Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Improving Flower Pollination Algorithm for Solving 0–1 Knapsack Problem

Improving Flower Pollination Algorithm for Solving 0 –1 Knapsack Problem

Ghalya Tawfeeq Basheer[1]and Zakariya Yahya Algamal[2*]

[1]Department of Operations Research and Intelligent Techniques, University of Mosul, Mosul, Iraq
[2]Department of Statistics and Informatics, University of Mosul, Mosul, Iraq

E-mail: zakariya.algamal@uomosul.edu.iq

Abstract: Binary knapsack problem has received considerable attention in combinational optimization. Various meta-heuristic algorithms are dedicated to solve this problem in the literature. Recently, a binary flower pollination algorithm (BFPA) was proposed, which has been successfully applied to solve 0-1 knapsack problem. In this paper, two new time-varying transfer functions are proposed to improve the exploration and exploitation capability of the BFPA with the best solution and short computing time. Based on small, medium, and high-dimensional scales of the knapsack problem, the computational results reveal that the proposed time-varying transfer functions not only to find the best possible solutions but also to have less computational time. Compared to the standard transfer functions, the efficiency of the proposed time-varying transfer functions is superior, especially in the high-dimensional scales.

1. Introduction

The process of optimization is searching and finding the optimal solution of a given problem [1, 2]. In general, based on the nature of the search space and decision variables, an optimization problem can be divided into three main classes: continuous, discrete and mixed integer optimization problem [3, 4]. The binary optimization problems are a set classes of the discrete optimization problem in which the decision variable is a set of bits.

The knapsack problem is an optimization problem that can be modelled as a discrete binary optimization problem. The knapsack problem has widely studied in many real world applications, such as project selection, cutting stock problems, scheduling problems, resource allocation, and investment decision making [5-7].

There are several methods that have been developed to solve knapsack problem which can be divided into two types: exact methods, such as branch and bound method and dynamic programming. These methods can give the exact solution, but it is effective for small sized problems. The second type includes approximate methods that can give an approximate solution, but at reasonable times compared to exact methods [8-10].

In recent years, several meta-heuristic methods were proposed for tackling 0-1 knapsack problem, such as monarch butterfly optimization (MBO)[11], bat algorithm (BA)[12,13], particle swarm optimization (PSO)[14-16], monkey algorithm (MA)[17], ant colony optimization (ACO)[18], amoeboid organism algorithm (AOA)[19], and harmony search (HS)[20].

The flower pollination algorithm is a bio-inspired algorithm that mimics the pollination characteristics of flowers in plants. Flower pollination algorithm is first proposed by Yang [21] for solving single objective optimization problems. In (2014) Yang et al.[22] extended flower pollination algorithm for solving multi objective optimization problems. In (2015) Yang [23] proposed a binary flower pollination algorithm to tackle a feature selection problem. Abdel-Basset et al. [9] proposed a binary version of flower pollination algorithm for solving both small and large scale knapsack problem. Compared with some other algorithms, the flower pollination algorithm can perform better in terms of the global convergence and the convergence speed.

In the binary flower pollination algorithm, a transformation function is used to convert the continuous values generated from the algorithm into binary ones, and, therefore it is able to provide a binary flower pollination algorithm a sufficient amount to balance between exploration and exploitation [24].

In this paper, two efficient time-varying transfer functions are proposed to solve the 0 –1 knapsack problem. The proposed transfer functions are based on combining the S-shaped and V-shaped transfer functions with time-varying concept.

The remainder of this paper is organized as follows. Section 2 describes the basic 0 –1 knapsack problem. Section 3 introduces binary flower pollination algorithm. In Section 4, the proposed time-varying transfer functions are presented. Section 5 presents and discusses the experimental results. In section 6, conclusion is drawn.

2. Knapsack problem

Knapsack problem is a NP-hard combinatorial optimization problem and defined as follows [8, 25]:

Given a set of n items, each item i has a profit c_i and weight w_i. The objective is to select a subset of the items such that the total profit is maximized without exceeding the knapsack's capacity M. Mathematically, the knapsack problem can be formulated as follows:

$$Max\ f(x)=\sum_{i=1}^{n} c_i x_i \qquad (1)$$

s.t.

$$\sum_{i=1}^{n} w_i x_i \le M \qquad (2)$$

where

$$x_i=\begin{cases}1 & \text{if item i is selecled}\\ 0 & \text{otherwise}\end{cases}$$

To solve knapsack problem, we select a subset of the items of the binary vector x_i such that the optimal solution satisfies the constraint in Eq. (2) and maximizes the objective function in Eq.(1).

In such a constrained optimization problem, the penalty function is used to handle the constrained knapsack problem. As known, the knapsack problem is a maximization problem that can be converted into minimization by multiplying Eq. (1) by -1. As a result, the penalty function can be written as follow:

$$Min\phi(x)=-f(x)+\lambda Max(0,h) \qquad (3)$$

Where $h=\sum_{i=1}^{n} w_i x_i - M$ and λ represents the penalty coefficient. In this paper λ set to 10^{10} for all tests. The penalty function can be described in Figure1.

Penalty function

- **input** solution x_i
- Calculate total weight of x_i by $\left(\sum_{i=1}^{n} w_i x_i\right)$
- **if** $\left(\sum_{i=1}^{n} w_i x_i \le M\right)$
- $\quad \phi(x) = -\sum_{i=1}^{n} c_i x_i$
- **else**
- $\quad \phi(x) = -\sum_{i=1}^{n} c_i x_i + \lambda\left(\sum_{i=1}^{n} w_i x_i - M\right)$
- **end**

Figure 1: Penalty function

A Repair operator is treated the infeasible solutions which violates the constraint in Eq. (2) by converting them into feasible solutions and also improve the feasible solutions. The repair operator algorithm can be applied in two stages. The first stage is to convert the infeasible solution into feasible by taking out the items of the lower c_i / w_i ratio so as the constraint in Eq. (2) is not to exceed the knapsack capacity. The second stage is to improve the feasible solution by adding the items of the high c_i / w_i ratio to the knapsack with the keeping of the constraint.

3. Binary flower pollination algorithm(BFPA)

Yang (2012) proposed a new algorithm for global optimization called flower pollination algorithm [21]. It is a meta-heuristic algorithm that mimics nature, inspired of the pollination process in flowers.

Pollination in flowers can be taken two forms: biotic pollination and abiotic pollination. In the first type, the pollen is transferred by pollination like insects and animals. While the second form based on wind and diffusion in the water.

Pollination can be divided into self-pollination and cross-pollination. Self-pollination is transferring the pollens from one flower to the same flower or different flowers in the same plant. Cross-pollination is transferring the pollens from one flower to another flower of a different plant. A flower and its pollen represented a solution to the optimization problem. In the flower pollination algorithm, four basic rules are used[21, 26, 27]:

1. The global pollination includes biotic and cross-pollination, the pollinators move in a way that follows a lévy flight distribution.
2. The local pollination includes abiotic and self-pollination.
3. Flower constancy can be considered as the reproduction probability that is proportional to the similarity of two flowers involved.
4. We use a switch probability $p \in [0,1]$ to switch between global pollination and local pollination.

Rules 1 and 3 can be expressed mathematically as:

$$x_i^{t+1} = x_i^t + \gamma L(\lambda)\left(x_i^t - g^*\right) \qquad (4)$$

where x_i^t is the solution vector or the pollen i at iteration t , g^* is the current best solution that is found at the current iteration, γ is a scaling factor to control the step size, $L(\lambda)$ is the step size in the lévy flights which is representing the strength of the pollination. Since pollinators move over a long distance with various distance steps, a lévy flight can be used to mimic this behaviour. That is, $L > 0$ from a lévy distribution as

$$L \sim \frac{\lambda\Gamma(\lambda)\sin\left(\frac{\pi\lambda}{2}\right)}{\pi}\left(\frac{1}{S^{1+\lambda}}\right)(S >> S_0 > 0) \tag{5}$$

Yang (2012) proposed $\Gamma(\lambda)$, the standard gamma function, and $\lambda = 1.5$. This distribution is valid for large steps $S > 0$. In (1994), Mantegna used the Gaussian distribution for generating the step size S by generating two random numbers U and V as follows [26]:

$$S = \frac{U}{|V|^{1/\lambda}} \quad U \sim N\left(0,\sigma^2\right), \ V \sim N\left(0,1\right) \tag{6}$$

$$\sigma^2 = \left(\frac{\Gamma(1+\lambda)}{\lambda\Gamma\left[(1+\lambda)/2\right]} * \frac{\sin(\pi\lambda/2)}{2^{\frac{\lambda-1}{2}}}\right)^{1/\lambda} \tag{7}$$

For local pollination, rules 2 and 3 can be expressed as:

$$x_i^{t+1} = x_i^t + k\left(x_j^t - x_k^t\right) \tag{8}$$

where x_j and x_k are the pollens (solution vectors) from different flowers of the same plant. k is the parameter draws from a uniform distribution in $[0,1]$. To switch between common global pollination to intensive local pollination, we used rule 4. In 2012,Yang suggested that switch probability or proximity probability $p = 0.8$ for most applications. The flower pollination algorithm can be presented in Figure 2.

```
Flower Pollination Algorithm
Begin
   -   Define the objective function max or min f (x) , and switch probability p ∈ [0,1]
   -   Initialize the population of n random flowers
   -   Evaluate each flowers in the population
   -   Find the best solution
   -   while (stopping criterion)
   -     for i=1:n
   -       if (r<p)
   -            New solution =global pollination Eq. (4)
   -       else
   -            New solution = local pollination Eq. (8)
   -       end if
   -       Evaluate new solution
   -       If new solution is better , update the population
   -     end for
   -     Find current best solution
   -   end while
```

Figure 2: Flower pollination algorithm

4. The proposed time-varying transfer functions

The knapsack problem can be modeled as a discrete problem in which the solution vector is binary, where 1 corresponds to that an item will be selected in the knapsack and 0 otherwise. In any binary algorithm, where one uses the step vector to calculate the probability of changing positions, the transfer functions significantly impact the balance between exploration and exploitation[24, 28].

In BFPA, the transfer function is used to map a continuous search space to a binary one, and the updating process is designed to switch positions of pollens between 0 and 1 in binary search spaces. In order to build this binary vector, a transfer function in Eq. (9) can be used after Eq. (8), in which the new solution is constrained to only binary values:

$$x_i^t = \begin{cases} 1 & if \quad T(x) > r \\ 0 & \quad o.w \end{cases} \tag{9}$$

where $r \in [0,1]$ is a random number, $T(x)$ is the transfer function.

Mirjalili and Lewis [29] introduced eight transfer functions and divided them into two families: S-Shaped transfer functions and V-Shaped transfer functions. These transfer functions are listed in Table 1. The transfer function was tested by Mirjalili and Lewis on 25 benchmark functions. The results show that V-shaped transfer functions are useful for the binary particle swarm optimization, especially V_4 function has merit for solving these functions. Teng et al.[30] has demonstrated the effect of V-shaped transfer function using binary particle swarm optimization on the feature selection problem. The experimental tests reveal that the efficiency of the proposed method.

In optimization algorithm, it is expected that the focus of the early stages of the implementation the algorithm will be on exploration to avoid falling into the local point, but in later stages of implementing the algorithm focuses more on exploitation to improve the quality of the solution[24, 28].

In this paper, two dynamic transfer functions are adapted form Mafarja, Aljarah [28] and Islam, Li [24], and proposed to improve the BFPA with the following considerations [24]:

1. In the early stages of the implementation, the transfer function should provide a high probability of flipping all the bits of x_i so that the BFPA can provide a stronger exploration.
2. In the intermediate stages of the implementation, the BFPA should start shifting from exploration to exploitation. This can achieved by using a transfer function that can reduce the probability of flipping all the bits of x_i .
3. In the final stages of the implementation, the transfer function should provide a low probability of flipping all the bits of x_i , so that the BFPA can provide a stronger exploitation capability.

In our proposed time-varying transfer function (TV), a new control parameter τ is added in the original transfer function. This τ is a time varying variable which starts with large value and gradually decreases over time. Two types of τ are proposed as follow:

$$\tau_1 = \left(1 - \frac{t}{T}\right)\tau_{1,\max} + \frac{t}{T}\tau_{1,\min} \tag{10}$$

and

$$\tau_2 = \tau_{2,\max} - t\left(\frac{\tau_{2,\max} - \tau_{2,\min}}{T}\right) \tag{11}$$

where $\tau_{\max}$ and $\tau_{\min}$ are the minimum and maximum values of the control parameter τ , and T is the maximum iteration of the BFPA. Table 2 lists the two proposed time-varying transfer functions.

Table 1: Families of transfer functions

	S-Shaped family		V-Shaped family
S1	$T(x)=\frac{1}{1+e^{-2x}}$	**V1**	$T(x)=\left\|erf\left(\frac{\sqrt{\pi}}{2}x\right)\right\|$
S2	$T(x)=\frac{1}{1+e^{-x}}$	**V2**	$T(x)=\left\|\tanh(x)\right\|$
S3	$T(x)=\frac{1}{1+e^{-\frac{x}{2}}}$	**V3**	$T(x)=\left\|\frac{x}{\sqrt{1+x^2}}\right\|$
S4	$T(x)=\frac{1}{1+e^{-\frac{x}{3}}}$	**V4**	$T(x)=\left\|\frac{2}{\pi}\arctan\left(\frac{\pi}{2}x\right)\right\|$

Table 2: The two proposed time-varying transfer functions.

	Time-varying 1		Time-varying 2
T1S1	$TVS(x)=\frac{1}{1+e^{\frac{-2x}{\tau_1}}}$	**T2S1**	$TVS(x)=\frac{1}{1+e^{\frac{-2x}{\tau_2}}}$
T1S2	$TVS(x)=\frac{1}{1+e^{\frac{-x}{\tau_1}}}$	**T2S2**	$TVS(x)=\frac{1}{1+e^{\frac{-x}{\tau_2}}}$
T1S3	$TVS(x)=\frac{1}{1+e^{-\frac{x}{2\tau_1}}}$	**T2S3**	$TVS(x)=\frac{1}{1+e^{-\frac{x}{2\tau_2}}}$
T1S4	$TVS(x)=\frac{1}{1+e^{-\frac{x}{3\tau_1}}}$	**T2S4**	$TVS(x)=\frac{1}{1+e^{-\frac{x}{3\tau_2}}}$

Table 3: The two proposed time-varying transfer functions.

T1V1	$TVV(x)=\left\|erf\left(\frac{\sqrt{\pi}x}{2\tau_1}\right)\right\|$	**T2V1**	$TVV(x)=\left\|erf\left(\frac{\sqrt{\pi}x}{2\tau_2}\right)\right\|$
T1V2	$TVV(x)=\left\|\tanh\left(\frac{x}{\tau_1}\right)\right\|$	**T2V2**	$TVV(x)=\left\|\tanh\left(\frac{x}{\tau_2}\right)\right\|$
T1V3	$TVV(x)=\left\|\frac{x/\tau_1}{\sqrt{1+x^2/\tau_1}}\right\|$	**T2V3**	$TVV(x)=\left\|\frac{x/\tau_2}{\sqrt{1+x^2/\tau_2}}\right\|$
T1V4	$TVV(x)=\left\|\frac{2}{\pi}\arctan\left(\frac{\pi x}{2\tau_1}\right)\right\|$	**T2V4**	$TVV(x)=\left\|\frac{2}{\pi}\arctan\left(\frac{\pi x}{2\tau_2}\right)\right\|$

5. Computational results

5.1. Parameter setting

For the binary flower pollination algorithm, the parameters were setting as follows: the population size =50, $\lambda = 1.5$, $p = 0.8$. In this paper, we use linear decreasing time varying where $\tau_{\max} = 4$, $\tau_{\min} = 0.1$, and T represents the maximum number of iterations.

5.2 Comparison results

To verify the feasibility and effectiveness of the proposed time-varying transfer functions method for solving 0–1 Knapsack problem, three scales of the knapsack problem are considered: low, medium, and high-dimensional scales. In this paper, all the results are obtained from 50 independent trials. The best solution, the worst solution, the mean and the standard deviation (SD) values, Mean iterations are reported as evaluation criteria. All of the computational experiments were conducted in Matlab 13a on a PC with an Intel Pentium Core i7-7500 processor (2.9 GHz) with 16GB of RAM in the Windows 10 OS.

5.2.1 Low scale 0-1 KP

The performance of the improved algorithm is investigated to solve ten low scale 0-1 KP instances (kp-1 to kp-10), which are taken from [9, 13]. The dimensions in this case are ranging from 4 to 23. The information dimension, capacity, weights and profits for these ten instances are described in Table S1 (Appendix). Table 4 shows the comparison results for all the used different transfer functions for the kp1 and kp10. The rest instances are listed in Table S2 (Appendix).

As observed from the results in Table 4 and Table S2, for the low scale knapsack problems, there is no difference among the results of using the proposed time-varying transfer functions and the standard transfer functions in terms of the best, worse, mean, and SD values. The major difference among the performance of the proposed time-varying transfer functions and the standard transfer functions in not expected because of relatively small numbered items. Contrary, the proposed time-varying transfer functions give optimal results with less number of iterations. The mean iterations of the proposed time-varying transfer functions are obviously better than the standard transfer functions for kp8, kp9, and kp10 where the number of items is higher than the others. Moreover, comparing between the two proposed transfer function, the required iterations to get the optimal solution using Eq. (11) is less than of Eq. (10) for kp8, kp9, and kp10.

Table 4: Results obtained by the transfer functions for the low scale 0–1 KP

Instance	Transfer function	Best	Mean	Worst	SD	Mean iterations
kp-1	S1	35	35	35	0	1
	S2	35	35	35	0	1
	S3	35	35	35	0	1
	S4	35	35	35	0	1
	V1	35	35	35	0	1
	V2	35	35	35	0	1
	V3	35	35	35	0	1
	V4	35	35	35	0	1
	T1S1	35	35	35	0	1
	T1S2	35	35	35	0	1
	T1S3	35	35	35	0	1
	T1S4	35	35	35	0	1

	T1V1	35	35	35	0	1
	T1V2	35	35	35	0	1
	T1V3	35	35	35	0	1
	T1V4	35	35	35	0	1
	T2S1	35	35	35	0	1
	T2S2	35	35	35	0	1
	T2S3	35	35	35	0	1
	T2S4	35	35	35	0	1
	T2V1	35	35	35	0	1
	T2V2	35	35	35	0	1
	T2V3	35	35	35	0	1
	T2V4	35	35	35	0	1
kp-10	S1	9767	9767	9767	0	2.28
	S2	9767	9767	9767	0	4.44
	S3	9767	9767	9767	0	9.8
	S4	9767	9767	9767	0	15.64
	V1	9767	9767	9767	0	1.2
	V2	9767	9767	9767	0	1.2
	V3	9767	9767	9767	0	1.24
	V4	9767	9767	9767	0	1.12
	T1S1	9767	9767	9767	0	4.39
	T1S2	9767	9767	9767	0	1.95
	T1S3	9767	9767	9767	0	7.19
	T1S4	9767	9767	9767	0	15.19
	T1V1	9767	9767	9767	0	1.12
	T1V2	9767	9767	9767	0	1.08
	T1V3	9767	9767	9767	0	1.15
	T1V4	9767	9767	9767	0	1
	T2S1	9767	9767	9767	0	2.95
	T2S2	9767	9767	9767	0	2.17
	T2S3	9767	9767	9767	0	8.42
	T2S4	9767	9767	9767	0	12.76
	T2V1	9767	9767	9767	0	1.15
	T2V2	9767	9767	9767	0	1
	T2V3	9767	9767	9767	0	1.04
	T2V4	9767	9767	9767	0	1

5.2.2 Medium scale 0-1 KP

To further evaluate the performance of proposed time-varying transfer functions in medium scale 0-1 Knapsack problem, ten medium size 0-1 KP instances (kp-11 to kp-20) are taken from [9, 13] in which the items are between 30 and 75. The description of these ten instances is described in Table S3 (Appendix). Table 5 summarizes the comparison results for all the used different transfer functions for the kp15 and kp20. The rest instances results are reported in Table S4 (Appendix).

Obviously, it is evident from Table 5 and Table S4 that the proposed time-varying transfer functions obtained the same best, worse, mean, and SD values as the standard transfer functions. From Tables 5 and S4, for the mean iterations, the proposed time-varying transfer functions are superior to the other eight standard transfer functions on kp11 to kp20. This indicates that the proposed time-varying transfer functions is comparatively fast. For example, in kp20, the reduction in mean iteration of T1S2 and T2S2 functions was 37.61% and 49.81% lower than that of S2, respectively. On the other hand, the reduction in mean iteration of T1V4 and T2V4 functions was 40.08% and 60.32% lower than that of V4, respectively.

Further, it was noted that the v-shaped transfer functions are usually yielded the least iterations compared to S-shaped transfer functions. On the other hand, comparing between the two proposed transfer functions, the required iterations to get an optimal solution using Eq. (11) is less than of Eq. (10) for all the 0-1 Knapsack problems. Additionally, the number of iterations of T2V1, T2V2, T2V3, and T2V4 are obviously small than T1V1, T1V2, T1V3, and T1V4.

Table 5: Results obtained by the transfer functions for the medium scale 0–1 KP

Instance	Transfer function	Best	Mean	Worst	SD	Mean iterations
kp-15	S1	2440	2440	2440	0	9.72
	S2	2440	2440	2440	0	29.4
	S3	2440	2440	2440	0	21.64
	S4	2440	2440	2440	0	15.6
	V1	2440	2440	2440	0	5.56
	V2	2440	2440	2440	0	3.4
	V3	2440	2440	2440	0	2.44
	V4	2440	2440	2440	0	1.8
	T1S1	2440	2440	2440	0	19.84
	T1S2	2440	2440	2440	0	17.04
	T1S3	2440	2440	2440	0	12.12
	T1S4	2440	2440	2440	0	8.48
	T1V1	2440	2440	2440	0	1.08
	T1V2	2440	2440	2440	0	1.16
	T1V3	2440	2440	2440	0	1.16
	T1V4	2440	2440	2440	0	1.12
	T2S1	2440	2440	2440	0	25.16
	T2S2	2440	2440	2440	0	16.8
	T2S3	2440	2440	2440	0	15.56
	T2S4	2440	2440	2440	0	9.92
	T2V1	2440	2440	2440	0	1.04
	T2V2	2440	2440	2440	0	1.04
	T2V3	2440	2440	2440	0	1.04
	T2V4	2440	2440	2440	0	1
kp-20	S1	3614	3614	3614	0	48.53
	S2	3614	3614	3614	0	125.41
	S3	3614	3614	3614	0	115.36
	S4	3614	3614	3614	0	137.4
	V1	3614	3614	3614	0	18.54
	V2	3614	3614	3614	0	8.46
	V3	3614	3614	3614	0	6.82
	V4	3614	3614	3614	0	4.89
	T1S1	3614	3614	3614	0	50.18
	T1S2	3614	3614	3614	0	78.24
	T1S3	3614	3614	3614	0	82.13
	T1S4	3614	3614	3614	0	92.74
	T1V1	3614	3614	3614	0	10.21

T1V2	3614	3614	3614	0	4.18
T1V3	3614	3614	3614	0	4.53
T1V4	3614	3614	3614	0	2.93
T2S1	3614	3614	3614	0	46.65
T2S2	3614	3614	3614	0	62.94
T2S3	3614	3614	3614	0	91.6
T2S4	3614	3614	3614	0	69.34
T2V1	3614	3614	3614	0	7.26
T2V2	3614	3614	3614	0	3.62
T2V3	3614	3614	3614	0	3.98
T2V4	3614	3614	3614	0	1.94

5.2.3 High-dimensional scale 0-1 KP

To further highlight the benefits of our proposed time-varying transfer functions, three cases have been investigated. The first case handles the uncorrelated problem (kp21 – kp25) where the weights w_i are uncorrelated with the profits c_i. Each w_i and c_i is randomly chosen from 5 to 20 and from 5 to 40, respectively. The second case handles the weakly correlated problem (kp26 – kp30). In this case, the weights w_i and the profits c_i can be expressed as follows: $w_i \in [5, 20]$ and $c_i \in [w_i - 5, w_i + 5]$. The third case handles the strongly correlated problem (kp31 – kp35). In this case, w_i and c_i can be calculated as: $w_i \in [5, 20]$ and $c_i \in [w_i + 5]$. The knapsack capacity for the kp-21-kp35 can be calculated as $M = 0.75 \times \sum_{i=1}^{n} w_i$. The dimension sizes varying from 100 to 2000 items. For all used transfer functions, the maximum iteration is set to 10000. Tables 6 – 8 reports the comparison results for all the used different transfer functions for the kp22, kp25, kp27, kp30, kp32, and kp35. The rest problems are listed in Tables S5 – S7 (Appendix). Based on the obtained results, several points are concluded.

(1) It can be seen that the proposed time-varying transfer functions significantly outperform the standard transfer functions on all evaluation measures including the best, mean, worst, and standard deviations.
(2) As observed from the results, the proposed time-varying V-shaped transfer functions, T1V1 – T2V4, can easily find the optimal values with small SD in all uncorrelated, weakly correlated, and strongly correlated problems.
(3) It is obvious that there is an improvement for searching the global optimal solution when using T2V1, T2V2, T2V3, and T2V4 compared to T1V1, T1V2, T1V3, and T1V4. This leads to the performance dominance of T2V1, T2V2, T2V3, and T2V4 against those performed on the T1V1, T1V2, T1V3, and T1V4.
(4) The mean iteration values of time-varying V-shaped transfer functions, T1V1 – T2V4, are obviously superior to S1, S2, S3, S4, V1, V2, V3, and V4 for all high-dimensional scale problems. The performance of T2V1, T2V2, T2V3, and T2V4 is better than that of T1V1, T1V2, T1V3, and T1V4.
(5) Compared to the proposed time-varying V-shaped transfer functions, T2V4 is significantly improving the performance metrics with lower SD and mean iterations.

Table 6: Comparison results of uncorrelated high-dimensional scale 0–1 KP

Instance	Dimension	Transfer function	Best	Mean	Worst	SD	Mean iterations

kp-22	500	S1	10340	10338	10335	2.739	1038
		S2	10338	10302.2	10225	25.46	2394
		S3	10338	10306.6	10225	26.923	2187
		S4	10338	10311	10225	20.01	2541
		V1	10345	10345	10345	0	185
		V2	10345	10345	10345	0	103
		V3	10345	10345	10345	0	161
		V4	10345	10345	10345	0	94
		T1S1	10345	10345	10345	0	1074
		T1S2	10343	10333.4	10319	13.145	1851
		T1S3	10343	10328.6	10319	13.145	2018
		T1S4	10343	10323.8	10319	10.733	1932
		T1V1	10345	10345	10345	0	110
		T1V2	10345	10345	10345	0	95
		T1V3	10345	10345	10345	0	104
		T1V4	10345	10345	10345	0	73
		T2S1	10345	10345	10345	0	1022
		T2S2	10343	10325	10319	10.392	1915
		T2S3	10343	10327.2	10319	9.96	1893
		T2S4	10343	10328.4	10319	9.044	1801
		T2V1	10345	10345	10345	0	100
		T2V2	10345	10345	10345	0	94
		T2V3	10345	10345	10345	0	99
		T2V4	10345	10345	10345	0	69
kp-25	2000	S1	40612	40598.2	40589	12.674	6285
		S2	40610	40554.8	40525	33.945	8136
		S3	40610	40549	40525	37.259	7985
		S4	40610	40544.9	40525	38.157	7612
		V1	40615	40611.8	40610	15.375	3485
		V2	40615	40612.4	40610	14.281	2952
		V3	40615	40611.2	40610	16.158	3624
		V4	40615	40612.8	40610	14.825	2531
		T1S1	40616	40610.6	40595	9.521	4258
		T1S2	40614	40599.5	40590	20.98	6395
		T1S3	40614	40611.8	40590	19.353	5927
		T1S4	40614	40610.2	40590	21.97	6042
		T1V1	40616	40614.6	40611	8.34	2493
		T1V2	40616	40615	40611	7.921	2051
		T1V3	40616	40614.2	40611	8.95	2964
		T1V4	40616	40615.2	40611	6.56	1950
		T2S1	40616	40611.2	40595	8.536	4381
		T2S2	40614	40610.6	40590	19.561	6134
		T2S3	40614	40611.4	40590	17.562	5630
		T2S4	40614	40610.8	40590	20.315	6729
		T2V1	40616	40614	40611	7.65	2654
		T2V2	40616	40614.8	40611	6.98	1983
		T2V3	40616	40614.6	40611	6.551	2452
		T2V4	40616	40615.4	40611	5.439	1875

Table 7: Comparison results of weakly correlated high-dimensional scale 0–1 KP

Journal of Physics: Conference Series **1879** (2021) 022097 doi:10.1088/1742-6596/1879/2/022097

Instance	Dimension	Transfer function	Best	Mean	Worst	SD	Mean iterations
kp-27	500	S1	5197	5185.2	5175	8.624	1002
		S2	5190	5186.2	5184	9.943	1980
		S3	5190	5185.4	5184	8.67	1542
		S4	5190	5185.6	5184	8.24	1834
		V1	5197	5193.4	5190	4.96	162
		V2	5197	5193	5190	5.63	90
		V3	5197	5194.2	5190	5.176	166
		V4	5197	5194.6	5190	4.641	75
		T1S1	5197	5197	5197	0	995
		T1S2	5197	5194.8	5194	2.67	1091
		T1S3	5197	5195.4	5194	1.954	1124
		T1S4	5197	5194.2	5194	1.37	1387
		T1V1	5197	5197	5197	0	123
		T1V2	5197	5197	5197	0	84
		T1V3	5197	5197	5197	0	98
		T1V4	5197	5197	5197	0	61
		T2S1	5197	5197	5197	0	981
		T2S2	5197	5195	5194	1.095	1192
		T2S3	5197	5194.8	5194	2.04	1184
		T2S4	5197	5194.4	5194	1.93	1207
		T2V1	5197	5197	5197	0	111
		T2V2	5197	5197	5197	0	88
		T2V3	5197	5197	5197	0	93
		T2V4	5197	5197	5197	0	56
kp-30	2000	S1	21044	21036.5	21018	20.652	6018
		S2	21025	21005.2	20890	44.67	9820
		S3	21025	21004.8	20890	46.391	9453
		S4	21025	21005.1	20890	45.752	9572
		V1	21068	21057	21033	17.297	3091
		V2	21068	21057.4	21033	18.632	2735
		V3	21068	21056.8	21033	15.348	3120
		V4	21068	21057.8	21033	14.982	2493
		T1S1	21081	21079.2	21070	12.358	5304
		T1S2	21078	21074	21062	27.291	7659
		T1S3	21078	21073	21062	28.9	7362
		T1S4	21078	21074.6	21062	25.367	7710
		T1V1	21081	21079	21075	9.452	2514
		T1V2	21081	21079.2	21075	8.651	2183
		T1V3	21081	21078	21075	10.12	2907
		T1V4	21081	21079.2	21075	8.654	2061
		T2S1	21081	21079	21070	11.92	5297
		T2S2	21078	21074.8	21062	29.381	7640
		T2S3	21078	21075	21062	25.648	7193
		T2S4	21078	21074	21062	28.31	7684
		T2V1	21081	21078	21075	11.51	2761
		T2V2	21081	21079	21075	8.372	2031
		T2V3	21081	21078.6	21075	9.62	2897
		T2V4	21081	21079.4	21075	8.05	2005

Table 8: Comparison results of strongly correlated high-dimensional scale 0–1 KP

Instance	Dimension	Transfer function	Best	Mean	Worst	SD	Mean iterations
kp-32	500	S1	6783	6783	6783	0	978
		S2	6779	6775.6	6768	5.681	1862
		S3	6779	6776	6768	6.16	1734
		S4	6779	6775	6768	5.935	1815
		V1	6783	6781.5	6779	3.94	138
		V2	6783	6782	6779	4.67	105
		V3	6783	6781.9	6779	4.914	133
		V4	6783	6782.4	6779	3.952	100
		T1S1	6783	6783	6783	0	1086
		T1S2	6783	6780	6776	3.68	1273
		T1S3	6783	6780.2	6776	2.942	1360
		T1S4	6783	6780.9	6776	4.37	1109
		T1V1	6783	6783	6783	0	97
		T1V2	6783	6783	6783	0	57
		T1V3	6783	6783	6783	0	96
		T1V4	6783	6783	6783	0	50
		T2S1	6783	6783	6783	0	1050
		T2S2	6783	6781.2	6776	4.518	1241
		T2S3	6783	6780.8	6776	5.63	1293
		T2S4	6783	6782	6776	4.09	1113
		T2V1	6783	6783	6783	0	101
		T2V2	6783	6783	6783	0	55
		T2V3	6783	6783	6783	0	90
		T2V4	6783	6783	6783	0	51
kp-35	2000	S1	27285	27279.6	27246	15.162	5506
		S2	27292	27276	27230	30.37	6931
		S3	27292	27277.2	27230	35.61	6654
		S4	27292	27276.1	27230	29.49	6837
		V1	27356	27349	27344	10.34	2741
		V2	27356	27350	27344	12.57	2652
		V3	27356	27349.8	27344	11.31	2903
		V4	27356	27351	27344	11.94	2511
		T1S1	27362	27359	27350	9.58	6347
		T1S2	27362	27358	27349	16.54	5028
		T1S3	27362	27357	27349	18.36	5391
		T1S4	27362	27357.6	27349	15.29	5727
		T1V1	27362	27360	27358	3.67	1890
		T1V2	27362	27361	27358	2.94	1682
		T1V3	27362	27360.2	27358	3.05	1954
		T1V4	27362	27361.6	27358	3.58	1509
		T2S1	27362	27358.2	27350	10.17	3583
		T2S2	27362	27357.6	27349	14.89	5126
		T2S3	27362	27358	27349	16.52	6084
		T2S4	27362	27357	27349	17.13	5737
		T2V1	27362	27361	27358	2.34	1903
		T2V2	27362	27360.9	27358	3.61	1530
		T2V3	27362	27361.4	27358	2.57	1827

	T2V4	27362	27361.8	27358	2.09	1493

6. Conclusion
In this paper, two time-varying transfer functions are proposed to improve the exploration and exploitation capability of the binary flower pollination algorithm in solving the 0–1 KP problem efficiently. The experimental results show that the introduction of time-varying parameter in the transfer function can improve the performance of BFPA in solving small, medium, and high-dimensional scales 0–1 KP problems. Additionally, the experimental results show that proposed time-varying V-shaped transfer functions outperform the other S-shaped transfer functions in terms of the best, worse, mean, SD values, and the mean iterations.

Appendix

Table S1: The description of the low scale 0-1 KP instances

Instance	dimension	capacity M	weights W	profits C
kp-1	4	20	w=[6 5 9 7]	c=[9 11 13 15]
kp-2	4	11	w=[2 4 6 7]	c=[6 10 12 13]
kp-3	5	80	w=[15 20 17 8 31]	c=[33 24 36 37 12]
kp-4	7	50	w=[31 10 20 19 4 3 6]	c=[70 20 39 37 7 5 10]
kp-5	10	269	w=[95 4 60 32 23 72 80 62 65 46]	c=[55 10 47 5 4 50 8 61 85 87]
kp-6	10	60	w=[30 25 20 18 17 11 5 2 1 1]	c=[20 18 17 15 15 10 5 3 1 1]
kp-7	15	375	w=[56.358531 80.874050 47.987304 89.596240 74.660482 85.894345 51.353496 1.498459 36.445204 16.589862 44.569231 0.466933 37.788018 57.118442 60.716575]	c=[0.125126 19.330424 58.500931 35.029145 82.284005 17.410810 71.050142 30.399487 9.140294 14.731285 98.852504 11.908322 0.891140 53.166295 60.176397]
kp-8	20	879	w=[84 83 43 4 44 6 82 92 25 83 56 18 58 14 48 70 96 32 68 92]	c=[91 72 90 46 55 8 35 75 61 15 77 40 63 75 29 75 17 78 40 44]
kp-9	20	878	w=[92 4 43 83 84 68 92 82 6 44 32 18 56 83 25 96 70 48 14 58]	c=[44 46 90 72 91 40 75 35 8 54 78 40 77 15 61 17 75]
kp-10	23	10000	w=[983 982 981 980 979 978 488 976 972 486 486 972 972 485 485 969 966 483 964 963 961 958 959]	c=[981 980 979 978 977 976 487 974 970 485 485 970 970 484 484 976 974 482 962 961 959 958 857]

Table S2: Results obtained by the transfer functions for the low scale 0–1 KP

Instance	Transfer function	Best	Mean	Worst	SD	Mean iterations
kp-2	S1	23	23	23	0	1
	S2	23	23	23	0	1
	S3	23	23	23	0	1
	S4	23	23	23	0	1
	V1	23	23	23	0	1
	V2	23	23	23	0	1

	V3	23	23	23	0	1
	V4	23	23	23	0	1
	T1S1	23	23	23	0	1
	T1S2	23	23	23	0	1
	T1S3	23	23	23	0	1
	T1S4	23	23	23	0	1
	T1V1	23	23	23	0	1
	T1V2	23	23	23	0	1
	T1V3	23	23	23	0	1
	T1V4	23	23	23	0	1
	T2S1	23	23	23	0	1
	T2S2	23	23	23	0	1
	T2S3	23	23	23	0	1
	T2S4	23	23	23	0	1
	T2V1	23	23	23	0	1
	T2V2	23	23	23	0	1
	T2V3	23	23	23	0	1
	T2V4	23	23	23	0	1
	S1	130	130	130	0	1
	S2	130	130	130	0	1
	S3	130	130	130	0	1
	S4	130	130	130	0	1
	V1	130	130	130	0	1
	V2	130	130	130	0	1
	V3	130	130	130	0	1
	V4	130	130	130	0	1
	T1S1	130	130	130	0	1
	T1S2	130	130	130	0	1
	T1S3	130	130	130	0	1
kp-3	T1S4	130	130	130	0	1
	T1V1	130	130	130	0	1
	T1V2	130	130	130	0	1
	T1V3	130	130	130	0	1
	T1V4	130	130	130	0	1
	T2S1	130	130	130	0	1
	T2S2	130	130	130	0	1
	T2S3	130	130	130	0	1
	T2S4	130	130	130	0	1
	T2V1	130	130	130	0	1
	T2V2	130	130	130	0	1
	T2V3	130	130	130	0	1
	T2V4	130	130	130	0	1
	S1	107	107	107	0	1
	S2	107	107	107	0	1.08
	S3	107	107	107	0	1.04
kp-4	S4	107	107	107	0	1.04
	V1	107	107	107	0	1
	V2	107	107	107	0	1
	V3	107	107	107	0	1
	V4	107	107	107	0	1

	T1S1	107	107	107	0	1
	T1S2	107	107	107	0	1
	T1S3	107	107	107	0	1
	T1S4	107	107	107	0	1
	T1V1	107	107	107	0	1
	T1V2	107	107	107	0	1
	T1V3	107	107	107	0	1
	T1V4	107	107	107	0	1
	T2S1	107	107	107	0	1
	T2S2	107	107	107	0	1
	T2S3	107	107	107	0	1
	T2S4	107	107	107	0	1
	T2V1	107	107	107	0	1
	T2V2	107	107	107	0	1
	T2V3	107	107	107	0	1
	T2V4	107	107	107	0	1
kp-5	S1	295	295	295	0	1.12
	S2	295	295	295	0	1.92
	S3	295	295	295	0	1.32
	S4	295	295	295	0	1.16
	V1	295	295	295	0	1.16
	V2	295	295	295	0	1.04
	V3	295	295	295	0	1.04
	V4	295	295	295	0	1.04
	T1S1	295	295	295	0	1
	T1S2	295	295	295	0	1
	T1S3	295	295	295	0	1
	T1S4	295	295	295	0	1
	T1V1	295	295	295	0	1
	T1V2	295	295	295	0	1
	T1V3	295	295	295	0	1
	T1V4	295	295	295	0	1
	T2S1	295	295	295	0	1
	T2S2	295	295	295	0	1
	T2S3	295	295	295	0	1
	T2S4	295	295	295	0	1
	T2V1	295	295	295	0	1
	T2V2	295	295	295	0	1
	T2V3	295	295	295	0	1
	T2V4	295	295	295	0	1
kp-6	S1	52	52	52	0	1
	S2	52	52	52	0	1
	S3	52	52	52	0	1
	S4	52	52	52	0	1
	V1	52	52	52	0	1
	V2	52	52	52	0	1
	V3	52	52	52	0	1
	V4	52	52	52	0	1
	T1S1	52	52	52	0	1
	T1S2	52	52	52	0	1

	T1S3	52	52	52	0	1
	T1S4	52	52	52	0	1
	T1V1	52	52	52	0	1
	T1V2	52	52	52	0	1
	T1V3	52	52	52	0	1
	T1V4	52	52	52	0	1
	T2S1	52	52	52	0	1
	T2S2	52	52	52	0	1
	T2S3	52	52	52	0	1
	T2S4	52	52	52	0	1
	T2V1	52	52	52	0	1
	T2V2	52	52	52	0	1
	T2V3	52	52	52	0	1
	T2V4	52	52	52	0	1
kp-7	S1	481.07	481.069	481.07	0	1
	S2	481.07	481.069	481.07	0	1
	S3	481.07	481.069	481.07	0	1
	S4	481.07	481.069	481.07	0	1
	V1	481.07	481.069	481.07	0	1
	V2	481.07	481.069	481.07	0	1
	V3	481.07	481.069	481.07	0	1
	V4	481.07	481.069	481.07	0	1
	T1S1	481.07	481.069	481.07	0	1
	T1S2	481.07	481.069	481.07	0	1
	T1S3	481.07	481.069	481.07	0	1
	T1S4	481.07	481.069	481.07	0	1
	T1V1	481.07	481.069	481.07	0	1
	T1V2	481.07	481.069	481.07	0	1
	T1V3	481.07	481.069	481.07	0	1
	T1V4	481.07	481.069	481.07	0	1
	T2S1	481.07	481.069	481.07	0	1
	T2S2	481.07	481.069	481.07	0	1
	T2S3	481.07	481.069	481.07	0	1
	T2S4	481.07	481.069	481.07	0	1
	T2V1	481.07	481.069	481.07	0	1
	T2V2	481.07	481.069	481.07	0	1
	T2V3	481.07	481.069	481.07	0	1
	T2V4	481.07	481.069	481.07	0	1
kp-8	S1	1025	1025	1025	0	1.36
	S2	1025	1025	1025	0	1.68
	S3	1025	1025	1025	0	1.24
	S4	1025	1025	1025	0	1.68
	V1	1025	1025	1025	0	1.68
	V2	1025	1025	1025	0	1.56
	V3	1025	1025	1025	0	1.52
	V4	1025	1025	1025	0	1.28
	T1S1	1025	1025	1025	0	1.95
	T1S2	1025	1025	1025	0	1.22
	T1S3	1025	1025	1025	0	1.15
	T1S4	1025	1025	1025	0	1.56

	T1V1	1025	1025	1025	0	1.49
	T1V2	1025	1025	1025	0	1.23
	T1V3	1025	1025	1025	0	1.07
	T1V4	1025	1025	1025	0	1.04
	T2S1	1025	1025	1025	0	2.14
	T2S2	1025	1025	1025	0	1.02
	T2S3	1025	1025	1025	0	1.04
	T2S4	1025	1025	1025	0	1.32
	T2V1	1025	1025	1025	0	1.59
	T2V2	1025	1025	1025	0	1.14
	T2V3	1025	1025	1025	0	1.04
	T2V4	1025	1025	1025	0	1
kp-9	S1	1024	1024	1024	0	1.6
	S2	1024	1024	1024	0	1.72
	S3	1024	1024	1024	0	1.16
	S4	1024	1024	1024	0	1.92
	V1	1024	1024	1024	0	1.92
	V2	1024	1024	1024	0	2
	V3	1024	1024	1024	0	2.04
	V4	1024	1024	1024	0	1.48
	T1S1	1024	1024	1024	0	2.5
	T1S2	1024	1024	1024	0	1.04
	T1S3	1024	1024	1024	0	1.04
	T1S4	1024	1024	1024	0	1.21
	T1V1	1024	1024	1024	0	1.07
	T1V2	1024	1024	1024	0	1.94
	T1V3	1024	1024	1024	0	1.83
	T1V4	1024	1024	1024	0	1.08
	T2S1	1024	1024	1024	0	1.91
	T2S2	1024	1024	1024	0	1.29
	T2S3	1024	1024	1024	0	1.12
	T2S4	1024	1024	1024	0	1.67
	T2V1	1024	1024	1024	0	1.54
	T2V2	1024	1024	1024	0	1.44
	T2V3	1024	1024	1024	0	1.75
	T2V4	1024	1024	1024	0	1

Table S3: Medium size 0–1 KP test problems

Instance	dimension	capacity M	weights W	profits C
kp-11	30	577	w=[46 17 35 1 26 17 17 48 38 17 32 21 29 48 31 8 42 37 6 9 15 22 27 14 42 40 14 31 6 34]	c=[57 64 50 6 52 6 85 60 70 65 63 96 18 48 85 50 77 18 70 92 17 43 5 23 67 88 35 3 91 48]
kp-12	35	655	w=[7 4 36 47 6 33 8 35 32 3 40 50 22 18 3 12 30 31 13 33 4 48 5 17 33 26 27 19 39 15 33 47 17 41 40]	c=[35 67 30 69 40 40 21 73 82 93 52 20 61 20 42 86 43 93 38 70 59 11 42 93 6 39 25 23 36 93 51 81 36 46 96]

kp-13	40	819	w=[28 23 35 38 20 29 11 48 26 14 12 48 35 36 33 39 30 26 44 20 13 15 46 36 43 19 32 2 47 24 26 39 17 32 17 16 33 22 6 12]	c=[13 16 42 69 66 68 1 13 77 85 75 95 92 23 51 79 53 62 56 74 7 50 23 34 56 75 42 51 13 22 30 45 25 27 90 59 94 62 26 11]
kp-14	45	907	w=[18 12 38 12 23 13 18 46 1 7 20 43 11 47 49 19 50 7 39 29 32 25 12 8 32 41 34 24 48 30 12 35 17 38 50 14 47 35 5 13 47 24 45 39 1]	c=[98 70 66 33 2 58 4 27 20 45 77 63 32 30 8 18 73 9 92 43 8 58 84 35 78 71 60 38 40 43 43 22 50 4 57 5 88 87 34 98 96 99 16 1 25]
kp-15	50	882	w=[15 40 22 28 50 35 49 5 45 3 7 32 19 16 40 16 31 24 15 42 29 4 14 9 29 11 25 37 48 39 5 47 49 31 48 17 46 1 25 8 16 9 30 33 18 3 3 3 4 1]	c=[78 69 87 59 63 12 22 4 45 33 29 50 19 94 95 60 1 91 69 8 100 32 81 47 59 48 56 18 59 16 45 54 47 84 100 98 75 20 4 19 58 63 37 64 90 26 29 13 53 83]
kp-16	55	1050	w=[27 15 46 5 40 9 36 12 11 11 49 20 32 3 12 44 24 1 24 42 44 16 12 42 22 26 10 8 46 50 20 42 48 45 43 35 9 12 22 2 14 50 16 29 31 46 20 35 11 4 32 35 15 29 16]	c=[98 74 76 4 12 27 90 98 100 35 30 19 75 72 19 44 5 66 79 87 79 44 35 6 82 11 1 28 95 68 39 86 68 61 44 97 83 2 15 49 59 30 44 40 14 96 37 84 5 43 8 32 95 86 18]
kp-17	60	1006	w=[7 13 47 33 38 41 3 21 37 7 32 13 42 42 23 20 49 1 20 25 31 4 8 33 11 6 3 9 26 44 39 7 4 34 25 25 16 17 46 23 38 10 5 11 28 34 47 3 9 22 17 5 41 20 33 29 1 33 16 14]	c=[81 37 70 64 97 21 60 9 55 85 5 33 71 87 51 100 43 27 48 17 16 27 76 61 97 78 58 46 29 76 10 11 74 36 59 30 72 37 72 100 9 47 10 73 92 9 52 56 69 30 61 20 66 70 46 16 43 60 33 84]
kp-18	65	1319	w=[47 27 24 27 17 17 50 24 38 34 40 14 15 36 10 42 9 48 37 7 43 47 29 20 23 36 14 2 48 50 39 50 25 7 24 38 34 44 38 31 14 17 42 20 5 44 22 9 1 33 19 19 23 26 16 24 1 9 16 38 30 36 41 43 6]	c=[47 63 81 57 3 80 28 83 69 61 39 7 100 67 23 10 25 91 22 48 91 20 45 62 60 67 27 43 80 94 47 31 44 31 28 14 17 50 9 93 15 17 72 68 36 10 1 38 79 45 10 81 66 46 54 53 63 65 20 81 20 42 24 28 1]
kp-19	70	1426	w=[4 16 16 2 9 44 33 43 14 45 11 49 21 12 41 19 26 38 42 20 5 14 40 47 29 47 30 50 39 10 26 33 44 31 50 7 15 24 7 12 10 34 17 40 28 12 35 3 29 50 19 28 47 13 42 9 44 14 43 41 10 49 13 39 41 25 46 6 7 43]	c=[66 76 71 61 4 20 34 65 22 8 99 21 99 62 25 52 72 26 12 55 22 32 98 31 95 42 2 32 16 100 46 55 27 89 11 83 43 93 53 88 36 41 60 92 14 5 41 60 92 30 55 79 33 10 45 3 68 12 20 54 63 38 61 85 71 40 58 25 73 35]
kp-20	75	1433	w=[24 45 15 40 9 37 13 5 43 35 48 50 27 46 24 45 2 7 38 20 20 31 2 20 3 35 27 4 21 22 33 11 5 24 37 31 46 13 12 12 41 36 44 36 34 22 29 50 48 17 8 21 28 2	c=[2 73 82 12 49 35 78 29 83 18 87 93 20 6 55 1 83 91 71 25 59 94 90 61 80 84 57 1 26 44 44 88 7 34 18 25 73 29 24 14 23 82 38 67 94 43 61 97 37 67 32 89 30 30 91

		44 45 25 11 37 35 24 9 40 45 8 47 1 22 1 12 36 35 14 17 5]	50 21 3 18 31 97 79 68 85 43 71 49 83 44 86 1 100 28 4 16]

Table S4: Results obtained by the transfer functions for the medium scale 0–1 KP

Instance	Transfer function	Best	Mean	Worst	SD	Mean iterations
	S1	1437	1437	1437	0	2.48
	S2	1437	1437	1437	0	7.8
	S3	1437	1437	1437	0	6.96
	S4	1437	1437	1437	0	4.92
	V1	1437	1437	1437	0	1.84
	V2	1437	1437	1437	0	2.04
	V3	1437	1437	1437	0	1.24
	V4	1437	1437	1437	0	1.12
kp-11	T1S1	1437	1437	1437	0	5.7
	T1S2	1437	1437	1437	0	3.56
	T1S3	1437	1437	1437	0	5.81
	T1S4	1437	1437	1437	0	4.17
	T1V1	1437	1437	1437	0	1.14
	T1V2	1437	1437	1437	0	1.94
	T1V3	1437	1437	1437	0	1.08
	T1V4	1437	1437	1437	0	1.04
	T2S1	1437	1437	1437	0	3.87
	T2S2	1437	1437	1437	0	3.92
	T2S3	1437	1437	1437	0	4.91
	T2S4	1437	1437	1437	0	2.55
	T2V1	1437	1437	1437	0	1.04
	T2V2	1437	1437	1437	0	1.37
	T2V3	1437	1437	1437	0	1
	T2V4	1437	1437	1437	0	1
	S1	1689	1689	1689	0	3.8
	S2	1689	1689	1689	0	7.96
	S3	1689	1689	1689	0	4.88
	S4	1689	1689	1689	0	3.76
	V1	1689	1689	1689	0	2.2
	V2	1689	1689	1689	0	1.92
kp-12	V3	1689	1689	1689	0	1.68
	V4	1689	1689	1689	0	1.36
	T1S1	1689	1689	1689	0	4.24
	T1S2	1689	1689	1689	0	3.6
	T1S3	1689	1689	1689	0	3.48
	T1S4	1689	1689	1689	0	3.8
	T1V1	1689	1689	1689	0	1.04

	T1V2	1689	1689	1689	0	1.16
	T1V3	1689	1689	1689	0	1.04
	T1V4	1689	1689	1689	0	1.04
	T2S1	1689	1689	1689	0	4.88
	T2S2	1689	1689	1689	0	3.8
	T2S3	1689	1689	1689	0	4.64
	T2S4	1689	1689	1689	0	3.16
	T2V1	1689	1689	1689	0	1.24
	T2V2	1689	1689	1689	0	1
	T2V3	1689	1689	1689	0	1.08
	T2V4	1689	1689	1689	0	1
kp-13	S1	1821	1821	1821	0	8.28
	S2	1821	1821	1821	0	37.8
	S3	1821	1821	1821	0	22.2
	S4	1821	1821	1821	0	17.92
	V1	1821	1821	1821	0	4.56
	V2	1821	1821	1821	0	3.72
	V3	1821	1821	1821	0	2.68
	V4	1821	1821	1821	0	1.96
	T1S1	1821	1821	1821	0	11.56
	T1S2	1821	1821	1821	0	22.41
	T1S3	1821	1821	1821	0	16.63
	T1S4	1821	1821	1821	0	12.65
	T1V1	1821	1821	1821	0	3.47
	T1V2	1821	1821	1821	0	2.34
	T1V3	1821	1821	1821	0	1.94
	T1V4	1821	1821	1821	0	1.04
	T2S1	1821	1821	1821	0	9.53
	T2S2	1821	1821	1821	0	19.3
	T2S3	1821	1821	1821	0	18.66
	T2S4	1821	1821	1821	0	10.06
	T2V1	1821	1821	1821	0	1.94
	T2V2	1821	1821	1821	0	1.15
	T2V3	1821	1821	1821	0	1.35
	T2V4	1821	1821	1821	0	1.23
kp-14	S1	2033	2033	2033	0	3.28
	S2	2033	2033	2033	0	22
	S3	2033	2033	2033	0	25.28
	S4	2033	2033	2033	0	22.88
	V1	2033	2033	2033	0	3.52
	V2	2033	2033	2033	0	2.32
	V3	2033	2033	2033	0	2.08
	V4	2033	2033	2033	0	1.48
	T1S1	2033	2033	2033	0	5.19
	T1S2	2033	2033	2033	0	14.55
	T1S3	2033	2033	2033	0	20.12

	T1S4	2033	2033	2033	0	19.49
	T1V1	2033	2033	2033	0	1.42
	T1V2	2033	2033	2033	0	1.42
	T1V3	2033	2033	2033	0	1.19
	T1V4	2033	2033	2033	0	1.04
	T2S1	2033	2033	2033	0	3.92
	T2S2	2033	2033	2033	0	12.95
	T2S3	2033	2033	2033	0	17.88
	T2S4	2033	2033	2033	0	14.94
	T2V1	2033	2033	2033	0	1.08
	T2V2	2033	2033	2033	0	1.12
	T2V3	2033	2033	2033	0	1.04
	T2V4	2033	2033	2033	0	1.04
kp-16	S1	2651	2651	2651	0	13.56
	S2	2651	2650.2	2647	1.69	524.3
	S3	2651	2651	2651	0	423.53
	S4	2651	2650.2	2643	2.53	360.6
	V1	2651	2651	2651	0	11.36
	V2	2651	2651	2651	0	9.12
	V3	2651	2651	2651	0	5.04
	V4	2651	2651	2651	0	3.2
	T1S1	2651	2651	2651	0	20.82
	T1S2	2651	2651	2651	0	347.16
	T1S3	2651	2651	2651	0	279.31
	T1S4	2651	2651	2651	0	318
	T1V1	2651	2651	2651	0	8.94
	T1V2	2651	2651	2651	0	6.78
	T1V3	2651	2651	2651	0	3.92
	T1V4	2651	2651	2651	0	1.9
	T2S1	2651	2651	2651	0	15.34
	T2S2	2651	2651	2651	0	328.5
	T2S3	2651	2651	2651	0	223.96
	T2S4	2651	2651	2651	0	6.95
	T2V1	2651	2651	2651	0	7.15
	T2V2	2651	2651	2651	0	4.52
	T2V3	2651	2651	2651	0	2.94
	T2V4	2651	2651	2651	0	1.08
kp-17	S1	2917	2917	2917	0	6.08
	S2	2917	2917	2917	0	81.88
	S3	2917	2917	2917	0	43.44
	S4	2917	2917	2917	0	42.88
	V1	2917	2917	2917	0	3.52
	V2	2917	2917	2917	0	2.88
	V3	2917	2917	2917	0	2.16
	V4	2917	2917	2917	0	1.8
	T1S1	2917	2917	2917	0	8.5
	T1S2	2917	2917	2917	0	50.73

	T1S3	2917	2917	2917	0	40.95
	T1S4	2917	2917	2917	0	39.27
	T1V1	2917	2917	2917	0	1.93
	T1V2	2917	2917	2917	0	1.96
	T1V3	2917	2917	2917	0	1.04
	T1V4	2917	2917	2917	0	1
	T2S1	2917	2917	2917	0	6.94
	T2S2	2917	2917	2917	0	48.33
	T2S3	2917	2917	2917	0	35.97
	T2S4	2917	2917	2917	0	31.94
	T2V1	2917	2917	2917	0	1.26
	T2V2	2917	2917	2917	0	1.75
	T2V3	2917	2917	2917	0	1.23
	T2V4	2917	2917	2917	0	1
kp-18	S1	2818	2818	2818	0	24.28
	S2	2818	2817.1	2817	0.32	978.5
	S3	2818	2817.1	2817	0.32	965.5
	S4	2818	2817.1	2817	0.32	905.5
	V1	2818	2818	2818	0	94.76
	V2	2818	2818	2818	0	39.6
	V3	2818	2818	2818	0	30.36
	V4	2818	2818	2818	0	18.52
	T1S1	2818	2818	2818	0	30.65
	T1S2	2818	2818	2818	0	766.3
	T1S3	2818	2818	2818	0	598.5
	T1S4	2818	2818	2818	0	832.9
	T1V1	2818	2818	2818	0	71.2
	T1V2	2818	2818	2818	0	23.9
	T1V3	2818	2818	2818	0	19.43
	T1V4	2818	2818	2818	0	8.69
	T2S1	2818	2818	2818	0	27.83
	T2S2	2818	2818	2818	0	515.8
	T2S3	2818	2818	2818	0	691.6
	T2S4	2818	2818	2818	0	536.4
	T2V1	2818	2818	2818	0	74.9
	T2V2	2818	2818	2818	0	21.95
	T2V3	2818	2818	2818	0	15.35
	T2V4	2818	2818	2818	0	6.94
kp-19	S1	3223	3223	3223	0	95.68
	S2	3223	3221.2	3221	0.632	985.5
	S3	3223	3221.2	3221	0.632	589.5
	S4	3223	3221.2	3221	0.632	765.3
	V1	3223	3223	3223	0	13.1
	V2	3223	3223	3223	0	9.43
	V3	3223	3223	3223	0	7.93
	V4	3223	3223	3223	0	4.1
	T1S1	3223	3223	3223	0	120.6

		T1S2	3223	3223	3223	0	645.35
		T1S3	3223	3223	3223	0	428.6
		T1S4	3223	3223	3223	0	594.82
		T1V1	3223	3223	3223	0	9.72
		T1V2	3223	3223	3223	0	9.18
		T1V3	3223	3223	3223	0	4.38
		T1V4	3223	3223	3223	0	2.14
		T2S1	3223	3223	3223	0	89.5
		T2S2	3223	3223	3223	0	583.8
		T2S3	3223	3223	3223	0	357.91
		T2S4	3223	3223	3223	0	455.91
		T2V1	3223	3223	3223	0	8.16
		T2V2	3223	3223	3223	0	6.28
		T2V3	3223	3223	3223	0	6.13
		T2V4	3223	3223	3223	0	1.91

Table S5: Comparison results of uncorrelated large size 0–1 KP

Instance	Dimension	Transfer function	Best	Mean	Worst	SD	Mean iterations
kp-21	100	S1	2060	2060	2060	0	142
		S2	2060	2038.4	2023	19.756	350
		S3	2060	2036.8	2023	17.387	353
		S4	2060	2043	2023	18.886	351
		V1	2060	2060	2060	0	95
		V2	2060	2060	2060	0	88
		V3	2060	2060	2060	0	90
		V4	2060	2060	2060	0	76
		T1S1	2060	2060	2060	0	169
		T1S2	2060	2049	2045	6.519	153
		T1S3	2060	2049.8	2045	6.14	215
		T1S4	2060	2048.4	2045	7.092	201
		T1V1	2060	2060	2060	0	57
		T1V2	2060	2060	2060	0	45
		T1V3	2060	2060	2060	0	62
		T1V4	2060	2060	2060	0	39
		T2S1	2060	2060	2060	0	185
		T2S2	2060	2049.6	2048	6.542	162
		T2S3	2060	2049	2048	6.633	224
		T2S4	2060	2049.6	2048	5.941	240
		T2V1	2060	2060	2060	0	61
		T2V2	2060	2060	2060	0	42
		T2V3	2060	2060	2060	0	59
		T2V4	2060	2060	2060	0	34
kp-23	1000	S1	20348	20339.6	20334	7.668	2859
		S2	20340	20327.2	20308	17.527	4781
		S3	20340	20329.4	20308	15.027	4618
		S4	20340	20326.8	20308	17.181	4237

		V1	20350	20347.6	20346	2.19	531
		V2	20350	20348	20346	2	498
		V3	20350	20348.4	20346	2.19	562
		V4	20350	20349.2	20346	1.789	415
		T1S1	20350	20344.8	20342	3.899	2094
		T1S2	20349	20338.8	20332	9.311	3586
		T1S3	20349	20339.4	20332	8.848	3428
		T1S4	20349	20338.8	20332	7.12	4057
		T1V1	20350	20348.8	20348	1.095	463
		T1V2	20350	20349.4	20348	0.894	401
		T1V3	20350	20348.8	20348	0.837	443
		T1V4	20350	20349.4	20348	0.894	381
		T2S1	20350	20346	20342	3.742	2099
		T2S2	20349	20342.2	20332	9.3	3308
		T2S3	20349	20343.6	20332	7.8	3258
		T2S4	20349	20340.2	20332	8.526	3974
		T2V1	20350	20349.2	20348	1.095	492
		T2V2	20350	20349.2	20348	0.837	391
		T2V3	20350	20349	20348	1	412
		T2V4	20350	20349.6	20348	0.894	328
kp-24	1500	S1	31255	31247.8	31241	6.686	4289
		S2	31250	31224.2	31198	26.04	6722
		S3	31250	31222	31198	23.63	5983
		S4	31250	31224	31198	25.76	5347
		V1	31257	31249.8	31245	5.215	1025
		V2	31257	31250.8	31245	4.49	992
		V3	31257	31250.4	31245	4.67	1349
		V4	31257	31252.2	31245	4.38	903
		T1S1	31257	31254.6	31253	2.19	2951
		T1S2	31256	31237.6	31228	13.45	4228
		T1S3	31256	31235.4	31228	14.69	3984
		T1S4	31256	31236.4	31228	15.646	4129
		T1V1	31257	31255.8	31255	1.095	751
		T1V2	31257	31256	31255	1	694
		T1V3	31257	31256.2	31255	0.837	789
		T1V4	31257	31256.4	31255	0.821	562
		T2S1	31257	31255	31253	2	2493
		T2S2	31256	31237.4	31228	12.605	4059
		T2S3	31256	31238.6	31228	12.954	4286
		T2S4	31256	31238.4	31228	11.06	4178
		T2V1	31257	31255.8	31255	0.837	842
		T2V2	31257	31256	31255	0.707	697
		T2V3	31257	31255.6	31255	0.894	981
		T2V4	31257	31256.4	31255	0.821	509

Table S6: Comparison results of weakly correlated large size 0–1 KP

Instance	Dimension	Transfer function	Best	Mean	Worst	SD	Mean iterations
kp-26	100	S1	1016	1016	1016	0	120

		S2	1016	997.2	989	11.278	325
		S3	1016	994.2	989	9.952	298
		S4	1016	998	989	10.677	337
		V1	1016	1016	1016	0	86
		V2	1016	1016	1016	0	79
		V3	1016	1016	1016	0	83
		V4	1016	1016	1016	0	62
		T1S1	1016	1016	1016	0	157
		T1S2	1016	1004	999	7.661	129
		T1S3	1016	1007	999	6.166	194
		T1S4	1016	1009.4	999	7.057	173
		T1V1	1016	1016	1016	0	65
		T1V2	1016	1016	1016	0	49
		T1V3	1016	1016	1016	0	64
		T1V4	1016	1016	1016	0	30
		T2S1	1016	1016	1016	0	160
		T2S2	1016	1009.6	1008	3.578	122
		T2S3	1016	1012	1008	4	188
		T2S4	1016	1011.2	1008	3.347	169
		T2V1	1016	1016	1016	0	64
		T2V2	1016	1016	1016	0	39
		T2V3	1016	1016	1016	0	61
		T2V4	1016	1016	1016	0	31
kp-28	1000	S1	10352	10345.4	10340	9.631	2156
		S2	10351	10348.2	10338	13.964	3924
		S3	10351	10347.8	10338	14.57	3615
		S4	10351	10348	10338	13.238	4025
		V1	10356	10353.6	10350	5.613	492
		V2	10356	10354	10350	4.679	385
		V3	10356	10353.8	10350	5.348	477
		V4	10356	10354.4	10350	4.069	336
		T1S1	10357	10354.2	10352	2.15	2294
		T1S2	10357	10352.1	10344	7.281	2960
		T1S3	10357	10351.6	10344	6.823	3458
		T1S4	10357	10351.4	10344	6.134	3512
		T1V1	10357	10355.1	10354	2.36	418
		T1V2	10357	10356	10354	1.068	304
		T1V3	10357	10355.4	10354	2.06	393
		T1V4	10357	10356.2	10354	0.985	300
		T2S1	10357	10354	10352	2.01	2341
		T2S2	10357	10351.8	10344	7.328	3001
		T2S3	10357	10351.2	10344	7.054	2923
		T2S4	10357	10532	10344	6.832	3194
		T2V1	10357	10355.4	10354	2.456	407
		T2V2	10357	10356.1	10354	1.37	318
		T2V3	10357	10356	10354	1.254	371
		T2V4	10357	10356.2	10354	0.895	303
kp-29	1500	S1	15446	15441.6	15432	7.627	4091
		S2	15400	15383.4	15365	21.975	5906
		S3	15400	15384	15365	19.713	5213

		S4	15400	15383.6	15365	20.668	5194
		V1	15496	15486	15454	10.634	1395
		V2	15496	15486.4	15454	9.248	1082
		V3	15496	15486.2	15454	11.573	1267
		V4	15496	15486.8	15454	8.275	990
		T1S1	15507	15491	15470	5.382	3924
		T1S2	15507	15495.6	15460	13.627	4615
		T1S3	15507	15494.8	15460	14.351	4208
		T1S4	15507	15495	15460	12.01	4792
		T1V1	15507	15502	15490	5.617	971
		T1V2	15507	15503.2	15490	4.319	898
		T1V3	15507	15502.6	15490	4.065	919
		T1V4	15507	15503.2	15490	4.319	805
		T2S1	15507	15491.6	15454	5.915	3902
		T2S2	15507	15494.2	15460	12.951	4593
		T2S3	15507	15494	15460	13.986	4321
		T2S4	15507	15495.4	15460	13.035	4435
		T2V1	15507	15502.6	15490	4.972	850
		T2V2	15507	15503	15490	3.658	777
		T2V3	15507	15502	15490	5.349	882
		T2V4	15507	15503.4	15490	4.568	692

Table S7: Comparison results of strongly correlated large size 0–1 KP

Instance	Dimension	Transfer function	Best	Mean	Worst	SD	Mean iterations
kp-31	100	S1	1332	1332	1332	0	95
		S2	1332	1332	1332	0	184
		S3	1332	1332	1332	0	176
		S4	1332	1332	1332	0	190
		V1	1332	1332	1332	0	54
		V2	1332	1332	1332	0	31
		V3	1332	1332	1332	0	50
		V4	1332	1332	1332	0	29
		T1S1	1332	1332	1332	0	109
		T1S2	1332	1332	1332	0	123
		T1S3	1332	1332	1332	0	125
		T1S4	1332	1332	1332	0	120
		T1V1	1332	1332	1332	0	40
		T1V2	1332	1332	1332	0	25
		T1V3	1332	1332	1332	0	33
		T1V4	1332	1332	1332	0	19
		T2S1	1332	1332	1332	0	137
		T2S2	1332	1332	1332	0	114
		T2S3	1332	1332	1332	0	122
		T2S4	1332	1332	1332	0	108
		T2V1	1332	1332	1332	0	47
		T2V2	1332	1332	1332	0	27
		T2V3	1332	1332	1332	0	30
		T2V4	1332	1332	1332	0	25
kp-33	1000	S1	13410	13405.2	13395	7.391	1945

		S2	13390	13386	13370	18.62	3026
		S3	13390	13385.9	13370	19.28	2915
		S4	13390	13386.4	13370	17.54	3108
		V1	13448	13441	13436	12.68	1154
		V2	13448	13442	13436	10.25	1093
		V3	13448	13441.6	13436	11.39	1251
		V4	13448	13442.4	13436	10.09	1001
		T1S1	13454	13449	13440	9.52	2397
		T1S2	13454	13448.5	13440	10.27	2108
		T1S3	13454	13448	13440	9.35	2293
		T1S4	13454	13447.2	13440	11.24	2604
		T1V1	13454	13454	13454	0	938
		T1V2	13454	13454	13454	0	756
		T1V3	13454	13454	13454	0	849
		T1V4	13454	13454	13454	0	720
		T2S1	13454	13447	13440	10.14	2253
		T2S2	13454	13448	13440	8.37	2204
		T2S3	13454	13447.8	13440	9.63	2157
		T2S4	13454	13448.2	13440	8.921	2384
		T2V1	13454	13454	13454	0	961
		T2V2	13454	13454	13454	0	770
		T2V3	13454	13454	13454	0	902
		T2V4	13454	13454	13454	0	738
kp-34	1500	S1	20295	20284	20252	10.692	3157
		S2	20289	20277.2	20246	25.34	5089
		S3	20289	20276	20246	27.137	4975
		S4	20289	20276.8	20246	26.821	5013
		V1	20310	20306	20302	9.38	1085
		V2	20310	20306.8	20302	8.937	1020
		V3	20310	20306.4	20302	9.036	1017
		V4	20310	20307	20302	10.02	1003
		T1S1	20314	20312	20300	6.35	3681
		T1S2	20314	20310	20296	16.93	3054
		T1S3	20314	20310.6	20296	18.37	3376
		T1S4	20314	20311	20296	20.589	3128
		T1V1	20314	20314	20314	0	984
		T1V2	20314	20314	20314	0	793
		T1V3	20314	20314	20314	0	980
		T1V4	20314	20314	20314	0	751
		T2S1	20314	20312.6	20300	5.83	3507
		T2S2	20314	20311	20296	19.34	3182
		T2S3	20314	20310.6	20296	15.61	3259
		T2S4	20314	20310.4	20296	13.29	3204
		T2V1	20314	20314	20314	0	901
		T2V2	20314	20314	20314	0	800
		T2V3	20314	20314	20314	0	826
		T2V4	20314	20314	20314	0	748

References

[1] Wang G, Guo L, Wang H, Duan H, Liu L and Li J 2012 Incorporating mutation scheme into krill

herd algorithm for global numerical optimization *Neural Computing and Applications* **24**(3-4) pp 853-871

[2] Feng Y, Yang J, Wu C, Lu M and Zhao X 2016 Solving 0–1 knapsack problems by chaotic monarch butterfly optimization algorithm with Gaussian Mutation *Memetic Computing* **10**(2) pp 135-150

[3] Tilahun S L and J M T Ngnotchouye 2017 Firefly algorithm for discrete optimization problems: A survey *KSCE Journal of Civil Engineering* **21**(2) pp 535-545

[4] Parsopoulos K E and Vrahatis M N 2010 Particle Swarm Optimization and Intelligence Advances and Applications *IGI Global*

[5] Abdel-Basset M, El-Shahat D and Sangaiah A K 2017 A modified nature inspired meta-heuristic whale optimization algorithm for solving 0–1 knapsack problem *International Journal of Machine Learning and Cybernetics*

[6] Abdel-Basset M, El-Shahat D and El-Henawy I 2017 Solving 0–1 Knapsack Problems by Binary Dragonfly Algorithm *Springer International Publishing AG* 10363 pp 491-502

[7] Cao J, Yin B, Lu X, Kang Y and Chen X 2017 A modified artificial bee colony approach for the 0-1 knapsack problem *Applied Intelligence* **48**(6) pp 1582-1595

[8] Layeb A 2013 A hybrid quantum inspired harmony search algorithm for 0–1 optimization problems *Journal of Computational and Applied Mathematics* 253 pp 14-25

[9] Abdel-Basset M, D El-Shahat and I El-Henawy 2018 Solving 0–1 knapsack problem by binary flower pollination algorithm *Neural Computing and Applications*

[10] Zouache D, A Moussaoui and F Ben Abdelaziz 2018 A cooperative swarm intelligence algorithm for multi-objective discrete optimization with application to the knapsack problem *European Journal of Operational Research* **264**(1) pp 74-88

[11] Feng Y, Wang G, Deb S, Lu M and Zhao X 2015 Solving 0–1 knapsack problem by a novel binary monarch butterfly optimization *Neural Computing and Applications* **28**(7) pp 1619-1634

[12] Zhou Y, L Li, and M Ma 2015 A Complex-valued Encoding Bat Algorithm for Solving 0–1 Knapsack Problem *Neural Processing Letters* **44**(2) pp 407-430

[13] Rizk-Allah R M and A E Hassanien 2017 New binary bat algorithm for solving 0–1 knapsack problem *Complex & Intelligent Systems* **4**(1) pp 31-53

[14] Haddar B, Khemakhem M, Hanafi S and Wilbaut C 2015 A hybrid heuristic for the 0–1 Knapsack Sharing Problem *Expert Systems with Applications* **42**(10) pp 4653-4666

[15] Haddar B, Khemakhem M, Rhimi H and Chabchoub H 2014 A quantum particle swarm optimization for the 0–1 generalized knapsack sharing problem *Natural Computing* **15**(1) pp 53-164

[16] Bansal J C and K Deep 2012 A Modified Binary Particle Swarm Optimization for Knapsack Problems *Applied Mathematics and Computation* **218**(22) pp 11042-11061

[17] Zhou Y, X Chen and G Zhou 2016 An improved monkey algorithm for a 0-1 knapsack problem *Applied Soft Computing* **6(**38) pp 817-830

[18] Changdar C, G S Mahapatra and R K Pal 2013 An Ant colony optimization approach for binary knapsack problem under fuzziness *Applied Mathematics and Computation* 223 pp. 243-253

[19] Zhang X, Huang S, Hu Y, Zhang Y, Mahadevan S and Deng Y 2013 Solving 0-1 knapsack problems based on amoeboid organism algorithm *Applied Mathematics and Computation* **219**(19) pp 9959-9970

[20] Zou D, Gao L, Li S and Wu J 2011 Solving 0–1 knapsack problem by a novel global harmony search algorithm *Applied Soft Computing* **11**(2) pp 1556-1564

[21] Yang X S 2012 Flower pollination algorithm for global optimization Unconventional *Computation and Natural Computation* 7445 pp 240-249

[22] Yang X S, M Karamanoglu and H Xingshi 2014 *Flower pollination algorithm: A noval approach for multiobjective optimization* Engineering Optimization

[23] Yang X S 2015 *Recent Advances in Swarm Intelligence and Evolutionary Computation* Springer International Publishing Switzerland

[24] Islam M J, X Li and Y Mei 2017 A time-varying transfer function for balancing the exploration and exploitation ability of a binary PSO *Applied Soft Computing* **59** pp 182-196
[25] Kulkarni A J and H Shabir 2014 Solving 0–1 Knapsack Problem using Cohort Intelligence Algorithm *International Journal of Machine Learning and Cybernetics* **7**(3) pp 427-441
[26] Abdel-Basset M, El-Shahat D, El-Henawy I and Sangaiah A A 2017 modified flower pollination algorithm for the multidimensional knapsack problem: human-centric decision making *Soft Computing* **22**(13) pp 4221-4239
[27] Bozorg-Haddad O 2018 *Advanced Optimization by Nature-Inspired Algorithms* Springer Nature Singapore Pte Ltd
[28] Mafarja M, Aljarah I, Heidari A, Faris H, Fournier-Viger P, Li X and Mirjalili S 2018 Binary dragonfly optimization for feature selection using time-varying transfer functions *Knowledge-Based Systems* 161 pp 185-204
[29] Mirjalili S and A Lewis 2013 S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization *Swarm and Evolutionary Computation* 9 pp 1-14
[30] Teng X, H Dong and X Zhou 2017 Adaptive feature selection using v-shaped binary particle swarm optimization *PLoS One* **12**(3) pp e0173907