Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **6**, November **2018**

Symbiotic Organisms Search algorithm based on Cloud Model Elite Search

Yan-Jiao Wang, Zhuang Ma

School of Electrical Engineering
Northeast Electric Power University
Jilin 132000, China
wangyanjiao1028@126.com|867382014@qq.com

Received April, 2018; revised August, 2018

Abstract. *In order to solve the problem of slow convergence speed and low convergence precision in Symbiotic Organisms Search (SOS) algorithm, in this paper, a Symbiotic Organisms Search algorithm based on Cloud Model Elite Search (CESOS) is proposed. On the basis of the SOS, CESOS introduces an Elite individual to conduct a deep search and adds a disturbing individual to guarantee the diversity of the population in the Commensalism phase. This strategy balances the development and exploration ability of the algorithm effectively. In the Parasitism phase, CESOS uses the Cloud Model to preserve evolutionary information of the current individual. This measure increases the diversity of the population and ensures the evolutionary direction of the individual. The two improved parts cooperate with each other to enhance the convergence speed and accuracy of the algorithm. Finally, the simulation results of 8 functions from CEC2005 and 15 functions from CEC2013, and the result of Friedman's Test and Holm's test show that, compared with Other optimization algorithms, the CESOS proposed in this paper has a good performance.*

Keywords: SOS; Elite individual; Cloud Model; Convergence speed and accuracy.

1. **Introduction.** Swarm intelligence optimization algorithms, such as Genetic Algorithm(GA), Particle Swarm Optimization(PSO) and Differential Evolution(DE) and so on, all the algorithms start with a set of initial variables and then perform the evolution of the variables until the optimal solution is found, which are widely used in Network Optimization, Engineering Applications, Artificial Intelligence and Medical Traffic etc. due to their advantages of high efficiency and simple structure. In order to solve the practical problems better, some algorithms with excellent performance have been proposed.

In 2014, a new swarm intelligence optimization algorithm – Symbiotic Organisms Search (SOS) based on the survival and evolutionary behavior of organisms in the natural world was proposed by Min-yuan Cheng et al[1]. In SOS, the relationship between organisms is divided into three types:Mutualism, Commensalism and Parasitism, the individuals of the population find the optimal solution through the evolution of the three stages. And document [1] compares SOS algorithm with Genetic Algorithm(GA), and Particle Swarm Optimization (PSO) through 26 standard test functions. The results show that SOS algorithm has obvious advantages in terms of convergence speed and accuracy. However, similar to the previously proposed swarm intelligence optimization algorithms, SOS also falls into local optimum easily when dealing with high-dimensional and multimodal complex problems. And the slow convergence speed in the middle and late stage of search are also the inevitable problems for SOS.

In order to further solve the problems in SOS and improve the performance of the algorithm. Some scholars have made a preliminary improvement, such as Hu Zhou et al. proposed a Multi Strategy Adaptive Symbiotic Organisms Search (MASSOS) [2], different strategies are taken for different subpopulations, which make the population evolve in good directions gradually. Although it plays a certain role in population diversity, the convergence performance is still to be improved. Nama S et al. proposed improved Symbiotic Organisms Search algorithm for solving unconstrained function optimization [3], on the basis of the original algorithm, the convergence speed of the algorithm is improved by increasing the reflection parameters and the predation phase, but at the same time, the complexity of the algorithm is also increased. Yan-jiao Wang et al. proposed a Symbiotic Organisms Search algorithm based on Rotating Learning Strategy[4], every dimension of the selected individual executes rotating learning, which enhances the ability of the algorithm to jump out of the local optimum. However, the algorithm is relatively complex to implement. In 2018, Gong S et al. proposed an opposition-based Symbiotic Organisms Search with a catastrophe phase algorithm, this method increases the diversity of the population, but it has great influence on the convergence speed of the algorithm[5]. At the same year, Prayogo D et al. proposed an improved version of the SOS algorithm named "Enhanced Symbiotic Organisms Search" for global numerical optimization. The algorithm applies the new search formula in the parasitism phase to produce a better searching capability. But in the later searching stage, the differences among the individuals get small, the population diversity can not be maintained with the parasitism phase changed alone, the algorithm still easily falls into local optimal[6].

To improve the optimization performance of SOS algorithm, the Symbiotic Organisms Search algorithm based on Cloud Model Elite Search (CESOS) is proposed in this paper. The improvements of CESOS and experimental methods are introduced as follows.

(1). In Commensalism phase, CESOS executes stretching operations by introducing an elite individual to guide the evolution of the current individual. Meanwhile in order to prevent the algorithm from falling into local optimum, another individual acts as a disturbing vector is also introduced in this phase. This will speed up the convergence of the algorithm and ensure the diversity of the population.

(2). In Parasitism phase, CESOS takes advantage of the stability characteristics of Cloud Theory Model to produce a parasitic individual, which will contain the excellent evolutionary information of the original individual, this measure increases the diversity of the population and avoids blind search of individual effectively in the process of evolution to ensure the evolution direction of the individual.

(3). As for the experiment methods, firstly, determine the number of individuals in the population and the evaluation times based on the experiments. Then, two experiments are carried out to verify the performance of the proposed algorithm. One is to compare the convergence speed and accuracy among CESOS , SOS and MASSOS on the functions from CEC2005. Another is that compare CESOS with GA with three-parent crossover(GA-TPC)[7], A Self-adaptive Heterogeneous PSO(f_k-PSO)[8], Self-adaptive Differential Evolution(SaDE)[9] and A new different evolution algorithm based on SSO algorithm and covariance matrix learning (SCDE)[10]on F1-F15 from CEC2013. Finally, Friedman's Test and Holm's test [11] are used to analyze the experimental data to show the differences among these algorithms. The experimental results show that the performance of the CESOS algorithm is much better than the existing improved SOS algorithm, and it is comparable to other excellent evolutionary algorithms.

The remaining parts of this paper are arranged as follows: Section 2 is the introduction of the SOS; Section 3 introduces the improved method of SOS and the implementation steps of CESOS; Section 4 is the comparative analysis of the experiment; Section 5 is the summary of the full text, the last two parts are the acknowledgement and the reference.

2. **Standard Symbiotic Organisms Search algorithm.**

SOS is a new algorithm, which can be divided into mutualism, commensalism and parasitism. The main evolutionary steps are as follows.

(1) Initialization population

The relevant parameters need to be set at the beginning of the algorithm, it is assumed that the number of population is: N; the upper and lower bounds of the search space are: U and L; the dimension of the individual is: D; the maximum number of iterations is: max_iter; then generate the initial solution: X according to formula (1):

$$X_i = L + rand(1, D) \times (U - L) \tag{1}$$

Where X_i represents the i-th $(i = 1, 2, 3, ..., N)$ individual in the population.

(2) Mutualism phase

In this stage, a biological individual X_i interacts with $X_j (i, j \in \{1, 2...N\}, j \neq i)$, which is selected randomly from the population, both of them promote their development under the guidance of the current optimal individual, then produce new individuals X_{inew} and X_{jnew}, the specific updating formula is shown below.

$$\begin{cases} X_{inew} = X_i + rand(0, 1) \times (X_{best} - MV \times BF_1) \\ X_{jnew} = X_j + rand(0, 1) \times (X_{best} - MV \times BF_2) \end{cases} \tag{2}$$

In the formula: rand (0,1) is a random number between [0,1], X_{best} is the optimal individual currently. $MV = (X_i + X_j)/2$ is the mutual benefit vector which represents the relationship between X_i and X_j, BF_1 and BF_2 are the benefit factors, their values are randomly determined either 1 or 2. X_{inew} is the updated result of X_i according to formula (2). Preserve the one with better fitness value between them and so do X_{jnew} and X_j.

The formula (2) shows that, X_i and X_j learn from the optimal individual and move towards the optimal position under the guidance of it, and achieve the global optimum gradually.

(3) Commensalism phase

The individual in the population carries out the Commensalism phase according to the formula (3):

$$X_{inew} = X_i + rand(-1, 1) \times (X_{best} - X_j) \tag{3}$$

In the formula, rand (-1,1) is a random number between [-1,1], X_{best} is the optimal individual currently. $X_j (j \in \{1, 2...N\}, j \neq i)$ is another individual, which is selected randomly from the current population.

Commensalism phase is similar to the Mutualism phase, the updated individual interacts with another individual, which is selected randomly from the population. However, what is different from Mutualism phase is that the interaction between X_i and X_j only benefits X_i, while there is neither benefit nor damage to the development of X_j.

(4) Parasitism phase

At this phase, a new parasitic individual named "Parasite_Vector" is produced by modifying one or several elements of X_i randomly, then select an individual $X_j (j \neq i)$ named "host" randomly from the current population. Compare the fitness values between "Parasite_Vector" and "host", if the fitness value of "Parasite_Vector" is better, the "host" will be replaced by "Parasite_Vector". Otherwise, the "host" will be considered as an immune individual to be retained.

3. Symbiotic Organisms Search algorithm based on Cloud Model Elite Search.

The SOS algorithm also has problems such as falling into local optimal easily and poor convergence precision. The root of the deficiencies above is found through in-depth study and the CESOS is proposed in this paper, which improves the convergence speed and accuracy of the algorithm. The details are as follows.

3.1. Commensalism phase based on Elite Search.

In the standard SOS algorithm, the updated individual evolves based on itself and absorbs benefits from another individual under the guidance of the current optimal individual in Commensalism phase. There is an obvious defect in this update strategy: As shown in Formula (3), the updated individual takes itself as the base vector to achieve the purpose of evolution by interacting the information between the social part $rand(-1,1) \times (X_{best} - X_j)$ and the best individual. This updating method is helpful to the evolution of the individual, but due to the lack of cognitive part of learning to other individuals, which ignores the effect of local search information on the algorithm. Without the help of other mechanisms, it is bound to fall into local optimum.

Due to the elite individual carries excellent evolution information, if use the elite individual to be the base vector that will improve the convergence speed of the algorithm. However, convergence too fast may also lead individuals to gather around the elite individual and will make the algorithm falls into local optimum easily. Consider that different individuals carry some local information, the introduction of these information is conducive to the maintenance of the diversity, which will effectively balance the development and exploration abilities of the algorithm and be beneficial for the algorithm to converge to the global optimum quickly. So in CESOS, an elite individual X_{best} and a disturbing individual $X_k (k \neq i \neq j)$ are introduced into the algorithm to solve the problems above respectively. X_{best} is the base vector used to make the algorithm search faster around the elite individual, X_k can provide the local search information, which can be used to guarantee the diversity of the population and prevent the algorithm from falling into local optimum. The specific updating strategies are shown in formula (4)-(5).

$$X_{inew} = X_{best} + c \times w \times (X_j - X_i) + c \times w \times (X_k - X_i) \tag{4}$$

$$c = \frac{Fit(i) - Fitbest}{Fitave - Fitbest - \alpha} \tag{5}$$

In formula (4)-(5), "$Fit(i)$" is the fitness value of X_i , "*Fitbest*" is the fitness value of the elite individual , "*Fitave*" represents the average fitness value of the current population. "α" is a positive number closed to zero to ensure the denominator is practical. The Stretching Factor "C" is a scalar which represents the difference between X_i and X_{best}. The larger the value of "C", the greater the influence of the updating formula. "w" is a random number between [-1,1].

Compared with formula(3), formula (4) allows the elite individual to participate in evolution as the basic vector directly will introduce more excellent evolution information and enable the evolutionary individual to search in a fine area. Under the leadership of elite individual, the convergence rate of the algorithm will improved greatly. At the same time, the introduction of the disturbing individual X_k brings more local search information and allows other areas in the search space to be explored, which makes the algorithm no longer search around the elite individual simply, so the diversity of the population can be guaranteed effectively in the whole evolutionary process of the algorithm. As a result, under the regulation of the elite individual X_{best} and disturbing individual X_k, the convergence speed is improved and the diversity of population is guaranteed, the performance of the algorithm has improved significantly. In addition, due to the evolutionary information carried by the disturbing individual iteself, learn from it has no harm to the convergence speed of the algorithm.

3.2. **Parasitism phase based on Cloud Model.** In the Parasitism phase of the SOS algorithm, select some dimensions randomly and replace them with random values in the search range, which is aimed at updating the individual as well as supplementing the diversity of the population. However, considering the current individual has some good evolutionary information, this blind random search is difficult to optimize the individual and preserve the original evolutionary information and can not supplement the individual diversity as expected. The SOS algorithm still falls into the local optimal easily.

A lot of experiments confirm that the characteristics of cloud model proposed by academician De-yi Li has a stable tendency that the generated cloud droplets change around their own positions with a higher probability and deviate those positions with a smaller probability[12]. Therefore, compared with replacing the original dimensions by the random numbers in Parasitism phase, generate the individual by the cloud model is more likely to achieve the same goal of maintaining population diversity, more importantly, the blindness of search is reduced accompanied with the preservation of the evolutionary direction.

The characteristics of the concept expressed by the cloud model can be described by the digital features of the cloud, which mainly contain Expectation (E_x), Entropy (E_n) and Hyper entropy (H_e). E_x determines the central position of the estimated data and reflects the inheritance and stability of the cloud. E_n reflects the degree of deviation between the cloud droplet and the center position, and represents the accepted range of cloud droplet in terms of the center position. H_e reflects the thickness of the cloud droplet, which is a measure of the uncertainty ofE_n. E_n and H_e reflect the randomness and fuzziness of the cloud droplet.

E_n, E_x and H_e are the interfaces between the cloud model and the outside, which are set artificially according to the actual situation. Produce the cloud droplet according to formula (6)-(7) based on the set E_n, E_x and H_e.

$$E_n^* = G(E_n, H_e) \tag{6}$$

$$X_i = G(E_x, E_n^*) \tag{7}$$

In formula (6), E_n^* is a normal random number generated by expected value E_n and the standard deviation H_e. In formula (7), X_i is a normal random number generated by expected value E_x and the standard deviation E_n^*. The random number generated by a cloud model is used to replace the element of the original individual, then produce a "Parasite_Vector". The parasitic individual generated by the Cloud Model preserves the evolution information of the original individual, and this way of generating an parasitic individual makes the "Parasite_Vector" still evolve towards the evolutionary direction of the original individual, so it can avoid the blind search. This keeps the diversity of the population and improves the convergence speed of the algorithm.

3.3. **Algorithm implementation steps.** The detailed process of CESOS is described as follows:

Step 1: Set the relevant parameters, generate primitive population according to formula (1).

Step 2: Calculate the fitness of the individuals in the population and determine the optimal individual X_{best}.

Step 3: The individual enters "Mutualism phase". Determine the retention of the new and the old individuals according to the formula (2).

Step 4: The individual enters "Commensalism phase" after "Mutualism phase", formula (4) and (5) replace the formula (3) as the updating formula for this stage, and move on to the next step.

Step 5: Set $E_n = 0.5$, $H_e = 0.05$, select an element $(X_i(k))$ from X_i as E_x. Generate a new element by formula (6)-(7) to replace $X_i(k)$ to generate the "Parasite_Vector". Compare the fitness value of the "Parasite_Vector" and "host" and update the individual.

Step 6: Execute the next step if all of the individuals have been updated, otherwise turn back to Step 2.

Step 7: If the iteration up to the upper limit or the fitness value satisfies the terminal condition, the algorithm stops, otherwise turn back to Step 2.

4. **Experiment and analysis.**

To verify the performance of the proposed CESOS algorithm in this paper, a series of experiments are carried out. All the experiments work on Intel (R) Core (TM) i5-3230M, 4G RAM, 2.60GHZ, Windows 8 and MatlabR2010b.

To make the experiment more representative and fully verify the performance of the algorithm proposed in this paper. First, check the influence of the parameter settings on the algorithm. Then, the contrastive experiment is divided into two parts. One is the comparison with the series of SOS on the functions from CEC2005, another is that compare with other algorithms on the functions from CEC2013. Finally, use Friedman's Test to check the experimental data to examine the differences among these algorithms.

4.1. **The influence of the parameters.**

To verify the impact of parameter settings on CESOS, select 8 test functions from CEC2005 (Include both Multimodal Functions and Unimodal Functions. Schaffer and Rastrigin are Multimodal Functions and the others are Unimodal Functions. Multimodal Functions with many wave valleys and peaks are more difficult to optimize compared with the Unimodal Functions. The optimal value of all functions is 0. These functions are shown in Table 1.) and use CESOS to optimize these functions in the case of changing the number of individuals in the population and the evaluation times respectively.

4.1.1. *The Test of the Number of Individuals in the Population.*

Test the performance of CESOS algorithm by changing the number of individuals in the population. For Schaffer and Rastrigin, the number of iterations is set to 50, the iterations of other functions are all set to 100. Each population runs 30 times independently, and calculates the Mean Value of the convergent results and Mean Time. The experimental results are shown in Table 2.

As is shown from table 2: for Schaffer and Rastrigin, under the same number of iterations, the algorithm can converge to the optimal value when the number of individuals in the population reaches 50. For other functions with the increase of the number of individuals, the convergence accuracy is also improving. However, when the number of individuals reaches 50, there is no longer any significant increase in the convergence accuracy or even down. This is because when the individual reaches a certain amount, some individuals will not get complete evolution at a specific number of iterations. At the same time, with the increase of individuals, the time spent on algorithm operation is also increasing. The number of the individuals should be set according to specific circumstances.

4.1.2. *The Test of the Number of Evaluations.*

To verify the influence of the number of evaluations on the algorithm, test the results of CESOS in optimizing these functions under the circumstance of changing the evaluation times. The number of individuals is

Table 1. Benchmark functions

Function	**Formulation**	**D**	**Range**
Schaffer	$f(x) = 0.5 + \frac{\sin^2(\sqrt{x_1^2+x_2^2})-0.5}{(1+0.001(x_1^2+x_2^2))^2}$	2	[-100,100]
Rastrigin	$f(x) = \sum\limits_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	30	[-5.12,5.12]
Elliptic	$f(x) = \sum\limits_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} x^2$	30	[-100,100]
Sphere	$f(x) = \sum\limits_{i=1}^{D} x_i^2$	30	[-100,100]
SumSquares	$f(x) = \sum\limits_{i=1}^{D} i x_i^2$	30	[-10,10]
Zakharov	$f(x) = \sum\limits_{i=1}^{D} x_i^2 + (\sum\limits_{i=1}^{D} 0.5 i x_i)^2 + (\sum\limits_{i=1}^{D} 0.5 i x_i)^4$	30	[-5,10]
Schwefel 1.2	$f(x) = \sum\limits_{i=1}^{D} (\sum\limits_{j=1}^{D} x_j)^2$	30	[-100,100]
Schwefel 2.22	$f(x) = \sum\limits_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	30	[-10,10]

Table 2. The influence of the number of individuals on the algorithm

Fun / **Pop**	**Mean Value (Mean Time)**						
	10	**20**	**30**	**40**	**50**	**60**	**70**
Schaffer	1.0841e-007 (0.4716s)	6.3187e-010 (0.5094s)	9.6467e-013 (0.5231s)	2.6367e-014 (0.5725s)	0 (0.6660s)	/ /	/ /
Rastrigin	1.3979e-011 (0.4467s)	8.7219e-012 (0.5103s)	1.7765e-014 (0.5789s)	5.7257e-015 (0.6415s)	0 (0.7428s)	/ /	/ /
Elliptic	1.9841e-142 (0.8274s)	4.7958e-223 (1.3636s)	6.0160e-248 (2.0419s)	5.8274e-254 (2.6976s)	2.8839e-262 (3.3678s)	3.8154e-264 (4.6900s)	8.8555e-268 (5.8432s)
Sphere	3.3173e-149 (0.7876s)	2.3264e-231 (1.1346s)	1.2907e-273 (1.4903s)	5.9914e-294 (1.8356s)	6.0289e-298 (2.3657s)	2.7255e-308 (2.7617s)	1.8068e-300 (3.2406s)
SumSquares	8.5727e-166 (0.7423s)	4.7152e-193 (1.1017s)	3.7439e-237 (1.4297s)	1.6852e-255 (1.9296s)	4.1627e-282 (2.3097s)	8.6117e-289 (2.9365s)	8.2269e-295 (3.4690s)
Zakharov	1.8859e-153 (0.8230s)	6.5000e-241 (1.3920s)	4.8603e-276 (2.0760s)	8.4112e-314 (2.8799s)	7.4240e-320 (3.9381s)	2.1203e-309 (4.8080s)	5.9637e-316 (5.8865s)
Schwefel 1.2	2.2092e-076 (0.4956s)	2.9203e-081 (0.7345s)	6.2523e-087 (1.2426s)	4.4125e-093 (1.5868s)	7.7396e-101 (1.9838s)	1.6134e-100 (2.5782s)	6.0789e-101 (2.9868s)
Schwefel 2.22	2.0276e-061 (0.7510s)	2.0181e-098 (1.0748s)	4.6723e-119 (1.5014s)	1.4030e-127 (1.9178s)	1.8541e-141 (2.3204s)	7.5746e-138 (2.9684s)	1.6206e-138 (3.3697s)

set to 50, the content of test is consistent with 4.1.1. The experimental results are shown in Table 3.

According to the Table 3: For Schaffer and Rastrigin, the algorithm converges to the optimal value when the number of evaluations reaches 3000. For other functions, with the increase of evaluation times, the convergence accuracy is also greatly improving. If ignore the time spent, the more evaluation times of the algorithm, the better performance of the algorithm. And the number of the evaluations should be set according to specific functions.

Table 3. The influence of the evaluation times on the algorithm

Times / Fun	Mean Value (Mean Time)						
	1000	3000	5000	7000	9000	11000	13000
Schaffer	2.8253e-010 (0.6779s)	0 (0.8083s)	/ /	/ /	/ /	/ /	/ /
Rastrigin	0.0089 (0.5069s)	0 (0.7131s)	/ /	/ /	/ /	/ /	/ /
Elliptic	0.0021 (0.5585s)	3.6503e-022 (0.8941s)	2.3714e-046 (1.2480s)	3.3153e-065 (1.6007s)	1.1426e-092 (1.9420s)	8.5718e-128 (2.2871s)	1.5665e-170 (2.5522s)
Sphere	8.2778e-005 (0.5490s)	4.8060e-028 (0.7047s)	1.2706e-064 (0.8940s)	4.2497e-085 (1.1070s)	5.4489e-123 (1.2983s)	1.0249e-153 (1.4553s)	9.8811e-192 (1.6589s)
SumSquares	4.1407e-005 (0.5021s)	1.6537e-034 (0.7188s)	1.2970e-049 (0.8814s)	1.5190e-076 (1.0751s)	1.3421e-110 (1.2644s)	1.4159e-146 (1.4919s)	1.4725e-174 (1.7425s)
Zakharov	1.9384e-006 (0.6632s)	4.0948e-033 (0.9762s)	4.7809e-057 (1.3254s)	8.4195e-082 (1.7245s)	1.8319e-137 (2.0515s)	2.3403e-140 (2.4369s)	3.6391e-193 (2.8359s)
Schwefel 1.2	9.6214e-004 (0.4754s)	6.9451e-027 (0.6637s)	1.0383e-065 (0.9037s)	1.0776e-098 (1.1182s)	1.4259e-132 (1.3044s)	3.5968e-163 (1.5500s)	5.4474e-191 (1.7921s)
Schwefel 2.22	0.0466 (0.5531s)	2.9160e-016 (0.7174s)	1.7919e-027 (0.8900s)	8.7093e-036 (1.0762s)	1.6875e-044 (1.2870s)	1.8245e-068 (1.6907s)	1.1527e-089 (2.0907s)

4.2. **Comparison With the Series of SOS.** In order to verify the performance of the proposed CESOS algorithm, compare it with SOS and MASSOS in terms of convergence speed and convergence accuracy. The parameters of each algorithm are set as follows: According to 4.1.1, the number of individuals is 50; the dimension is set according to the specific functions in Table 1; the other specific parameters of these algorithms are set according to the corresponding documents.

4.2.1. *Comparison of Convergence Accuracy.* In this section, the accuracy is compared with the algorithms above. According to 4.1.2, the number of evaluations is as many as possible to fully show the performance of the algorithms. So for Schaffer and Rastrigin, the number of evaluation is set to 20000; the evaluation times of other functions are all set to 60000. The test items of convergence accuracy include the Best Value, the Worst Value, the Mean Value, Standard Deviation and the Success Rate of arriving at 10^{-10} in the 30 independent experiments. The experimental results are shown in Table 4.

For table 4 above: First, all functions obtain the theoretical optimal value through CESOS; SOS and MASSOS obtain the theoretical optimal value only on Rastrigin. Obviously, CESOS has a great advantage in convergence precision. Second, it can be seen that the stability of the proposed algorithm in this paper is better by comparing the Standard Deviation of the results and the Success rate.

To verify whether there is a significant difference among the algorithms, use Friedman's Test to analyze the experimental results (Significant level $\alpha = 0.05$). In order to make the test more objective, the rank of the mean value of each algorithm on each function in table 4 is taken as the statistical value and evaluated from low to high. (Take Schaffer as an example, the mean values of the SOS, MASSOS and CESOS algorithms are: 7.1296e-010, 7.4469e-005 and 0, so the rank of the mean values is: 2,1,3). In the case of the same mean value, it is arranged according to the standard deviation. The specific data are shown in Table 5.

Table 4. The results of convergence accuracy

Function	Methods	Best	Mean	Worst	SD	Rate %
Schaffer	SOS	5.5514e-017	7.1296e-010	1.4803e-008	2.7577e-009	86.67
	MASSOS	4.2936e-013	7.4469e-005	8.7305e-004	1.7358e-004	23.33
	CESOS	0	0	0	0	100
Rastrigin	SOS	0	0	0	0	100
	MASSOS	0	0.2140	1.2549	0.1208	33.33
	CESOS	0	0	0	0	100
Elliptic	SOS	1.5452e-122	1.2027e-116	3.0232e-115	5.5393e-116	100
	MASSOS	3.7464e-204	8.5976e-190	2.5791e-188	0	100
	CESOS	0	0	0	0	100
Sphere	SOS	7.6753e-125	1.4713e-119	3.1238e-118	5.8340e-119	100
	MASSOS	6.6475e-234	2.9693e-222	2.6607e-221	0	100
	CESOS	0	0	0	0	100
SumSquares	SOS	7.1823e-127	4.1910e-120	1.2299e-118	2.2439e-119	100
	MASSOS	9.4766e-245	7.8122e-234	1.0378e-232	0	100
	CESOS	0	0	0	0	100
Zakharov	SOS	1.5698e-126	1.1087e-121	1.4061e-120	3.1356e-121	100
	MASSOS	7.0113e-253	3.5946e-240	2.2685e-240	0	100
	CESOS	0	0	0	0	100
Schwefel 1.2	SOS	6.0725e-101	2.8955e-097	1.2768e-096	6.6201e-097	100
	MASSOS	1.5514e-169	2.4165e-163	7.6541e-162	1.2807e-164	100
	CESOS	0	0	0	0	100
Schwefel 2.22	SOS	1.6896e-063	7.2445e-061	6.7210e-060	1.3787e-060	100
	MASSOS	7.1015e-110	3.3133e-103	9.5395e-102	1.7394e-102	100
	CESOS	0	0	0	0	100

Table 5. The Test Statistics

Methods	The Rank of the MeanValue								Average Rank
SOS	2	2	1	1	1	1	1	1	1.25
MASSOS	1	1	2	2	2	2	2	2	1.75
CESOS	3	2	3	3	3	3	3	3	2.875

The formula (8) is used to calculate the test statistics.

$$\chi_r^2 = \frac{12n}{k(k+1)} \sum_j R_j^2 - 3n(k+1) \tag{8}$$

In the formula, the meaning of "k" and "n" is that "k" learning algorithms on "n"data sets. In this test: $k = 3$, $n = 8$, R_j^2 is the square of the Average Rank. Finally, $\chi_r^2 = 7.125$ is calculated. Due to $\alpha = 0.05$, $df = 3\text{-}1$, it can be seen from the standard table $\chi_{0.05}^2 = 5.99 < \chi_r^2 = 7.125$. Therefore, there are significant differences between the three algorithms under the level of 5%.

In order to verify the specific differences between the proposed algorithm and the other two algorithms, compare CESOS with MASSOS and SOS through Holm's test respectively. The specific comparison formula is shown in formula (9).

$$z = (R_i - R_j) \Bigg/ \sqrt{\frac{k(k+1)}{6n}} \tag{9}$$

Find out the probability P corresponding to Z according to the standard normal distribution table. Arrange the value of P from small to large. Compare P_i and $\alpha/(k\text{-}i)$ (The settings for α and k are the same as in Friedman's Test) under the hypothesis that the two algorithms corresponding to P_i have the same performance, if P_i is smaller than $\alpha/(k\text{-}i)$, the hypothesis is thought to be rejected, in other words, there are obvious differences between the two algorithms.

According to formula (9), $se = \sqrt{\frac{k(k+1)}{6n}} = \sqrt{\frac{3\times(3+1)}{6\times 8}} = 0.5$ can be calculated. It is assumed that the proposed algorithm in this paper is superior to the other two algorithms. The comparison process of Holm's test is shown in Table 6.

TABLE 6. The procedure of Holm's test

i	Methods	$Z = (R_i - R_j)/se$	P_i	$a/(k-i)$
1	CESOS,SOS	(2.875-1.25)/0.5 = 3.25	0.0012	0.0250
2	CESOS,MASSOS	(2.875-1.75)/0.5 = 2.25	0.0244	0.0500

It can be seen from table 6: $P_1 < \alpha/(k\text{-}1)$ and $P_2 < \alpha/(k\text{-}2)$, so the Holm's test rejects the original hypothesis. This shows that the algorithm proposed in this paper has obvious advantages in the case of significance level $\alpha = 0.05$.

4.2.2. *Comparison of Convergence Speed.* As for the comparison of convergence speed, test the convergence accuracy of each algorithm under different evaluation times. The evaluation times for Schaffer and Rastrigin are set to 4000, 8000 and 12000, the other functions are set to 20000, 30000 and 40000.

TABLE 7. The results of convergence Speed

Function	Evaluation times	Methods		
		SOS	MASSOS	CESOS
Schaffer	4000	0.0019	0.0060	0
	8000	2.3842e-004	4.1812e-004	0
	12000	1.6511e-007	1.2811e-005	0
Rastrigin	4000	0.1286	21.8956	0
	8000	2.5208e-005	0.1723	0
	12000	2.1316e-014	2.0378e-006	0
Elliptic	20000	1.0276e-038	4.2580e-005	3.9137e-275
	30000	8.0338e-058	4.7898e-049	0
	40000	3.2861e-077	4.3136e-096	0
Sphere	20000	3.0235e-037	1.0616e-029	7.6928e-279
	30000	1.2455e-058	1.4516e-078	0
	40000	1.7486e-079	2.0354e-126	0
SumSquares	20000	1.5135e-038	6.2985e-042	2.2073e-301
	30000	5.9364e-062	1.9105e-091	0
	40000	2.2176e-081	4.0962e-140	0
Zakharov	20000	1.2375e-039	3.5113e-046	1.2849e-254
	30000	1.7769e-060	4.4989e-096	0
	40000	2.5504e-082	4.8090e-146	0
Schwefel 1.2	20000	2.5628e-030	2.1891e-039	2.6108e-316
	30000	9.6532e-048	4.1557e-071	0
	40000	2.1612e-064	2.6274e-103	0
Schwefel 2.22	20000	4.4451e-020	1.5794e-010	5.5669e-122
	30000	2.0947e-030	9.1826e-035	1.5084e-211
	40000	4.0941e-041	3.9076e-059	9.9185e-288

From table 7: the convergence value of CESOS is obviously better than the other algorithms under the same evaluation times, which shows that the convergence speed of CESOS is faster.

Analyze the results of 4.2.1 and 4.2.2, the algorithm proposed in this paper has an obvious improvement in convergence precision and speed. At the same time, CESOS is more excellent in stability by comparing and analyzing the Mean Value, the Worst Value and the Standard Deviation. Therefore, in the series of SOS, CESOS performs best on the functions from CEC2005.

4.3. **Comparison with the Series of non-SOS.** In order to further test the performance of the proposed algorithm in this paper, 15 functions F1-F15 are selected from the CEC2013. Compared with the functions from CEC2005 above, these functions are more complex. And F1-F5 are Unimodal Functions, F6-F15 are Multimodal Functions.

4.3.1. *Contrast algorithms and Parameters setting.* To make the test results more objective, compare CESOS with GA-TPC, f_k-PSO, SaDE and SCDE. The parameters of each algorithm are set as follows: The dimension(D) is set to 30, the search range is from -100 to 100 and the evaluation times of each function is D*10000. Each algorithm runs 51 times independently. The other specific parameters of these algorithms are set according to the corresponding documents.

4.3.2. *Experimental results and analysis.* This paper evaluates the performance of each algorithm by the Mean Error: [Me $f(x_{best})$]-$f(x^*)$[13]. The [Me $f(x_{best})$] is the average value of the 51 independent experiments, x^* is the global optimal solution. The Standard Deviation is used to evaluate the stability of the algorithm. The experimental results are shown in table 8.

From table 8: For Unimodal Functions: F1-F5, CESOS performs great on F1/F2/F3/F5 , only the result on F4 is modest. GA-TPC performs great on F1/F4/F5. f_k-PSO does well on F1/F5. SCDE perform best on F1. SaDE is the worst performance. For Multimodal Functions: F6-F15, CESOS performs best on F7/F8/F9/F12/F13/F15. GA-TPC only dose well on F10. SaDE has the best results on F6/F11/F14. f_k-PSO and SCDE are modest. Only analyze these results in the table above, CECSOS still has a better advantage.

However, in order to further analyze the difference among the algorithms, use Friedman's Test to analyze the experimental datas, the experimental method is the same as 4.2.1. In this test, $k = 5$, $n = 15$, R_j^2 is the square of the Average Rank($R_{CESOS} = 3.8$, $R_{GA-TPC} = 2.87$, $R_{fk-PSO} = 2.27$, $R_{SaDE} = 3.6$,$R_{SCDE} = 2.07$), take these parameters into the formula(8). Finally, $\chi_r^2 = 0.4482$ is calculated. $\alpha = 0.05$, $df = 5\text{-}1$, it can be seen from the standard table $\chi_{0.05}^4 = 9.49 > \chi_r^2 = 0.4482$. The Friedman's Test shows that there is no obvious difference among the five algorithms under the significance level of 5%. The main reason for this phenomenon is that the structure of CEC2013 is too complicated, the difference between algorithms is not as obvious as CEC2005.

In order to further test the specific performance of each algorithm, use Holm's test mentioned in 4.2.1 to analyze those data. According to formula (9), $se = \sqrt{\frac{k(k+1)}{6n}} = \sqrt{\frac{5\times(5+1)}{6\times 15}} = 0.5774$ is calculated. Assuming that CESOS is superior to the other four algorithms. The comparison process of Holm's test is shown in Table 9.

For table 9: $P_1 < \alpha/(k\text{-}1)$ and $P_2 < \alpha/(k\text{-}2)$, so the Holm's test rejects these two original hypotheses, this means that the CESOS is superior to SCDE and f_k-PSO under the significance level of 5%. However, $P_3 > \alpha/(k\text{-}3)$and $P_4 > \alpha/(k\text{-}4)$, the Holm's test

Table 8. The results on CEC2013 functions

No	CESOS	GA-TPC	f_k-PSO	SaDE	SCDE
	Mean/SD	Mean/SD	Mean/SD	Mean/SD	Mean/SD
$F1$	0.00e+00/ 0.00e+00	0.00e+00/0.00e+00	0.00e+00/0.00e+00	1.00e-08/0.00e-00	0.00e+0.00/ 3.22e-14
F2	2.77e+04/ 5.06e+04	1.55e+05/1.37e+05	1.59e+06/8.03e+05	3.40e+04/1.63e+04	8.15e+06/ 3.44e+06
F3	1.03e+06/ 2.49e+07	3.28e+07/7.55e+07	2.40e+08/3.71e+08	3.32e+06/5.33e+06	1.13e+08/ 1.11e+08
F4	1.36e+03/ 2.31e+02	9.08e-01/1.26e+00	4.78e+02/1.96e+02	1.03e+02/1.52e+02	2.95e+04/ 4.28e+03
F5	0.00e+00/ 0.00e+00	0.00e+00/0.00e+00	0.00e+00/0.00e+00	1.00e-08/0.00e-00	1.99e-01/ 1.10e+00
F6	2.05e+02/ 3.84e+01	2.04e+01/7.92+00	2.99e+01/1.76e+01	8.72e+00/1.03e+01	2.83e+01/ 5.09e+00
F7	1.51e+00/ 7.40e-003	4.58e+01/2.97e+01	6.39e+01/3.09e+01	1.92e+01/1.06e+01	1.43e+01/ 8.35e+00
F8	3.58e+00/ 2.46e-002	2.10e+01/5.34e-02	2.09e+01/6.28e-02	2.09e+01/5.21e-02	2.09e+01/ 5.14e-02
F9	1.07e+01/ 1.92e+00	3.70e+01/6.44e+00	1.85e+01/2.69e+00	1.69e+01/3.81e+00	3.84e+01/ 4.47e+00
F10	7.31e+01/ 3.56e-01	8.35e-02/4.66e-02	2.29e-01/1.32e-01	1.52e-01/1.02e-01	1.03e+01/ 6.79e+00
F11	1.90e+01/ 3.42e-06	2.13e+01/1.07e+01	2.36e+01/8.76e+00	5.85e-02/2.36e-01	1.15e+01/ 4.26e+00
F12	2.82e+01/ 1.88e+00	3.77e+01/9.54e+00	5.64e+01/1.51e+01	3.34e+01/8.92e+00	1.60e+02/ 1.05e+01
F13	3.35e+01/ 2.29e+01	8.10e+01/1.95e+01	1.23e+02/2.19e+01	7.21e+01/2.02e+01	1.56e+02/ 1.04e+01
F14	6.83e+01/ 6.95+01	1.01e+03/4.74e+02	7.04e+02/2.38e+02	8.34e-02/2.25e-01	2.20e+02/ 1.30e+02
F15	3.47e-01/ 5.10e-01	4.10e+03/6.93e+02	3.42e+03/5.16e+02	4.82e+03/4.08e+02	7.16e+03/ 2.53e+02

Table 9. The procedure of Holm's test

i	Methods	$Z = (R_i\text{-}R_j)/se$	P_i	$a/(k\text{-}i)$
1	CESOS, SCDE	(3.8-2.07)/0.5774 = 2.9962	0.0028	0.0125
2	CESOS, f_k-PSO	(3.8-2.27)/0.5774 = 2.6498	0.0082	0.0167
3	CESOS, GA-TPC	(3.8-2.87)/0.5774 = 1.6107	0.1073	0.0250
4	CESOS, SaDE	(3.8-3.6)/0.5774 = 0.3464	0.2710	0.0500

accepts these two original hypotheses. This indicates that CESOS, GA-TPC and SaDE have the considerable performance when optimizing the CEC2013.

Through the analysis of the experimental results in 4.2 and 4.3, compare with SOS and MASSOS, the CESOS algorithm proposed in this paper shows a great advantage on the 8 functions from CEC 2005. Meanwhile, compare with the improved algorithms of other algorithms, CESOS also shows considerable performance in optimizing the functions from CEC2013.

5. **Conclusions.** In order to solve the deficiency of SOS in convergence speed and convergence precision, Symbiotic Organisms Search algorithm based on Cloud Model Elite Search is proposed in this paper. First, the introduction of the Elite Individual, Stretching Factor "C" and Disturbing Individual in Commensalism phase promote the speed of

the convergence as well as guarantee the diversity of the population. Then, at Parasitism phase, the introduction of Cloud Model retains the original evolutionary information in a way and avoids the blindness of search. Finally, compare CESOS with SOS and MASSOS on the test functions from CEC2005, and compare it with GA-TPC, f_k-PSO, SaDE and SCDE on CEC 2013, as well as use Friedman's Test and Holm's test to statistically analyze experimental data, the results show that the algorithm proposed in this paper has a good performance.

Acknowledgments. The authors disclosed receipt of the following financial support for the research, authorship of this article: This work was supported in part by the National Natural Science Foundation of China under grants NO.61501107 and the Education Department of Jilin province science and technology research project of "13th Five-Year" under grants [2016] No.95. And the Project of Scientific and Technological Innovation Development of Jilin NO.201750227 and NO.201750219

REFERENCES

[1] M. Y. Cheng, D. Prayogo , Symbiotic Organisms Search: A new metaheuristic optimization algorithm, *Computers& Structures,* vol.139, pp. 98-112, 2014.

[2] H. Zhou, H. Zhao, et al. Multi Strategy Adaptive Symbiotic Organisms Search, *Journal of Air Force Engineering University (Natural Science Edition)*, vol.17, no. 4, pp.101-106, 2016.

[3] S. Nama, A. K. Saha , S. Ghosh, Improved symbiotic organisms search algorithm for solving unconstrained function optimization, *Decision Science Letters*, vol.5, no.3, pp.361-380, 2016.

[4] Y. J. Wang , H. H. Tao, Symbiotic Organisms Search algorithm based on Rotating Learning Strategy, *Computer Application Research*, vol. 34, no. 9, pp. 2614-2617, 2017.

[5] S. Gong, R. Huang, Cao Z, An improved symbiotic organisms search algorithm for low-yield stepper scheduling problem, *Automation Science and Engineering IEEE*, pp.289-294, 2018.

[6] D. Prayogo, F. T. Wong , Sugianto S Enhanced symbiotic organisms search (ESOS) for global numerical optimization *International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation. IEEE*, 2018.

[7] S. M. Elsayed, Sarker R A, et al. A genetic algorithm for solving the CEC'2013 competition problems on real-parameter optimization, *Evolutionary Computation IEEE*, pp.356-360, 2013.

[8] Nepomuceno F V, Engelbrecht A P, A Self-adaptive Heterogeneous PSO for real-parameter optimization, *Evolutionary Computation IEEE*, pp. 361-368, 2013.

[9] A. K. Qin , Li X, Pan H, et al. Investigation of Self-adaptive Differential Evolution on the CEC-2013 real-parameter single-objective optimization testbed, *Evolutionary Computation IEEE*, pp.1107-1114, 2013.

[10] L. B. Zhu , et al. A new different evolution algorithm based on SSO algorithm and covariance matrix learning, *Computer Engineering and Science,* vol.39, no.11, pp.2122-2130, 2017.

[11] Demšar J, Statistical comparisons of classifiers over multiple data sets, *Journal of Machine Learning Research*, vol.7, no.1, pp.1-30, 2006.

[12] D. Y. Li , Zhang G W, et al, Evolution algorithm based on cloud model, *Journal of Computer Science*, vol.31, no.7, pp.1082-1091, 2008.

[13] M. G. Epitropakis, D. K. Tasoulis , et al. Enhancing Differential Evolution Utilizing Proximity-Based Mutation Operators, *IEEE Transactions on Evolutionary Computation*, vol.15, no.1, pp. 99-119, 2011.