Parallel Binary Cat Swarm Optimization with Communication Strategies for Traveling Salesman Problem

Jeng-Shyang Pan[1], Xiao-Fang Ji[1], Anhui Liang[1], Kuan-Chun Huang[2], Shu-Chuan Chu[1,3]

[1] College of Computer Science and Engineering, Shandong University of Science and Technology, China
[2] Department of Industrial and Systems Engineering, Chung Yuan Christian University, Taiwan
[3] College of Science and Engineering, Flinders University Sturt Rd, South Australia
jengshyangpan@gmail.com, xiaofangji163@163.com, Liangah8@sdust.edu.cn, westie888@gmail.com, scchu0803@gmail.com

Abstract

This paper studies the Parallel Binary Cat Swarm Optimization (PBCSO) algorithm and its application to the Traveling Salesman Problem (TSP). It investigates the performance of PBCSO with two evolution strategies employing neighborhood merge comparison and dynamic adaptation, aiming to optimize population fitness. This parallel mechanism randomly divides initial solutions into few groups, and shares the information in various groups following every fixed iteration. In this way, the defects of the original BCSO premature convergence and easy to fall under the local optimal search space can be significantly reduced. We have conducted repeated tests on 23 functions of three types. The results demonstrate that the proposed PBCSO can solve the optimization problem more specifically. It can maximize the diversity of the population, so that the optimal solution is most likely to be obtained, which is easy to break through the limitations of local convergence to achieve the global optimal. TSP is a classic NP-hard problem, which has been extensively studied in the literature. This article shows that PBCSO can be successfully used in the analysis of TSP problems and has broad application prospects. PBCSO was simulated on Matlab, and the result proved the feasibility and effectiveness.

Keywords: Binary Cat Swarm Optimization, Parallel Binary Cat Swarm Optimization, Communication strategy, Traveling salesman problem

1 Introduction

Metaheuristic algorithms [1] are classified into trajectory-based algorithms and population-based algorithms. Reasons for the widespread use of metaheuristic algorithms contain its ease of implementation and broad applicability. The Cat Swarm Optimization (CSO) algorithm is a bionic intelligence optimization algorithm inspired by the foraging behavior of cat populations. As one of the most extensively used meta-heuristic algorithms, it was first propound by Chu et al. in 2006 [2-4]. There are many optimization problems in the discrete binary search space, and their solution requires binary algorithms. Following these directions, many other swarm algorithms have been proposed, such as Differential Evolution (DE) [5-6], Genetic Algorithm (GA) [7-8], Particle Swarm Optimization (PSO) [9-10], Ant Colony Optimization (ACO) [11-12], Grey Wolf Optimization (GWO) [13], Whale Optimization Algorithm (WOA) [14-15], Firefly Algorithm (FA) [16-17], and Memetic Algorithm [18]. They all have binary versions that enable them to be executed in binary space. In 2013, Y. Sharafi et al proposed the Binary Cat Swarm Algorithm (BCSO) [19]. Whereas, the original BCSO is prone to fall under local optimum when addressing practical issues. In this paper, PBCSO is improved based on BCSO. To reinforce global search ability and quality of solutions, it can effectively enhances the stability and solves the problem of low optimization accuracy of the BCSO. The focus of the present paper is that the parallel process aims at exchanging useful information among individuals through a certain communication strategy in each group.

The remainder of this paper is structured as follows: Sect 2 presents the specific background of the CSO and BCSO. Section 3 details two communication mechanisms that exist in PBCSO. Section 4 shows the eight functions of the transfer function family. Section 5 presents and investigates PBCSO and BCSO experimental results for benchmark function testing. Section 6 analysis of TSP problem. The new algorithm performance test are presented in Sect 7. Eventually, Section 8 makes a summary for the research, and points out future research direction.

2 Related Works

This section focuses on the basic models of traditional

*Corresponding Author: Shu-Chuan Chu; E-mail: scchu0803@gmail.com
DOI: 10.53106/160792642021122207015

CSO and BCSO. Their introduction is as follows:

2.1 Cat Swarm Optimization

The CSO algorithm has similarities with a large quantity of swarm algorithms. It simulates the process of a group of cats looking for food. Each cat is a particle (point) in the CSO, in other words, it is a possible solution to the problem we need to solve. In the process of searching for food, they constantly change their position and velocity. In CSO, the cats are scattered at first, and then the cats follow a limited number of neighbors, and finally the entire group of cats is under the control of a center to find the optimal solution. CSO was originally developed for the continuous value space, the cat changes its position and moves according to its speed, through the information sharing between cats and the individual's own optimization experience to modify the individual's action strategy. Finally find the solution of the optimization problem. There are two forces that affect the location update process: one is the best position obtained by oneself (self-experience), and the other is the best position (group experience) that a certain particle finds in its neighborhood. Through this combination, an intelligent collective behavior emerges.

2.2 Binary Cat Swarm Optimization

In contrast to standard CSO where the goal is to solve the continuity problem, the task in discretization optimization frequently amounts to tracking extreme values for iterative optimization as possible. In BCSO, the speed and position of the cat are both binary vectors in the search space. Cats have certain attributes, for example, the position and fitness function in the multi-dimensional search space are usually performance indicators. To execute the algorithm, the mixture ratio MR indicates a proportional relationship between the two behavior patterns of cats.

2.2.1 Structure Coding

The structural coding of the BCSO algorithm, as the name suggests, is designed in the form of binary coding. As shown in Table 1.

Table 1. The genetic code of the BCSO algorithm

	Dimension 1	…	Dimension *d*-1	Dimension *d*
Velocity code	$V_{k,1}$	…	$V_{k,d-1}$	$V_{k,d}$
Position code	$X_{k,1}$	…	$X_{k,d-1}$	$X_{k,d}$

In the binary coding structure, cat_k is a d-dimensional binary vector, which can be regarded as a binary string. When the algorithm continues to iterate, the velocity vector V_k of the kth cat can be expressed as $V_k = \left(V_{k,1}, V_{k,2}, V_{k,3}, \ldots, V_{k,d-1}, V_{k,d}\right)$. Expressed by the same principle, position vector X_k of the kth cat can be denoted as $X_k = \left(X_{k,1}, X_{k,2}, X_{k,3}, \ldots, X_{k,d-1}, X_{k,d}\right)$. The feasible solution space of velocity can be expressed as $V = \left\{\left(V_1, V_2, V_3, \ldots, V_{d-1}, V_d\right) \mid V_i \in \{0,1\}, \sum_{i=1}^{d} V_i = e\right\}$, that is, V is regarded as a d-bit binary string composed of 0 and 1. Among them, the number of $e(\leq d)$ positions is 1, and remaining is 0.

2.2.2 Seeking Mode

In this mode, four basic factors are delimit as shown in Figure 1. The change in the current position of cat may be considered the binary mutation. Under such circumstances, the parameter PMO will replace SRD in the original CSO version. Remaining parameters are totally the same with those in CSO.

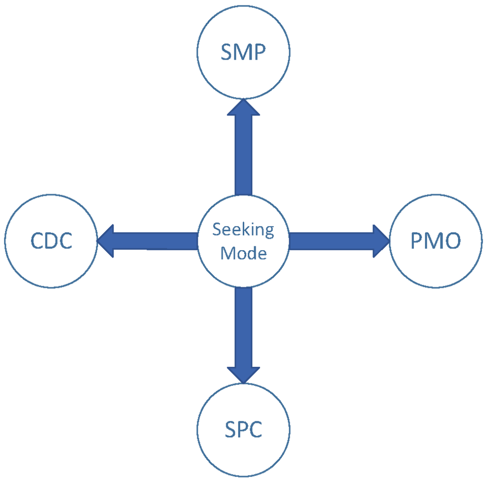


Figure 1. Four important factors of Seeking Mode

SMP defines the size of the search memory of each cat, denoting the location point that the cat has searched, and the cat will choose the best location point from the memory pool in accordance with the fitness level
PMO denotes mutation operation
CDC represents the number of dimensions to be mutated, which is a value in the interval (0, 1)
SPC expresses a boolean value, which indicates whether the cat takes the past position as one of the candidate positions to be moved to, and its value does not affect the value of SMP

The procedure of BCSO can be divided into the following steps:

Step 1: When the SPC flag is true, the original position of cat_k is potentially the candidate position, and thus, the additional SMP-1 copy of the current position of every cat is required, and the current position should be used as a candidate position. While when the SPC flag is not true, SMP copy should be made for the current position of every cat.

Step 2: The procedure marks the major distinction of BCSO and CSO. For every SMP copy, the CDC dimension should be chosen as much as possible as per the PMO and replace the original one. The step implies that, due to the binary property of the BCSO value, SRD converts to mutation probability PMO.
Step 3: Take into account the cost function to seek the fitness value (FS) of overall candidate points.
Step 4: The optimal copy position can be found from the non-dominated solution of the memory pool. If all FS are not completely equal, calculate the selection probability of each candidate solution using a formula. A variety of setting methods can be used for the selection operator. For example, Equation (1) is applied to obtain the probability of the selected candidate point P_i. Otherwise, set the probability of all candidate points to 1.

$$P_i = \begin{cases} 1, & if\ FS_{\max} = FS_{\min} \\ \dfrac{|FS_i - FS_b|}{FS_{\max} - FS_{\min}}, & where\ 0 < i < j, otherwise. \end{cases} \quad (1)$$

Among them, if $FS_b = FS_{\max}$ indicates that the numerator of fraction in Equation (1) is minimized, which is smallest for minimization, vice versa.
Step 5: Implement roulette on candidate points, and choose one candidate point to replace the current position.

2.2.3 Tracing Mode

In CSO, the diversity of the current and previous positions of the cat is expressed in terms of velocity. At the same time, the velocity change is achieved by adding a random disturbance. However, the velocity vector here turns its meaning into a possibility of mutation, leading to the position component is either 1 or 0. Therefore, the velocity here no longer represents the size of the position change, it reflects the probability of the position change. The two velocity vectors of each cat are defined as $V_{k,d}^0$ and $V_{k,d}^1$. $V_{k,d}^0$ is the probability of a bit becoming zero, and $V_{k,d}^1$ is the probability of a bit becoming one.

The update process of $V_{k,d}^0$, $V_{k,d}^1$ is as follows:

$$\begin{aligned} V_{k,d}^1 &= \omega \cdot V_{k,d}^1 + d_{k,d}^1 \\ V_{k,d}^0 &= \omega \cdot V_{k,d}^0 + d_{k,d}^0 \quad d = 1,\ldots,M \end{aligned} \quad (2)$$

$d_{k,d}^0$ and $d_{k,d}^1$ are temporary values determined by the Equation (3).

$$\begin{aligned} &if\ X_{gbest,d} = 1\ Then\ d_{k,d}^1 = r_1 \cdot c_1\ and\ d_{k,d}^0 = -r_1 \cdot c_1 \\ &f\ X_{gbest,d} = 0\ Then\ d_{k,d}^1 = -r_1 \cdot c_1\ and\ d_{k,d}^0 = r_1 \cdot c_1 \end{aligned} \quad (3)$$

Among them, ω enotes the inertia weight, r_1 has a random value in the interval of [0, 1], c_1 stands for a constant defined by the user, and its value needs to be determined according to different problems. It is emphasized that these speeds are not complementary [20] The final speed of the cat (the probability of changing the d position of the ith cat) is given by Equation (4). Using the current position $X_{k,d}$, the speed can be bounded in the interval $[V_{\min}, V_{\max}]$. According to the current position cat_k, the speed of cat_k is calculated as:

$$V_{k,d}^{'} = \begin{cases} V_{k,d}^1 & if\ X_{k,d} = 0 \\ V_{k,d}^0 & if\ X_{k,d} = 1 \end{cases} \quad (4)$$

Figure 2 depicts the Tracing Mode:

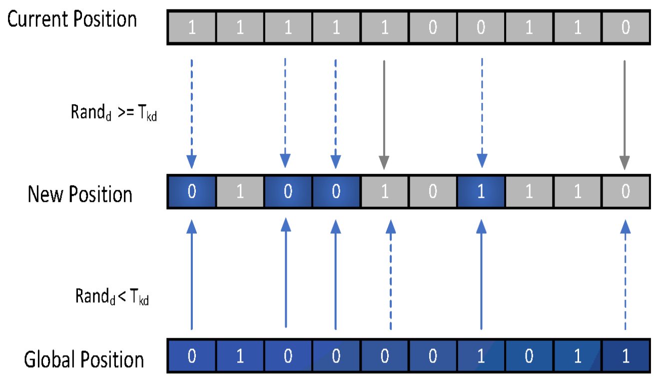


Figure 2. Tracing Mode

The biggest difficulty of discrete variables is to constantly redefine the position and speed of the cat. Therefore, transforming discrete variables into continuous variables can avoid this difficulty. These vectors are mapped to the same space using the transfer function. The most general method is to apply a sigmoid function to the velocity vector called "S-shaped" function as defined by Equation (5). The velocity V is processed by sigmod transformation, and the obtained parameter t represents the probability of sudden change in each dimension. They can be measured using the formula below:

$$t_{k,d} = sig\left(V_{k,d}^{'}\right) = \frac{1}{1 + e^{-V_{k,d}^{'}}} \quad (5)$$

Among them, $t_{k,d}$ obtains values at [0, 1] intervals. A smaller value $V_{\max}$ is beneficial to promote exploration, and a larger value is beneficial to restrict exploration. If $V_{\max} = 0$ it will enter a random search, if $V_{\max} = 5$, then $sig\left(V_{\max}\right) = 0.99331$ is a high probability that the position will be updated to the optimal value searched for by the entire cat group. The purpose of adding a limit to the variation of each dimension is to prevent the variation from being too large, otherwise, causing the algorithm to search blindly and randomly in the solution space.

Update the new position of each dimension of cat_k according to the value of $t_{k,d}$ as shown below:

$$X_{k,d} = \begin{cases} X_{gbest,d} & if\ rand < t_{k,d} \\ X_{k,d} & otherwise \end{cases} \quad (6)$$

Use the velocity to update the position and set a threshold. When the speed is higher than the threshold, the position is taken as the global optimum of the cat group. Otherwise, the individual's own historical optimum is taken. Another unique method is to use the "V-shaped" function Equation (7).

$$\left|\tanh\left(V_{k,d}^{'}\right)\right| = \left|\frac{e^{V_{k,d}^{'}} - e^{-V_{k,d}^{'}}}{e^{V_{k,d}^{'}} + e^{-V_{k,d}^{'}}}\right| \quad (7$$

We can generate $X_{k,d}^{t+1}$ should be expressed according to the following Equation (8).

$$X_{k,d}^{t+1} = \begin{cases} X_{k,d}^{t} & if\ rand < \left|\tanh\left(V_{k,d}^{'}\right)\right| \\ X_{k,d} & otherwise \end{cases} \quad (8)$$

where superscripts indicate time. As shown below, Figure 3 depicts the flow chart of BCSO:

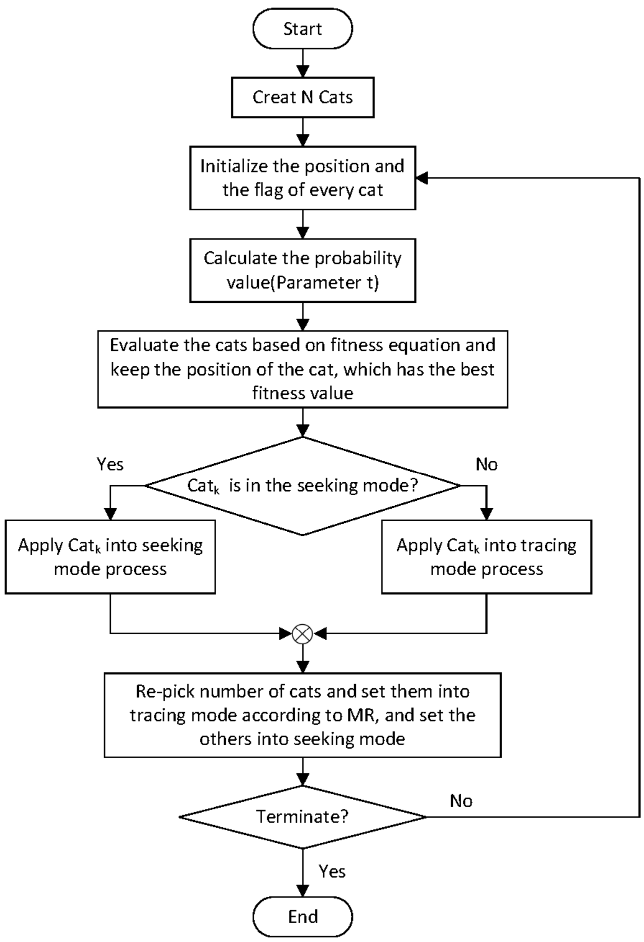


Figure 3. The flow chart of BCSO

3 Parallel Binary Cat Swarm Optimization

This part discusses the main ideas and implementation schemes of PBCSO in detail. Several parallel algorithms have been proposed for some meta-heuristic optimization algorithms and applied to some engineering problems such as the Parallel Multi-Verse Optimizer [21], Parallel Symbiotic Organism Search Algorithm [22], Parallel Grasshopper Optimization Algorithm [23], Parallel Genetic Algorithm [24], and other parallel methods [25-28]. The original BCSO has many advantages, simple structure and can optimize the objective function when the search space is unknown and limited, etc. However, when dealing with some specific problems, such as multi-modal function or compound function optimization problem, its performance is not satisfactory. In view of these shortcomings, the concept of multiple groups is introduced, which is conductive to maintaining the diversity of the population, thereby ensuring that the optimal solution is found as much as possible in the optimization process. Since the search process is from one point set (population) to another point set in space, it is actually a parallel search. This not only helps to jump out of local optimization, but also enables large-scale parallel computing. The specific scheme is as follows: First, construct a parallel processing structure to group the entire population to obtain several sub-populations. Then each subpopulation evolves independently according to the iteration rules. The evaluation of the solution is based on the fitness value. After triggering a certain inter-population communication plan, use corresponding strategies to replace inappropriate individuals in the population to develop or explore promising areas. Communication strategy can transmit effective information between groups, and thus algorithm performance can be promoted. PBCSO is a population-based, cooperative, global search, swarm intelligence and meta-heuristic algorithm. Based on this idea, this article designs two PBCSO communication strategies. The following is a detailed description of the two communication strategies:

3.1 Neighborhood Merger Comparison Strategy

First, the population size *nPop* is divided into *N* groups. Each group is independently optimized BCSO. This method first randomly divides the initial population into *N* subgroups and optimizes them in each subgroup in parallel with the proposed algorithm. At the same time, to avoid falling into local extremum, a binary mutation operator is used to exchange information between the subgroups, and to update the position of related particles to ensure the algorithm's global search capability and maintain the diversity of the population. In the optimization process of each subgroup, when the communication conditions are triggered, migration and merger will occur between neighbors. After communication, the solutions of the merged subgroups become random solutions. In the end, the poor individual can be substituted for the local best in each group, and then the local best is replaced by the global best. Analyses of the behavior of communication strategy in PBCSO rely on the mutation and update of subgroups that is illustrated in Figure 4. In this strategy, the subgroups using the same algorithm have the same parameters. It finds the optimal solution widely in the search space and avoids falling into the local optimal, also improving the convergence speed of the algorithm.

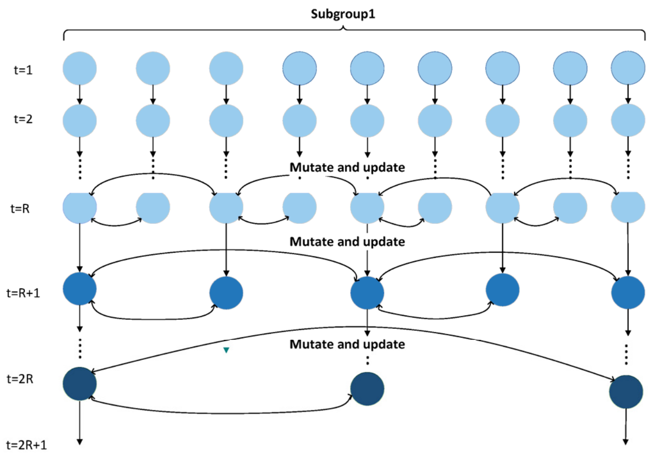


Figure 4. Neighborhood merger comparison strategy

Step 1. Initialization: Generate individuals for the jth group, $j = 1,2,3,...,N$, $P_{i,j}^{(t)}$ is the ith cat position in the jth group at the tth iteration.

Step 2. Evaluation: Evaluate the $F(x)$ value of each particle in each group.

Step 3. Update: Update the speed and position.

$$V_{i,j}^{'(t+1)} \begin{cases} V_{i,j}^{1(t+1)} = \omega \cdot V_{i,j}^{1(t)} + d_{i,j}^{1(t)} \\ V_{i,j}^{0(t+1)} = \omega \cdot V_{i,j}^{0(t)} + d_{i,j}^{0(t)} \end{cases} \tag{9}$$

$$P_{i,j}^{(t+1)} = P_{i,j}^{(t)} + V_{i,j}^{'(t+1)} \tag{10}$$

Step 4. Communication strategy: Once every R iterations, individual neighbors are compared with each other and the neighbors at intervals are compared, and the poorer performance particles in each group are replaced. $G_j^{(t)}$ is the best solution of the jth group obtained. $G^{(t)}$ is the best solution compared among all groups. The formula is as follows:

$$f\left(G^{(t)}\right) \leq f\left(G_j^{(t)}\right) \tag{11}$$

Step 5. Termination: Repeat Steps 2 to 5 until the function reaches the preset value or reaches the maximum number of iterations, and then record the best solution.

3.2 Dynamic Adjustment of the Number of Subgroups

PBCSO algorithms are population-based strategies that frequently place great emphasis on the adaptability. Under normal circumstances, in major shopping malls or stores, the crowds of customers at the doors of hot businesses can be observed. However, there may be a lot of products that are not what we need, and they will still join the ranks of buyers' involuntarily. This is the so-called "follow the trend" phenomenon. In addition, the phenomenon may cause us to purchase undesirable goods and lead to unnecessary waste. Even though the cat swarm algorithm is simple and not difficult to understand, for evolutionary algorithms, there is often a phenomenon of group loss in subgroups during the experiment. The phenomenon of group loss is a phenomenon in which the majority of subgroups support wrong views resulted from the asymmetry of herd mentality and information. When reflected in the cat group algorithm, the following two situations can be shown:

(1) Dynamic changes in the environment may contribute to group loss;

(2) The gap in the speed of movement between individuals may lead to group loss.

On the basis of the above situation, the individual may be stagnant or no better solution appears after a certain number of iterations. The present study puts forward a dynamic adjustment propagation strategy based on the number of subgroups that mimics the above phenomenon, improving the search quality and speed of subgroups. The cat group is divided into N groups. When the number of iterations reaches R, Equation (12) is applied to update the positions of randomly selected individuals.

$$X_{k,d} = \left(1 - l / Maxiterations\right) \cdot X_{k,d} \tag{12}$$

This propagation strategy increases the diversity of the population by updating the current position of the subgroup particles. As a result, individuals can have more choices to prevent the cat group from having similar fitness values.

4 Families of Transfer Functions

There are two types of transfer functions in the binary version called S-shaped and V-shaped [29]. They are responsible for mapping the discrete search space to the continuous search space. They define the probability of changing the elements of the position vector from 0 to 1, and vice versa. The following characteristics of them and shown in Table 2 and Figure 5 shows the curves of the derivative functions.

The meaning of transfer function can be understood as forcing cats to move in binary space. As a result, the transfer function mapping the speed value to the

Table 2. S-shaped and V-shaped families of transfer function

	Name	Transfer function
S-shaped	S1	$\frac{1}{1+e^{-2x}}$
	S2	$\frac{1}{1+e^{-x}}$
	S3	$\frac{1}{1+e^{-x/2}}$
	S4	$\frac{1}{1+e^{-x/3}}$
V-shaped	V1	$\left\lvert\frac{\sqrt{2}}{\pi}\int_0^{(\sqrt{\pi}/2)x} e^{-t^2}dt\right\rvert$
	V2	$\lvert\tanh(x)\rvert$
	V3	$\left\lvert(x)/\sqrt{1+x^2}\right\rvert$
	V4	$\left\lvert\frac{2}{\pi}\arctan\left(\frac{\pi}{2}x\right)\right\rvert$

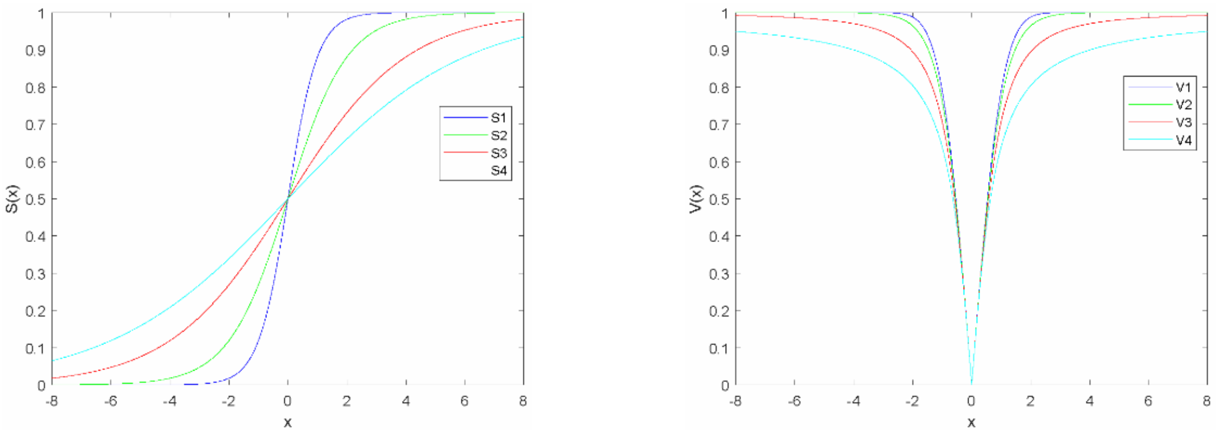
probability value will possess the following characteristics:

(1) The transfer function should provide a large absolute value of velocity with a large probability of changing position. In addition, particles with larger absolute speeds may be far from the optimal solution. Therefore, they should switch positions in the next iteration;

(2) The return value of the transfer function decreases or increases concurrently with the speed. Individual cats distant from the optimal solution are supposed to have a greater likelihood to change their position vector and return to the previous position.

5 Experimental Results and Analysis

In this section, for verifying the efficiency of the projected algorithm, 23 mathematical optimization functions were used to compare PBCSO and BCSO. These typical test equations are listed in the Table 3 to Table 5 used by many scholars [30], unimodal, multimodal, fixed-dimension and composite functions, respectively. *Space* denotes the boundary of search space; $D_{\min}$ means the dimension of function and $f_{\min}$ represents the optimum.



(a) The curves of S-shaped

(b) The curves of V-shaped

Figure 5. The curves of S-shaped (a) and V-shaped (b) transfer functions

Table 3. Unimodal benchmark functions

Name	Function	Space	D_{im}	$f_{\min}$
Sphere	$F1(x)=\sum_i^n x_i^2$	[-100, 100]	30	0
Schwefel's function 2.21	$F2(x)=\sum_{i=1}^n \lvert x_i\rvert+\prod_{i=1}^n \lvert x_i\rvert$	[-10, 10]	30	0
Schwefel's function 1.2	$F3(x)=\sum_{i=1}^n\left(\sum_{j-1}^i x_j\right)^2$	[-100, 100]	30	0
Schwefel's function 2.22	$F4(x)=\max_i\{\lvert x_i\rvert, 1\le i\le n\}$	[-100, 100]	30	0
Rosenbrock	$F5(x)=\sum_{i=1}^{n-1}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right]$	[-30, 30]	30	0
Step	$F6(x)=\sum_{i=1}^n\left([x_i+0.5]\right)^2$	[-100, 100]	30	0
Dejong's noisy	$F7(x)=\sum_{i=1}^n ix_i^4+random[0,1)$	[-1.28, 1.28]	30	0

Table 4. Multimodal benchmark functions

Name	Function	Space	D_{im}	f_{min}
Schwefel	$F8(x)=\sum_{i=1}^{n}-x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$	[-500, 500]	30	-12569
Rastringin	$F9(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)+10\right]$	[-5.12, 5.12]	30	0
Ackley	$F10(x)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)$	[-32, 32]	30	0
Griewank	$F11(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	[-600, 600]	30	0
Generalized penalized 1	$F12(x)=\frac{\pi}{n}\left\{\begin{matrix}10\sin(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2\\\left[1+10\sin^2(\pi y_{i+1})+(y_n-1)^2\right]\end{matrix}\right\}$ $+\sum_{i=1}^{n}u(x_i,10,100,4)y_i=1+\frac{x_i+1}{4}u(x_i,a,k,m)$ $=\begin{cases}k(x_i-a)^m & x_i>a\\0 & -a<x_i<a\\k(-x_i-a)^m & x_i<-a\end{cases}$	[-50, 50]	30	0
Generalized penalized 2	$F13(x)=0.1\left\{\begin{matrix}\sin^2(3\pi x_1)+\sum_{i=1}^{n}(x_i-1)^2\left[1+\sin^2(3\pi x_i+1)\right]\\+(x_n-1)^2\left[1+\sin^2(2\pi x_n)\right]\end{matrix}\right\}$ $+\sum_{i=1}^{n}u(x_i,10,100,4)$	[-50, 50]	30	0

Table 5. Fixed-dimension multimodal benchmark functions

Name	Function	Space	D_{im}	f_{min}
Fifth of Dejong	$F14(x)=\left(\frac{1}{500}\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right)^{-1}$	[-65, 65]	2	1
Kowalik	$F15(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\right]^2$	[-5, 5]	4	0.00030
Six-hump camel back	$F16(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	[-5, 5]	2	-1.0316
Branins	$F17(x)=\left(x_2-\frac{5.1}{4\pi^2}x_1^{-2}+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos x_1+10$	[-5, 5]	2	0.398
Goldstein–Price	$F18(x)=\left[1+(x_1+x_2+1)^2\begin{pmatrix}19-14x_1+3x_1^2\\-14x_2+6x_1x_2+3x_2^2\end{pmatrix}\right]$ $\times\left[\begin{matrix}30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2)\\+48x_2-36x_1x_2+27x_2^2\end{matrix}\right]$	[-2, 2]	2	3
Hartman 1	$F19(x)=-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2\right)$	[1, 3]	3	-3.86
Hartman 2	$F20(x)=-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2\right)$	[0, 1]	6	-3.32
Shekel 1	$F21(x)=-\sum_{i=1}^{5}\left[(X-a_i)(X-a_i)^T+c_i\right]^{-1}$	[0, 10]	4	-10.1532
Shekel 2	$F22(x)=-\sum_{i=1}^{7}\left[(X-a_i)(X-a_i)^T+c_i\right]^{-1}$	[0, 10]	4	-10.4028
Shekel 3	$F23(x)=-\sum_{i=1}^{10}\left[(X-a_i)(X-a_i)^T+c_i\right]^{-1}$	[0, 10]	4	-10.5363

According to Table 6 to Table 9, from the optimization accuracy of the 23 mathematical optimization functions, the proposed PBCSO algorithm is superior to the BCSO algorithms. It can be seen that Table 6 to Table 9, through comparison with BCSO algorithm. The proposed PBCSO achieves 15 better performances, and 8 similar performances in 23 benchmarks under the perspective of "Ave". While from the "Std" viewpoint, it has achieved 9 better performances, 6 worse performances, as well as 8 similar performances.

Table 6. The statistical results of BCSO in the S-shaped transfer function

Function	BCSO_S1		BCSO_S2		BCSO_S3		BCSO_S4	
	Ave	Std	Ave	Std	Ave	Std	Ave	Std
F1	3.03	0.3	5.07	0.4976	8	0	3.16	0.7347
F2	1.09	0.4516	2.13	0.6139	2.2	0.8989	2.16	0.7208
F3	42.02	58.7996	70.71	28.6139	282	0	50.27	79.9.69
F4	1	0	1	0	1	0	1	0
F5	551	95.219	419.04	20.4	408.36	41.2863	428.72	56.396
F6	15.54	0.4	13.76	1.2603	13.72	0.9804	13.9	1.6576
F7	10.6535	19.1266	2.9326	13.1299	58.102	9.9977E-14	77.9568	8.5695E-14
F8	-24.1418	0.8682	-23.435	0.691	-21.0115	0.2524	-23.4855	0.5622
F9	6	0	6	0	4.05	0.3589	4.04	0.2429
F10	1.7137	0.0251	1.4177	0.0508	1.4188	0.0628	1.25	0.14
F11	0.2221	0.0011	0.0934	0.0153	0.1156	0.0237	0.1153	0.0337
F12	2.0724	0.1565	2.1466	0.1622	2.4859	0.0753	1.9941	0.1147
F13	0.404	0.04	0.702	0.02	0.7	1.23E-15	0.501	0.01
F14	12.6705	1.61E-14	12.6705	1.61E-14	12.6705	1.61E-14	12.6705	1.61E-14
F15	0.1484	2.51E-16	0.1484	2.51E-16	0.1484	2.51E-16	0.1484	2.51E-16
F16	0	0	0	0	0	0	0	0
F17	27.7029	1.43E-14	27.7029	1.43E-14	27.7029	1.43E-14	27.7029	1.43E-14
F18	600	0	600	0	600	0	600	0
F19	-0.3348	6.14E-16	-0.3348	6.14E-16	-0.3348	6.14E-16	-0.3348	6.14E-16
F20	-0.1653	0.0039	-0.1657	1.12E-16	-0.1657	1.12E-16	-0.1657	1.12E-16
F21	-5.0552	8.03E-15	-5.0552	8.03E-15	-5.0552	8.03E-15	-5.0552	8.03E-15
F22	-5.0877	0	-5.0877	0	-5.0877	0	-5.0877	0
F23	-5.1285	1.16E-14	-5.1285	1.16E-14	-5.1285	1.16E-14	-5.1285	1.16E-14

Table 7. The statistical results of PBCSO in the S-shaped transfer function

Function	PBCSO_S1		PBCSO_S2		PBCSO_S3		PBCSO_S4	
	Ave	Std	Ave	Std	Ave	Std	Ave	Std
F1	**0.0871**	0.6984	**0.02**	0.2	**0.04**	0.2429	**0.01**	0.1
F2	**0.05**	0.2611	**0**	0	**0.04**	0.2814	**0.03**	0.3
F3	**4.6649**	12.1776	**1.7642**	15.1975	**0**	0	**0.6763**	4.0425
F4	1	0	1	0	1	0	1	0
F5	**4.13**	29.0557	**8.13**	51.5106	**9.39**	39.3433	**8.07**	63.6535
F6	**6.7054**	1.2436	**7.5**	0	**7.5**	0	**7.52**	0.2
F7	**0.0145**	0.1428	**0.122**	0.817	**1.0766**	3.4104	**0.1437**	1.4009
F8	**-25.2441**	4.9989E-14	**-25.2441**	4.9989E-14	**-25.2189**	0.1874	**-25.2273**	0.1184
F9	**0.02**	0.2	**0.03**	0.3	**0.03**	0.2227	**0.01**	0.1
F10	**0.0101**	0.1007	**0.0072**	0.0717	**0.0123**	0.1226	**0.0072**	0.0717
F11	**0.00040781**	0.0041	**0.00047794**	0.0048	**0**	0	**0.00023965**	0.0024
F12	**1.6705**	0.0151	**1.384**	0.1072	**1.5397**	0.1063	**1.6705**	0.0151
F13	**0.003**	0.03	**0.0001**	0.01	**1.35E-32**	2.48E-47	**0.001**	0.01
F14	12.6705	1.61E-14	12.6705	1.61E-14	12.6705	1.61E-14	12.6705	1.61E-14
F15	0.1484	2.51E-16	0.1484	2.51E-16	**0.0425**	4.88E-17	0.1484	2.51E-16
F16	0	0	0	0	0	0	0	0
F17	27.7029	1.43E-14	27.7029	1.43E-14	27.7029	1.43E-14	27.7029	1.43E-14
F18	600	0	**248.3359**	5.71E-13	600	0	600	0
F19	**-0.9352**	2.23E-16	-0.3348	6.14E-16	-0.3348	6.14E-16	-0.3348	6.14E-16
F20	**-0.1657**	1.12E-16	-0.1657	1.12E-16	-0.1657	1.12E-16	-0.1657	1.12E-16
F21	-5.0552	8.03E-15	-5.0552	8.03E-15	-5.0552	8.03E-15	-5.0552	8.03E-15
F22	-5.0877	0	-5.0877	0	-5.0877	0	-5.0877	0
F23	-5.1285	1.16E-14	-5.1285	1.16E-14	-5.1285	1.16E-14	-5.1285	1.16E-14

Table 8. The statistical results of BCSO in the V-shaped transfer function

Function	BCSO_V1		BCSO_V2		BCSO_V3		BCSO_V4	
	Ave	Std	Ave	Std	Ave	Std	Ave	Std
F1	7.01	0.1	5.08	0.4858	1.4	0.9949	2.26	1.1859
F2	2.1	0.5025	4.09	0.4516	2.1	0.5946	5.01	0.1
F3	389	0	30.86	45.9638	127.9	49.1157	63.29	65.108
F4	1	0	1	0	1	0	1	0
F5	542.44	127.5659	511.09	33.521	418.1	85.0447	573.63	48.6178
F6	11.96	2.1622	11.74	1.4573	19.54	0.4	17.5	0
F7	38.4574	10.1734	28.372	8.9341	26.5224	3.2791	13.9813	20.2611
F8	-21.8446	0.2044	-22.6187	0.6581	-22.4841	1.0703	-21.0284	0.0841
F9	4.08	0.4858	6	0	6.06	0.4454	1.21	0.9134
F10	1.0422	0.1776	1.4101	0.016	1.7112	0	1.8418	1.7853E-15
F11	0.0843	0.0039	0.0353	0.0406	0.107	0.0071	0.0803	0.0175
F12	2.0156	0.193	1.9971	0.1724	2.6783	0.0601	1.7126	0.1673
F13	0.404	0.0315	0.6	8.93E-16	0.505	0.0359	0.049	0.1432
F14	12.6705	1.61E-14	12.6705	1.61E-14	12.6705	1.61E-14	12.6705	1.61E-14
F15	0.1484	2.51E-16	0.1484	2.51E-16	0.1484	2.51E-16	0.1484	2.51E-16
F16	0	0	0	0	0	0	0	0
F17	27.7029	1.43E-14	27.7029	1.43E-14	27.7029	1.43E-14	27.7029	1.43E-14
F18	600	0	600	0	600	0	600	0
F19	-0.3348	6.14E-16	**-0.3348**	6.14E-16	-0.3348	6.14E-16	-0.3348	6.14E-16
F20	-0.1657	1.12E-16	**-0.1318**	0.0212	-0.1657	1.12E-16	-0.1657	1.12E-16
F21	-5.0552	8.03E-15	-5.0552	8.03E-15	-5.0552	8.03E-15	-5.0552	8.03E-15
F22	-5.0877	0	-5.0877	0	-5.0877	0	-5.0877	0
F23	-5.1285	1.16E-14	-5.1285	1.16E-14	-5.1285	1.16E-14	-5.1285	1.16E-14

Table 9. The statistical results of PBCSO in the V-shaped transfer function

Function	PBCSO_V1		PBCSO_V2		PBCSO_V3		PBCSO_V4	
	Ave	Std	Ave	Std	Ave	Std	Ave	Std
F1	**0.083**	0.572	**0.0401**	0.3165	**0.03**	0.2231	**0.02**	0.2
F2	**0.0247**	0.174	**0.02**	0.2	**0**	0	**0**	0
F3	**2.3802**	11.9805	**2.6902**	1.4055	**1.813**	2.4819	**3.811**	20.3002
F4	1	0	1	0	1	0	1	0
F5	**18.2954**	82.3694	**4.9068**	40.8017	**69.2836**	95.5807	**7.03**	53.8857
F6	**5.7482**	1.2744	**6.5831**	0.5188	**7.0784**	1.2936	**7.5**	0
F7	**1.749**	6.9382	**1.1915**	4.822	**2.1988**	7.9975	**0.0006772**	0.0045
F8	**-25.2273**	0.1683	**-25.2273**	0.1683	**-25.2357**	0.00841	**-25.2357**	0.0841
F9	**0.03**	0.2227	**0.03**	0.2227	**0.0115**	0.1149	**0.04**	0.3153
F10	**0.0172**	0.123	**0.0101**	0.1007	**0.0106**	0.0791	**0.0284**	0.1721
F11	**0.0017**	0.0097	**0.00083863**	0.0049	**0.0061**	0.0329	**0.00098831**	0.0062
F12	**1.2316**	0.0329	**1.0677**	0.0908	**1.4797**	0.0301	**1.2665**	0.0974
F13	**1.3498E-32**	2.48E-47	**1.3498E-32**	2.48E-47	**1.3498E-32**	2.48E-47	**0.002**	0.02
F14	12.6705	1.61E-14	12.6705	1.61E-14	12.6705	1.61E-14	12.6705	1.61E-14
F15	**0.0144**	3.49E-17	**0.0021**	4.30E-04	0.1484	2.51E-16	0.1484	2.51E-16
F16	0	0	0	0	**-0.69**	4.46E-16	0	0
F17	27.7029	1.43E-14	27.7029	1.43E-14	27.7029	1.43E-14	27.7029	1.43E-14
F18	**89.8751**	7.14E-14	**294.8854**	4.00E-13	**173.2234**	8.57E-14	600	0
F19	-0.3348	6.14E-16	-0.3351	9.48E-16	-0.3348	6.14E-16	-0.3348	6.14E-16
F20	**-0.4575**	1.06E-16	-0.5108	8.93E-16	**-0.6557**	4.46E-16	-0.1657	1.12E-16
F21	-5.0552	8.03E-15	-5.0552	8.03E-15	-5.0552	8.03E-15	-5.0552	8.03E-15
F22	-5.0877	0	-5.0877	0	-5.0877	0	-5.0877	0
F23	-5.1285	1.16E-14	-5.1285	1.16E-14	-5.1285	1.16E-14	-5.1285	1.16E-14

As mentioned above, simulation results prove the superiority of PBCSO to BCSO in accuracy and convergence speed.

6 Traveling Salesman Problem

The present section aims to referral the traveling salesman problem studied in the remainder of this article. Intelligent transportation system is widely used

[31-35]. The earliest mathematical planning of the traveling salesman problem was proposed by Dantzig et al. [36] in 1959, and it is a well-known and extensively studied problem in discrete optimization or combination. The main problem of TSP is that the salesman must return to the original city from the original city to the next city. However, the rule is: Given n cities and their coordinates, the traveling salesman returns to the starting point again after passing through all the cities, and finds the shortest path plan that passes through all the cities once and only once. This problem is how to establish an optimal route by considering these rules to obtain the smallest total mileage, thus impacting saving transportation costs.

Currently, the method of solving TSP is mainly divided into two parts, respectively, traditional method and cluster intelligent algorithm. As mentioned earlier, the traditional TSP solving algorithm has certain limitations. With the development of cluster intelligent algorithms, a lot of numerical optimization algorithms have appeared. The search ability of the algorithm needs to be considered to solve the effect of TSP needs using optimization algorithms. Algorithms with strong optimization capabilities will generate better results.

From the perspective of graph theory, the essence of the problem is to find a Hamiltonian loop with the smallest weight in a weighted completely undirected graph $G=(V,E)$. Simultaneously, $V=(1,2,3,...,n)$ means the set of all vertices (or cities). Edge set E contains the distance between vertices, where $d_{i,j}\,(i,j=1,2,3,...,n)$. The weight of the edge is represented by (i,j). The mathematical model is as follows:

Objective function:

$$\min F=\sum_{i\neq j} d_{ij}x_{ij} \tag{13}$$

Decision variables:

$$X_{i,j}=\begin{cases}1 & if\ \ i,j\ connected\\ 0 & if\ \ i,j\ unconnected\end{cases} \tag{14}$$

$$\sum_{i\neq j} x_{ij}=1 \quad i,j\in V \tag{15}$$

The Equation (14) gives the definition of $x_{i,j}$, and each vertex of the Equation (15) represents each city can be visited once.

The flow chart of PBCSO for TSP is shown in Figure 6:

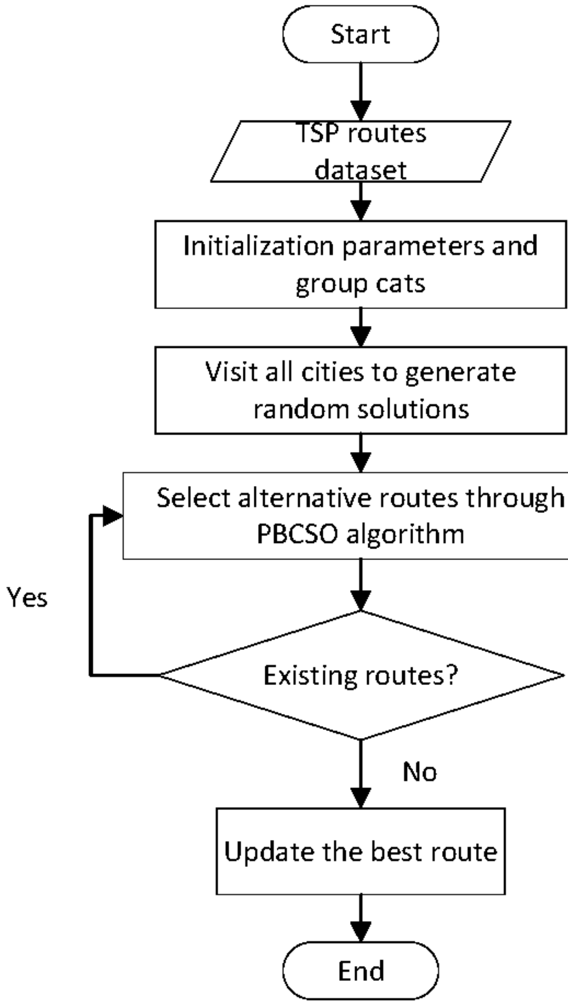


Figure 6. The flow of PBCSO for TSP

TSP is widely used in practical fields such as printed circuit board rotation, vehicle routing, logistics and distribution. For the TSP of n cities, there is $(n+1)!/2$ possible path. Therefore, it is also called NP (non-polynomial) problem because it has a search space in a set of permutations in these cities.

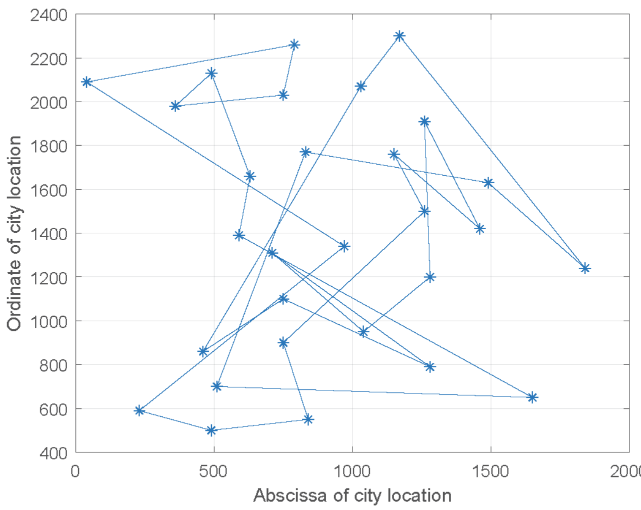
7 Algorithm Performance Test

To verify the practicability of the PBCSO algorithm, this essay selects some cases in the TSPLIB test library. Use MATLAB (R2018b) to program. Test and experiment on win10 PC with the processor Inter (R) Core (TM) i7-8550U CPU @1.80 GHz 2.00 GHz and 8G RAM. The number of cats in the experiment is set to 500, and the maximum number of iterations is set to 1000. The Table 10 records the statistical results of PSO, BCSO and PBCSO to optimize VRP by comparing the best-known solutions.

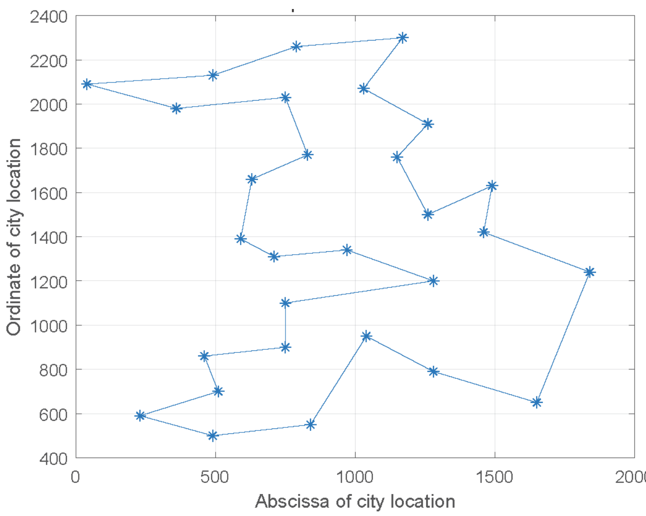
Table 10. PBCSO algorithm and PSO, BCSO algorithm experimental results

Instance	Algorithm	Known best	Optimal value
att48	PSO	10628	54830
	BCSO		34874
	PBCSO		32116
bagy29	PSO	1610	8977
	BCSO		6206
	PBCSO		5528
eil51	PSO	426	689
	BCSO		607
	PBCSO		513
dantzig42	PSO	699	1065
	BCSO		873
	PBCSO		723

Figure 7 shows the path diagram of the initial random solution and the final optimal solution of the optimal solution obtained by PBCSO solving the Bagy29 case.



(a) Random solution



(b) Optimal solution

Figure 7. Random solution (a) and Optimized solution (b) obtained by PBCSO solving example bagy29

8 Conclusion

In this paper, PBCSO solves the discretization problems of TSP. It contains two communication strategies including the neighborhood merger comparison strategy and the dynamic adjustment of the number of subgroups. Finally, we apply the novel algorithm in TSP and obtained results are compared with that of PSO and BCSO. Not only maintains the randomness and diversity of the particles in the search range, but also makes the particles have a greater possibility to retain shorter edges when adjusting their positions. Although the problem is extremely simple, PBCSO also obtains excellent results. A further avenue for future research is to consider different forms of communication strategies, we believe that they can get good results.

References

[1] N. Ramachandran, V. Perumal, Proficient communication between sensor devices using heuristic approaches in iot environment, *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 29, No. 1-2, pp. 15-27, August, 2018.

[2] S.-C. Chu, P.-W. Tsai, J.-S. Pan, Cat Swarm Optimization, in: Q. Yang, G. Webb (Eds.), *PRICAI 2006: Trends in Artificial Intelligence*, Springer, 2006, pp. 854-858.

[3] P.-W. Tsai, J.-S. Pan, S.-M. Chen, B.-Y. Liao, S.-P. Hao, Parallel cat swarm optimization, *2008 international conference on machine learning and cybernetics*, Kunming, China, 2008, pp. 3328-3333.

[4] P.-W. Tsai, J.-S. Pan, S.-M. Chen, B.-Y. Liao, Enhanced parallel cat swarm optimization based on the taguchi method, *Expert Systems with Applications*, Vol. 39, No. 7, pp. 6309-6319, June, 2012.

[5] H. Wang, S. Rahnamayan, H. Sun, M. G. Omran, Gaussian bare-bones differential evolution, *IEEE Transactions on Cybernetics*, Vol. 43, No. 2, pp. 634-647, April, 2013.

[6] Z. Meng, J.-S. Pan, K.-K. Tseng, Pade: An enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization, *Knowledge-Based Systems*, Vol. 168, pp. 80-99, March, 2019.

[7] G. R. Harik, F. G. Lobo, D. E. Goldberg, The compact genetic algorithm, *IEEE transactions on evolutionary computation*, Vol. 3, No. 4, pp. 287-297, November, 1999.

[8] J. Pan, F. McInnes, M. Jack, Application of parallel genetic algorithm and property of multiple global optima to VQ codevector index assignment for noisy channels, *Electronics Letters*, Vol. 32, No. 4, pp. 296-297, February, 1996.

[9] J. Kennedy, R. C. Eberhart, Particle swarm optimization, *IEEE International Conference on Neural Networks*, Vol. 4, Perth, WA, Australia, 1995, pp. 1942-1948.

[10] Eberhart, Y. Shi, Particle swarm optimization: developments, applications and resources, *Congress on Evolutionary Computation*, Vol. 1, Seoul, Korea, 2001, pp. 81-86.

[11] S.-C. Chu, J. F. Roddick, C.-J. Su, J.-S. Pan, Constrained Ant Colony Optimization for Data Clustering, in: C. Zhang, H. W. Guesgen, W.-K. Yeap (Eds.), *Pacific Rim International Conference on Artificial Intelligence*, Springer, 2004, pp. 534-543.

[12] S.-C. Chu, J. F. Roddick, J.-S. Pan, Ant colony system with communication strategies, *Information Sciences*, Vol. 167, No. 1-4, pp. 63-76, December, 2004.

[13] P. Hu, J. S. Pan, S. C. Chu, Improved binary grey wolf optimizer and its application for feature selection, *Knowledge-Based Systems*, Vol. 195, Article No. 105746, May, 2020.

[14] S. Mirjalili, A. Lewis, The whale optimization algorithm, *Advances in Engineering Software*, Vol. 95, pp. 51-67, May, 2016.

[15] Q.-w. Chai, S.-C. Chu, J.-S. Pan, P. Hu, W.-m. Zheng, A parallel WOA with two communication strategies applied in DV-Hop localization method, *EURASIP Journal on Wireless Communications and Networking*, Vol. 2020, pp. 1-10, February, 2020.

[16] X. Xue, A compact firefly algorithm for matching biomedical ontologies, *Knowledge and Information Systems*, Vol. 62, No. 7, pp. 2855-2871, July, 2020.

[17] X. Xue, J. Chen, Optimizing sensor ontology alignment through compact co-firefly algorithm, *Sensors*, Vol. 20, No. 7, Article No. 2056, April, 2020.

[18] X. Xue, Y. Wang, Using memetic algorithm for instance coreference resolution, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 28, No. 2, pp. 580-591, February, 2016.

[19] Y. Sharafi, M. A. Khanesar, M. Teshnehlab, Discrete binary cat swarm optimization algorithm, *IEEE International Conference on Computer*, *Control and Communication*, Karachi, Pakistan, 2013, pp. 1-6.

[20] A. Srivastava, S. Maheswarapu, Optimal pmu placement for complete power system observability using binary cat swarm optimization, *International Conference on Energy Economics and Environment*, Greater Noida, India, 2015, pp. 1-6.

[21] X. Wang, J.-S. Pan, S.-C. Chu, A parallel multi-verse optimizer for application in multilevel image segmentation, *IEEE Access*, Vol. 8, pp. 32018-32030, February, 2020.

[22] S.-C. Chu, Z.-G. Du, J.-S. Pan, Symbiotic organism search algorithm with multi-group quantum-behavior communication scheme applied in wireless sensor networks, *Applied Sciences*, Vol. 10, No. 3, Article No. 930, February, 2020.

[23] J.-S. Pan, X. Wang, S.-C. Chu, T. Nguyen, A multi-group grasshopper optimization algorithm for application in capacitated vehicle routing problem, *Data Science and Pattern Recognition*, Vol. 4, No. 1, pp. 41-56, July, 2020.

[24] X. J. Lin, F. Q. Zhang, L. Xu, Research on high-accuracy gait recognition technique based on parallel genetic algorithm, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 9, No. 1, pp. 244-255, January, 2018.

[25] B.-Q. Jiang, J.-S. Pan, A parallel quasi-affine transformation evolution algorithm for global optimization, *Journal of Network Intelligence*, Vol. 4, No. 2, pp. 30-46, May, 2019.

[26] M. Gao, J.-S. Pan, J.-p. Li, Z.-p. Zhang, Q.-W. Chai, 3-d terrains deployment of wireless sensors network by utilizing parallel gases brownian motion optimization, *Journal of Internet Technology*, Vol. 22, No. 1, pp. 13-29, January, 2021.

[27] Z. Min, G. Yang, J. Wang, G.-j. Kim, A privacy-preserving bgn-type parallel homomorphic encryption algorithm based on lwe, *Journal of Internet Technology*, Vol. 20, No. 7, pp. 2189-2200, December, 2019.

[28] B. Zhang, D. Qin, X. Diao, K. Ding, H. Yan, Efficient searching with a tcam-based parallel architecture, *Journal of Internet Technology*, Vol. 19, No. 7, pp. 2215-2226, December, 2018.

[29] S. Mirjalili, A. Lewis, S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization, *Swarm and Evolutionary Computation*, Vol. 9, pp. 1-14, April, 2013.

[30] M. Molga, C. Smutnicki, *Test functions for optimization needs*, April, 2005. https://robertmarks.org/Classes/ENGR5358/Papers/functions.pdf

[31] M. Sankaranarayanan, C. Mala, S. Mathew, Partition tolerant collaborative gateway architecture for interoperable its applications, *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 35, No. 4, pp. 222-240, November, 2020.

[32] R. Pradhan, T. De, A selective forwarding technique for data dissemination in vehicular ad hoc networks based on traffic parameters, *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 35, No. 4, pp. 179-190, November, 2020.

[33] D. N. M. Dang, Q. T. Ngo, Q. Le-Trung, L. B. Le, An adaptive and cooperative mac protocol in vehicular ad hoc network: design and performance analysis, *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 35, No. 4, pp. 191-204, November, 2020.

[34] K. Yao, Z. Li, L. Yao, K. Lang, Popularity prediction caching based on logistic regression in vehicular content centric networks, *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 35, No. 3, pp. 150-161, October, 2020.

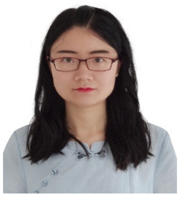
[35] G. Ye, X. Rui, An improved simulated annealing andgenetic algorithm for TSP, *IEEE International Conference on Broadband Network & Multimedia Technology*, Guilin, China, 2013, pp. 6-9.

[36] G. B. Dantzig, J. H. Ramser, The truck dispatching problem, *Management science*, Vol. 6, No. 1, pp. 80-91, October, 1959.

Biographies

Jeng-Shyang Pan received the B.S. degree in electronic engineering from the National Taiwan University of Science and Technology in 1986, the M.S. degree in communication engineering from National Chiao Tung University, Taiwan, in 1988, and the Ph.D. degree in electrical engineering from the University of Edinburgh, U.K., in 1996. He is currently the Director of the Fujian Provincial Key Lab of Big Data Mining and Applications, and an Assistant President with the Fujian University of Technology.

He is also the Professor with the Harbin Institute of Technology. He is the IET Fellow, U.K., and has been the Vice Chair of the IEEE Tainan Section. He was offered Thousand Talent Program in China in 2010.



Xiao-Fang Ji received the B.S. degree from the School of electronic communication engineering, Anhui Xinhua University, China, in 2018. She is currently pursuing the master's degree with the Shandong University of Science and Technology, Qingdao, China. Her recent research interests include swarm intelligence and transportation logistics.

Anhui Liang is the second level professor, Shandong University of Science and Technology, China. He held several positions, e.g. Chief Scientist, FiberHome Technologies; Chief Scientist, WTD; Deputy Director of University Academic Committee, Nanjing University of Posts and Telecommunications, and Tyco Submarine Systems Ltd. in USA etc. He has published more than 100 papers and patents. He has made significant contributions in the fields of optical fiber communications, vision，biological optical AI, quantum mechanics and Chinese meridian, chromosome optical fibers and biological fibers. He is China Overseas Chinese Contribution Award recipient (2014); Yearly Person of "Scientific Chinese"(2015).

Kuan-Chun Huang received the B.S. degree in department of Mechanical Engineering from National Central University in Taiwan He received the Ph.D. degree in Mechanical Engineering from the University of Edinburgh, U.K, in 1998. He is an assistant professor in the department of Industrial and Systems Engineering in Chung Yuan Christian University in Taiwan. His research interests include soft computing, artificial intelligence, and system integration etc.

Shu-Chuan Chu received the Ph.D. degree in 2004 from the School of Computer Science, Engineering and Mathematics, Flinders University of South Australia. She joined Flinders University in December 2009 after 9 years at the Cheng Shiu University, Taiwan. She is the Research Fellow in the College of Science and Engineering of Flinders University, Australia from December 2009. Currently, she is the Research Fellow with PhD advisor in the College of Computer Science and Engineering of Shandong University of Science and Technology from September 2019. Her research interests are mainly in Swarm Intelligence, Intelligent Computing and Data Mining.