposed by Holland in 1992, GA simulates population evolution in nature. Differential evolution [5]–[7] is a powerful algorithm as well, which is based on vector differences and therefore primarily applicable to numerical optimization problems.

Physics-based algorithms typically mimic physical rules. Gravitational search algorithm (GSA) [8]–[10] is one of the physics-based algorithms. The search agents of GSA are a collection of masses that interact with each other based on the Newtonian gravity and the laws of motion.

SI is the collective behavior of decentralized and self-organized systems which may be natural or artificial, and the SI algorithm is obtained by simulating these behaviors. The search agents of the SI algorithm navigate using the simulated collective and social intelligence of creatures. Some of the most popular SI algorithms are ant colony optimization [11] and artificial bee colony [12], [13].

SE [14] is a typical EA with a spherical search style. It’s widely accepted that an excellent meta-heuristic algorithm is expected to achieve a good balance between exploration and exploitation [15]. In order to strengthen the exploitation ability of SE, we innovatively propose a new population update strategy for SE. As the number of iterations increases, the number of the parent individuals participating in the iteration decreases in a step-down manner. Individuals who have the opportunity to produce offspring are reduced from all individuals in the population to the current optimal one in the population. This new population generation method, namely Ladder Descent (LD) method, enables SE to concentrate the search around the promising solutions, thus accelerating its convergence and obtaining better solutions.

The main contributions of this paper can be summarized as follows:

1) To our best knowledge, we are the first to improve the update strategy of SE population, which not only enhances the optimization ability of SE, but also gives more insights into the search dynamics of SE.

2) Experimental results based on thirty IEEE CEC2017 benchmark function instances show the superiority of the proposed method in comparison with the original one. The results can increase the interests in research on meta-heuristic algorithms, thereby promoting the development of related fields.

2. Spherical Evolution

Firstly, SE is briefly introduced. There are seven variants of SE. In this study, we choose SE04 [14] as it performs the best among all variants. According to Tang [14], the search operators of metaheuristic algorithms have some common characteristics or patterns, which can be represented as:

\[
X_{i,j}^{new} = X_{i,j}^{old} + \sum_{k=1}^{n} SS \left( C_{i,j}^k + D_{i,j}^k \right) \tag{1}
\]

where \(X^{new}, X^{old}, C\) and \(D\) denote four solution sets by four populations and each population is represented with \(\text{popsize}\)
rows and \( \text{Dim} \) columns. \( n \) denotes the number of updating units; \( SS \left( C^k, D^k \right) \) represents the updating units in the search operator and decides the search style.

The update formula of a single individual in the matrix of SE can be represented as:

\[
X^\text{new}_j = X^\text{old}_j + \sum_{k=1}^{n} SS \left( C^k_j + D^k_j \right)
\]

\( j = 1, 2, \ldots, \text{Dim} \)

where \( X^\text{old}_{R,j} \) is an individual randomly selected from the parent matrix. \( R \) represents the serial number of the individual, and it is a random integer between 1 and \( \text{popsize} \). This way of selecting individual parents focuses on ensuring the diversity of the population in the iterative process.

The updating units of SE can be represented as Eqs. (3)–(7) in high-dimensions, two-dimensions and one-dimension, respectively [16].

\[
SS_{\geq 3} \left( C_{i,j}, D_{i,j} \right) = SF() \cdot \left\| C_{i,*} - D_{i,*} \right\|_2 \cdot \prod_{k=j}^{\text{dim}-1} \sin \left( \theta_i \right), j = 1
\]

\[
SS_{\geq 3} \left( C_{i,j}, D_{i,j} \right) = SF() \cdot \left\| C_{i,*} - D_{i,*} \right\|_2 \cdot \cos \left( \theta_{j-1} \right) \cdot \prod_{k=j}^{\text{dim}-1} \sin \left( \theta_i \right), 1 < j \leq \text{dim} - 1
\]

\[
SS_{\leq 2} \left( C_{i,j}, D_{i,j} \right) = SF() \cdot \left\| C_{i,*} - D_{i,*} \right\|_2 \cdot \cos \left( \theta_{j-1} \right), j = \text{dim}
\]

\[
SS_{\leq 2} \left( C_{i,j}, D_{i,j} \right) = SF() \cdot \left\| C_{i,*} - D_{i,*} \right\|_2 \cdot \sin \left( \theta \right)
\]

\[
SS_{\leq 1} \left( C_{i,j}, D_{i,j} \right) = SF() \cdot \left[ C_{i,j} - D_{i,j} \right] \cdot \cos \left( \theta \right)
\]

where \( SF() \) denotes a scale factor that can appropriately adjust the radius length, and \( \left\| C_{i,*} - D_{i,*} \right\|_2 \) denotes the radius of sphere computed by Euclidean norm in high dimensions. \( \theta \) denotes the angle between \( C_{i,*} \) and \( D_{i,*} \), and it is generated by a random number of uniform distribution between \([0, 2\pi]\). The dimension size is denoted as \( \text{dim} \) (\( \text{dim} \geq 3 \)).

The main characteristics of spherical search style are the larger search range and undirected search trajectory due to changes in \( \| C_{i,*} - D_{i,*} \|_2 \) and \( \theta \). These characteristics give SE a stronger search ability and enable it to avoid local optimal. In two-dimensional space, the search space of the spherical search style is a whole region of the circle when the angle is adjusted from 0 to \( 2\pi \). The length of the radius is adjusted from 0 to \( |D_i - C_i| \).

3. A Ladder Spherical Evolution Search Algorithm

Based on the study of the SE, a novel LSE algorithm is proposed in this paper. LD changes the way of selecting individuals in the parent population, which not only ensure the diversity of the population in the early stage of the iterative process, but also enhances the convergence ability of the algorithm in the latter of the iterative process. The operators of LD can be represented as:

\[
X^\text{new}_j = X^\text{old}_{A,j} + \sum_{k=1}^{n} SS \left( C^k_j + D^k_j \right)
\]

\( j = 1, 2, \ldots, \text{Dim} \)

where \( X^\text{old}_{A,j} \) denotes the parent individual who is eligible to participate in the iteration. We first numbered all the individuals in the population in the order of fitness value in a descending order. Then we use \( A \) to screen the individuals in the parent population to generate \( X^\text{old}_{A,j} \). The symbol \( A \) represents the serial number of the individual, and it is a random integer between 1 and \( P \). For example, when the value of \( A \) is 1, \( X^\text{old}_{A,j} \) represents the parent individual ranked first in the population, that is, the parent individual with the best fitness value. \( P \) is an upper limit, and its value decreases stepwise as the number of iterations increases. It can be represented as:

Fig. 1 Different population update methods of SE and LSE.
where \( P = [(1 - \frac{it}{maxiter}) \cdot popsize] \) (9)

where \( it \) represents the current number of iterations, and \( maxiter \) is the maximum number of iterations.

The difference of population update strategy of the original SE and LSE are illustrated in Fig. 1, where the red dot represents the parent individual eligible to produce offspring. \( X_{a,j} \) represents the individual in the population, \( a \) is the serial number of the individual, and the smaller the \( a \), the closer to the current optimal. We can find that during the iteration process of SE, any parent individual has the opportunity to participate in the iteration, thereby generating a diverse of offspring individuals. Different from SE, as the number of iterations increases, the range of parent individuals in LSE who are eligible to produce offspring gradually narrows to the current optimal solution, which thus makes the search concentrate on more promising search regions.

4. Experimental Results

To verify the performance of the proposed LSE, 30 benchmark functions of IEEE CEC2017 composed of unimodal functions (F1–F3), simple multimodal functions (F4–F10), hybrid functions (F11–F20) and composition functions (F21–F30) are applied to the experiment. We tested all functions with the max iteration number of 15000, a population size of 20, a dimension of 30 and each function was run for 30 times.

The experiment results on IEEE CEC2017 are summarized in Table 1 where the highlighted ones are the best. For each function, the symbol + (i.e., win) which means that LSE performs significantly better (i.e., the obtained \( p \)-value is smaller than 0.05) than SE on the current function via the Wilcoxon rank-sum test at a significant level \( \alpha = 0.05 \) [17]. Similarly, symbol - (i.e., lose) means that the \( p \)-value of the comparative algorithm is less than or equal to 0.05, suggesting that the LSE is worse than SE on the function. Symbol ~ (i.e. tie) denotes that the \( p \)-values of two algorithms are greater than 0.05, indicating that there is no significant difference on one function between both algorithms. Totally, W/T/L shows the number of win, tie and lose between the two algorithms according to statistical analysis. Its value of 18/12/0 suggests that LSE performs better than SE on 18 functions out of all 30 tested functions.

Furthermore, the analysis results of the Wilcoxon matched-pairs signed-rank test are also presented in Table 2 by only considering the mean value of each function. We can find that the exact \( p \)-values is less than 0.05, which means that LSE performs significantly better than SE on IEEE CEC2017 test suit.

Figure 2 illustrates convergence graphs for three typical functions F10, F12, and F30. The abscissa and ordinate represent the number of iteration and the average of best-so-far

![Fig. 2](image-url)
solutions, respectively. From these results, we can find that LSE converges faster than SE. Figure 3 depicts box-and-whisker graphs. The top and bottom line in the figure represents the maximum and minimum value in fitness except for outliers, respectively. The upper and lower borderline of the rectangle represents the upper and lower quantiles, respectively. The line in the middle of a rectangle represents the median. The results show that the values of all types of lines of LSE are smaller than SE. These characteristics reveal that LSE can not only converge quickly and get rid of local optimum, but also provide the best solutions.

Above all, we can conclude that the experimental results demonstrate that the incorporation of the LD method into SE is effective.

5. Conclusions

In this paper, we propose a LSE algorithm, in which the LD method is added to the SE, thus improving the population update method of the SE population. To assess the performance of LSE, we compare it with original SE on benchmark functions in IEEE CEC2017. Experimental results suggest that the proposed LSE is efficient. The results give a guidance for the related research in the evolution of the population update method in evolutionary algorithms.

References

[1] X. Yang. Nature-inspired metaheuristic algorithms, Luniver Press, 2010.
[2] G. Beni and J. Wang. “Swarm intelligence in cellular robotic systems,” Robots and Biological Systems: Towards a New Bionics?, NATO ASI, vol.102, pp.703–712, Springer, 1993.
[3] J.H. Holland. “Genetic algorithms,” Scientific American, vol.267, no.1, pp.66–73, 1992.
[4] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang. “Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction,” IEEE Trans. Neural Netw. Learn. Syst., vol.30, no.2, pp.601–614, 2019.
[5] R. Storn and K. Price. “Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces,” Journal of Global Optimization, vol.11, no.4, pp.341–359, 1997.
[6] S. Gao, Y. Wang, J. Wang, and J. Cheng. “Understanding differential evolution: A Poisson law derived from population interaction network,” Journal of Computational Science, vol.21, pp.140–149, 2017.
[7] S. Gao, Y. Yu, Y. Wang, J. Wang, J. Cheng, and M. Zhou. “Chaotic local search-based differential evolution algorithms for optimization,” IEEE Trans. Syst. Man Cybern. Syst., 2019. doi:10.1109/TSMC.2019.2956121.
[8] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi. “GSA: A gravitational search algorithm,” Information Sciences, vol.179, no.13, pp.2232–2248, 2009.
[9] Y. Wang, Y. Yu, S. Gao, H. Pan, and G. Yang. “A hierarchical gravitational search algorithm with an effective gravitational constant,” Swarm and Evolutionary Computation, vol.46, pp.118–139, 2019.
[10] Z. Lei, S. Gao, S. Gupta, J. Cheng, and G. Yang. “An aggregative learning gravitational search algorithm with self-adaptive gravitational constants,” Expert Systems with Applications, vol.152, 113396, 2020.
[11] S. Gao, Y. Wang, J. Cheng, Y. Inazumi, and Z. Ting. “Ant colony optimization with clustering for solving the dynamic location routing problem,” Applied Mathematics and Computation, vol.285, pp.149–173, 2016.
[12] D. Karaboga and B. Basturk. “A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm,” Journal of Global Optimization, vol.39, no.3, pp.459–471, 2007.
[13] J. Ji, S. Song, C. Tang, S. Gao, Z. Tang, and Y. Todo. “An artificial bee colony algorithm search guided by scale-free networks,” Information Sciences, vol.473, pp.142–165, 2019.
[14] D. Tang. “Spherical evolution for solving continuous optimization problems,” Applied Soft Computing, vol.81, 105499, 2019.
[15] B. Morales-Castañeda, D. Zaldívar, E. Cuevas, F. Fausto, and A. Rodríguez. “A better balance in metaheuristic algorithms: Does it exist?,” Swarm and Evolutionary Computation, vol.54, 100671, 2020.
[16] D. Mustard. “Numerical integration over the n-dimensional spherical shell,” Mathematics of Computation, vol.18, no.88, pp.578–589, 1964.
[17] J. Carrasco, S. García, M.M. Rueda, S. Das, and F. Herrera. “Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review,” Swarm and Evolutionary Computation, vol.54, 100665, 2020.