Solving 0 –1 knapsack problem by an improved binary Pigeon Inspired Optimization Algorithm

Rana Bashar Hussein
Department of management Information systems, College of Administration and Economics, University of Mosul, Mosul, Iraq
E-mail: rana_bashar@uomosul.edu.iq
ZakariyaYahya Algamal *
Department of Statistics and Informatics, University of Mosul, Mosul, Iraq
E-mail: zakariya.algamal@uomosul.edu.iq
ORCID: 0000-0002-0229-7958
*Corresponding Author
Zakariya YahyaAlgamal
E-mail: zakariya.algamal@uomosul.edu.iq
Telephone number: +964 7701640834

Article Info
Page Number: 312-324
Publication Issue:
Vol 71 No. 3s2 (2022)

Article History
Article Received: 28 April 2022
Revised: 15 May 2022
Accepted: 20 June 2022
Publication: 21 July 2022

Abstract
The binary pigeon inspired optimization Algorithm (BPOA) is a meta-heuristic algorithm that has been applied widely in combinational optimization problems. Binary knapsack problem has received considerable attention in the combinational optimization. In this paper, a new time-varying transfer function is proposed to improve the exploration and exploitation capability of the BPOA with the best solution and short computing time. Based on small, medium, and high-dimensional sizes of the knapsack problem, the computational results reveal that the proposed time-varying transfer functions obtain the best results not only by finding the best possible solutions but also by yielding short computational times. Compared to the standard transfer functions, the efficiency of the proposed time-varying transfer functions is superior, especially in the high-dimensional sizes.

Keywords: 0-1 knapsack problem; pigeon inspired optimization Algorithm; transfer function; time-varying parameter.

1. Introduction

The knapsack problem is considered as one of the NP-hard combinatorial optimization problems. “The knapsack problem cannot be solved efficiently in a practically acceptable time scale using the exact algorithms because the computational time increases exponentially with the problem size. This leads to use approximate algorithms such as meta-heuristic algorithms to getting a good solutions, not necessarily optimal, in a reasonable time [1, 2].

The meta-heuristic algorithms are simple, flexible and they can be deal with the problems with different objective function properties, either discrete problems, continuous problems, or mixed problems [2]. These algorithms include genetic algorithm (GA) [3], particle swarm optimization (PSO) [4], artificial fish swarm algorithm (AFSA) [5], harmony search algorithm (HAS) [6, 7], gravitational search algorithm (GSA) [8], moth search algorithm (MSA) [9], cuckoo search algorithm (CSA) [10, 11], firefly algorithm (FA) [12], artificial bee colony algorithm (ABCA) [13], bat algorithm (BA) [14, 15], flower pollination algorithm (FPA) [1], and pigeon inspired optimization Algorithm (POA) [16].

The pigeon optimization algorithm (POA), which was proposed by Duan and Qiao [17], has certain outstanding merits, such as a simple computational process, simple implementation, and easy understanding with only a few parameters for tuning. Due to its good properties, POA has become a useful tool for many real-world problems [18-23]. The POA simulates the homing behavior of pigeons.

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

In this paper, an efficient time-varying transfer function is proposed to solve the 0 –1 knapsack problem. The proposed transfer function is 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 the POA. In Section 4, the proposed time-varying transfer function is presented. Section 5 presents and discusses the experimental results. In section 6, conclusions are drawn.

2. Knapsack problem

Knapsack problem is one of the NP-hard combinatorial optimization problems, which has been widely studied in operation research. Knapsack problem consists of a set of n items where each item i has a profit c_i, weight w_i, and maximum weight capacity M. The objective is to maximize the total profit of the selected items in the knapsack such that the total weights of these items are achieved by Eq.(2). Mathematically, the knapsack problem can be written as [25, 26]:

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

s.t.

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

where

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

Using the penalty function, the knapsack problem can be written as follows:

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

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

Penalty function

- **for** each x_i
- Calculate total weight of x_i by $\left(\sum_{i=1}^{n} w_i x_i\right)$
- **if** $(total\_weight \le knapsack\_capacity)$
- $\phi(x) = -f(x)$
- **else**
- $\phi(x) = -f(x) + \lambda(total\_weight - knapsack\_capacity)$
- **end**

Figure 1: Penalty function

3. Pigeon optimization algorithm

The POA mainly consists of two operators: the map and compass operator and the landmark operator. In the map and compass operator, pigeons sense the geomagnetic field to shape the map for homing. Suppose that the search space is N-dimensional, and then the i-th pigeon of the swarm can be represented by a N-dimensional vector $X_i = (X_{i,1}, X_{i,2}, ..., X_{i,N})$. The velocity of this pigeon, which represents the position change of this pigeon, can be represented by another N-dimensional vector $V_i = (V_{i,1}, V_{i,2}, ..., V_{i,N})$. The best previously visited position of the i-th pigeon is denoted as $P_i = (P_{i,1}, P_{i,2}, ..., P_{i,N})$.The global best position of the swarm is $g = (g_1, g_2, \ldots, g_N)$. Each pigeon is flying according to the following two equations:

$$V_i(t+1) = V_i(t) \times e^{-Rt} + rand \times (X_g - X_i(t)) \tag{4}$$

$$X_i(t+1) = X_i(t) + V_i(t+1), \tag{5}$$

where R is a map and compass factor, while $rand$ is a uniform random number in the range $[0, 1]$, X_g is the global best solution, $X_i(t)$ denotes the current position of a pigeon at instance t, and $V_i(t)$ denotes the current velocity of a pigeon at iteration t.

In landmark operator, all the pigeons are ranked according to their fitness value. In each generation, the number of pigeons is updated by Eq. (4), where only half number of pigeons is considered to calculate the desired position of the centered pigeon, while all other pigeons adjust their destination by following the desirable destination position.

$$N_p(t+1) = \frac{N_p(t)}{2}, \tag{6}$$

where N_p is the number of pigeons in the current iteration t.

The position of the desired destination is calculated by Eq. (5), while all other pigeons update their position toward this position by Eq. (6) [17].

$$X_c(t+1) = \frac{\sum X_i(t+1) \times \text{Fitness}(X_i(t+1))}{N_p \sum \text{Fitness}(X_i(t+1))} \quad (7)$$

$$X_i(t+1) = X_i(t) + rand \times \left(X_c(t+1) \times X_i(t)\right), \quad (8)$$

where X_c is the position of the centered pigeon (desired destination).

To perform the variable selection, a BPOA was proposed [27]. Unlike the standard POA, in which the solutions are updated in the search space towards continuous-valued positions, in the BPOA, the search space is modeled as an n-dimensional Boolean lattice and the solutions are updated across the corners of a hypercube. In addition, as the problem is to select or not a given variable, a solution binary vector is employed, where 1 corresponds whether a variable will be selected to compose the new dataset, 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].

The proposed time-varying transfer functions

The standard POA was originally proposed to handle a continuous optimization problems. In discrete optimization problems, such as knapsack problem, the standard method cannot be applied directly to deal for such this problems. Therefore the transfer functions are usually employed to convert the continuous search space to discrete search space. There are two families of transfer functions: S-shaped and V-shaped transfer function which were proposed by [29]. The V-shaped transfer functions have also been studied by [30] to tackle the feature selection problem. The most common transfer functions from the S-shaped family is the sigmoid function [31, 32]:

$$S\left(x_i^t\right) = \frac{1}{1+e^{-x_i^t}} \quad (9)$$

$$x_i^t = \begin{cases} 1 & if \quad S\left(x_i^t\right) > rand \\ 0 & \text{O.W} \end{cases} \quad (10)$$

On the other hand, the inverse tangent hyperbolic function is the most common used transfer function from the V-shaped family. It is defined as:

$$V\left(x_i^t\right) = \left|\frac{2}{\pi} \arctan\left(\frac{\pi}{2} x_i^t\right)\right| \quad (11)$$

$$x_i^t = \begin{cases} 1 & if \quad V\left(x_i^t\right) > rand \\ 0 & \text{O.W} \end{cases} \quad (12)$$

The transfer function is the main key to the balance between exploitation and exploration [24, 28]. In our proposed time-varying transfer function, a new control parameter φ is added in the original transfer function. This φ is a time-varying variable which starts with a large value and gradually decreases over time and it is expressed in Eq.(16).

$$\varphi = \varphi_{\min} + (\varphi_{\max} - \varphi_{\min}) \times e^{-t} \qquad (13)$$

where $\varphi_{\max}$ and $\varphi_{\min}$ are, respectively, the minimum and maximum values of the control parameter φ, and T is the maximum iteration of the BPOA. Accordingly, the two proposed transfer functions are defined as, respectively,

$$TVS\left(x_i^{\,t}\right) = \frac{1}{1 + e^{\frac{-x_i^t}{\varphi}}} \qquad (14)$$

and

$$TVV\left(x_i^{\,t}\right) = \left|\frac{2}{\pi}\arctan\left(\frac{\pi x_i^{\,t}}{2\varphi}\right)\right| \qquad (15)$$

Figure 1 explains the behavior of the proposed time-varying transfer function for both Eq. (12) and Eq. (14), respectively. It is obvious that these proposed functions coverage to be a vertical line when iteration increasing.

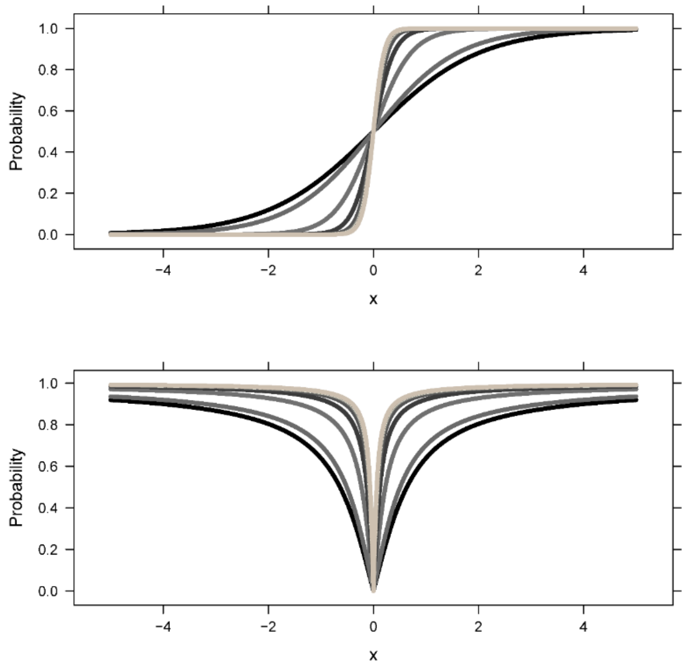

Figure 1: Explanation of the time-varying transfer function when $\varphi_{\max} = 2$ and $\varphi_{\min} = 0.1$ during 10 iteration. The top panel is the sigmoid transfer function and the bottom panel is the inverse tangent hyperbolic transfer function.

4. Computational results

5.1 Parameter setting

For the binary POA, we set the parameters as follows: the population size =50, migration ratio = 5/12, migration period= 1.2, butterfly adjusting rate = 5/12, and Max step = 1. In addition, we used linear decreasing time varying with $\varphi_{\max} = 2$ and $\varphi_{\min} = 0.1$.

4.2 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 sizes. In this paper, all the results are obtained from 50 independent trials. The Best, Mean, Worst, SD, 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.

4.2.1 Low size 0-1 KP

The performance of improved algorithm is investigated to solve ten low scale 0-1 KP instances (kp-1 to kp-10), which are taken from [1, 14]. 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 (supplementary file). Table 1 shows the comparison results for all the used different transfer functions for the kp1 - kp10.

As observed from the results in Table 1, 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. 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 kp4, kp5, 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 optimal solution using TVV is less than of TVS for kp4, kp5, kp6, kp8, kp9, and kp10.

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

Instance	Transfer function	Best	Mean	Worst	SD	Mean iterations
kp-1	S	35	35	35	0	1
	V	35	35	35	0	1
	TVS	35	35	35	0	1
	TVV	35	35	35	0	1
kp-2	S	23	23	23	0	1
	V	23	23	23	0	1

	TVS	23	23	23	0	1
	TVV	23	23	23	0	1
kp-3	S	130	130	130	0	1
	V	130	130	130	0	1
	TVS	130	130	130	0	1
	TVV	130	130	130	0	1
kp-4	S	107	107	107	0	2.17
	V	107	107	107	0	1.11
	TVS	107	107	107	0	1
	TVV	107	107	107	0	1
kp-5	S	295	295	295	0	3.38
	V	295	295	295	0	2.15
	TVS	295	295	295	0	1
	TVV	295	295	295	0	1
kp-6	S	52	52	52	0	1.17
	V	52	52	52	0	1.14
	TVS	52	52	52	0	1
	TVV	52	52	52	0	1
kp-7	S	481.07	481.069	481.07	0	1
	V	481.07	481.069	481.07	0	1
	TVS	481.07	481.069	481.07	0	1
	TVV	481.07	481.069	481.07	0	1
kp-8	S	1025	1025	1025	0	2.63
	V	1025	1025	1025	0	1.95
	TVS	1025	1025	1025	0	1.66
	TVV	1025	1025	1025	0	1
kp-9	S	1024	1024	1024	0	2.94
	V	1024	1024	1024	0	1.21
	TVS	1024	1024	1024	0	1.12
	TVV	1024	1024	1024	0	1
kp-10	S	9767	9767	9767	0	5.36
	V	9767	9767	9767	0	3.31
	TVS	9767	9767	9767	0	4.22
	TVV	9767	9767	9767	0	2.41

4.2.2 Medium size 0-1 KP

To further evaluate the performance of proposed time-varying transfer functions in medium size 0-1 Knapsack problem, ten medium size 0-1 KP instances (kp-11 to kp-20) are taken from [1, 14] in which the items are between 30 and 75. The description of these ten instances are described in Table S2 (supplementary file). Table 2 summarizes the comparison results for all the used different transfer functions.

Obviously, it is evident from Table 2 that the proposed time-varying transfer functions obtained the same best, worse, mean, and SD values as the standard transfer functions. From Tables 2, for the mean iterations, the proposed time-varying transfer functions are superior to the 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 TVS function was 63.15% lower than that of S function. On the other hand, the reduction in mean iteration of TVV function was 57.94% lower than that of V function.

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 function, the required iterations to get optimal solution using TVV is less than of TVS for all the 0-1 Knapsack problems.

Table 2: Results obtained by the transfer functions for the medium size 0–1 KP

Instance	Transfer function	Best	Mean	Worst	SD	Mean iterations
kp-11	S	1437	1437	1437	0	8.83
	V	1437	1437	1437	0	4.92
	TVS	1437	1437	1437	0	5.74
	TVV	1437	1437	1437	0	2.95
kp-12	S	1689	1689	1689	0	10.19
	V	1689	1689	1689	0	4.62
	TVS	1689	1689	1689	0	5.51
	TVV	1689	1689	1689	0	2.63
kp-13	S	1821	1821	1821	0	41.76
	V	1821	1821	1821	0	13.9
	TVS	1821	1821	1821	0	29.33
	TVV	1821	1821	1821	0	6.51
kp-14	S	2033	2033	2033	0	30.25
	V	2033	2033	2033	0	9.64
	TVS	2033	2033	2033	0	20.03
	TVV	2033	2033	2033	0	3.82
kp-15	S	2440	2440	2440	0	35.87
	V	2440	2440	2440	0	13.04
	TVS	2440	2440	2440	0	22.96
	TVV	2440	2440	2440	0	7.25
kp-16	S	2651	2648.5	2643	2.86	699.08
	V	2651	2651	2651	0	17.93
	TVS	2651	2651	2651	0	476.29
	TVV	2651	2651	2651	0	9.96
kp-17	S	2917	2917	2917	0	196.41

	V	2917	2917	2917	0	26.04
	TVS	2917	2917	2917	0	75.99
	TVV	2917	2917	2917	0	10.28
kp-18	S	2818	2815.6	2794	1.73	894.78
	V	2818	2818	2818	0	12.26
	TVS	2818	2818	2818	0	529.38
	TVV	2818	2818	2818	0	6.57
kp-19	S	3223	3221.6	3219	0.93	785.18
	V	3223	3223	3223	0	11.89
	TVS	3223	3223	3223	0	6.5
	TVV	3223	3223	3223	0	6.86
kp-20	S	3614	3614	3614	0	590.38
	V	3614	3614	3614	0	9.93
	TVS	3614	3614	3614	0	217.98
	TVV	3614	3614	3614	0	4.57

4.2.3 High-dimensional size 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 3 – 5 reports the comparison results for all the used different transfer functions. 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, TVV, 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 TVV compared to TVS. This leads to the performance dominance of the inverse tangent hyperbolic transfer function against the sigmoid transfer function.
(4) The mean iteration values of time-varying V-shaped transfer functions, TVV, are obviously superior to S and V functions for all high-dimensional size problems.

(5) Compared to the proposed time-varying V-shaped transfer functions, TVV is significantly improve the performance metrics with lower SD and mean iterations".

Table 3: Comparison results of uncorrelated high-dimensional size 0–1 KP

Instance	Dimension	Transfer function	Best	Mean	Worst	SD	Mean iterations
kp-21	100	S	2126	2120.8	2116	158	1088
		V	2126	2126	2126	0	72
		TVS	2126	2126	2126	0	523
		TVV	2126	2126	2126	0	37
kp-22	500	S	11025	11017.5	11012	18.5	3028
		V	11025	11024.2	11023	1.51	130
		TVS	11025	11023.8	11022	2.20	1585
		TVV	11025	11025	11025	0	68
kp-23	1000	S	21963	21958.6	21950	17.1	6856
		V	21968	21967.1	21965	2.11	842
		TVS	21969	21966.9	21963	4.92	2979
		TVV	21969	21969	21969	0	494
kp-24	1500	S	32633	32626.2	32620	20.65	5631
		V	32639	32637.8	32636	2.61	1960
		TVS	32637	32635.4	32634	3.48	3276
		TVV	32640	32639.2	32638	0.34	981
kp-25	2000	S	43711	43705	43692	31.6	8857
		V	43725	43722.6	43720	5.22	3764
		TVS	43723	43720.7	43718	3.67	5845
		TVV	43726	43725	43722	1.93	2075

Table 4: Comparison results of weakly correlated high-dimensional size 0–1 KP

Instance	Dimension	Transfer function	Best	Mean	Worst	SD	Mean iterations
kp-26	100	S	2015	2012.3	2009	4.32	931
		V	2015	2015	2015	0	53
		TVS	2015	2015	2015	0	357
		TVV	2015	2015	2015	0	30
kp-27	500	S	10450	10447.2	10446	6.52	1859
		V	10450	10449.3	10448	0.84	103
		TVS	10450	10447.8	10446	1.71	856
		TVV	10450	10450	10450	0	54
kp-28	1000	S	20856	20852.1	20849	6.85	3463
		V	20856	20854.6	20853	0.99	220
		TVS	20856	20854	20852	2.31	2856
		TVV	20856	20856	20856	0	142

kp-29	1500	S	31625	31620.3	31618	8.02	4188
		V	31630	31629.5	31626	1.85	1962
		TVS	31632	31628.7	31626	2.94	3278
		TVV	31632	31631.4	31631	0.26	900
kp-30	2000	S	42050	42046	42041	10.93	7799
		V	42055	42053.1	42049	2.04	2541
		TVS	42057	42051.6	42047	7.32	4587
		TVV	42057	42056	42054	1.34	1062

Table 5: Comparison results of strongly correlated high-dimensional size 0–1 KP

Instance	Dimension	Transfer function	Best	Mean	Worst	SD	Mean iterations
kp-31	100	S	2669	2669	2669	0	287
		V	2669	2669	2669	0	40
		TVS	2669	2669	2669	0	111
		TVV	2669	2669	2669	0	16
kp-32	500	S	13657	13654.1	13652	0.67	568
		V	13657	13657	13657	0	54
		TVS	13657	13657	13657	0	326
		TVV	13657	13657	13657	0	32
kp-33	1000	S	27164	27162.5	27159	1.29	925
		V	27166	27164.6	27164	0.90	131
		TVS	27166	27164.4	27162	0.22	623
		TVV	27166	27166	27166	0	92
kp-34	1500	S	40461	40459.8	40455	2.58	1770
		V	40466	40465	40463	1.61	297
		TVS	40468	40466.3	40464	1.58	839
		TVV	40468	40468	40468	0	150
kp-35	2000	S	42050	42048.9	42042	3.01	3909
		V	42054	42053.1	42048	2.18	563
		TVS	42057	42055	42051	2.33	2380
		TVV	42057	42056.2	42055	0.89	385

5. Conclusion

In this paper, a time-varying transfer function was proposed to improve the exploration and exploitation capability of the binary POA 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 binary POA in solving small, medium, and high-dimensional sizes 0–1 KP problems. Additionally, the experimental results show that proposed time-varying V-shaped transfer function outperforms the S-shaped transfer function in terms of the best, worse, mean, SD, and the mean iterations.

References

1. Abdel-Basset, M., D. El-Shahat, and I. El-Henawy, Solving 0–1 knapsack problem by binary flower pollination algorithm. Neural Computing and Applications, 2018.
2. Yang, X.-S., Cuckoo Search and Firefly Algorithm_ Theory and Applications Springer International Publishing Switzerland, 2014.
3. Zhao, J.F., et al., Genetic Algorithm Bassed on Greedy Strategy in the 0-1 Knapsack problem. Third international conference on genetic and evolutionary computing, 2009.
4. Beheshti, Z., S.M. Shamsuddin, and S.S. Yuhaniz, Binary Accelerated Particle Swarm Algorithm (BAPSA) for discrete optimization problems. Journal of Global Optimization, 2012. **57**(2): p. 549-573.
5. Azad, M.A.K., A.M.A.C. Rocha, and E.M.G.P. Fernandes, A simplified binary artificial fish swarm algorithm for 0–1 quadratic knapsack problems. Journal of Computational and Applied Mathematics, 2014. **259**: p. 897-904.
6. Kong, X., et al., A simplified binary harmony search algorithm for large scale 0–1 knapsack problems. Expert Systems with Applications, 2015. **42**(12): p. 5337-5355.
7. Tuo, S., L. Yong, and F. Deng, A novel harmony search algorithm based on teaching-learning strategies for 0-1 knapsack problems. ScientificWorldJournal, 2014. **2014**: p. 637412.
8. Sajedi, H. and S.F. Razavi, DGSA: discrete gravitational search algorithm for solving knapsack problem. Operational Research, 2016. **17**(2): p. 563-591.
9. Feng, Y., H. An, and X. Gao, The Importance of Transfer Function in Solving Set-Union Knapsack Problem Based on Discrete Moth Search Algorithm. Mathematics, 2018. **7**(1): p. 17.
10. Feng, Y., K. Jia, and Y. He, An improved hybrid encoding cuckoo search algorithm for 0-1 knapsack problems. Comput Intell Neurosci, 2014. **2014**: p. 970456.
11. Feng, Y., et al., An effective hybrid cuckoo search algorithm with improved shuffled frog leaping algorithm for 0-1 knapsack problems. Comput Intell Neurosci, 2014. **2014**: p. 857254.
12. Feng, Y., G.-G. Wang, and L. Wang, Solving randomized time-varying knapsack problems by a novel global firefly algorithm. Engineering with Computers, 2017. **34**(3): p. 621-635.
13. He, Y., et al., A novel binary artificial bee colony algorithm for the set-union knapsack problem. Future Generation Computer Systems, 2018. **78**: p. 77-86.
14. Rizk-Allah, R.M. and A.E. Hassanien, New binary bat algorithm for solving 0–1 knapsack problem. Complex & Intelligent Systems, 2017. **4**(1): p. 31-53.
15. Zhou, Y., L. Li, and M. Ma, A Complex-valued Encoding Bat Algorithm for Solving 0–1 Knapsack Problem. Neural Processing Letters, 2015. **44**(2): p. 407-430.
16. Wang, G.-G., S. Deb, and Z. Cui, Monarch butterfly optimization. Neural Computing and Applications, 2015.
17. Duan, H. and P. Qiao, Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning. International Journal of Intelligent Computing and Cybernetics, 2014. **7**(1): p. 24-37.
18. Fu, X., et al., A multi-objective pigeon inspired optimization algorithm for fuzzy production scheduling problem considering mould maintenance. Science China Information Sciences, 2019. **62**(7).
19. Qiu, H. and H. Duan, A multi-objective pigeon-inspired optimization approach to UAV distributed flocking among obstacles. Information Sciences, 2020. **509**: p. 515-529.

20. Sushnigdha, G. and A. Joshi, Re-entry trajectory optimization using pigeon inspired optimization based control profiles. Advances in Space Research, 2018. **62**(11): p. 3170-3186.
21. Yan, L., et al., Dynamic economic emission dispatch based on multi-objective pigeon-inspired optimization with double disturbance. Science China Information Sciences, 2019. **62**(7).
22. Yang, Z., et al., Automatic Carrier Landing System multilayer parameter design based on Cauchy Mutation Pigeon-Inspired Optimization. Aerospace Science and Technology, 2018. **79**: p. 518-530.
23. Zhong, Y., et al., Discrete pigeon-inspired optimization algorithm with Metropolis acceptance criterion for large-scale traveling salesman problem. Swarm and Evolutionary Computation, 2019. **48**: p. 134-144.
24. Islam, M.J., X. Li, and Y. Mei, A time-varying transfer function for balancing the exploration and exploitation ability of a binary PSO. Applied Soft Computing, 2017. **59**: p. 182-196.
25. Cao, J., et al., A modified artificial bee colony approach for the 0-1 knapsack problem. Applied Intelligence, 2017. **48**(6): p. 1582-1595.
26. Abdel-Basset, M., D. El-Shahat, and A.K. Sangaiah, A modified nature inspired meta-heuristic whale optimization algorithm for solving 0–1 knapsack problem. International Journal of Machine Learning and Cybernetics, 2017.
27. Alazzam, H., A. Sharieh, and K.E. Sabri, A feature selection algorithm for intrusion detection system based on Pigeon Inspired Optimizer. Expert Systems with Applications, 2020. **148**: p. 113249.
28. Mafarja, M., et al., Binary dragonfly optimization for feature selection using time-varying transfer functions. Knowledge-Based Systems, 2018. **161**: p. 185-204.
29. Mirjalili, S. and A. Lewis, S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization. Swarm and Evolutionary Computation, 2013. **9**: p. 1-14.
30. Teng, X., H. Dong, and X. Zhou, Adaptive feature selection using v-shaped binary particle swarm optimization. PLoS One, 2017. **12**(3): p. e0173907.
31. Tilahun, S.L. and J.M.T. Ngnotchouye, Firefly algorithm for discrete optimization problems: A survey. KSCE Journal of Civil Engineering, 2017. **21**(2): p. 535-545.
32. Bhattacharjee, K.K. and S.P. Sarmah, Modified swarm intelligence based techniques for the knapsack problem. Applied Intelligence, 2016. **46**(1): p. 158-179.