Journal of Computers Vol. 34 No. 5, October 2023, pp. 305-316
doi: 10.53106/199115992023103405023

An Improved Cuckoo Search Algorithm Based on Elite Opposition-based Learning for Indoor Visible Light Positioning

Yang Yang, Mao-Sheng Fu, Chao-Chuan Jia*, Wang Miao, Zong-Ling Wu

College of Electronics and Information Engineering, West Anhui University,
Lu'an, 237012, China

yangyy@wxc.edu.cn, 49593838@qq.com, ccjia@hfcas.ac.cn,
{124414198, 1057287263}@qq.com

Received 22 April 2022; Revised 5 May 2022; Accepted 11 May 2022

Abstract. In the cuckoo search algorithm, the structure is simple, and the parameters are not much, but it is easy to trap into the local optimum, and in the later period, the convergence speed is plodding. Aiming at the shortcomings of the standard cuckoo algorithm, a modified cuckoo algorithm (EACSDAM) is presented in this paper, which adopts elite reverse learning to enhance the population diversity, and increases the step factor and discovery probability to improve the global detection and local searchability. Eight standard test functions are used to simulate the EACSDAM algorithm. Compared with the standard cuckoo algorithm and the other two improved algorithms, the accuracy and convergence speed of EACSDAM are greatly improved. In the end, EACSDAM is used to optimize the indoor 3D visible light positioning. The simulation results indicate that EACSDAM has a more powerful ability for global optimization, and more accurate positioning, and the positioning error is significantly reduced.

Keywords: cuckoo search algorithm, elite reverse learning, step size factor, discovery probability, indoor three-dimensional visible light positioning

1 Introduction

As we all know, optimization algorithms are usually divided into two categories: conventional optimization algorithms and intelligent optimization algorithms. Intelligent optimization algorithms are more popular than conventional optimization algorithms. Intelligent optimization algorithms contain such advantages as agility, reliability, and internal evolution. Secondly, the use of intelligent optimization algorithm is extensive. Almost all research and engineering fields are involved in its knowledge, such as computers, biology, medicine, transportation, aviation, communication, and so on. Intelligent optimization algorithm mainly discusses the stability, convergence speed, and complexity of the algorithm, and the parameter setting of the algorithm is an essential object of research. Researchers have made many improvements to the optimization algorithm, adjusting strategies and resetting parameters to make the algorithm more efficient. The improved algorithm dynamically modifies some parameters to achieve a certain relationship balance between exploration and development. For the intelligent optimization algorithms, the balance degree of exploration and development directly affects the convergence speed and optimization ability of the algorithm, and also determines the advantages and disadvantages of the algorithm. At the end of the 20th century, meta-heuristic algorithms have gradually become prominent, such as Genetic Algorithm [1] (GA), Ant Colony Optimization Algorithm [2] (ACO), Particle Swarm Optimization algorithm [3] (PSO), Glowworm Swarm Optimization algorithm [4] (GSO), etc. The meta-heuristic intelligent optimization algorithms imitate the reproduction and evolution process of various organisms in nature, such as fish, ant colonies, fireflies, etc., to find the optimal solution to the problem. In the research and development of meta-heuristic algorithms, the swarm intelligence algorithm develops rapidly and is the most active one. The swarm intelligence algorithm, which combines life evolution closely with evolutionary strategy, has been well applied in search and discrete optimization problems.

In 2009, researchers Xin-She Yang and Squash Deb simulated cuckoo nesting behavior and proposed Cuckoo Search Algorithm (CS) [5-6]. It adopts Levy flight and has excellent adaptability with few parameters, optimal random search path, and robust searching ability [7].

In recent years, CS has been generally applied in various engineering optimization problems [8], and it has

* Corresponding Author

great application value. CS algorithm has a simple structure and is easy to implement. Still, it also has the disadvantage that the convergence is slow in the later period, and it is trapped to local optimal [9]. The CS algorithm is based on Levy Flights and cross-variation within the population and its capacity has reached specific level of optimization. The random walk mechanism in Levy flight is controlled by the random step length, which switches randomly between the relatively stable short step length and the occasional long step length. In this case, the cuckoo's searching appears to jump between large locations. It is this kind of jump that brings better global exploration ability for the algorithm, but it can't rule out that it also has certain blindness. At the same time, in order to ensure the continuity of the population, the CS algorithm carries out the reservation update to the search target, which leads to the weak search ability in the local area of the target search. The convergence rate of the algorithm in the late optimization is slow, and the convergence accuracy is not high.

To increase the accuracy and convergence speed of the CS, many kinds of literature put forward some improved methods to optimize the CS. The detection probability was resized as the number of iterations changed in the paper [10]. The step size factor was divided into three categories according to the range of fitness value, and it was adjusted respectively to balance the global and local search capabilities. It updated the step size and discovery probability in the literature [11], which effectively improved the accuracy and convergence speed. In the paper [12], the difference of multi-strategy was used to optimize the initial population. Queuing optimization mechanism and greedy algorithm were used to improve ion diversity and algorithm accuracy. In the literature [13], it optimized some individuals of Levy flight with the elite reverse learning strategy. The individuals which are generated by preference wandering were optimized by a simple cross. And it improved the search ability and convergence speed. In reference [14], the sorting layer, the allocation layer, and the selection layer in the workshop were coded and decoded respectively. These two initial population strategies, optimization time and randomness, were adopted to classify the population fitness values. It optimized the algorithm and improved the search efficiency. In literature [15], the step size factor is replaced by the optimal results of 12 kinds of chaotic mapping. When the non-optimal solution is discovered, a new strategy is adopted for the local update, which increases the individual difference within the group and effectively avoids the premature fall into the local optimal. In the global search process, literature [16] adopted the positive disturbance radius parameter with multi-stage variation to control the search range, and combined the inertia weight with dynamic variation to improve the search speed and accuracy, thus improving the local optimization ability. Literature [17] uses the combining the triangulation flip strategy with the change of iteration number. Using the practical selection of penalty factors, the satisfaction degree of the optimal solution is improved, and the convergence speed of the algorithm is accelerated.

A modified cuckoo algorithm EACSDAM is put forward for the deficiency of the CS. The second section of this paper, abstracts the principles of the CS, and the third part introduces the EACSDAM algorithm implementation strategy. The fourth part shows the simulation results of the improved algorithm and other three kinds of algorithm, and comparative analysis of their solution precision and convergence. The fifth part optimizes the indoor optical three-dimensional positioning method with the EACSDAM algorithm, and proves the advantages.

2 The Standard Cuckoo Search Algorithm (CS)

A cuckoo spawns in another birds' nest [18]. If the nest owner finds these exotic birds' eggs, they will abandon their young or look for a new nest [19].

Yang and Deb set three perfect rules [5]: (1) A cuckoo produces only one offspring once and places it in other nests; (2) The best one survives to the next generation; (3) The count of nests is specific. The owner may find that the offspring are not his own, with the probability of P_a. After the discovery, the host may discard the offspring or abandon the nest and rebuild the nest.

On the basis of the above three points, Yang and Deb proposed Cuckoo Search (CS), which mimics cuckoo's reproductive behavior by finding suitable nests of other birds, placing eggs, and incubating them. In finding suitable nests, the CS algorithm completes several iterations to update the nests. In each iteration, the CS algorithm carries out two population updates, namely the update strategy of Lévy flights and preferred random walk, because each update will consider the advantages and disadvantages of the current position and the new position. If the new position is better, the current position will be updated; otherwise, the current position will remain unchanged.

Update the nesting position of each generation, as shown in Equation (1):

$$X_i^{(t+1)} = X_i^{(t)} + \alpha \oplus levy(\lambda)\ i = 1, 2, ..., n. \quad \textbf{(1)}$$

$\chi_i^{(t)}$ represents the nest position of t generation; α is the change in step size; $\oplus$ is point-to-point multiplication; $levy(\lambda)$ is Lévy flights, as shown in Equation (2):

$$levy(\lambda)\tilde{}\ \mu = t^{-\lambda} (1 < \lambda \leq 3). \quad \textbf{(2)}$$

Lévy flights, named after The French mathematician Paul Levy, is also known as a random walk renewal strategy. The cuckoo search algorithm adopts this updating strategy. When the cuckoo is looking for nests, the flight direction is random, and there is a certain deviation between the current flight track and the previous flight track. Lévy flights makes the cuckoo's step size change in searching for optimization, so the CS has the robust global search ability and it can avoid falling into local optimal.

Formula (3) represents Lévy flights [20]:

$$levy(\lambda) = \frac{\varphi * \mu}{|\upsilon|^{\frac{1}{\beta}}}. \quad \textbf{(3)}$$

υ and μ the normal distribution obeying the standard, $\beta = 1.5$,

$$\phi = \left[\frac{\Gamma\ (1 + \beta) \times \sin\left(\pi \dfrac{\beta}{2}\right)}{\Gamma\left(\left(\dfrac{1 + \beta}{2}\right) \times \beta \times 2^{\frac{\beta - 1}{2}}\right)} \right]^{\frac{1}{\beta}}. \quad \textbf{(4)}$$

The step size factor was updated by reference [6]:

$$\alpha = \alpha_0 * \left(\chi_{\mathrm{i}}^{(\mathrm{t})} - \chi_{\mathrm{best}}\right). \quad \textbf{(5)}$$

α_0 is 0.01, X_{best} is the optimal value.

Combine formula (1) ~ (5) to update the nest position:

$$\chi_i^{(\mathrm{t+1})} = \chi_i^{(t)} + \alpha_0 \frac{\phi * \mu}{|\upsilon|^{\frac{1}{\beta}}} * \phi * \left(\chi_{\mathrm{i}}^{(\mathrm{t})} - \chi_{\mathrm{best}}\right). \quad \textbf{(6)}$$

The probability that the host finds the alien offspring is P_a, and if other offspring are found, it will update the location of the nest with Lévy flights. It will generate the compression factor γ in the interval of (0,1). When $\gamma > P_a$, the preference walk is adapted to create the solution of the t+1 generation, $X_i^{(t\ \ 1)}$ as shown in (6):

$$X_i^{(t+1)} = X_i^{(t)} + \gamma * (X_j^{(t)} - X_k^{(t)}). \quad \textbf{(7)}$$

$X_j^{(t)}$ and $X_k^{(t)}$ represent two random solutions for generation t.

This is an iterative process of the CS algorithm, which contains three essential updates: first, Lévy flights, the step size keeps changing during the optimization process; second, the preference for the random walk is to cross and mutate the two random solutions. Third, the current optimal solution is updated iteratively. The CS algorithm also adopts the strategy of reserving better value, so that the optimal solution in the population is always retained.

3 Improved Cuckoo Search Algorithm (EACSDAM)

Swarm intelligence optimization algorithms have the ability of "global exploration" and "local exploitation". When the relationship between the two reaches a balance, the algorithm can find the optimal solution faster and more accurately. "Exploration" is searching random locations to update the location of the optimal solution. "Exploitation" refers to searching in the updated local solution space to find a locally optimal solution. So exploration is also called "global exploration", and development is also called "local exploitation". In the global exploration stage, the local exploitation ability of the algorithm is not strong, and it is easy to fall into local optimization. In the local exploitation phase, the global exploration ability is weak and the optimization speed is slow.

In the SWARM intelligence optimization CS algorithm, a random method is adopted to initialize the population before iteration, with an extensive solution range and a long search time, affecting the search performance [21-22]. At the same time, the parameter sum in the CS is a fixed value, and has fallen into optimal local with the iteration. This paper adopts the strategy in ACSDAM [11], and is optimized by elite opposition-based learning (EOBL), so the improved algorithm is named EACSDAM.

3.1 Optimize the Initial Population using Elite Opposition-Based Learning

It can accelerate the convergence rate of the algorithm and find the optimal global value faster if it has a high-quality initial population [22]. The paper [23] proposed Elite Opposition-Based Learning (EOBL) in 2005. The opposition-based learning method adopts the principle of bidirectional evaluation. In the each iteration, not only the candidate solution of the current iteration is evaluated, but also the candidate solution in the opposite direction is evaluated. Then, the best-evaluated solution is selected as the next-generation candidate solution after the current solution is compared with the opposition-based solution. The inverse solution is 50% more likely to approach the optimal global solution than the current solution. Opposition-based learning can improve the probability of finding the optimal global solution greatly.

Assuming that an initial population to be optimized is X_i, the elite opposition-based learning strategy is adopted to optimize the population, and the opposition-based solution of the individual population $\overline{X}_i$ is generated, which is expressed in the paper [24].

$$\overline{X}_i = k(L+U) - X_i. \tag{8}$$

Where, X_i is the current individual, L is the maximum of feasible solutions. U is the minimum. k is a random number between (0,1).

$$X_i = \begin{cases} \overline{X}_i, f(\overline{X}_i) < f(X_i) \\ X_i,\ else \end{cases}. \tag{9}$$

f is the evaluation function of the fitness value of the candidate solution. The current values are compared to the opposition-based values, and then the optimal one was chosen as the initial population [25].

The basic steps of the elite opposition-based learning strategy to initialize the population are as follows:

(1) Randomly initialize population S, and select the first N/2 individuals with better fitness to form elite population E.

(2) Find the opposition-based population OE of elite population E.

(3) A new population {S∪OE} is obtained by combining S and OE, and N individuals are selected to form the initial population.

In this paper, the elite opposition-based learning is introduced in the initial population and the each population iteration. Firstly, for the initial population, the elite opposition-based learning is introduced to sort the initial random population and its opposition-based population based on the adaptive value, and the top H individuals are selected as the next generation population to improve the diversity of the initial population. It lays a foundation for better global optimization. Secondly, in the population iteration, the current population and its opposition-based population are sorted according to the adaptive value, and the first H individuals are selected as the next-generation population to enhance the algorithm's ability to jump out of local optimization and prevent the algorithm from being premature. Finally, elite opposition-based learning dynamic boundary search is used to gradually reduce the search space with the increase of iteration times, and the convergence speed and precision of the algorithm are enhanced.

3.2 Dynamically Adjust Step Size Factor and Discovery Probability

At the beginning of the iteration of the cuckoo algorithm, to ensure the global search ability of the algorithm, the step factor a_0 should be set to a higher value. In contrast the discovery probability p_a should be set to a smaller value, so that the search range should be the whole solution space as far as possible. As the number of iterations increases, the local exploration capability must be strengthened. In this case, a smaller value should be set for a_0, while a higher value should be set for p_a. In this case, the search range will be smaller, the search speed will be faster, and the accuracy must be improved. Therefore, in the whole iteration process, the step size and discovery probability change dynamically with the change of iteration times, a_0 from high to small, and p_a from small to high.

The strategy (ACSDAM) proposed in the paper [11] is adopted to increase the step size of dynamic adjustment a_0, as shown in Formula (10):

$$a_0 = 1 - \frac{1}{1.5 + exp\left(\frac{t - tmax}{tmax}\right)^2} . \tag{10}$$

And the introduction of dynamic adjustment factors into the probability of discovery Pw [11],

$$Pa = P_0 * Pw . \tag{11}$$

Pw as shown in Equation (12):

$$Pw = \frac{1}{1 + exp\left(ln1.5 - ln19 * \left(\frac{t}{tmax}\right)^2\right)} . \tag{12}$$

t is the current number of iterations and $tmax$ is the total number of iterations. In the iterative process, with the change of t, gradually changes from large to small, and p_w grows from small to high progressively. P_a and p_w synchronous change just keep an excellent global exploration in the early stage to ensure the diversity of the population. In the late stage, local development becomes more prominent and accelerates the convergence rate.

3.3 Improved Cuckoo Algorithm (EACSDAM) Process

The EACSDAM algorithm flow is as follows:

(1) Set the number of nests and the maximum number of iterations;

(2) Using elite reverse learning method to optimize the flock;

(3) Calculate the objective function and get the fitness value of each individual;

(4) Within the maximum number of iterations, Formula (13) and Formula (2) are used to calculate Formula (1) to find the next position of the nest;

(5) Using the results of Formula (14) and Formula (15), discuss the relationship between the size of and, calculate Formula (7), find the next position of the nest;

(6) Judge whether the maximum number of iterations is reached in step (4). If yes, the optimal position is output; otherwise, return to Step (3) to continue searching for optimization.

4 Comparative Experimental Analysis of Several Cuckoo Algorithms

4.1 Algorithm Accuracy Analysis

To prove the performance of EACSDAM, eight standard test functions (as shown in Table 1) are selected to conduct simulation experiments, and the EACSDAM is compared with the standard Cuckoo search algorithm (CS), paper [10] (IACS), and paper [11] (ACSDAM) to analyze the convergence rate and resolution accuracy. The population number is n = 30, and the maximum of iterations is $tmax$ = 600.This experiment is repeated 50 times in the same environment. It is shown in Table 2 that the worst, mean, and variance values of EACSDAM are slightly higher than the ACSDAM in function f7, but the optimal value of EACSDAM is the minimum. Among the other seven functions, the best, worst, mean, and variance of the EACSDAM are all lower than those of the three different algorithms, indicating the EACSDAM has optimized the convergence accuracy and has the best fitness value.

Table 1. Eight standard test functions

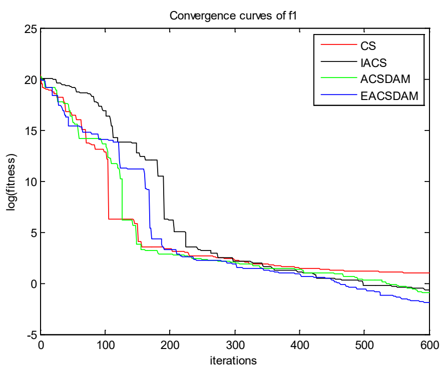
Function name	Dimensionality	Range	Min
$f_1(x)=\frac{\pi}{n}\left\{10\sin(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2\left[1+10\sin^2(\pi y_{i+1})\right]+(y_n-1)^2\right\}+\sum_{i=1}^{n}u(x_i,10,100,4)$ $y_i=1+\frac{x_i+1}{4}u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m\ x_i>a\\0-a<x_i<a\\k(-x_i-a)^m\ x_i<-a\end{cases}$	30	[-50,50]	0
$f_2(x)=\sum_{i=1}^{n}\lvert x\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	30	[-10,10]	0
$f_3(x)=\sum_{i=1}^{n}x_i^2$	30	[-100,100]	0
$f_4(x)=\max_i\{\lvert x_i\rvert,1\le i\le n\}$	30	[-100,100]	0
$f_5(x)=\sum_{i=1}^{n\text{-}1}\left[100(x_{i+1}-x_i^2)^2+(x_i-1)^2\right]$	30	[-30,30]	0
$f_6(x)=\sum_{i=1}^{n}([x_i+0.5])^2$	30	[-100,100]	0
$f_7(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30	[-600,600]	0
$f_8(x)=\left(\frac{1}{500}+\sum_{i=1}^{25}\frac{1}{j+\sum_{I=1}^{2}(x_i-a_{ij})^6}\right)^{-1}$	2	[-65,65]	1

4.2 Convergence Analysis of the Algorithm

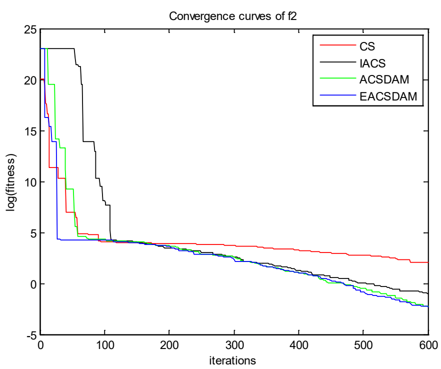
Each algorithm solves the convergence curves of eight standard test functions, as shown in Fig. 1. During the initial iteration period, the convergence rate of EACSDAM in f3 and f5 is higher than that of the other three algorithms. At the late iteration period, the four different algorithms tend to fall into local optimal premature convergence. The EACSDAM still maintains a fast convergence speed and searches for the optimal global value earlier than the four algorithms. The convergence rate of the four algorithms is similar in the early stage in f1, f2, f4, f6, and f7. In the later stage, the convergence speed of EACSDAM is still fast and the fitness value is best. In f8, the early convergence speed of the four algorithms is similar. EACSDAM algorithm searches for the optimal global value at about the 20th iteration, while ACSDAM, IACS and CS algorithms search for the optimal global value after the 30th, 80th, and 95th iteration, respectively. By comparing the convergence curves, it is found that EACSDAM has better stability and convergence speed than other improved cuckoo algorithms.

Table 2. Simulation results of eight functions

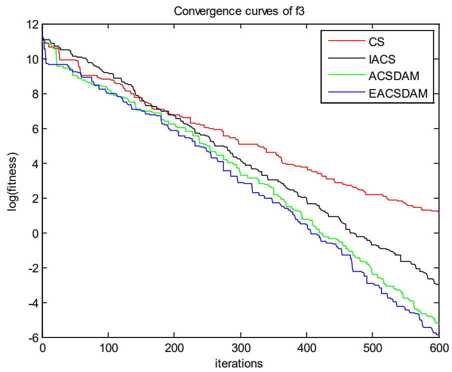
f	Algorithm	Best	Worst	Mean	Std
f1	CS	3.9541	12.0123	5.2014	4.2574
	IACS	0.0959	10.015	2.5641	1.2035
	ACSDAM	0.0857	5.0115	1.9256	1.0456
	EACSDAM	0.0321	3.2508	0.0214	0.5124
f2	CS	2.1984	9.6757	4.4647	2.8410
	IACS	0.3985	9.2680	1.6336	1.4243
	ACSDAM	0.0317	0.3174	0.1715	0.0870
	EACSDAM	0.0291	0.1324	0.0726	0.0733
f3	CS	0.8702	5.2354	1.8036	1.2719
	IACS	0.0292	0.0713	0.0455	0.0156
	ACSDAM	0.0090	0.0094	0.0066	0.0040
	EACSDAM	0.0040	0.0025	0.0045	0.0018
f4	CS	1.0949	11.7039	3.1030	3.1100
	IACS	0.0207	13.3492	1.3793	4.2058
	ACSDAM	0.0019	4.8323	0.4872	1.5267
	EACSDAM	0.0014	3.4018	0.3438	1.0745
f5	CS	1.0949	190.8579	21.0184	59.6800
	IACS	0.0207	118.0130	11.8457	37.3034
	ACSDAM	0.0019	56.2042	5.6244	17.7719
	EACSDAM	0.0014	40.4122	4.0448	12.7782
f6	CS	0.8428	3.4484	2.0169	0.8422
	IACS	0.0171	0.0795	0.0461	0.0221
	ACSDAM	0.0069	0.0086	0.0042	0.0021
	EACSDAM	0.0020	0.0074	0.0043	0.0019
f7	CS	1.0258	3.4484	2.0352	0.8154
	IACS	0.0207	0.1112	0.0555	0.0277
	ACSDAM	0.0019	0.0089	0.0048	0.0024
	EACSDAM	0.0014	0.0656	0.0101	0.0196
f8	CS	0.9980	7.2292	4.7171	2.3517
	IACS	0.9980	7.0259	4.3592	2.4832
	ACSDAM	0.9980	5.8491	3.3507	2.2457
	EACSDAM	0.1280	0.9980	0.3240	0.3789



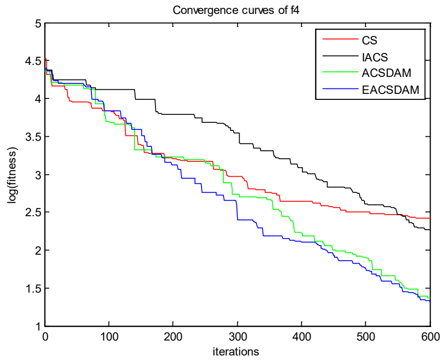
(a)



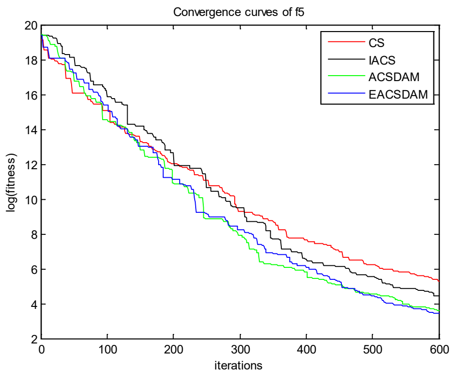
(b)



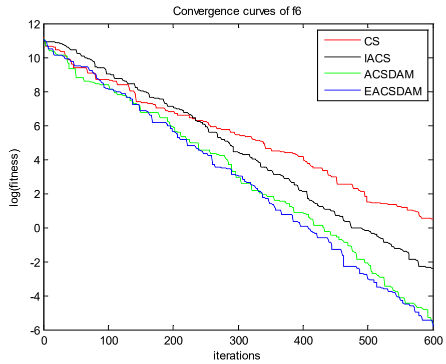
(c)



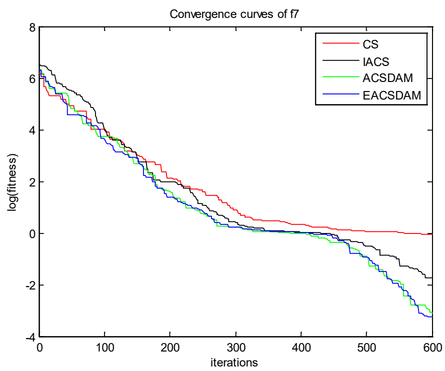
(d)



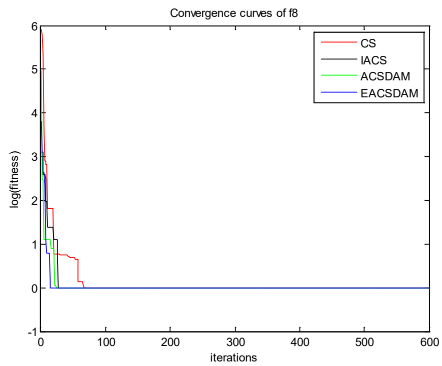
(e)



(f)



(g)



(h)

Fig. 1. Convergence curves of eight functions

5 Optimize the 3D Indoor Visible Light Positioning Method using the EACSDAM Algorithm

In recent years, visible light communication (VLC) technology has developed rapidly. More researchers are using VLC to solve indoor positioning problems [26]. The visual light positioning (VLP) system dramatically increases the utilization rate of luminescence, and Light-emitting diodes (LEDs) are free from electromagnetic interference and consume low energy [10]. Positioning systems based on VLC can be divided into two categories according to different receivers: the image sensor [27-28] and the photodiode (PD). Positioning systems based on VLC can be divided into two categories on the basis of different receiving devices, and the essential category is a photodiode (PD) [29]. In the indoor VLP, the LED lamp constitutes the signal source for data transmission, while PD includes the receiver and is responsible for receiving data [10]. Each LED sends an ID message modulated by code division multiple access (CDMA), which is received by the receiver PD, sampled, and algorithmic to estimate the precise location of the PD.

5.1 Positioning Performance Analysis with Non-rotating PD

In this paper, it conducts simulation experiments on the VLP system used simulation software of Matlab. A room of 5×5×6(m) is established. Four LED lights are set as signal sources, located in the ceiling, and the position coordinates are [5,0,6], [0,0,6], [0,5,6], [5,5,6]. The test is carried out at the height of 1.0m with a resolution of 0.25m, and it generates 121 test points with no rotation for the multi-point positioning experiment. Related parameter Settings of the simulation experiment are in Table 3. Due to the randomness of the optimization algorithm, the positioning algorithm is run 50 times. Positioning results are in Fig. 2. An asterisk is a LED position. Crosses are the actual positions. Triangles are the estimated positions. Positioning results are accurate, and the estimated position approaches the real position.

Table 3. Related parameter settings

Parameter	Value
Emitting the power of LED, P_t	2.2W
Photo-electric conversion efficiency of PD, R_p	0.5 A/W
Equivalent noise bandwidth, B	100 Mb/s
Background photocurrent, I_{bg}	5100 μA
Noise bandwidth factor, I_2	0.562
Noise bandwidth factor, I_3	0.0868
Absolute temperature, T_k	295K
Open-loop voltage gain, G	10
Transconductance of FET, g_m	30mS
Channel noise coefficient of FET, Γ	1.5
Unit-area capacitance of PD, η	112pF/cm2
Population size, N_p	100
Max iteration number, $G_{\max}$	60

CDF curve is in Fig. 3. Positioning error histogram is in Fig. 4. Fig. 3 assumes an acceptable location service coverage rate of 96.1219%. The 3D positioning accuracy of the EACSDAM is 1.3143 cm, and horizontal and vertical positioning accuracy is 1.0706 cm and 1.1325 cm, respectively. In Fig. 4, the average error is 0.2653cm, the standard deviation is 0.1780cm, the maximum error is 2.15cm, and the minimum error is 0.0254cm.

It can be seen from the experiment that optimizing the 3D indoor VLP system using the EACSDAM has a good effect. The results were accurate down to within three millimeters.

5.2 Comparison of Positioning Performance of Four Algorithms

To verify the implementation of the EACSDAM in an indoor positioning system, it selected the meta-heuristic algorithm (Jaya), standard Cuckoo algorithm (CS), and paper [10] (IACS) for comparative analysis with the improved EACSDAM algorithm.

It shows the actual position of PD and the estimated position distribution calculated with four algorithms in Fig. 5. The position calculated with EACSDAM is the closest to the exact position of PD and has the best coverage.

The estimated position error curve of four algorithms is in Fig. 6. It is easy to see that, among 121 test points, the estimated position error of the EACSDAM is the smallest, followed by the IACS, Jaya, and CS.

Fig. 7 is a histogram of the average error of four algorithms' estimated positions. By comparing, the average error calculated by the CS is close to 0.07cm, the average error calculated by the Jaya is close to 0.05cm, and the average error calculated by the IACS is close to 0.0007cm. The average error of the estimated position calculated by the EACSDAM is close to 0.0001cm. The EACSDAM algorithm has the most stable localization.

The results show that among the four algorithms the EACSDAM has the best optimization effect on an indoor 3D VLP system.

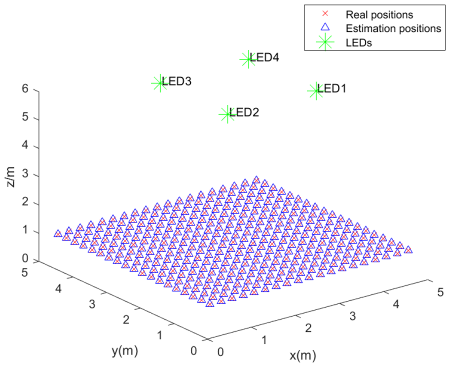


Fig. 2. Real positions and their estimated 3D positions

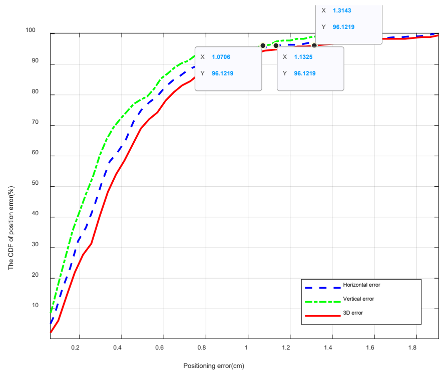


Fig. 3. CDF curves of positioning error

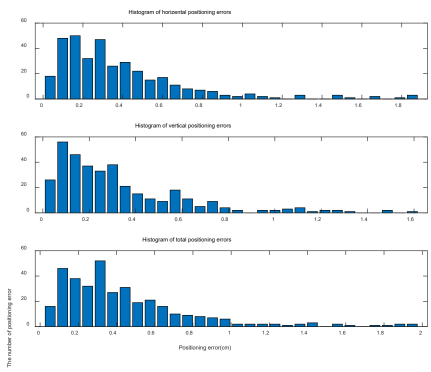


Fig. 4. Histogram of positioning error

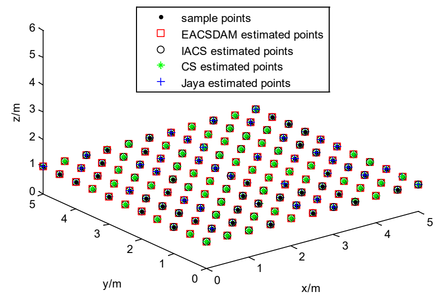


Fig. 5. Distribution diagram of actual location

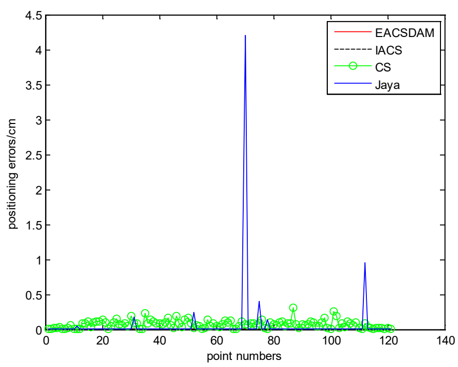


Fig. 6. Curve of estimated position error

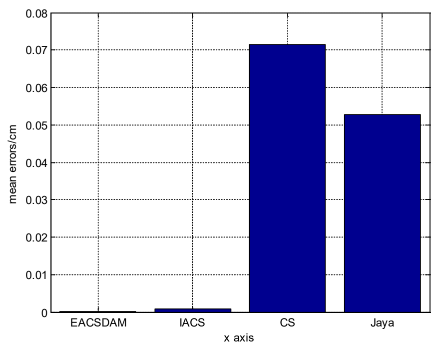


Fig. 7. Comparison of average errors of estimated positions

6 Conclusion

Because of the disadvantages of the standard CS algorithm, which is prone to fall into local optimization and has low search speed, this paper proposes an improved cuckoo search algorithm (EACSDAM) based on elite reverse learning. The EACSDAM adopts the elite opposition-based learning method to optimize the initial population and combines the improved strategy of step size factor and discovery probability in literature [7]. Then, eight standard test functions are selected as the objective function to conduct simulation experiments on the standard Cuckoo search algorithm (CS), literature [6] (IACS), literature [7] (ACSDAM), and the EACSDAM algorithm proposed in this paper. It compares the optimal value, worst value, average value, and variance of the four algorithms, and analyzes their convergence curves. The results show that the improved EACSDAM algorithm can effectively improve the accuracy and convergence speed of the optimization, and balance the global and local search ability.

Finally, the EACSDAM algorithm is applied to the indoor THREE-DIMENSIONAL visible light positioning system. Through the simulation experiment, the estimated position is very close to the actual position, and it indicates that the three-dimensional positioning accuracy is perfect. The CDF curve and the histogram of positioning error are further analyzed, which shows that the 3d positioning accuracy can reach the centimeter level. It selects Meta-heuristic algorithm (Jaya), standard Cuckoo algorithm (CS), and literature [6] (IACS) to carry out experiments with the EACSDAM algorithm. The distribution diagram, estimated position error curve and, histogram of average estimated position error calculated by actual position and its algorithm were compared and analyzed. The results show that, the position of the EACSDAM algorithm is closest to the actual position. The coverage is the best. The estimated position error is the smallest. The positioning is the most stable. In conclusion, the improved EACSDAM algorithm reduces the average positioning error and optimizes the positioning effect in the indoor 3D visible light positioning system, which has a good application value.

7 Acknowledgment

This study is funded by Special Scientific Research zone - Development and application of Ta-pieh Mountain characteristic resources informatization and intelligent equipment (0044020001), the Natural Science Research Project of West Anhui University (0041021002, 0041021003). The authors thank all the agencies for extending their help to conduct this work.

References

[1] Y.-G. Xi, T.-Y. Chai, W.-M. Yun, Survey on Genetic Algorithm, Control Theory and Applications 13(6)(1996) 697-708.
[2] J.-H. Zhang, Q.-S. Gao, X.-H. Xu, A Self-Adaptive Ant Colony Algorithm, Control Theory and Applications 17(1) (2000) 1-3.
[3] N.-G. Zhao, J.-S. Deng, A Review of Particle Swarm Optimization Algorithms, Science and Technology Innovation Herald 12(26)(2015) 216-217. DOI: 10.3969/j.issn.1674-098X.2015.26.104
[4] J.-Q. Wang, F.-L. Wang, Improvement Analysis and Application of Firefly Algorithm, Computer Applications 34(9) (2014) 2552-2556.
[5] X.S. Yang, S. Deb, Cuckoo Search via Levy Flights, in: Proc. 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), 2009.
[6] X.S. Yang, S. Deb, Engineering Optimization by Cuckoo Search, International Journal of Mathematical Modeling and Numerical Optimization 1(4)(2010) 330-343.
[7] P.R. Srivastava, R. Khandelwal, S. Khandelwal, S. Kumar, S.S. Ranganatha, Automated Test Data Generation Using Cuckoo Search and Tabu Search (CSTS) Algorithm, Journal of Intelligent Systems 21(2)(2012) 195-224.
[8] X.-F. Zhang, X.-Y. Wang, Survey of cuckoo search algorithm, Computer engineering and applications 54(18)(2018) 8-16.
[9] H.-N. Zhang, X.-M. You, S. Liu, Z.-Q. Liu, Interactive Learning Cuckoo Search Algorithm, Computer engineering and applications 56(7)(2020) 147-154.
[10] C.-C. Jia, T. Yang, C.-J. Wang, M.-L. Sun, HighAccuracy 3D Indoor Visible Light Positioning Method Based on the Improved Adaptive Cuckoo Search Algorithm, Arabian Journal for Science and Engineering 47(2)(2022) 2479-2498.
[11] Y. Yang, X.-K. Zuo, M.-S. Fu, S.-H. Yu, C.-C. Jia, Adaptive Cuckoo Search Algorithm based on Dynamic Adjustment Mechanism, Journal of Computers 32(5)(2021) 171-183.

[12] C. Huang, J.-Y. Fei, L.-Y. Wang, X.-D. Liu, Particle Filter Method Based on Multi-strategy Difference Cuckoo Search Algorithm, Transactions of the Chinese society for agricultural machinery 49(4)(2018) 265-272.

[13] Y.-H. Lin, L.-J. Wang, Elite Opposition-based Learning Based Simplex Crossover Cuckoo Search Algorithm, Journal of Zhengzhou University (engineering science) 38(6)(2017) 33-38.

[14] H.-T. Tang, J.-Y. Liu, An Improved Cuckoo Algorithm for Distributed Flexible Flow-shop Scheduling Problem with Transport Time Consideration, Operations Research and Management Science 30(11)(2021) 76-83.

[15] Y.-Y. Xiang, W.-M. Jia, F.-G. Zhang, Sparsely-distributed linear array pattern optimization based on modified chaotic cuckoo search algorithm, Modern Electronics Technique 44(13)(2021) 7-12.

[16] Z.-Z. Zhang, X.-S. He, Q.-L. Yu, X.-S. Yang, Cuckoo Algorithm for Multi-stage Dynamic Disturbance and Dynamic Inertia Weight, Computer Engineering and Applications 58(1)(2022) 79-88.

[17] R. Zang, X.-X. Liu, Segment Weighted Cuckoo Algorithm and Its Application, Computer Science 47(6A)(2020) 119-123.

[18] Y. Li, Z.-Y. Shang, J.-S. Liu, Improved cuckoo search algorithm for function optimization problem, Computer science 47(1)(2020) 219-230.

[19] Z. Wang, C.-X. Jia, Y.-H. Sun, Parasitized breeding and nestlings growth in oriental cuckoo, Chinese Journal of Zoology 39(1)(2004) 103-105.

[20] F. Wang, H.-S. He, Y. Wang, The Cuckoo search Algorithm based on Gaussian Disturbance, Journal of Xi'an Polytechnic University 25(4)(2011) 566-569.

[21] K. Liu, L.-L. Zhao, H. Wang, Whale Optimization Algorithm Based on Elite Opposition-based and Crisscross Optimization, Journal of Chinese Computer Systems 41(10)(2020) 2092-2097.

[22] X.-M. Meng, C.-C. Cai, Whale Optimization algorithm based on Elite Reverse Learning and Levy Flight, Electronic measurement technology 44(20)(2021) 82-87. https://kns.cnki.net/kcms/detail/11.2175.TN.20211125.0843.004.html.

[23] H.R. Tizhoosh, Opposition-based learning: a new scheme for machine intelligence, in: Proc. of IEEE Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), 2005.

[24] Q. He, M.-M. Huang, X. Wang, Elite Opposition Learning-Based Dimension by Dimension Improved Dragonfly Algorithm, Journal of Nanjing normal university (natural science) 42(3)(2019) 65-72.

[25] C.-W. Xie, Z.-J. Wang, X.-W. Xia, Multi-Objective Evolutionary Algorithm Based on Archive-Elite Learning and Opposition-Based Learning, Chinese journal of computers 40(3)(2017) 757-772.

[26] P.-F. Wang, W.-P. Guan, S.-S. Wen, Y.-J. Xie, Y.-X. Wu, M.-Q. Zhang, High Precision Indoor Visible Three-Dimensional Positioning System Based on Immune Algorithm, Acta Optica Sinica 38(10)(2018) 1006007-1-1006007-11.

[27] P. Huynh, M. Yoo, VLC-based positioning system for an indoor environment using an image sensor and an accelerometer Sensor, Sensors (Basel) 16(6)(2016) 783.

[28] P. Davidson, R. Piché, A survey of selected indoor positioning methods for Smartphones, IEEE Communications Surveys & Tutorials 19(2)(2017) 1347-1370.

[29] H. Steendam, T.Q. Wang, J.T. Armstrong, Theoretical lower bound for indoor visible light positioning using received signal strength measurements and an aperture-based Receiver, Journal of Lightwave Technology 35(2)(2017) 309-319.