Research Article

A Novel Crow Swarm Optimization Algorithm (CSO) Coupling Particle Swarm Optimization (PSO) and Crow Search Algorithm (CSA)

Ying-Hui Jia,1 Jun Qiu,2,3 Zhuang-Zhuang Ma,2 and Fang-Fang Li1

1College of Water Resources & Civil Engineering, China Agricultural University, Beijing 100083, China
2State Key Laboratory of Hydroscience & Engineering, Tsinghua University, Beijing 100084, China
3State Key Laboratory of Plateau Ecology and Agriculture, Qinghai University, Xining 810016, China

Correspondence should be addressed to Fang-Fang Li; lff@cau.edu.cn

Received 9 December 2020; Revised 9 April 2021; Accepted 8 May 2021; Published 22 May 2021

Academic Editor: José Alfredo Hernández-Pérez

Copyright © 2021 Ying-Hui Jia et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The balance between exploitation and exploration essentially determines the performance of a population-based optimization algorithm, which is also a big challenge in algorithm design. Particle swarm optimization (PSO) has strong ability in exploitation, but is relatively weak in exploration, while crow search algorithm (CSA) is characterized by simplicity and more randomness. This study proposes a new crow swarm optimization algorithm coupling PSO and CSA, which provides the individuals the possibility of exploring the unknown regions under the guidance of another random individual. The proposed CSO algorithm is tested on several benchmark functions, including both unimodal and multimodal problems with different variable dimensions. The performance of the proposed CSO is evaluated by the optimization efficiency, the global search ability, and the robustness to parameter settings, all of which are improved to a great extent compared with either PSO and CSA, as the proposed CSO combines the advantages of PSO in exploitation and that of CSA in exploration, especially for complex high-dimensional problems.

1. Introduction

Bio-inspired optimization algorithms have become increasingly popular over the past decade due to their simplicity of implementation, robustness, and ability of parallel computation [1]. Although the specific principles and procedures of bio-inspired optimization algorithms are various, it is a consensus that an effective search technique must strike a balance between exploring new regions in the search space and exploiting known promising regions [2, 3].

Particle swarm optimization (PSO), developed by Kennedy [4] in 1995, is one of the most popular bio-inspired algorithms with wide applications in industrial design [5], energy distribution [6], economic dispatch [7], and so on. PSO is inspired by the social behavior of bird flocking or fish schooling. A swarm of particles moves inside a bounded search space and cooperates to identify the best solution under the guidance of social attraction and cognitive attraction, which are used with the aim of exploiting and controlling the cooperation within the swarm. Meanwhile, the exploration of the search space is considered by an inertia factor weighing the movement of particles and the magnitude of their velocities. Nevertheless, similar to other bio-inspired algorithms, PSO also suffers from the bane of premature convergence and entrapment in local optima when solving complex multimodal problems [8, 9], i.e., the algorithm has difficulty exploring all regions, and some of the peaks are easily missed. Various improvements to PSO have been made to balance exploration and exploitation search.

Crow search algorithm (CSA), proposed by Askarzadeh [10] in 2016, is a newly developed algorithm inspired by the strategic behavior of crows while searching food, thievery, and chasing behavior. CSA is characterized by easy implementation, less parameter setting, and relatively strong development capacity in the searching process [11].
However, CSA also suffers from low search precision, high possibility of getting into the local optimum, and premature convergence, especially for multidimensional optimization problems [11], which may result from two important features in the basic CSA [12, 13]: (1) there is no criterion for choosing the destination and the selection is done randomly between all crows; (2) the amount of flight length is a constant value which may cause inappropriate searching by the crows in the solution space that results in trapping in the local optimum. In the past few years, most of the recent research studies focused on the application of the algorithm in different scientific fields with suitable modification. In case of feature selection, Sayed et al. [3] combined chaos with CSA to enhance the performance and convergence speed, while Chaudhuri and Sahu [14] proposed time varying flight length to obtain better results. As for energy optimization, Makhdoom and Askarzadeh [15] incorporated adaptive chaotic awareness probability into CSA to optimize operation of photovoltaic/diesel hybrid energy system. To overcome unbalanced exploration and exploitation phases, Shekhawat and Saxena [13] improved the algorithm by a cosine function and opposition-based learning concept. CSA to enhance the performance and convergence speed, while Chaudhuri and Sahu [14] proposed time varying flight length to obtain better results. As for energy optimization, Makhdoom and Askarzadeh [15] incorporated adaptive chaotic awareness probability into CSA to optimize operation of photovoltaic/diesel hybrid energy system. To overcome unbalanced exploration and exploitation phases, Shekhawat and Saxena [13] improved the algorithm by a cosine function and opposition-based learning concept. CSA variants have been categorized into modified versions [16–18] and hybrid versions [19–21].

Both PSO and CSA are population-based techniques. For optimization problems, the most critical part is to find as many local optimal solutions as possible and gradually move to the global best. Therefore, the most important thing is to give the solutions some degree of freedom during iteration while making the optimization process efficient, i.e., to balance between exploration and exploitation. It can be inferred from the previous studies that PSO is more capable in controlling exploitation by social and cognitive factors, but its ability of exploration in the unknown region is highly influenced by the current best solutions in spite of the existence of random numbers affecting the acceleration coefficients. In contrast, the individual in CSA moves completely randomly when it is aware of being tracked to get rid of the stalker. Hence, CSA performs better in exploring new regions in the search space. A few studies have proposed hybrid models combining PSO and CSA. Babu et al. [22] have proposed a model that connects the update procedure of CSA with that of PSO, i.e., every particle updates its position twice for each iteration. Crow Search Algorithm Auto-Drive PSO [23] uses CSA as the outer algorithm to optimize the sizing of renewable distributed generations; then, PSO is applied for the optimal power flow of the power distribution system as inner optimization. Both algorithms execute the position updating procedure using formulas from PSO and CSA during each iteration. In [22], it is used to address the same problem, while Farh et al. [23] divide a problem into two parts which are optimized by CSA and PSO, respectively. However, their algorithms are applied on specific situations without detailed evaluation on further performance. To take advantage of the ability of CSA in exploration and the ability of PSO in exploitation, this paper proposed a new Crow Swarm Optimization algorithm (CSO). Besides, moving towards the (current) best particle in the swarm and the best position is autonomously found so far, and each individual also has a probability to stalk the best ever solution of another individual. The proposed CSO essentially adds the information sharing mechanism in CSA and explores the unknown region between the two individuals using PSO. Numerical benchmark problems are tested using the proposed algorithm, and the results prove that it performs better in balancing exploitation and exploration with a few parameters.

The rest of the paper is organized as follows. Section 2 introduces the proposed CSO algorithm as well as traditional PSO and CSA. Section 3 is the performance evaluation on several benchmark functions with an example of application in real-world problem. Section 4 draws the conclusions.

2. Crow Swarm Optimization Algorithm (CSO)

2.1. Standard PSO. PSO imitates the foraging behavior of bird flocks and consists of a collection of agents, so-called particles. The position of each particle represents a solution, and each particle retains the information of its current position, velocity, and its personal best found position within the search space. The particles move to new positions with a velocity vector, which is iteratively updated based on its attraction towards the best position found by the particle and the best position found by any particle within the search space, i.e., the position of particle $i$ at the $r$th iteration is determined by the position vector $x_i^r$ and the velocity vector $v_i^r$. The updating formulas for particle velocity and position are shown in Equations (1) and (2):

$$v_{i}^{r+1} = \omega v_{i}^{r} + c_1 r_1(\text{pbest}_i - x_{i}^{r}) + c_2 r_2(\text{gbest} - x_{i}^{r}),$$

$$x_{i}^{r+1} = x_{i}^{r} + v_{i}^{r+1},$$

where $n$ is the number of the particles in population, $\omega$ is the inertia weight coefficient, $c_1$ and $c_2$ are called acceleration coefficients, indicating the cognition degree of the particle to the individual and the society, respectively, $r_1$ and $r_2$ are random numbers in $[0, 1]$, $\text{pbest}_i$ represents the current optimal position of the $i$th particle, $\text{gbest}$ represents the best position of the current population, $v_{\text{min}}$ and $v_{\text{max}}$ are the lower and upper limit of particle update velocity, respectively; in this paper, $v_{\text{min}} = -v_{\text{max}}$, $v_{\text{min}}$ and $v_{\text{max}}$ are the minimum and maximum positions of particles, respectively.

The key idea of PSO is that, under the dominant action of better particles, each particle is close to the global optimal position. Although there exist random numbers $r_1$ and $r_2$ to avoid that the movements of the particles are completely decided by its current optimal position and the best position of the current population, it can be inferred that these two factors have great impact on the movement of particles. Thus, the initial position of the particles affects the search process greatly, and a large part of unknown regions can be hardly paid attention to. Such characteristic provides a strong capacity of exploiting the known information, while it is weaker in the exploration of new regions, which lead to
premature convergence, i.e., the optimization easily converges to a local optimum.

2.2. Standard CSA. CSA is a new search algorithm inspired by the behavior of crows hiding and stealing food. It is non-greed and can increase the variety of generated solutions. The principles of CSA are listed as follows: (1) crows live in the form of flock, (2) each crow can memorize the position of its hiding places and steal food from other crows, and (3) with a certain probability, the crow can be aware of being stalked and then protect its food from being stolen by flying randomly. The position of each crow represents a solution \( x^j \). During the iteration, crow \( i \) tracks a random crow \( j \). If crow \( j \) is not aware of being tracked (i.e., \( r_j \geq \text{AP} \)), crow \( i \) approaches the hiding place (\( pb^j \)) of crow \( j \); while if crow \( j \) knows that crow \( i \) is chasing it, crow \( j \) fools crow \( i \) by going to a random location in the search space to protect its hiding place from detection. The mathematical expression is

\[
\begin{align*}
\mathbf{x}^{i+1} = \mathbf{x}^i + r_1 \cdot f_l \cdot (pb^j - \mathbf{x}^i), & \quad r_j \geq \text{AP}, \\
\text{a random position,} & \quad \text{else},
\end{align*}
\tag{3}
\]

where \( r_j \) and \( r_3 \) are random numbers with uniform distribution between 0 and 1, \( f_l \) is the length of a crow’s flight, \( \text{AP} \) is the perceptual probability of crow \( j \), and \( pb^j \) is the location where the current crow \( j \) stores food, which is equivalent to the historical optimal solution the current crow \( j \) has found.

CSA is a population-based optimization algorithm that is fairly simple with only two adjustable parameters (flight length \( f_l \) and perceived probability \( \text{AP} \)), which in turn makes it very attractive for different engineering applications. In CSA, the parameters of perceived probability are directly used to control the diversity of the algorithm. Compared with genetic algorithm (GA), PSO, and harmony search algorithm (HS), CSA has fewer parameters to adjust and is easier to implement. Moreover, the individual in CSA has the possibility of reaching a total random position, and thus, it has stronger ability of exploring newly unknown regions. However, CSA has lack of the criterion for choosing the destination and the selection is done randomly between crows; besides, the flight length is a constant value. Such characteristics of CSA lead to a weaker ability of exploitation of current information compared with PSO, resulting in low search precision, high possibility of getting into the local optimum, and premature convergence, especially for multidimensional optimization problems.

2.3. Proposed CSO. To take advantage of both the PSO in exploitation and CSA in exploration, this study proposes a new crow swarm optimization algorithm (CSO). The movement of particles in CSO is influenced by the best position found so far by the particle itself and the best position identified so far by the whole swarm, as in PSO. Meanwhile, the particle keeps an eye on each other, i.e., there exists a possibility that its movement is determined by the best position identified by the whole swarm and the best solution found so far by another particle; in CSA, it tracks the position where another crow hides the food. The formula updating the flying velocity of crows in CSO is

\[
\begin{align*}
\mathbf{v}^{i+1} = \mathbf{v}^i + c_3 r_3 (pb^j - \mathbf{x}^i) + c_2 r_2 (gb^j - \mathbf{x}^i), & \quad r_j \geq \text{AP}, \\
\mathbf{v}^{i+1} = \mathbf{v}^i + c_1 r_1 (pb^j - \mathbf{x}^i) + c_2 r_2 (gb^j - \mathbf{x}^i), & \quad \text{else},
\end{align*}
\tag{4}
\]

where \( c_3 \) represents the degree of the influence of individual \( j \) on individual \( i \) and \( r_3 \) is a random number within [0, 1]. The updating velocity in equation (4) is also limited by \( \mathbf{v}_{\text{min}} \) and \( \mathbf{v}_{\text{max}} \).

When \( r_j \geq \text{AP} \), the individual \( i \) decided to track individual \( j \), and the velocity of individual \( i \) is affected by inertia velocity, global optimal solution, and current optimal solution of individual \( j \). Otherwise, the velocity of individual \( i \) is composed of inertia velocity, global optimal solution, and local optimal solution of individual \( i \). When the velocity of individual \( i \) is obtained, the position of individual \( i \) in the next iteration is calculated by equation (2).

Figure 1 shows the flowchart of the proposed CSO. It can be seen from Figure 1 that the difference between proposed CSO and other two algorithms mainly comes from the method of updating the particle speed and position. PSO pays more attention to the optimization efficiency and aims to be closer to the current optimal solution in the iterative process, thus resulting in a strong ability of exploiting the current known information, while CSA gives greater freedom to the algorithm to ensure the diversity of solutions, which produces greater ability of exploring unknown regions. The proposed CSO combines the advantages of both the PSO and the CSA to reach a better balance between increasing randomness and improving efficiency, i.e., between exploration and exploitation.

2.4. Individual Movement. Individual movement directly affects the performance of the swarm intelligence algorithm. Figure 2 shows the schematics of how an individual updates its position in (a) PSO, (b) CSA, and (c) CSO. The movement of the individuals in PSO is rather fixed and determined by its inertia, its current best position, and the best position identified by the whole swarm. Both CSA and CSO provide alternatives with a certain probability to better maintain the diversity of solutions. However, it can be seen that CSA is less efficient compared to CSO due to the divergence of the way the solutions update. CSO preserves the optimization efficiency of PSO with a possibility of exploring larger regions. In CSO, the larger the value of the probability parameter \( \text{AP} \), the stronger directional characteristic it has. When \( \text{AP} = 1 \), CSO degenerates into standard PSO; while when \( \text{AP} = 0 \), the individuals in CSO always randomly select the historical optimal solution of an individual in the population to track with less exploitation of other known information. It should be illustrated that when \( \text{AP} \) is equal to 0, CSO does not degenerate into CSA, but it retains the characteristics of rich diversity of CSA.
3. Experiments and Discussion

3.1. Standard Benchmark Functions. Extensive experiments are conducted on a set of well-known benchmark functions to ascertain the performance of the proposed CSO, including the global optimization problems shown in Table 1. The experimental settings are as follows: the maximum iteration number is 1000, and the swarm size is set to be 100; the parameters in equations (1) to (4) are set to be $c_1 = c_2 = c_3 = 2$, $\omega = 0.5$, $AP = 0.2$, $fI = 2$, and $v_{max} = [2]^D$.

In Table 1, $D$ indicates the dimension of the problem and $x_i$ represents the value of the decision vector in dimension $i$.

These test functions have included both unimodal functions (such as Sphere, Schwefel’s problem 1.2, Step, and Rosenbrock) and multimodal functions (such as Ackley and Griewank). Unimodal functions with one global optimum and no local optima are used to investigate the exploitation level and the convergence rate of the algorithms, while multimodal benchmark functions with multiple local optima are used to test the ability of algorithms to avoid entrapment in local optima and explore new unknown regions.

In addition, a parameter representing the improvement efficiency is introduced to evaluate the progress of CSO compared with standard PSO or CSA:

$$\eta = \frac{\text{optimal}_{\text{PSO or CSA}} - \text{optimal}_{\text{CSO}}}{\text{optimal}_{\text{CSO}}}$$  \hspace{1cm} (5)

3.2. Optimization Efficiency. Table 2 shows the optimization results of the three algorithms for different target functions, respectively. In addition to the above-described three methods, recently, well-accepted bio-inspired optimization algorithms MBO and MS are also examined to better illustrate the performance of CSO. Each test conducts 25 trials to eliminate the influence of the initial population and
| Benchmark functions       | Formula                                                                 | Dimension | Up boundary | Lower boundary | Optima |
|---------------------------|--------------------------------------------------------------------------|-----------|-------------|---------------|--------|
| Sphere                    | $f(x_i) = \sum_{i=1}^{D} x_i^2$                                          | 10        | 100         | -100          | 0      |
| Schwefel’s problem 1.2    | $f(x_i) = \sum_{i=1}^{D} (x_i - 0.5)^2$                                   | 10        | 100         | -100          | 0      |
| Rosenbrock                | $f(x_i) = \sum_{i=1}^{D} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2]$            | 10        | 2           | -2            | 0      |
| Rastrigin’s               | $f(x_i) = \sum_{i=1}^{D} [x_i^2 - 10 \cos(2\pi x_i) + 10]$               | 10        | 5.12        | -5.12         | 0      |
| Ackley                    | $f(x_i) = -20 \exp(-0.2\sqrt{\sum_{i=1}^{D} x_i^2}) - \exp(\sum_{i=1}^{D} \cos(2\pi x_i)) + 20 + e$ | 10        | 32          | -32           | 0      |
| Michalewicz               | $f(x) = -\sum_{i=1}^{D} \sin(x_i)\sin(20\pi x_i/\pi)$                   | 10        | $\pi$       | 0             | -9.66015 |
| Branin                    | $f(x) = (x_2 - 5.1/4\pi x_1^2 + 5/\pi x_1 - 6)^2$                       | 15        | -5          | 0.397887      |        |
| 10 $(1 - 1/8\pi) \cos(x_i) + 10$ | $f(x) = \sum_{i=1}^{D} x_i^2/4000 - \prod_{i=1}^{D} \cos(x_i/\sqrt{7}) + 1$ | 10        | 600         | -600          | 0      |
| Griewank                  | $f(x) = D/4 \sum_{i=1}^{D} (x_i - 0.5)^2$                                 | 10        | 1           | 0             | 0      |
| Normalised paraboloid     | $f(x) = \sum_{i=1}^{D} x_i^2/D$                                         | 10        | 100         | -100          | 0      |
| Step                      | $f(x) = \sum_{i=1}^{D} (x_i + 0.5)^2$                                    | 10        | 5           | -5            | -1.03628 |
| Six hump camel bsck       | $f(x) = 4x_1^2 - 2.1x_1^4 + 1/3x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$       | 2         | 5           | -5            | -1.03628 |
| CEC2011 problem1          | CEC2011 problem1: parameter estimation for frequency-modulated (FM) sound waves | 6         | 6.35        | -6.4          | 0      |
| CEC2011 problem2          | Lennard-Jones potential problem [24]                                     | 15        | [4, 4, pi, . . . | [0, 0, 0, . . . | 0      |
| Function   | Algorithm | Optimum | Mean      | Standard |
|------------|-----------|---------|-----------|----------|
|            | MBO       | 2.64E-14| 1.11E-10  | 2.23E-10 |
| Sphere     | MBO       | 1.62E-08| 2.40E+03  | 2.75E+03 |
|            | CSA       | 3.21E+01| 1.63E+02  | 8.38E+01 |
|            | PSO       | 3.10E-14| 8.72E-03  | 2.95E-02 |
|            | CSO       | 1.75E-24| 4.27E-23  | 5.69E-23 |
| Schwefel's | MBO       | 5.45E-20| 1.91E-18  | 2.66E-18 |
|            | CSA       | 1.27E+03| 2.27E+03  | 6.48E+02 |
|            | PSO       | 4.55E-05| 9.96E-01  | 2.10E+00 |
|            | CSO       | 2.46E-24| 9.28E-23  | 1.22E-22 |
| Rosenbrock | MBO       | 1.63E-02| 1.64E+00  | 1.81E+00 |
|            | MS        | 3.28E+00| 7.99E-01  | 1.04E+00 |
|            | CSA       | 3.09E-01| 8.27E-01  | 4.20E-01 |
|            | PSO       | 1.98E-05| 6.31E+00  | 1.39E+01 |
|            | CSO       | 2.50E-26| 1.79E+01  | 2.77E+01 |
| Ackley     | MBO       | 6.53E-08| 1.23E-05  | 1.37E-05 |
|            | MS        | 1.16E-11| 9.98E-10  | 1.16E-09 |
|            | CSA       | 3.50E+00| 5.82E+00  | 1.29E+00 |
|            | PSO       | 4.04E-03| 2.59E+00  | 1.29E+00 |
|            | CSO       | 7.39E-13| 2.90E-12  | 2.23E-12 |
| Rastrigin's| MBO       | 2.82E-12| 8.25E-01  | 1.48E+00 |
|            | MS        | 0.00E+00| 1.42E-16  | 4.82E-16 |
|            | CSA       | 1.68E+01| 2.86E+01  | 5.72E+00 |
|            | PSO       | 4.08E+00| 1.84E+01  | 9.29E+00 |
|            | CSO       | 3.98E+00| 1.46E+01  | 9.18E+00 |
| Michalewicz| MBO       | -9.38E00| -9.10E-00 | 1.41E-01 |
|            | MS        | -9.58E+00| -8.69E+00 | 5.44E-01 |
|            | CSA       | -7.14E+00| -6.31E+00 | 4.16E-01 |
|            | PSO       | -9.12E+00| -7.74E+00 | 8.73E-01 |
|            | CSO       | -9.29E+00| -7.64E+00 | 9.12E-01 |
| Brannin    | MBO       | 3.98E-01| 4.02E-01  | 1.25E-02 |
|            | MS        | 3.98E-01| 3.98E-01  | 1.11E-16 |
|            | CSA       | 3.98E-01| 3.98E-01  | 1.11E-08 |
|            | PSO       | 3.98E-01| 3.98E-01  | 1.11E-16 |
|            | CSO       | 3.98E-01| 3.98E-01  | 1.11E-16 |
| Griewank   | MBO       | 6.66E-15| 2.61E+00  | 5.22E+00 |
|            | MS        | 0.00E+00| 0.00E+00  | 0.00E+00 |
|            | CSA       | 1.37E+00| 2.52E+00  | 7.30E-01 |
|            | PSO       | 6.16E-02| 2.75E-01  | 1.47E-01 |
|            | CSO       | 2.96E-02| 1.29E-01  | 1.03E-01 |
| Normalised paraboloid | MBO | 3.23E+15| 4.28E-11  | 7.50E-11 |
|            | MS        | 1.39E-14| 9.55E-14  | 6.63E-14 |
|            | CSA       | 3.11E-04| 2.00E-03  | 1.34E-03 |
|            | PSO       | 1.27E-18| 3.00E-08  | 1.21E-07 |
|            | CSO       | 4.92E-30| 6.13E-28  | 1.98E-27 |
| Step       | MBO       | 0.00E+00| 0.00E+00  | 0.00E+00 |
|            | MS        | 0.00E+00| 0.00E+00  | 0.00E+00 |
|            | CSA       | 1.80E+01| 1.72E+02  | 1.08E+02 |
|            | PSO       | 0.00E+00| 3.56E+00  | 2.98E+00 |
|            | CSO       | 0.00E+00| 0.00E+00  | 0.00E+00 |
| Six hump camel back | MBO | 1.17E-16| 1.70E-13  | 4.04E-13 |
|            | MS        | 5.13E-09| 4.00E-24  | 8.99E-24 |
|            | CSA       | -1.03E+00| -1.03E+00 | 4.46E-05 |
|            | PSO       | -1.03E+00| -1.03E+00 | 0.00E+00 |
|            | CSO       | -1.03E+00| -1.03E+00 | 2.09E+07 |
enhance the reliability of the optimization results. Ten out of thirteen test functions show that CSO finds the best optimum values among compared to other algorithms. However, the performance of CSO on Rastrigin’s function, Michalewicz function, and Griewank function is inferior to that of MS or MBO. For Rosenbrock function and CEC 2011 problem of parameter estimation for frequency-modulated (FM) sound waves, CSO has its best performance extremely close to 0, which is never reached by neither MS nor MBO. However, how to maintain this potential in each run still needs further research. It can be seen from the data that, in most cases, CSO performs better than both PSO and CSA, with not only a smaller value of the optimum but also less standard deviation. The results indicate that CSO has stronger optimization ability with higher stability. For some benchmark problems, the minimum value obtained by CSO is even different in magnitude from both PSO and CSA, indicating that the optimization performance of CSO has been greatly improved. CSA performs better in specific problems where the feasible region is relatively small, such as CEC2011 problem2. However, when the feasible range is in the hundreds of orders of magnitude, CSO’s performance can degrade dramatically. Although we tried to change the parameters of the algorithm, it was time-consuming and often difficult to obtain the desired results.

Figure 3 shows the variation of the optimums with the increase of the variables dimensions for (a) the unimodal problem of sphere function and (b) step function, as well as (c) the multimodal problem of Rastrigin’s function and (d) Ackley function under other fixed conditions. It can be found from Figure 3 that CSO is able to maintain a relative high exploitation level for unimodal function with high dimensions. To improve the quality of the solution for high-dimensional problems, the size of the population is usually enlarged, which leads to a great increase of the calculation cost. Figure 3(a) indicates that, for a 10D problem, the optimum resulting from CSO still remains in the order of $10^{-23}$, while those from CSA and PSO rise to $10^2$ and $10^{-3}$, respectively. Moreover, CSO also has the shortest error bars, which represent the stability of the algorithm. Figure 3 demonstrates that CSO performs superiorly for high-dimensional unimodal problems.

The improvement made by CSO on the multimodal problems varies with different problems. For Ackley function, Table 1 shows that the optimization efficiency of CSO is still significantly better than that of CSA and PSO. The three algorithms perform the worst when solving Rastrigin’s problem, in which case, CSO is still the best among them. Table 1 also indicates that the improvement efficiency $\eta$ of CSO is as high as 0.96 and 0.26 compared with CSA and PSO, respectively. Figure 3(c) shows the optimization results of Rastrigin’s function with different dimensions. It is clear that the optimization results degrade significantly as the dimension increases. Nevertheless, CSO has the weakest degradation among the three algorithms. The same trend could also be observed in Figure 3(d).

### 3.3. Global Search Abilities

Figure 4 shows the evolution of the global optimum in the optimization process of the three algorithms. As suggested in Figure 4(a), PSO converges earliest in the iteration, and there is almost no improvement after 100 iterations. However, for CSA and CSO, improvements can still be observed even at the end of the iteration. Such phenomenon indicates that PSO is prone to premature convergence, while CSA and CSO still preserve the diversity of solutions to a certain extent during the optimization process, i.e., have a stronger ability of exploring new regions in the search space. It can be seen from Figure 4(a) that although CSA can avoid being trapped in local optima, and its ability to find a better solution is weak. In the later iteration, the solution obtained by CSO is smaller compared to CSA. To better investigate the optimization process, a concept of turning point is proposed here. When the optimum of a certain generation is better than that of the previous generation with a certain amplitude (in this study, it refers to the fourth decimal point or above of the objective value being improved), the current generation is defined as a turning point. Table 3 shows the statistics of turning points for the three algorithms in 25 trials. For Rastrigin’s function, the average number of the turning points of PSO is much higher than that of CSA and CSO, but most of them are centralized in the first 100 iterations and the change for each turn is small. As for CSA and CSO, the last turning point is

### Table 2: Continued.

| Function          | Algorithm | Optimum       | Mean         | Standard |
|-------------------|-----------|---------------|--------------|----------|
| CEC2011 problem1  | MBO       | $1.18E + 01$  | $1.95E + 01$ | $3.55E + 00$ |
|                   | MS        | $1.67E + 02$  | $1.91E + 01$ | $6.20E + 00$ |
|                   | CSA       | $1.60E + 01$  | $2.06E + 01$ | $2.53E + 00$ |
|                   | PSO       | $8.42E + 00$  | $1.87E + 01$ | $4.82E + 00$ |
|                   | CSO       | $1.27E - 21$  | $1.14E + 01$ | $8.68E + 00$ |

Note. The bold fonts indicate the algorithm performs best for the corresponding problem.
over 800, verifying that both CSA and CSO have a better ability to jump out of the local optimal solution, i.e., the ability of exploration. Since CSA gives too much freedom to the algorithm, the amount of turning point is least, indicating its low efficiency. CSO possesses a moderate number of turn with relative high change, which makes it a more balanced algorithm.

Figure 4(b) shows the optimization process of the three algorithms for Ackley function. For PSO, there are several times when the optimum is trapped around 2. The optimization process of CSA looks like a staircase and improved every several iterations. It can be inferred that a certain number of iterations is still in need to get to 0. The best results are obtained from CSO, for which the optimums of all trails are quite close to 0. The calculated turning point in Table 3 shows that, after about 365 iterations, CSO almost reaches the global optimum value. The purple line (CSO) first sandwiched between the yellow line (PSO) and the blue line (CSA), and then, it passes these two lines coming to the bottom of the diagram in both Figures 4(a) and 4(b). CSA introduces an antitrack mechanism so that more randomness is given to the algorithm, which greatly reduces the possibility of entrapment in a local optimum at the expense of efficiency [13]. PSO employs local and global optimal solutions to direct the optimization process, so it has higher optimization efficiency but also higher probability of getting stuck. Based on the above results, it can be concluded that CSO absorbs the advantages of PSO and CSA while
overcoming the shortcomings of both, so it is able to approach the global optimal solution quickly while maintaining a certain diversity of solutions.

3.4. Robustness to Parameter Settings. Table 4 is the optimization results of two unimodal functions, indicating the impact of the setting of parameter $AP$ on the algorithm performance with other parameters unchanged. Given that $AP \in [0, 1]$, when $AP$ takes the values of the two endpoints, the optimization result is always the worst. For $AP = 1$, the result of CSO is close to that of PSO. For unimodal problems such as sphere functions, the performance of CSA deteriorates with the increase of $AP$, while for Schwefel’s problem 1.2, the results of CSA is unstable and do not have obvious rule; thus, the tuning of $AP$ is a big challenge.

Figure 5 shows the impact of $AP$ on solving more complex multimodal problems. It can be seen that the performance of CSO is not stable when $AP$ is close to endpoints. For the case $AP = 1$, CSO degenerates into PSO. For CSA algorithm, when $f1 = 2$, the optimal value of $AP$ is concentrated between 0 and 0.2 [10]. If $AP$ is too large, the

| function   | Parameter | Average number of turning points | Iteration time of the last turning point | Averaged optimal value | Averaged value change for each turn |
|------------|-----------|----------------------------------|------------------------------------------|------------------------|-------------------------------------|
| Rastrigin’s | PSO       | 86                               | 112                                      | 19.07                  | 0.84                                |
|            | CSA       | 16                               | 839                                      | 28.92                  | 6.58                                |
|            | CSO       | 36                               | 839                                      | 11.72                  | 3.08                                |
| Ackley     | PSO       | 73                               | 93                                       | 2.64                   | 0.19                                |
|            | CSA       | 32                               | 922                                      | 6.18                   | 0.46                                |
|            | CSO       | 83                               | 365                                      | 2.58e-12               | 0.20                                |

Figure 4: Optimization process for Rastrigin’s function (a) and Ackley function (b) with a dimension of $D = 10$. | Objective function value | CSA | PSO | CSO |
|-------------------|-----|-----|-----|
| 0 600 800 400 200 1000 |     |     |     |
| 140 | 120 | 100 | 80  |
| 60  | 40  | 20  | 0   |
| 0   | 20  | 40  | 60  |
| 80  | 100 | 120 | 140 |

| Objective function value | CSA | PSO | CSO |
|--------------------------|-----|-----|-----|
| 0 600 800 400 200 1000 |     |     |     |
| 25 | 20  | 15  | 10  |
| 5  | 0   | 5   | 10  |
| 15 | 20  | 25  | 30  |
| 30 | 35  | 40  | 45  |
optimization ability of CSA becomes quite weak. When the AP gets closer to 1, CSO gradually approaches PSO, and the algorithm can be easily trapped in a local optimum. Both Table 4 and Figure 5 indicate that the value of AP should be far from the two endpoints for CSO.

The impact of AP on the performance of CSO and CSA is discussed above given that $f_t = 2$. $f_t$ is another important parameter for CSA algorithm. The combination of these two parameters together determines the optimization capability of CSA. Askarzadeh [10] has mentioned that the parameters

| Sphere function                      | AP    |
|--------------------------------------|-------|
| CSO                                  |       |
| 0                     | $1.29E - 318$ | $2.44E - 323$ | $2.85E - 331$ | $8.62E - 342$ | $5.22E - 354$ | $4.52E - 3$ |
| CSA                                  | 122.51 | 202.91           | 318.31       | 408.18        | 881.13        | 3029.16     |
| PSO                                  | $1.42E - 33$ |

| Schwefel's problem 1.2 function      | AP    |
|--------------------------------------|-------|
| CSO                                  |       |
| 0                     | $1.44E - 318$ | $3.91E - 323$ | $7.77E - 331$ | $1.94E - 341$ | $5.88E - 354$ | $8.28E - 301$ |
| CSA                                  | 3452.62 | 2900.35          | 2986.59      | 2531.65       | 2660.23       | 3.23E + 3   |
| PSO                                  | $1.74$ |

Figure 5: The impact of AP on solving multipeak problems: (a) Rastrigin’s function and (b) Ackley function.
of \( fl \) and AP played an essential role in the CSA search process as the solution obtained by the right parameter was far superior to the one using the improper value. For different optimization problems, it is necessary to choose among different parameters to get the reasonable optimal solution.

Figure 6: Optimization results of CSA and CSO with different parameter settings for (a) sphere function \((D = 6)\) and (b) Rastrigin’s function \((D = 4)\) (the decision space is set to be \([-10, 10]\)).
Figure 6 compares the results obtained from CSA and CSO under different parameter combinations. In the previous discussion, it has been realized that a large AP would greatly undermine the optimization ability of CSA. Therefore, the upper bound of AP is set to 0.5. Askarzadeh [10] also mentioned that the best choice of parameter $f_1$ maybe around 2. Thus, the range of $f_1$ is set as $[1.25, 2.75]$. Figure 6(a) shows the optimization result of sphere function with dimension of 6, and Figure 6(b) is that of Rastrigin’s function with dimension of 4. The upper surface of the figures represents the running result of CSA algorithm, while the lower surface is the optimized result of CSO. As parameters $f_1$ and $v_{\text{max}}$ play the similar role in the optimization process as they both restrict the updating velocity of the solution, the parameter $v_{\text{max}}$ (assuming $v_{\text{max}}$ has the same value in each dimension, so $v_{\text{max}}$ is treated as a scalar) in CSO is taken as $f_1$. As can be explored from Figure 6, the upper surface is always steeper than the lower surface, indicating that the performance of CSA greatly depends on the setting of parameters. For the sphere function, of all parameter combinations applied in this study, the worst result (1.61) is 10 times worse than the best result (0.15). However, for CSO, all parameter combinations reach extremely close to 0, making almost a flat surface, i.e., CSO has stronger robustness to parameter settings compared with CSA.

3.5. Application in Real-World Problem. Optimization algorithm is often used in the optimal operation of reservoir. This section describes the ability of the CSO algorithm to solve real-world problem. It is a short-term scheduling of a hydrothermal power system described in CEC 2011 competition. The optimal goal is to meet the load demand and reduce the cost in accordance with the various constraints put on the hydraulic systems and the power system networks. Hourly discharge management for four hydro units requires 96 variables, i.e., $D = 96$.

The objective function for this problem is to minimize the cost of thermal power plant. The operation is subjected to several limitations, including demand constraints, thermal generator constraints, hydro generator constraints, storage capacity constraints, and reservoir limit constraints. These constraints are added in the fitness function as penalty term. The detailed description and data could be found in [24].

Each algorithm was run for 25 times, and the average value of the optimal solution for each run was calculated to
get the results, as shown in Figure 7. From the figure, we can see PSO performs badly since several penalties being relatively large, leading to the worst total objective value. MBO occupies the second largest circle, indicating that its performance is slightly better than that of PSO. As for CSA, the total objective value is small compared to PSO, while it violates the requirements of the discharge to a large extent. The best result comes from MS, which takes up the smallest area. CSO also meets almost every requirement with slight penalty. Table 5 shows the optimal and average values of 25 runs. The best total objective value comes from CSO and MS, which reduced by 57.6%, 70.9%, and 85.5% compared to CSA, PSO, and MBO, respectively. Optimized total cost does not make huge differences, and the penalty term is deterministic factor that influence the optimization results. Both CSO and MS could make every penalty equal or close to 0. Thus, it is believed that CSO has the same strength when tackling this problem, but sometimes it is trapped in local optima, which needs further improvements. As a conclusion, the operation scheme obtained by CSO could be a powerful alternative. Combined with above analysis, CSO may also be strong in real-world problems since it is explorative and efficient.

4. Conclusion

The balance between exploitation and exploration essentially determines the performance of a population-based optimization algorithm, which is also a big challenge in algorithm design. PSO is a popular bio-inspired optimization algorithm with strong ability in exploiting the existing information, i.e., every movement of the particles is under a certain guidance resulting from the previous knowledge, which leads to the weakening of the ability of exploring. CSA is a recent proposed algorithm with characteristics of simplicity, less parameters, and more randomness. The probability of flying totally randomly makes CSA has stronger ability of exploring new regions in the search space, while also results in a large computational cost and low search precision. This study proposes a new crow swarm optimization algorithm coupling PSO and CSA, which provides the individuals the possibility of exploring the unknown regions under the guidance of another random individual. The proposed CSO algorithm is tested on several benchmark functions, including both unimodal and multimodal problems with different variable dimensions. An example of its application in hydrothermal scheduling problem is also presented. The performance of the proposed CSO, the standard PSO, and CSA is evaluated by the optimization efficiency, the global search abilities, and the robustness to parameter settings. The results verify that the proposed CSO has better optimization ability compared with PSO and CSA, and it is less likely to fall into the local optimal solution than PSO algorithm and more efficient than CSA algorithm, i.e., CSO is good at approaching the global optimum quickly while maintaining a certain diversity of solutions during iteration. It can be concluded that the proposed PSO combines the advantages of PSO in exploitation and that of CSA in exploration. In addition, CSO has stronger robustness to the parameter $f_I$ and AP than CSA, which was regarded a major drawback of CSA previously. Although the proposed algorithm performs better than traditional PSO and CSO, it does sometimes become inferior to other algorithms. CSO, in this paper, is based on original PSO, but it could be applied in other modified version of PSO instead of traditional ones for possible further improvements. It is impossible for this algorithm to beat all the other algorithms, but it can be improved further, such as setting adaptive parameters, including subswarms. In addition to above algorithms, many other representative computational intelligence algorithms can be used to solve the problems, such as Harris hawks optimization (HHO) [25], which are not fully discussed in this paper. The source code of the proposed CSO algorithm is provided in this study, and more application and tests of CSO is welcome.

Data Availability

The data used to support the findings of the study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by National Natural Science Foundation of China (Grant nos. 91847302, 51879137, and 51979276).

References

[1] J. Del Ser, E. Osaba, D. Molina et al., “Bio-inspired computation: where we stand and what’s next,” Swarm and Evolutionary Computation, vol. 48, pp. 220–250, 2019, in English.
[2] K. R. Harrison, A. P. Engelbrecht, and B. M. Ombuki-Berman, “Optimal parameter regions and the time-dependence of control parameter values for the particle swarm optimization algorithm,” Swarm and Evolutionary Computation, vol. 41, pp. 20–35, 2018, in English.
[3] G. I. Sayed, A. E. Hassanien, and A. T. Azar, “Feature selection via a novel chaotic crow search algorithm,” Neural Computing and Applications, vol. 31, no. 1, pp. 171–188, 2019, in English.
[4] E. R. Kennedy, "Particle swarm optimization," in Proceedings of ICNN’95-International Conference on Neural Networks, Perth, WA, Australia, November 1995.
[5] T. Jiang, J. Li, and K. Huang, “Longitudinal parameter identification of a small unmanned aerial vehicle based on modified particle swarm optimization,” Chinese Journal of Aeronautics, vol. 28, no. 3, pp. 865–873, 2015, in English.
[6] A. Khare and S. Rangnekar, “A review of particle swarm optimization and its applications in Solar Photovoltaic system,” Applied Soft Computing, vol. 13, no. 5, pp. 2997–3006, 2013, in English.
[7] A. Mahor, V. Prasad, and S. Rangnekar, “Economic dispatch using particle swarm optimization: a review,” Renewable and Sustainable Energy Reviews, vol. 13, no. 8, pp. 2134–2141, 2009, in English.
[8] J. Gou, Y.-X. Lei, W.-P. Guo, C. Wang, Y.-Q. Cai, and W. Luo, “A novel improved particle swarm optimization algorithm based on individual difference evolution,” Applied Soft Computing, vol. 57, pp. 468–481, 2017, in English.

[9] M. He, Y. Hu, H. Chen et al., “Lifecycle coevolution framework for many evolutionary and swarm intelligence algorithms fusion in solving complex optimization problems,” Swarm and Evolutionary Computation, vol. 47, pp. 3–20, 2019, in English.

[10] A. Askarzadeh, “A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm,” Computers & Structures, vol. 169, pp. 1–12, 2016, in English.

[11] C. Qu and Y. Fu, “Crow search algorithm based on neighborhood search of non-inferior solution set,” IEEE Access, vol. 7, pp. 52871–52895, 2019, in English.

[12] F. Mohammadi and H. Abdí, “A modified crow search algorithm (MCSA) for solving economic load dispatch problem,” Applied Soft Computing, vol. 71, pp. 51–65, 2018, in English.

[13] S. Shekhawat and A. Saxena, “Development and applications of an intelligent crow search algorithm based on opposition based learning,” Isa Transactions, vol. 99, pp. 210–230, 2020, in English.

[14] A. Chaudhuri and T. P. Sahu, “Feature selection using Binary Crow Search Algorithm with time varying flight length,” Expert Systems with Applications, vol. 168, Article ID 114288, 2020, in English.

[15] S. Makhdoomi and A. Askarzadeh, "Optimizing operation of a photovoltaic/diesel generator hybrid energy system with pumped hydro storage by a modified crow search algorithm," Journal of Energy Storage, vol. 27, Article ID 101040, 2019, in English.

[16] D. Gupta, J. J. P. C. Rodrigues, S. Sundaram, A. Khanna, V. Korotaev, and V. H. C. de Albuquerque, "Usability feature extraction using modified crow search algorithm: a novel approach," Neural Computing & Applications, vol. 32, no. 15, pp. 10915–10925, 2018, in English.

[17] H. Wu, P. Wu, K. Xu, and F. Li, “Finite element model updating using crow search algorithm with Levy flight,” International Journal for Numerical Methods in Engineering, vol. 121, no. 13, pp. 2916–2928, 2020, in English.

[18] A. Javidi, E. Salajegheh, and J. Salajegheh, “Enhanced crow search algorithm for optimum design of structures,” Applied Soft Computing, vol. 77, pp. 274–289, 2019, in English.

[19] S. Arora, H. Singh, M. Sharma, S. Sharma, and P. Anand, “A new hybrid algorithm based on grey wolf optimization and crow search algorithm for unconstrained function optimization and feature selection,” IEEE Access, vol. 7, pp. 26343–26361, 2019, in English.

[20] N. Mahesh and S. Vijayachitrir, “DECSA: hybrid dolphin echolocation and crow search optimization for cluster-based energy-aware routing in WSN,” Neural Computing & Applications, vol. 31, no. S1, pp. 47–62, 2019, in English.

[21] A. M. Anter and M. Ali, “Feature selection strategy based on hybrid crow search optimization algorithm integrated with chaos theory and fuzzy c-means algorithm for medical diagnosis problems,” Soft Computing, vol. 24, no. 3, pp. 1565–1584, 2020, in English.

[22] N. R. Babu, L. C. Saikia, S. K. Bhagat, and A. Saha, “Maiden application of hybrid crow-search algorithm with particle swarm optimization in LFC studies,” Advances in Intelligent Systems and Computing in Proceedings of International Conference on Artificial Intelligence and Applications, vol. 1164, pp. 427–439, 2021, in English.