Binary Particle Swarm Optimization Algorithm Based on Z-shaped Probability Transfer Function to Solve 0-1 Knapsack Problem

Wei-Zhong Sun, Min Zhang, Jie-Sheng Wang *, Sha-Sha Guo, Min Wang, Wen-Kuo Hao

***Abstract*—Binary particle swarm optimization (BPSO) algorithm can map the original continuous searching space to the binary searching space by introducing a new velocity transfer function. For the 0-1 knapsack problem, a binary particle swarm optimization algorithm based on the Z-shaped probability transfer function was proposed. In order to solve the shortcomings of BPSO algorithm based on S-shaped and V-shaped probability functions that it is easy to fall into local optima and slow convergence speed, a new probability function (Z-shaped transfer function) was proposed. Then a penalty function strategy is adopted to deal with the violation of the constraint solutions. In order to verify the effectiveness of the proposed algorithm, the BPSO algorithm based on the Z-shaped transfer function with different parameters was used to solve the typical 0-1 knapsack problems,which is compared with the BPSO algorithm based on the S-shape and the V-shape transfer functions. Simulation experiment results show that the proposed Z-shaped probability transfer function improves the convergence speed and optimization accuracy of the BPSO algorithm.**

***Index Terms*—0-1 Knapsack Problem, Binary Particle Swarm Optimization Algorithm, Transfer Function**

I. Introduction

THE knapsack problem (KP) and its variables are part of the classic NP-hard combinatorial optimization problems[1]. The current classification of knapsack problems includes 0-1 knapsack problem, multi-choice knapsack problem, multi-constraint knapsack problem, bounded and unbounded knapsack problems, etc. The most basic knapsack problem is 0-1 knapsack problem (0-1 KP), which plays an important role in many fields, such as resource allocation, warehouse loading and project selection [2-3].

With the development of computer technology, many heuristic algorithms have been used to solve the 0-1 knapsack problems in recent decades. Zou et al. [4] proposed a new global harmony search algorithm for the 0-1 knapsack problem (NGHS). The algorithm includes two important operations: location update and small probability genetic mutation. The former can make the worst harmony of the harmony memory quickly move to the global optimal harmony in each iteration, and the latter can effectively prevent the NGHS from falling into the local optimal harmony. Zhang et al. [5] proposed a new bionic model to solve this problem. There are three main steps in the new method. First, the 0-1 knapsack problem is transformed into a directed graph problem using a network conversion algorithm. Then, the longest path problem is transformed into the shortest path problem by using the amoeba biological model. Finally, the amoeba algorithm can be used to solve the shortest path problem well. Bhattacharjee et al. [1] proposed an modified discrete shuffled frog leaping (MDSFL) algorithm to solve the 0-1 knapsack problems. The proposed algorithm includes two important operations: the "particle swarm optimization" technology of local search and the information mixed competitiveness of "shuffle complex evolution" technology. In order to improve the performance of the harmony search algorithm in solving discrete optimization problems, Tuo et al. [6] proposed a new harmony search algorithm based on teaching-learning strategies to solve the 0-1 knapsack problem. In the HSTL algorithm, a method of dynamically adjusting the dimension of the selected harmony vector in the optimization process is first proposed. In addition, four strategies, including harmony memory, teaching-learning strategies, local pitch adjustment and random mutation, are used to improve the performance of the HS algorithm. Another improvement of the HSTL method is the use of dynamic strategies to change parameters, which effectively maintains the balance between global exploration and local development. Truong et al. [7] proposed a new artificial chemical reaction optimization algorithm based on the greedy strategy for the 0-1 knapsack problem. The artificial chemical reaction optimization (ACROA) that stimulates the chemical reaction process is used to realize local and global search. The repair operator combining greedy strategy and random selection is used to repair the in-feasible solution. Zhou et al.

Manuscript received November 15, 2020; revised February 15, 2021. This work was supported by the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province (Grant No. 2017FWDF10), and the Project by Liaoning Provincial Natural Science Foundation of China (Grant No. 20190550263).

Wei-Zhong Sun is a lecturer of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China. (e-mail: weizhongsun@126.com).

Min Zhang is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China. (e-mail:zhangmin09118@163.com).

Jie-Sheng Wang is a professor of School of International Finance and Banking, University of Science and Technology Liaoning, Anshan, 114051, P. R. China. (Corresponding author, phone: 86-0412-2538246; fax: 86-0412-2538244; e-mail: wang_jiesheng@126.com).

Sha-Sha Guo is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China. (e-mail: gss420924@163.com).

Min Wang is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China. (e-mail: minw996@163.com).

Wen-Kuo Hao is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China. (e-mail: 864664273@qq.com).

[8] aimed at the problem that many algorithms have low accuracy in solving 0-1 knapsack problems and are easy to fall into local optimal solution, a binary version of monkey algorithm based on greedy strategy was proposed to enhance the local search ability, which is modified in the process of transformation to avoid falling into the local optimal solution, and the cooperative process is adopted to accelerate the convergence speed of the algorithm. Zhou et al. [9] proposed a new complex-valued coding bat algorithm (CPBA), where an effective global optimization strategy-complex value coding method was introduced. According to the two-dimensional nature of complex numbers, the real and imaginary parts of the complex numbers are updated respectively. This algorithm can effectively disperse the bat population and improve the convergence performance. CPBA improves detection capabilities and is effective for solving small-scale and large-scale 0-1 knapsack problems.

In the past two years, many scholars have proposed many other strategies to solve 0-1 KP problems to further improve the performance of the algorithm. Gao et al.[10] proposed a quantum-encoding-based wolf packet algorithm (QWPA) to improve its performance and solve the 0-1 knapsack problems. There are two important operations in quantum mechanics: quantum rotation and quantum collapse. The first step moves the population to the global optimal value, and the second step helps to avoid individuals from falling into the local optimal value. Rizk-Allah et al. [11] proposed a new binary bat algorithm (NBBA) to solve the 0-1 knapsack problems, which combines two important stages of binary bat algorithm (BBA) and local search scheme (LSS). Bat algorithm aims to improve the exploration ability of bats, while LSS aims to improve the utilization trend, so it can prevent BBA-LSS from falling into local optimum. In addition, LSS starts searching from the BBA that has been found so far. In this way, BBA-LSS enhances the diversity of bats and improves the convergence performance of bats. Huang et al. [12] proposed a binary multi-scale quantum harmonic oscillator algorithm (BMQHOA) based on genetic operators to solve the 0-1 knapsack problems. The framework consists of three nested stages: energy level stabilization, energy level drop and scale adjustment. In BMQHOA, the number of different bits between solutions is defined as the distance between solutions, which is used to map a continuous search space to a discrete search space. In order to ensure the constraint of the knapsack capacity, a greedy strategy repair operator was used in BMQHOA. This paper proposes a binary particle swarm optimization algorithm based on Z-shaped probability transfer function to solve the 0-1 knapsack problems. The simulation results verify the effectiveness of the proposed algorithm.

II. 0-1 Knapsack Problem

A. Mathematical Model of 0-1 Knapsack Problem

Knapsack problem is a combinatorial optimization problem with a wide range of applications. It is not only used in investment decision-making, loading, inventory and so on, but also often appears in the form of sub problems in the large-scale optimization problems[1]. The general description of the knapsack problem is described as follows.

Item set $U=\{u_1,u_2,\cdots,u_n\}$ is a set of items to be put into a backpack with a capacity of $C\in Z^+$. Select some items in U (denoted as set A) and load them into the backpack so that the total weight $\sum_{u_i\in A}^{'} w_i$ of these items does not exceed C , and make the total value reach the maximum $\sum_{u_i\in A}^{'} v_i$. Generally speaking, the 0-1 KP can be defined as follows [13].

$$f(x)=\sum_{i=1}^{n} v_i x_i \tag{1}$$

The constraints are described as follows:

$$\sum_{i=1}^{n} w_i x_i \le C \tag{2}$$

where, $x_i = 0$ *or* 1 , $i=1,2,\cdots,n$. The 0-1 variable x_i has the following meanings:

$$x_j=\begin{cases}1, & if\ u_i\in A\\ 0, & if\ u_i\notin A\end{cases} \tag{3}$$

where, w_i indicates the weight of the i -th item, v_i indicates the value of the i -th item, and x_i indicates whether the i -th item is put in the backpack.

B. Coding Method and Constraint Handling of 0-1 Knapsack Problem

(1) Coding Method

Based on the characteristics of the 0-1 knapsack problem, as described in Eq. (3), a binary coding method is used, that is to say the code 1 means the item is put into the backpack and the code 0 means the item was not put in the backpack.

(2) Constraint Processing

According to Eq. (3), it can be seen that when solving the 0-1 KP, we will encounter a constraint violation. In this paper, the punishment function will be adopt to deal with the solution that violates the constraint.

$$f(x)=\begin{cases}f(x) & if\quad w_i x_i\le C\\ f(x)-\alpha(w_i x_i-C) & if\quad w_i x_i > C\end{cases} \tag{4}$$

where, α is the penalty factor. After experimental analysis, the penalty factor is set as 2 in the following simulation experiments.

III. Particle Swarm Optimization Algorithm

Particle swarm optimization (PSO) algorithm is an swarm intelligent optimization algorithm proposed by J. Kennedy and R. C. Eberhart in 1995 [14-15]. It simulates the flight and foraging behavior of birds and optimizes the colony of birds through collective cooperation between birds. In the PSO algorithm, the potential solution Is a bird in the searching space, named as a particle. Later, Y. Shi and R. C. Eberhart added a new impact factor w to improve detection and exploration, and formed the current standard PSO algorithm [16]. All particles have an appropriate value determined by the optimization function. Each particle has a speed, which determines the direction and distance of their flight,and then the particle follows the current optimal particle to search in the solution space. The velocity and position of the particles are updated by:

$$v_{id} = w * v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \quad (5)$$

$$x_{id} = x_{id} + v_{id} \quad (6)$$

where, v_{id} is the velocity of the i -th particle in the d -th dimension; p_{id} is the optimal position of the i -th particle so far; x_{id} is the current position of the i -th particle in the d -th dimension; p_{gd} is the best position searched by PSO algorithm so far; w is the inertia weight, and linearly decreasing weights are used in this article, that is to say $w_{\max} - (t * (w_{\max} - w_{\min})) / t_{\max}$, where $w_{\max}$ represents the maximum value of the inertia weight, $w_{\min}$ represents the minimum value of the inertia weight, t represents the current iteration number, and $t_{\max}$ represents the maximum number of iterations. This principle is still used in the binary version PSO algorithm; r_1 and r_2 are two acceleration weight coefficients randomly generated between [0,1]; c_1 and c_1 are acceleration numbers, and the value 2 is used in this paper [17].

IV. Binary Particle Swarm Optimization Algorithm

A. Typical Binary Particle Swarm Optimization Algorithm

In binary particle swarm optimization (BPSO) algorithm, the velocity update equation has not changed, but a new velocity transfer function is introduced to map the original continuous searching space to the binary searching space. The concept of transfer function was originally proposed by Kennedy and Eberhart [18], which allows PSO algorithm to run in a binary searching space. In this version, the particle position vector can only be 0 or 1.The effect of velocity is the probability that the indication bit takes 0 or 1, so they propose a Sigmoid transfer function. As shown in Eq. (7), this function can convert all real values of velocity into probability values [0,1].

$$T(v_i^k(t)) = \frac{1}{1 + e^{-v_i^k(t)}} \quad (7)$$

where, $v_i^k(t)$ represents the velocity of the particle i at the k -th dimension and t -th iteration.

After converting the velocity into a probability value, the position vector can be updated with the probability of its velocity described as follows.

$$x_i^k(t+1) = \begin{cases} 0 & if \quad rand < T(v_i^k(t)) \\ 1 & if \quad rand \geq T(v_i^k(t)) \end{cases} \quad (8)$$

where, $v_i^k(t)$ represents the velocity of the particle i at the k -th dimension and t -th iteration.

This transfer function is called the S -shaped transfer function, and a set of S -shaped transfer functions are formed by changing the parameter. Their expressions are shown in Table 1 [17].

In 2009, Rashedi et al. proposed a new transfer function named V-shaped transfer function and a new position update strategy [19], which are shown in Eq. (9).

$$x_i^k(t+1) = \begin{cases} (x_i^k(t+1))^{-1} & if \quad rand < T(x_i^k(t)) \\ x_i^k(t) & if \quad rand \geq T(x_i^k(t)) \end{cases} \quad (9)$$

where, $x_i^k(t)$ is the position of the i -th particle in the k -th dimension and the t -th iteration; $(x_i^k(t+1))^{-1}$ is the complement of $x_i^k(t)$.

According to the characteristics of the V-shaped transfer function, a series of V-shaped transfer functions are proposed by using different functional equations. Their expressions are shown in Table 2 [19].

B. Improved Binary Particle Swarm Optimization Algorithm

According to the characteristics of the BPSO algorithm, the continuous searching space is mapped to the discrete binary space. The purpose of the transfer function is to express the probability of the elements of the position vector from 0 to 1, so the transfer function must be a bounded function [0,1]. In addition, when the velocity value is 0, the probability of change should be relatively small, because when the particle finds the optimal value, the velocity should be reduced to 0, and the probability of particle position change should be 0. According to the characteristics of the transfer function, a new Z-shaped transfer function is proposed [20], which is defined as follows.

$$T(x_i^k(t)) = \sqrt{1 - a^{x_i^k(t)}} \quad (10)$$

TABLE 1. S-SHAPED TRANSFER FUNCTIONS

Algorithm name	Name	Expression
BPSO1	S_1	$T_1(x) = 1/(1+e^{-2x})$
BPSO2	S_2	$T_2(x) = 1/\left(1+e^{-x}\right)$
BPSO3	S_3	$T_3(x) = 1/\left(1+e^{-x/2}\right)$
BPSO4	S_4	$T_3(x) = 1/\left(1+e^{-x/3}\right)$

TABLE 2. V-SHAPED TRANSFER FUNCTIONS

Algorithm name	Name	Expression
BPSO5	V_1	$T_5(x) = \left\|erf\left((\pi/2)x\right)\right\|$
BPSO6	V_2	$T_6(x) = \left\|\tanh(x)\right\|$
BPSO7	V_3	$T_7(x) = \left\|(x)/\sqrt{1+x^2}\right\|$
BPSO8	V_4	$T_8(x) = \left\|(2/\pi)\arctan(\pi/2)\right\|$

where, $x_i^k(t)$ is the position of the i -th particle in the k -th dimension and the t -th iteration; a is a positive integer. By changing the value of a , a set of Z -shaped functions are obtained. Their expressions and curves are shown in Table 3 and Fig. 1 respectively [20].

V. Improved BPSO Algorithm to Solve 0-1 Knapsack Problems

This paper has conducted a lot of experimental research to verify the performance of the improved BPSO algorithm. We conducted small-scale (Test 1-Test 6), medium-scale (Test 7-Test 9) and large-scale (Test 10) random tests on BPSO algorithms. This part is divided into two groups of experiments. The first group of simulation experiments are used to test the performance of the Z-shaped transfer function under different parameters, and the second group of simulation experiments is based on the first group, that is to say that the Z-shaped transfer function under the best parameter in the first group is selected and compared with the S-shaped and V-shaped transfer function, which verifies the effectiveness of the improved BPSO algorithm.

A. Solving 0-1 Knapsack Problem by BPSO Algorithm Based on Z-shaped Transfer Functions with Different Parameters

In order to make the experimental results more accurate and objective, in this simulation experiments, each set of 0-1 NP problems are run 10 times, then record the best value (Best) and worst value (Worst) of the 10 results, and calculate the average (Ave) and variance (Std) of the results. The simulation results are listed in Table 4 and the convergence curves are shown in Fig. 2.

Table 3. Z-shaped transfer functions

Algorithm name	Name	Expression
BPSO9	Z_1	$T_9(x)=\sqrt{1-2^x}$
BPSO10	Z_2	$T_{10}(x)=\sqrt{1-5^x}$
BPSO11	Z_3	$T_{11}(x)=\sqrt{1-8^x}$
BPSO12	Z_4	$T_{12}(x)=\sqrt{1-20^x}$

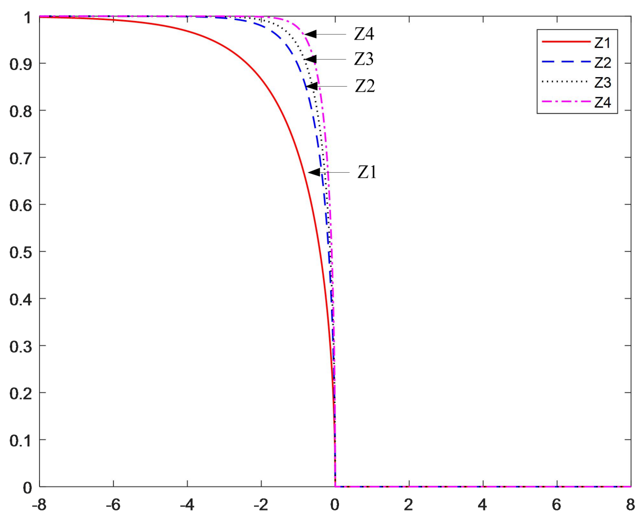
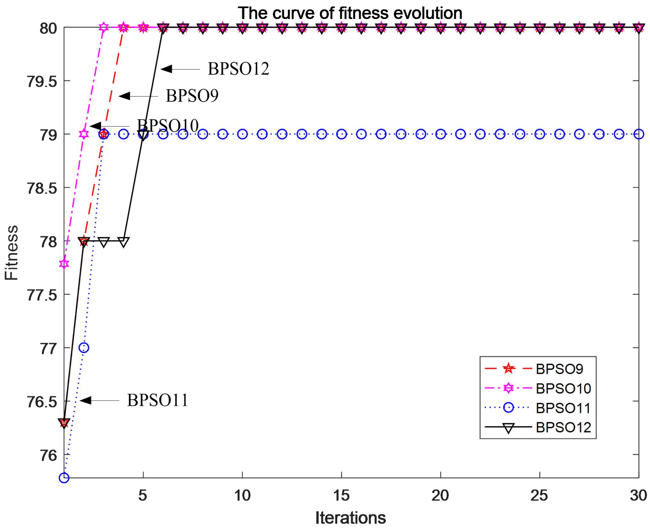


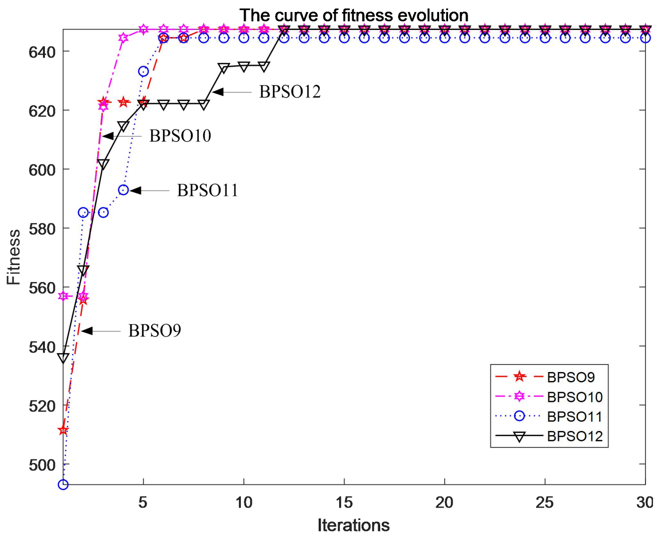
Fig. 1 Z-shaped transfer functions.

Table 4. 0-1 knapsack problems solved by BPSO algorithms based on Z-shaped Probability Transfer Function with different parameters

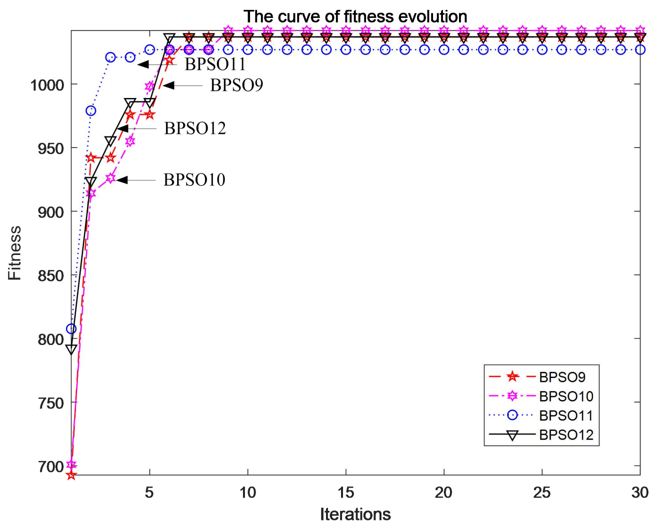
		Best	Worst	Ave	Std
Test 1	BPSO9	**80.00**	80.00	80.00	0.00
	BPSO10	**80.00**	80.00	80.00	0.00
	BPSO11	**80.00**	79.00	79.90	0.30
	BPSO12	**80.00**	80.00	80.00	0.00
Test 2	BPSO9	**647.40**	647.40	647.40	1.14E-13
	BPSO10	**647.40**	647.40	647.40	1.14E-13
	BPSO11	**647.40**	644.50	646.53	1.33
	BPSO12	**647.40**	644.50	647.11	0.87
Test 3	BPSO9	**1042.00**	1037.00	1040.00	2.45
	BPSO10	**1042.00**	1037.00	1041.50	1.50
	BPSO11	**1042.00**	1027.00	1036.50	5.22
	BPSO12	**1042.00**	1032.00	1038.50	3.20
Test 4	BPSO9	**1042.00**	1037.00	1040.00	2.45
	BPSO10	**1042.00**	1042.00	1042.00	0.00
	BPSO11	**1042.00**	1032.00	1038.50	3.20
	BPSO12	**1042.00**	1027.00	1038.00	4.36
Test 5	BPSO9	440.00	408.00	431.00	10.06
	BPSO10	**452.00**	440.00	450.30	3.74
	BPSO11	**452.00**	423.00	442.00	9.87
	BPSO12	**452.00**	422.00	436.60	9.40
Test 6	BPSO9	10911.00	10898.00	10905.90	7.00
	BPSO10	**10913.00**	10909.00	10912.00	1.61
	BPSO11	**10913.00**	10906.00	10909.80	2.18
	BPSO12	**10913.00**	10895.00	10908.90	5.19
Test 7	BPSO9	2932.00	2721.00	2797.10	66.84
	BPSO10	**2997.00**	2906.00	2949.20	28.90
	BPSO11	2928.00	2825.00	2875.00	29.43
	BPSO12	2955.00	2862.00	2903.60	26.50
Test 8	BPSO9	3756.00	3479.00	3652.20	97.92
	BPSO10	**4002.00**	3794.00	3896.20	53.33
	BPSO11	3887.00	3696.00	3825.40	56.25
	BPSO12	3944.00	3719.00	3837.90	57.25
Test 9	BPSO9	3450.00	3167.00	3327.60	68.52
	BPSO10	**3671.00**	3535.00	3630.70	37.74
	BPSO11	3649.00	3456.00	3558.20	51.86
	BPSO12	3666.00	3505.00	3573.50	45.32
Test 10	BPSO9	6502.00	6139.00	6349.20	123.92
	BPSO10	**7032.00**	6701.00	6862.40	115.36
	BPSO11	6886.00	6533.00	6718.10	94.12
	BPSO12	6919.00	6486.00	6752.20	123.58



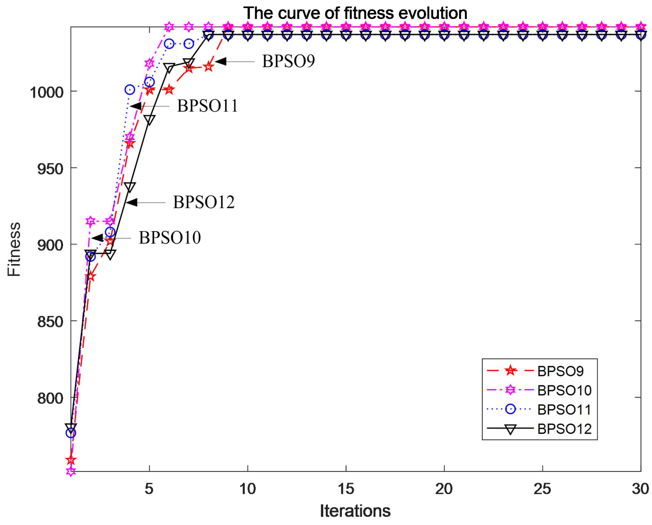
(a) Test 1



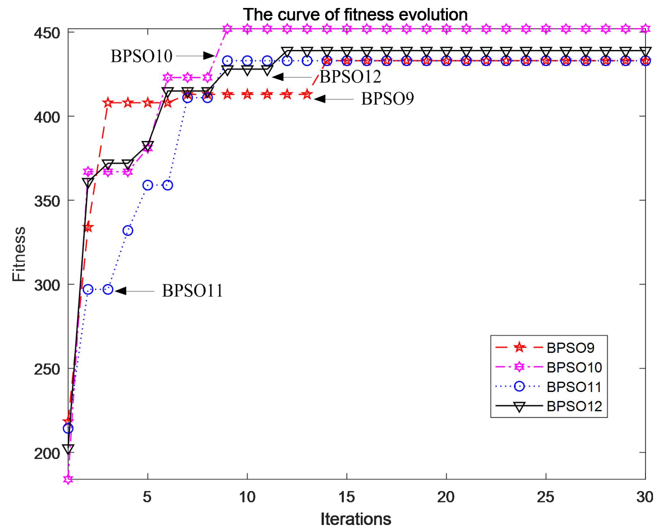
(b) Test 2



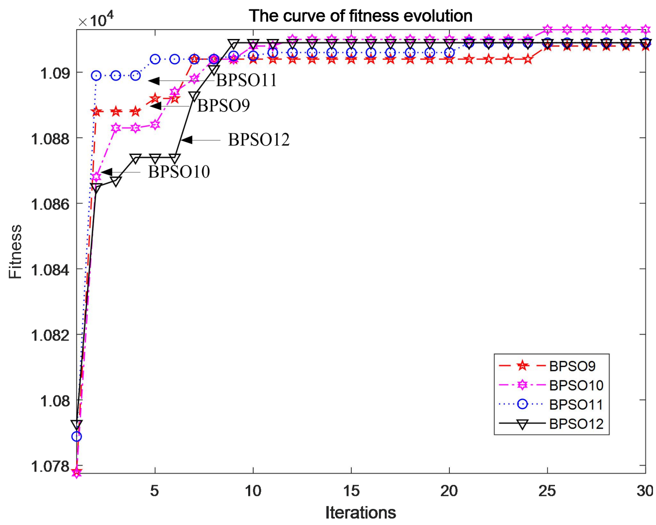
(c) Test 3



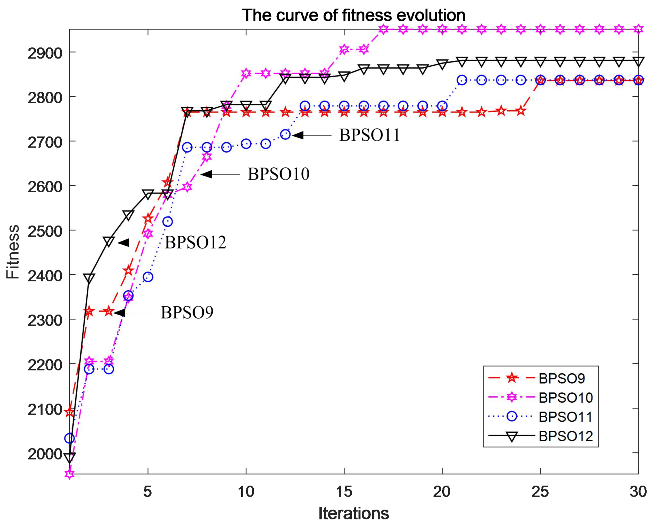
(d) Test 4



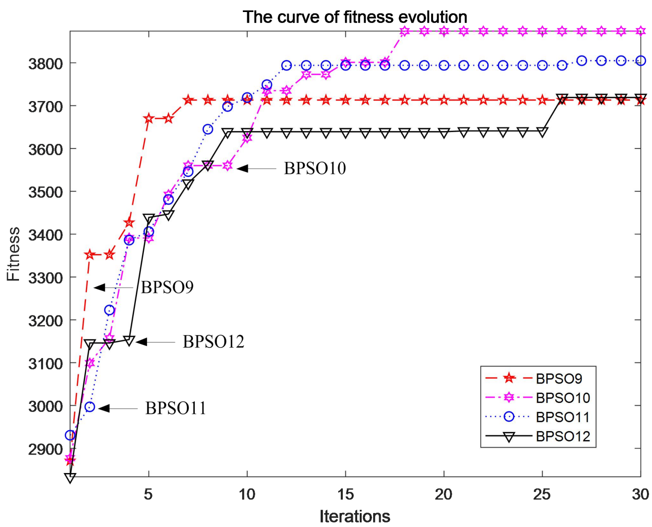
(e) Test 5



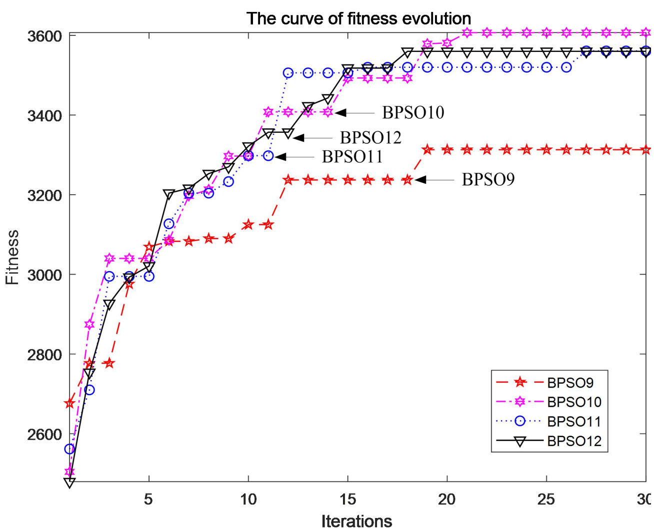
(f) Test 6



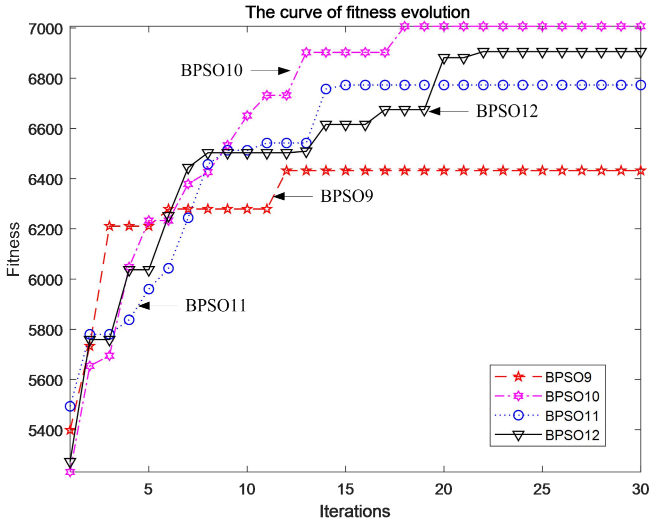
(g) Test 7



(h) Test 8



(i) Test 9



(j) Test 10

Fig. 2 0-1 knapsack problems solved by BPSO algorithms with Z-shaped Probability Transfer Function with different parameters.

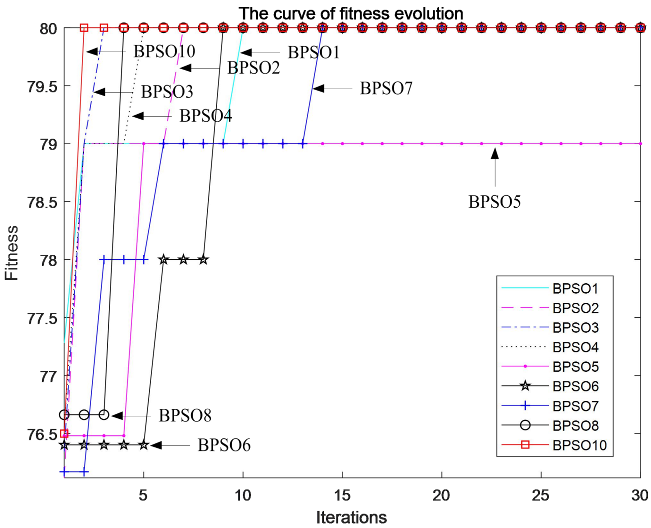
It can be seen from Table 4 that for the small-scale knapsack problems, changing the parameters of the Z-shaped transfer function has little effect on the results of the experiments. Especially for Test 1-Test 4, the results obtained by the Z-shaped transfer function under different parameters are the same. However, as the scale increases, such as Test 5- Test 6, when the parameter of BPSO10 is set as 5, it shows better performance. For the medium-scale knapsack problems, as shown in Test 7-Test 9, BPSO10 has found a better optimal value. It is also seen from the Std data that the value of BPSO10 is the smallest, that is to say that when the parameter is set to 5, the algorithm is the best stable and has better robustness. For the large-scale knapsack problems, it can be seen from the test results that the data of BPSO9-BPSO12 first becomes larger and then smaller, reaching the optimal value at BPSO10, and the best solution is obtained. It can be seen from Fig. 2 that for small-scale problems, although the experimental results are similar, in most cases BPSO10 can find the optimal value first and has a faster convergence speed. For medium-scale problems, BPSO10 has the highest convergence accuracy, and Test 7 and Test 9 have the fastest convergence speed. For large-scale problems, BPSO10 has the highest convergence accuracy. In terms of convergence speed, although it is slower than BPSO9 and BPSO11, it may converge when the number of iterations is 17, so it is within an acceptable range.

B. Solving 0-1 Knapsack Problem by BPSO Algorithm Based on Different Transfer Functions

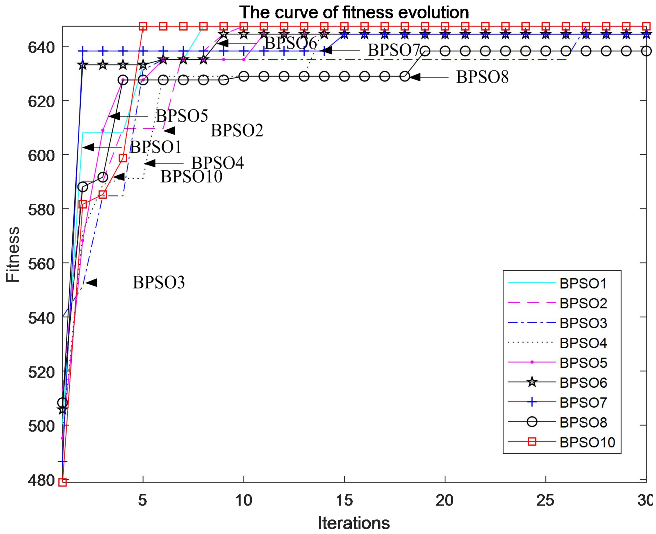
In the experiments in this section, in order to further verify the performance of the improved algorithm, the best Z-shaped transfer function (BPSO10 with parameter 5) was carried out comparison experiments with S-shaped and V-shaped transfer function. In the simulation experiments, the above 10 sets of data are also selected, and each set of data was run 10 times. The experimental results are shown in Table 5, and the convergence curves are shown in Fig. 3.

TABLE 5. 0-1 KNAPSACK PROBLEMS SOLVED BY BPSO ALGORITHMS WITH DIFFERENT TRANSFER FUNCTIONS

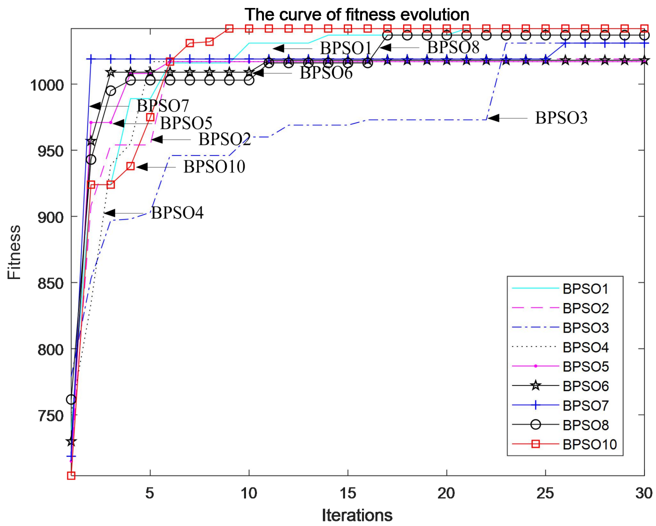
		Best	Worst	Ave	Std
Test 1	BPSO1	**80.00**	80.00	80.00	0.00
	BPSO2	**80.00**	80.00	80.00	0.00
	BPSO3	**80.00**	80.00	80.00	0.00
	BPSO4	**80.00**	80.00	80.00	0.00
	BPSO5	**80.00**	79.00	79.90	0.30
	BPSO6	**80.00**	80.00	80.00	0.00
	BPSO7	**80.00**	80.00	80.00	0.00
	BPSO8	**80.00**	80.00	80.00	0.00
	BPSO10	**80.00**	80.00	80.00	0.00
Test 2	BPSO1	**647.40**	647.40	647.40	0.00
	BPSO2	**647.40**	647.38	647.39	0.01
	BPSO3	**647.40**	644.50	647.10	0.87
	BPSO4	**647.40**	627.91	645.11	5.82
	BPSO5	**647.40**	635.13	643.45	4.95
	BPSO6	**647.40**	644.50	646.52	1.32
	BPSO7	**647.40**	633.16	644.52	4.02
	BPSO8	**647.40**	634.67	644.29	4.82
	BPSO10	**647.40**	647.40	647.40	0.00
Test 3	BPSO1	**1042.00**	1016.00	1036.40	7.50
	BPSO2	**1042.00**	1007.00	1026.20	9.95
	BPSO3	1037.00	1013.00	1026.70	9.71
	BPSO4	1032.00	994.00	1009.10	11.12
	BPSO5	1037.00	1008.00	1025.60	9.93
	BPSO6	1037.00	1016.00	1025.00	7.80
	BPSO7	1037.00	1017.00	1032.10	5.75
	BPSO8	1037.00	985.00	1028.00	15.74
	BPSO10	**1042.00**	1042.00	1042.00	0.00
Test 4	BPSO1	**1042.00**	1016.00	1035.20	7.78
	BPSO2	1037.00	1015.00	1027.00	8.38
	BPSO3	1027.00	992.00	1012.00	9.15
	BPSO4	**1042.00**	993.00	1016.10	15.51
	BPSO5	1037.00	1019.00	1030.00	7.80
	BPSO6	1037.00	1009.00	1026.40	9.54
	BPSO7	1037.00	1010.00	1023.80	9.05
	BPSO8	1037.00	997.00	1021.90	10.18
	BPSO10	**1042.00**	1037.00	1041.50	1.50
Test 5	BPSO1	452.00	423.00	439.80	9.14
	BPSO2	452.00	421.00	431.40	8.67
	BPSO3	439.00	387.00	418.60	14.16
	BPSO4	428.00	392.00	413.00	13.57
	BPSO5	376.00	237.00	318.80	38.79
	BPSO6	406.00	237.00	343.00	52.69
	BPSO7	423.00	322.00	386.30	26.59
	BPSO8	435.00	366.00	395.10	21.78
	BPSO10	**452.00**	452.00	452.00	0.00
Test 6	BPSO1	10910.00	10899.00	10903.70	2.90
	BPSO2	10910.00	10898.00	10903.50	3.47
	BPSO3	10906.00	10891.00	10898.10	5.07
	BPSO4	10904.00	10889.00	10896.60	4.90
	BPSO5	10911.00	10870.00	10885.95	11.70
	BPSO6	10896.00	10873.00	10888.20	6.24
	BPSO7	10901.00	10877.00	10891.90	6.77
	BPSO8	10904.00	10888.00	10896.90	5.54
	BPSO10	**10913.00**	10909.00	10911.40	1.74
Test 7	BPSO1	2717.00	2584.00	2651.80	42.15
	BPSO2	2686.00	2504.00	2578.70	57.29
	BPSO3	2661.00	2437.00	2520.70	70.98
	BPSO4	2537.00	2436.00	2488.50	33.52
	BPSO5	2568.00	2230.00	2382.50	82.31
	BPSO6	2537.00	2324.00	2385.70	63.14
	BPSO7	2644.00	2329.00	2434.70	86.01
	BPSO8	2551.00	2370.00	2444.60	53.90
	BPSO10	**2952.00**	2840.00	2898.50	32.41
Test 8	BPSO1	3645.00	3482.00	3568.40	56.32
	BPSO2	3538.00	3414.00	3480.00	31.43
	BPSO3	3466.00	3311.00	3384.80	45.11
	BPSO4	3535.00	3298.00	3420.20	67.08
	BPSO5	3317.00	3144.00	3244.70	52.40
	BPSO6	3384.00	3093.00	3230.20	85.83
	BPSO7	3455.00	3178.00	3315.30	79.72
	BPSO8	3488.00	3272.00	3366.20	65.00
	BPSO10	**3937.00**	3677.00	3830.90	78.73
Test 9	BPSO1	3382.00	3218.00	3292.30	44.77
	BPSO2	3301.00	3125.00	3214.00	55.28
	BPSO3	3205.00	3047.00	3111.60	49.13
	BPSO4	3145.00	2993.00	3050.60	50.43
	BPSO5	3094.00	2728.00	2918.90	87.43
	BPSO6	3091.00	2850.00	2955.60	69.86
	BPSO7	3049.00	2919.00	2990.30	32.98
	BPSO8	3166.00	2927.00	3065.20	61.58
	BPSO10	**3659.00**	3417.00	3551.20	82.34
Test 10	BPSO1	6581.00	6357.00	6461.00	63.41
	BPSO2	6412.00	6121.00	6277.00	91.67
	BPSO3	6276.00	6014.00	6104.40	74.31
	BPSO4	6223.00	5997.00	6099.30	64.32
	BPSO5	6302.00	5914.00	6066.40	105.03
	BPSO6	6264.00	5878.00	6089.20	135.38
	BPSO7	6360.00	5884.00	6078.50	118.91
	BPSO8	6431.00	6014.00	6177.90	110.52
	BPSO10	**7023.00**	6639.00	6806.60	137.26



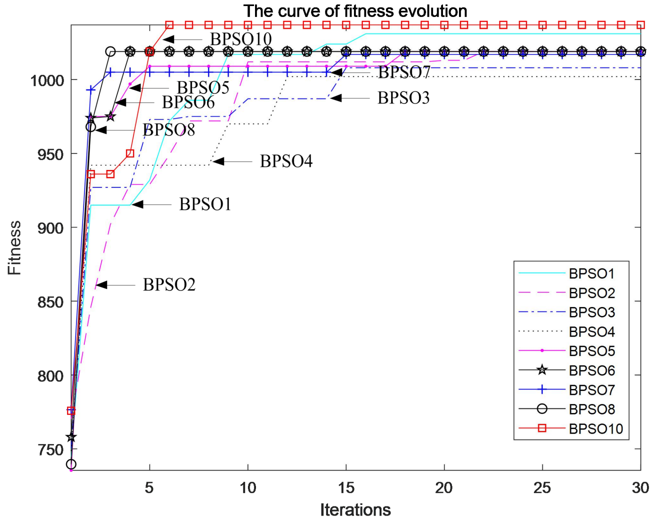
(a) Test 1



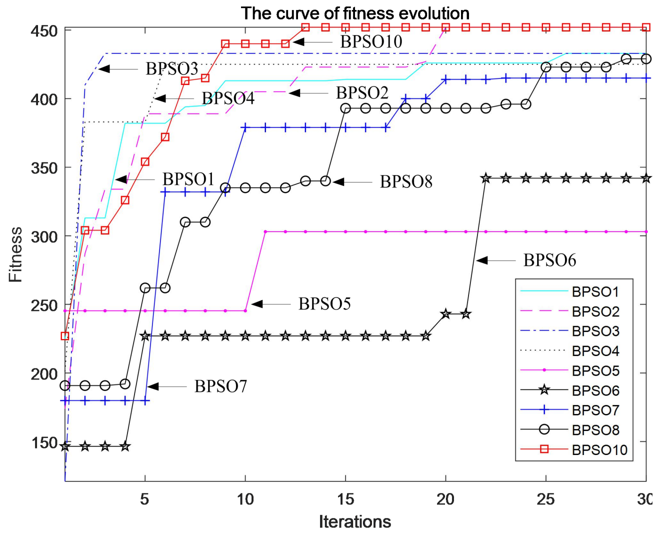
(b) Test 2



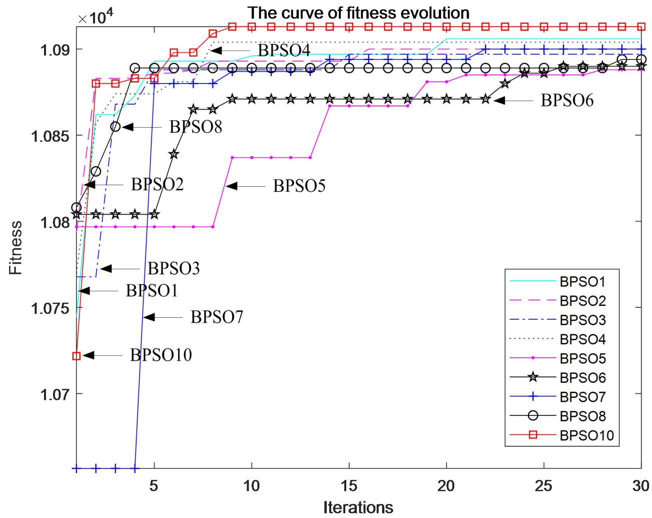
(c) Test 3



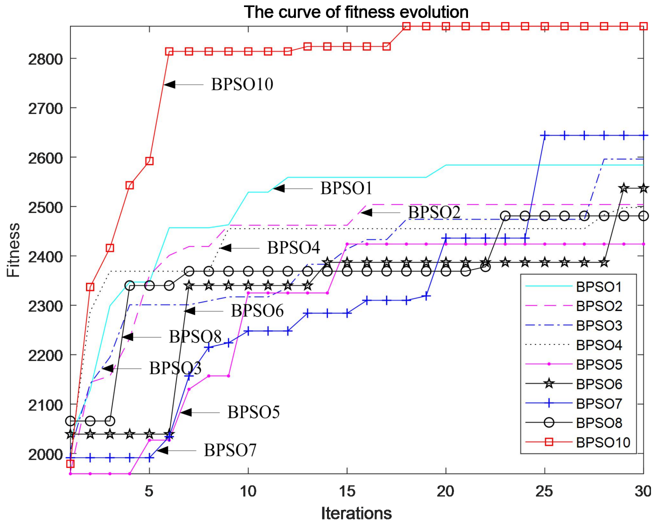
(d) Test 4



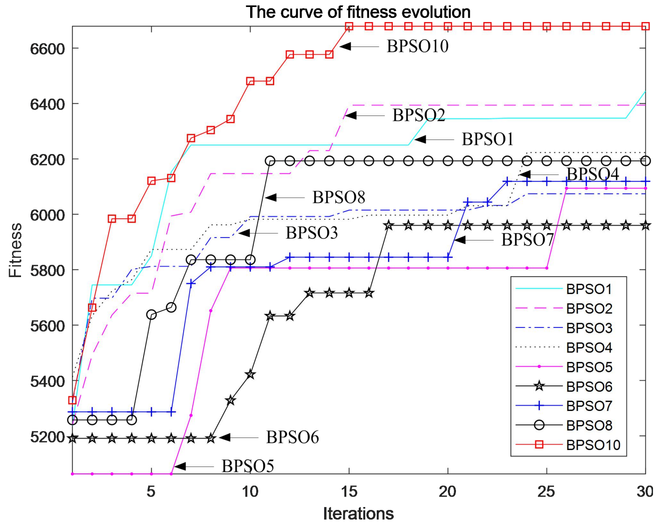
(e) Test 5



(f) Test 6

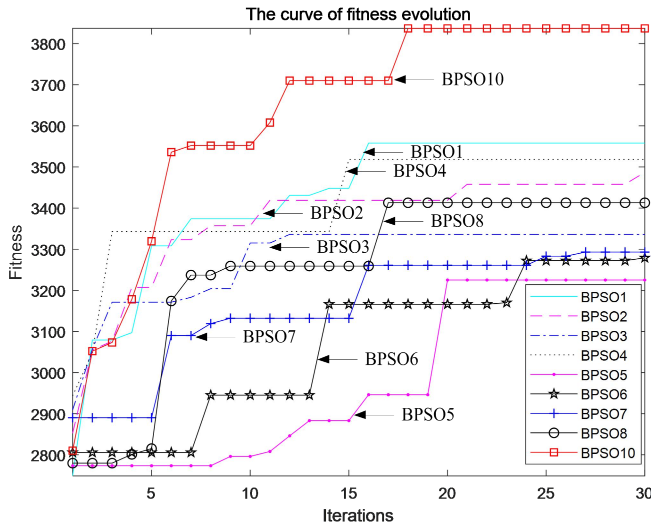


(g) Test 7

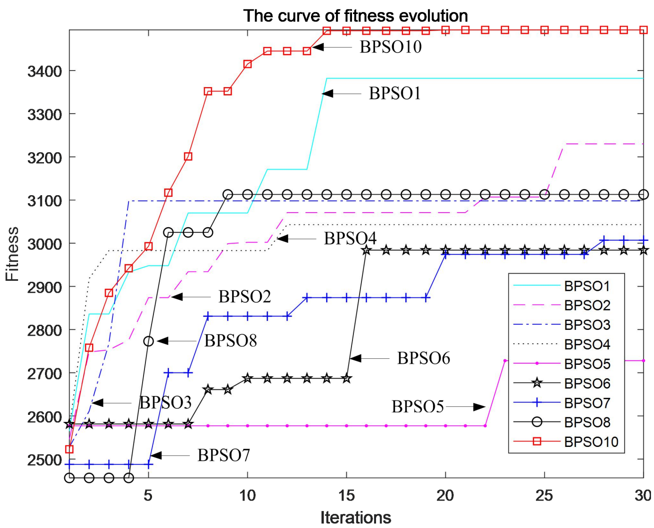


(j) Test 10

Fig. 3 0-1 knapsack problems solved by BPSO algorithms with different transfer functions.



(h) Test 8



(i) Test 9

It can be seen from Table 5 that for small-scale 0-1 KP problems, the test data Test 1 and Test 2, BPSO1-BPSO9 and BPSO10 all found the optimal value, but the Std data shows that the Std value of BPSO10 in the two sets of experiments is 0, that is to say in 10 experiments, BPSO10 can find the optimal value every time and show the strong robustness. For the test set Test 3-Test 6, BPSO10 finds the optimal value, and the Std value is the smallest, especially in Test 3 and Test 5 with zero. So the improved algorithm has better stability than other algorithms. For the medium-scale problems Test 7-Test 9, BPSO10 shows the best effect in the two sets of data of optimal value and variance, that is to say it has the highest convergence accuracy and stability. For large-scale problems, BPSO10 finds a better optimal value than other algorithms. However, the Std value of this algorithm is larger, which needs to be further improved in future learning research. It can be seen from Fig. 3 that for small-scale, medium-scale and large-scale 0-1 KP problems, in most cases, BPSO10 is better than other algorithms in terms of convergence speed, but the convergence speed is slow and needs further improvement.

VI. Conclusions

This paper proposes a series of improved binary particle swarm optimization algorithm based on Z-shaped transfer functions to solve the 0-1 knapsack problem. Aiming at the fact that the existing BPSO algorithm is prone to fall into the local optimal solution, a new Z-shaped probability transfer function is proposed to map the continuous searching space to a binary space. The effectiveness of the proposed strategy is verified by solving the typical 0-1 knapsack problems based on the proposed Z-shaped probabilistic transfer function, V-shaped and S-shaped transfer functions. Simulation experiment results show that the proposed probability transfer function improves the convergence speed and optimization accuracy of the BPSO algorithm.

REFERENCES

[1] G. B. Dantzig, "Discrete-Variable Extremum Problems," *Operations Research*, vol. 5, no. 2, pp. 266-277, 1957.

[2] G. Mavrotas, D. Diakoulaki, and A. Kourentzis, "Selection among Ranked Projects under Segmentation, Policy and Logical Constraints," *European Journal of Operational Research*, vol. 187, no. 1, pp. 177-192, 2008.

[3] L. Yu, S. Wang, F. Wen, and K. K. Lai, "Genetic Algorithm-based Multi-criteria Project Portfolio Selection," *Annals of Operations Research*, vol. 197, pp. 71-86, 2012.

[4] D. Zou, L. Gao, S. Li, and J. Wu, "Solving 0–1 Knapsack Problem by a Novel Global Harmony Search Algorithm," *Applied Soft Computing*, vol. 11, no. 2, pp. 1556-1564, 2011.

[5] X. Zhang, S. Huang, Y. Hu, Y. Zhang, S. Mahadevan, and Y. Deng, "Solving 0-1 Knapsack Problems Based on Amoeboid Organism Algorithm," *Applied Mathematics & Computation*, vol. 219, no. 19, pp. 9959-9970, 2013.

[6] Y. Feng, G. G. Wang, and X. Z. Gao, "A Novel Hybrid Cuckoo Search Algorithm with Global Harmony Search for 0-1 Knapsack Problems," *International Journal of Computational Intelligence Systems*, vol. 9, no. 6, pp. 1174-1190, 2016.

[7] T. K. Truong, K. Li, and Y. Xu, "Chemical Reaction Optimization with Greedy Strategy for the 0–1 Knapsack Problem," *Applied Soft Computing Journal*, vol. 13, no. 4, pp. 1774-1780, 2013.

[8] Y. Zhou, X. Chen, and G. Zhou, "An Improved Monkey Algorithm for a 0-1 Knapsack Problem," *Applied Soft Computing*, vol. 38, no. C, pp. 817-830, 2016.

[9] Y. Q. Zhou, L. L. Li, and M. Z. Ma, "A Complex-valued Encoding Bat Algorithm for Solving 0-1 Knapsack Problem," *Neural Processing Letters*, vol. 44, no. 2, pp. 407-430, 2016.

[10] Y. J. Gao, F. M. Zhang, Y. Zhao, and C. Li, "A Novel Quantum-inspired Binary Wolf Pack Algorithm for Difficult Knapsack Problem," *International Journal of Wireless and Mobile Computing*, vol. 16, no. 3, pp. 222-232, 2019.

[11] R. M. Rizk-Allah, and A. E. Hassanien, "New Binary Bat Algorithm for Solving 0–1 Knapsack Problem," *Complex & Intelligent Systems*, vol. 4, no. 1, pp. 31-53, 2018.

[12] Y. Huang, P. Wang, J. Li, X. Chen, and T. Li, "A Binary Multi-Scale Quantum Harmonic Oscillator Algorithm for 0-1 Knapsack Problem with Genetic Operator," *IEEE Access*, vol. 7, pp. 137251-137265, 2019.

[13] Y. Feng, G. G. Wang, S. Deb, M. Lu, and X. J. Zhao，"Solving 0–1 Knapsack Problem by a Novel Binary Monarch Butterfly Optimization," *Neural Computing and Applications*, vol. 28, no. 7, pp. 1619-1634, 2017.

[14] M. Clerc, and J. Kennedy, "The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58-73, 2002.

[15] Z. Beheshti, and S. M. H. Shamsuddin, "CAPSO: Centripetal Accelerated Particle Swarm Optimization," *Information Sciences*, vol. 258, no. 3, pp. 54-59, 2014.

[16] Y. X. Lu, J. S. Wang, and S. S. Guo, "Solving Path Planning Problem Based on Particle Swarm Optimization Algorithm with Improved Inertia Weights," *IAENG International Journal of Computer Science*, vol. 46, no. 4, pp. 628-636, 2019.

[17] S. Mirjalili, and A. Lewis, "S-shaped Versus V-shaped Transfer Functions for Binary Particle Swarm Optimization," *Swarm and Evolutionary Computation*, vol. 9, pp. 1-14, 2013.

[18] J. S. Wang, and J. D. Song, "A Hybrid Algorithm Based on Gravitational Search and Particle Swarm Optimization Algorithm to Solve Function Optimization Problems," *Engineering Letters*, vol. 25, no. 1, pp. 22-29, 2017.

[19] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "BGSA: Binary Gravitational Search Algorithm," *Natural Computing*, vol. 9, no. 3, pp. 727-745, 2010.

[20] S. S. Guo, J. S. Wang, and M. W. Guo, "Z-shaped Transfer Functions for Binary Particle Swarm Optimization Algorithm," *Computational Intelligence and Neuroscience*, vol. 2020, no. 19, pp. 1-21, 2020.

Wei-Zhong Sun received his B. Sc. degree in Electronic and Information Engineering and his M. Sc. degree in Control Science from University of Science and Technology Liaoning, China in 2004 and 2008, respectively. He is a lecturer of School of Computer Science and Software Engineering, University of Science and Technology Liaoning. His main research interest is image processing, pattern recognition and embedded system.

Min Zhang is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China. Her main research interest is swarm intelligent optimization algorithm.

Jie-Sheng Wang received his B. Sc. And M. Sc. degrees in Control Science from University of Science and Technology Liaoning, China in 1999 and 2002, respectively, and his Ph. D. degree in Control Science from Dalian University of Technology, China in 2006. He is a professor and Doctor's Supervisor in School of Electronic and Information Engineering, University of Science and Technology Liaoning. His main research interest is intelligent control and computer integrated manufacturing.

Sha-Sha Guo is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China. Her main research interest is swarm intelligent optimization algorithm.

Min Wang is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China. Her main research interest is swarm intelligent optimization algorithm.

Wen-Kuo Hao is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China. His main research interest is swarm intelligent optimization algorithm.