A Modified Binary Pigeon-Inspired Algorithm for Solving the Multi-dimensional Knapsack Problem

Article *in* Journal of Intelligent Systems · January 2021

DOI: 10.1515/jisys-2018-0450

Asaju La'aro Bolaji
Federal University Wukari
42 PUBLICATIONS **643** CITATIONS
SEE PROFILE

Okwonu Friday Zinzendoff
Universiti Utara Malaysia
16 PUBLICATIONS **20** CITATIONS
SEE PROFILE

Peter Shola
University of Ilorin
12 PUBLICATIONS **65** CITATIONS
SEE PROFILE

Obinna Adubisi
Federal University Wukari
21 PUBLICATIONS **24** CITATIONS
SEE PROFILE

Research Article

Asaju La’aro Bolaji*, Friday Zinzendoff Okwonu, Peter Bamidele Shola, Babatunde Sulaiman Balogun, and Obinna Damian Adubisi

A Modified Binary Pigeon-Inspired Algorithm for Solving the Multi-dimensional Knapsack Problem

https://doi.org/???
Received ; accepted

Abstract: The pigeon-inspired optimization algorithm is a category of a newly proposed swarm intelligence-based algorithm that belongs to the population-based solution technique. The MKP is a class of complex optimization problems that have many practical applications in the fields of engineering and sciences. Due to the practical applications of MKP, numerous algorithmic-based methods like local search and population-based search algorithms have been proposed to solve the MKP in the past few decades. This paper proposes a modified binary pigeon-inspired optimization algorithm named (Modified-BPIO) for the 0 - 1 multidimensional knapsack problem (MKP). The utilization of the binary pigeon-inspired optimization (BPIO) for solving the multidimensional knapsack problem came with huge success. However, it can be observed that the BPIO converges prematurely due to lost diversity during the search activities. Given the above, the crossover operator is integrated with the landmark component of the BPIO to improve the diversity of the solution space. The MKP benchmarks from the Operations Research (OR) library are utilized to test the performance of the proposed binary method. Experimentally, it is concluded that the proposed Modified-BPIO has a better performance when compared with the BPIO and existing state-of-the-arts that worked on the same MKP benchmarks.

Keywords: Pigeon-inspired optimization, Multidimensional Knapsack problem, Crossover mechanism, Nature-inspired algorithm, Population-based metaheuristic

1 Introduction

Complex optimization is a class of problems that cut across different disciplines such as engineering design, production and manufacturing systems, economics and so on. Due to their practical applicability, numerous efficient and robust computational and soft computing techniques have been proposed to solve different classes of these problems. One of the classical examples of complex optimization problems that has numerous applications in different areas is the multidimensional knapsack problem (MKP). Basically, the practical applications of MKP can be found in cryptography [10, 21, 24], warehouse location problem, production scheduling problem, assignment problem, and reliability problem [23]. In the optimization context, the MKP

***Corresponding Author: Asaju La’aro Bolaji:** Department of Computer Sciences, Federal University Wukari, Wukari, Taraba State, Nigeria; Email: lbasaju@fuwukari.edu.ng
Friday Zinzendoff Okwonu: School of Quantitative Sciences, College of Arts and Sciences, Universiti Utara Malaysia, Sintok, Kedah, Malaysia; Email: o.friday.zinzendoff@uum.edu.my
Peter Bamidele Shola: Department of Computer Science, University of Ilorin, Ilorin, Nigeria; Email: shola.bp@unilorin.edu.ng
Babatunde Sulaiman Balogun: Department of Computer Science, University of Ilorin, Ilorin, Nigeria; Email: aceslimz@gmail.com
Obinna Damian Adubisi: Department of Mathematics and Statistics, Federal University Wukari, Wukari, Taraba State, Nigeria; Email: adubisiobinna@fuwukari.edu.ng

as a resource allocation model focused on selecting a subset of objects that generate the maximum profit, whilst satisfying the set of constraints on capacities of the knapsack. Generating efficient techniques for the MKP as one of the categories of the NP-hard problem has been subject of investigation over the past few decades by researchers in the fields of artificial intelligence and operational research. The complexity of the MKP motivated the workers in the domain to proposed several techniques that are classified into traditional-based and metaheuristic-based techniques. The traditional-based technique could be utilized to generates the exact solution for the MKP, however, its time complexity increases exponentially with the size of the problem and thus could exhibit some weaknesses when utilized for large-scale problems of high dimensionality. The common examples of the traditional-based techniques employed to tackle the MKP are branch and bound [30], dynamic programming [33] and so on. Whereas the metaheuristic-based method could be employed to produce a near-optimal solution within reasonable computational times when compared to the traditional-based method and thus makes the metaheuristic-based method to be more suitable when utilized to handle large-scale problems. Typically, this method could be classified into two: local search-based methods that are designed to handle a single solution at a time and some examples of local search-based methods utilized for the MKP are simulated annealing [11], tabu search [17]; [39]. Similarly, the population-based methods, developed to tackle many solutions at a time are categorized into Evolutionary-based Algorithms (EAs) and Swarm Intelligence-based (SI) algorithms. Few of examples of the population-based algorithms that have been successfully utilized, modified and hybridized to solve different combinatorial optimization and engineering applications like MKP are ant algorithms [18], artificial bee colony algorithm [31], bat algorithm [45], cuckoo search algorithm [13], flower pollination algorithm [1], fruit fly optimization algorithm [26], monkey algorithm [44], differential evolution algorithm [32], genetic algorithm [28], harmony search algorithm [20]. particle swarm optimization [4, 8, 9, 15], symbiotic organisms search algorithm [38], wind-driven optimization [43]. It is worthy of mentioning that from the literature, numerous swarm-intelligence algorithmic techniques have successfully been proposed, modified and hybridized to tackled the different formulations of 0 - 1 multidimensional knapsack problem. Summarily, the SI - based algorithms have the following advantages which include: simplicity, highly adaptable, highly scalable, self-organized, flexible and collective robustness. Due to these advantages, many studies have proposed their usage in solving complex problems in the field of artificial intelligence and operational research [5, 6, 29, 34–36]. However, few limitations trace to SI - Based algorithms are time-critical applications, parameter tuning, premature convergence and stagnation in local optimum [2].

The pigeon-inspired optimization (PIO) algorithm is a category of swarm intelligence algorithm that is newly introduced to tackle the air robot path planning proposed in 2014 by Duan and Qiao [12]. It is a stochastic nature-inspired technique that has proven to be an easy but powerful population-based search technique inspired by the behavior of a swarm of pigeons. Homing pigeons can easily locate their homes in accordance with three homing operators: magnetic field, sun, and landmarks. In the context of this algorithm, a map and compass model is formulated according to the magnetic field and sun, while the presentation .of the landmark operator model is done based on landmarks [40]. Note that pigeons perceived the magnetic field through the nose from the magnetic particles taken to the brain by trigeminal nerve [25]. It is observed that the pigeons probably employ the usage of different navigational tools during diverse parts of their journey. At the initial stage of the journey, each pigeon could possibly depend more on compass-like tools, while in the middle and sometimes switch to landmarks in order to reexamine their routes and thus makes amendments [14]. Note that the PIO algorithm has been successfully utilized to tackle many complex optimization problems like orbital space-craft formation reconfiguration [41], unmanned aerial vehicles [16, 42]. Studies have proven that the PIO algorithm had a better or comparable performance with some of the existing nature-inspired algorithms like the artificial bee colony, genetic algorithm, particle swarm optimization [27]. In recent time, the binary version of the classical PIO is proposed to tackle the MKP in [7] where its performance is better than some of the existing methods. However, from the experimental results, it can be observed that the BPIO converges prematurely due to lost diversity during the search activities.

The main motivation of this paper is to modify the BPIO algorithm with integration of simple crossover element of evolutionary algorithm to improve the performance. It is worthy to mention that the contribution of this research is in two folds which include:

– Modification of Landmark operator of the BPIO to improve the diversity of the search space and exploring the interaction between pigeons to navigate the most promising regions of the search space effectively.
– " The performance of the proposed modified is evaluated on the standard MKP benchmark datasets published by Operations Research Library (OR-Library).

Therefore, this paper presents a modified version binary pigeon-inspired optimization (Modified-BPIO) algorithm in which its landmark operator is modifies by the incorporation of crossover concept from the evolutionary algorithm when utilized to solve the 0-1 multidimensional knapsack problem. The performance of the modified-BPIO is evaluated on the standard MKP benchmark datasets published by Operations Research Library (OR-Library). Experimentally, the computational results show that the performance of the modified-BPIO is better when compared with some of the existing nature-inspired algorithms that worked on the same dataset.

The rest of the paper is organized as follows: Section 2 presents the description of the MKP, while the description of the basic PIO is given in Section 3, this is followed by the proposed binary approach as discusses with detailed information in Section 4. Next, the experimental results and discussions are given in Section 5. Finally, conclusions are considered in Section 6.

2 Multidimensional Knapsack problem (MKP) formulation

The MKP consists of a set of m knapsacks with a set of m capacities $\mathfrak{c} = \{c_i | i = 0, \ldots, m-1\}$, and a set of n entities $\mathfrak{e} = \{e_i | i = 0, \ldots, n-1\}$. The binary variables $X_i (i = 0, \ldots, n-1)$ correspond to the chosen items to be carried in m knapsacks. The X_i assumes value of 1 if entity i is in the knapsack and 0 otherwise. Each item e_i has an associated profit $P_i \geq 0$ and weight $W_i j \geq 0$ for each knapsack j. The goal is to find the best combination of n entities by maximizing the sum of profits P_i multiplied by the binary variable X_i, which is mathematically represented as shown in Eq. (1).

$$\max \left(\sum_{i=0}^{n-1} (P_i \times X_i) \right) \tag{1}$$

Their constraints are the capacity $C_j \geq 0$ of each knapsack. Therefore, the sum of the values of X_i multiplied by W_{ij} must be less than or equal to C_j as given in Eq. (2). Note that this formulation is adopted from [3].

$$\sum_{j=0}^{m-1} \left(W_{ij} \times X_i \right) \leq C_j \tag{2}$$

3 Pigeon-Inspired Optimization Algorithm

PIO is a novel nature-inspired, metaheuristic algorithm that has been utilized for solving global optimization problems. It is based on imitating the natural homing pigeon behavior. Studies on the behavior of pigeon in the recent time, have shown that the pigeon can follow their path using some landmark features such as like main terrains, railways, and rivers rather than move directly for their destination. In the optimization context, the migration of pigeons can be formulated using the two mathematical models: map and compass operator, and landmark operator.

3.1 Map and Compass Operator

Basically, the concept of map and compass operator is to assist pigeons to find their position as well as determine their direction. Analogously, in the map and compass operator, rules are formulated to determine which position X_i and the velocity V_i of pigeon *i*, and the positions and velocities in a D-dimension search space are updated in each iteration. The formulations of the new position X_i and velocity V_i of pigeon *i* at the t^{th} iteration are given in Eqs. 3 and 4 as follows:

$$V_i(t) = V_i(t-1) \cdot e^{-rt} + rand \cdot \left(X_g - X_i(t-1)\right) \tag{3}$$

$$X_i(t) = X_i(t-1) + V_i(t) \tag{4}$$

where *r* represent the map and compass factor, *rand* is a random number, and X_g represent the current global best position that is obtained by comparing all the positions among all the pigeons.

3.2 Landmark Operator

In the landmark operator, the total number of pigeons is reduced to half N_p in every generation, where pigeons in the lower half of the line sorted by fitness values are abandoned. This is due to the fact that they are believed to be far from the destination and unfamiliar with the landmarks. Let $X_c(t)$ represent the center of some pigeons' position at the t^{th} iteration, and that every pigeon can fly straight to the destination. The position updating rule for pigeon *i* at t^{th} iteration can be formulated as shown in Eq. 5:

$$N_p(t) = \frac{N_p(t-1)}{2} \tag{5}$$

$$X_c(t) = \frac{\sum X_i(t) \cdot fitness(X_i(t))}{N_p \sum fitness(X_i(t))} \tag{6}$$

$$X_i(t) = X_i(t-1) + rand \cdot (X_c(t) - X_i(t-1)) \tag{7}$$

where *fitness* is the object value (i.e. quality) of the pigeon individual. Note that for the minimization problems, then the fitness cost is given as shown in Eq 8:

$$f(X_i(t)) = \frac{1}{f(X_i(t)) + \varepsilon} \tag{8}$$

Whereas the fitness cost for the maximization problem is given in Eq. 9

$$f(X_i(t)) = f(X_i(t)) \tag{9}$$

The key phases of the PIO algorithm as proposed in 2014 by Duan and Qiao [12] is given in Algorithm 1:

4 Modified-BPIO Optimization for the MKP

In this section, the proposed Modified-BPIO version that is based on the integration of a crossover strategy within the landmark operation of the BPIO for solving the MKP is presented. The solution approach utilized in order to achieve the aim of this paper is presented in the next subsections:

Algorithm 1 PIO algorithm

```
1: Initialization of PIO parameters: $P_N$, D, $t_{1max}$, $t_{2max}$.
2: Initialize the path and the velocity of each pigeon. {Initialized the population of pigeons}
3: Calculate the fitness cost of each pigeon and compare the individual best fitness costs
   of all the pigeons to obtain the global best path {Evaluate each pigeon}
4: **while** $t \le t1max$ **do**
5:     Operate the map and compass operator. Update the velocities and paths of each
       pigeon, and update the globally best path.
6: **end while**
7: **while** $t \le t2max$ **do**
8:     Operate the landmark operator. Sort all pigeons according to their fitness costs. Those pigeons with
       low fitness costs will follow those with high fitness cost using Eq. (5). Then determine the center of all
       pigeons based on Eq. (6), and this center is the desired destination. Adjust all pigeons fly directions in
       accordance with the Eq. (7).
9:     Memorize the best pigeon parameters and the best fitness cost.
10: **end while**
```

4.1 Initialization of the PIO and MKP parameters

In this step, the control parameters of Modified-BPIO which are necessary to tackle the MKP are initialized. These parameters include the population size, map and compass factor the maximum number of iteration etc.

- Population size: the number of pigeons (or solutions) in the population.
- Map and compass factor
- Maximum number of iteration which it refers to as the total number of generations

The MKP parameters such as the number of knapsacks, the capacities, the set of items and the constraints are extracted from the problem instances. The fitness functions stated in Eqs. 1 and 2 are utilized to evaluate each pigeon during the search process of the Modified-BPIO.

4.2 Initialization of the Pigeons Memory (PM)

The Pigeon Memory (PM) is referred to as the allocation of memory space of size where each row consists of a solution vector representing an MKP solution as in (10).

$$\mathbf{PM} = \begin{bmatrix} x_1(1) & x_1(2) & \cdots & x_1(N) \\ x_2(1) & x_2(2) & \cdots & x_2(N) \\ \vdots & \vdots & \ddots & \vdots \\ x_{PN}(1) & x_{PN}(2) & \cdots & x_{PN}(N) \end{bmatrix} \begin{bmatrix} f(\boldsymbol{x}_1) \\ f(\boldsymbol{x}_2) \\ \vdots \\ f(\boldsymbol{x}_{PN}) \end{bmatrix}. \tag{10}$$

Note that for the problem under consideration, the pigeons in PM are generated using a 0-1 binary representation which is an obvious choice since it represents the underlying 0-1 integer variables. Therefore, this study utilizes a *n*-bit binary string representation, where *n* is the number of variables in the MKP, a value of 0 or 1 at the i^{th} path means that or 1 in the MKP solution, respectively. Figure 1 shows the binary representation of each pigeon (i.e. solution) in the search space for the MKP.

i	*0*	*1*	*2*	*3*	*4*	....	*n-1*
Y_i	*0*	*1*	*0*	*1*	*1*	....	*0*

Figure 1: MKP solution representation

It is worthy of mentioning that in some case the solution (i.e. pigeon) generated based on the above procedure may not be feasible. Thus in order to avoid the utilization of such solutions in the population, a penalty function method is employed. Therefore, the fitness cost function is given in Eq. (11)

$$fitness(S) = \sum_{i=1}^{n} p_i s[i] \times Pen[i] \tag{11}$$

Where *Pen* is a penalty cost for an infeasible pigeon.

Accordingly, a population of pigeon (i.e. the population of solution) is initialized randomly with their paths and velocities. Then, fitness cost of each pigeon are evaluated and sorted in ascending order in PM based to their fitness costs (i.e. $f(x_1) \geq f(x_2) \geq \ldots \geq f(x_{PN})$) to obtain the global best path.

4.3 Activate the Map and Compass Component

In this step, each pigeon searches through the solution space by altering its flight based on its own personal best position or by moving towards the direction of the global best path as determined by the evaluation of the paths of all pigeons based on this component. The process of exploring the solution space is repeated until the maximum iterations of the map and compass component is achieved.

4.4 Activate the Landmark Component

The center of the pigeon is determined in this step which is achieved by raking the pigeons based on their fitness costs and those pigeons with the lowest fitness costs will move towards those highest fitness costs. However, the concept of moving low-quality pigeons towards those of high quality is modified in this paper where the crossover concept from the Evolutionary Algorithm is employed to diversify the solution space and thus prevent the premature convergence. The procedure of integrating the crossover concept is provided as follows:

4.4.1 Integration of Crossover Component

Crossover operator is one of the typical component of the Evolutionary Algorithm (EA) which solely aimed at introducing some element of diversity through the generation of new members (i.e. individuals) into the population. Basically, the main operation of crossover involves the combination of the traits of two entities (i.e. parents) in order to produce one or two new entities (offspring). In the Modified-BPIO, the crossover operator is embedded into the landmark component immediately after the centre of the pigeons have been determined. In this step, two pigeons are randomly selected from those with good qualities and those pigeons of lowest fitness costs. Then, the elements from the good pigeon are randomly chosen and interchanged with the corresponding elements from those with the lowest fitness cost and placed in the same order in the second pigeon. Then, the capacities of the knapsacks are evaluated and if it is available for both pigeons, otherwise, if any capacity of knapsack exceeds by adding a new element from the other parent pigeon, the value of this element is changed zero. The capacity of the new pigeon is completed from the remaining elements that are not included by any knapsack based on their profit which is provided by including in the new pigeon. The new pigeons are accepted into the swarm if their fitness cost is better or equal to the pigeon with lowest fitness cost and thus the search is diversified. The best path and the fitness costs are stored. The search process in this phase is repeated until the maximum iterations of the landmark component is reached.

4.5 Memorize the best pigeon in the population

In this step, the position of the best pigeon, the velocity and the best fitness cost are memorized

4.6 Stopping Condition

Steps 4.2 to 4.4 are repeated until a stop criterion is met. This is originally determined by the maximum iterations value.

5 Time complexity of the proposed Modified BPIO algorithm

The time complexity of the proposed BPIO algorithm is calculated based on section 4 as itemized in Table 1. Note that the computational time required to calculate the objective function is neglected because it is different from one problem to another. As shown in Table 1, the time complexity of the proposed Modified BPIO algorithm is $O(MIN \times C_R \times MC_R \times d)$. TThe step that requires a large time complexity in the algorithm is Step 4.3 that takes more computational time

Table 1: Time complexity of each step in the proposed Modified BPIO

No	Step Number	Time Complexity
1	Step 4,2: Initialize the BPIO population	$O(d \times P_N)$
2	Step 4.3: Activate the Map and Compass Component	$O(P_N \times MC_R \times t_1 max)$
3	Step 4.4: Activate the Landmark Component with crossover integration	$O(C_R \times P_N \times t_2 max)$
4	Step 4.5 Memorize the position of the best pigeon	$O(P_N)$
5	Step 4.5: Stopping Condition	$O(MIN \times C_R \times MC_R \times d)$

6 Computational Experiments, Results, and Discussions

In this section, the performance of the proposed Modified-BPIO for solving multidimensional knapsack problem is presented and it is coded using Visual Basic.net and run on a Personal Computer with Intel Core i3 1.9 GHz, 4 GB memory running on Windows 10. The MKP benchmarks obtained from the OR library that is utilized to evaluate the proposed Modified-BPIO are one small and two big data sets namely: MKNAP 1, MKNAPCB 1 and MKNAPCB 4. Note that seven instances from a small dataset and ten problem instances from both big datasets are utilized in the evaluation. The characteristics of these datasets are provided in Tables 3 and 2. As shown in Table 1, the first row shows the problem index, while the second row indicates the size of the problem i.e. number of objects and lastly, the third row represents the number of knapsack dimensions.

Table 2: Characteristics of MKNAP 1 problem instances

Instance S_p	1	2	3	4	5	6	7
n	6	10	15	20	28	39	50
M	10	10	10	10	10	5	5

Similarly, the characteristic of the problem instances MKNNAPCB 1 and MKNNAPCB 4 are provided in Table 3, where column 1 and 2 represents the instance name and the problem size for the MKNNAPCB 1 respectively while column 3 and 4 shows the instance name and the problem size for the MKNNAPCB 4 respectively

Table 3: Characteristics of MKNAPCB 1 and MKNAPCB 4 instances

PROBLEM INSTANCE	PROBLEM SIZE	PROBLEM INSTANCE	PROBLEM SIZE
MKNAPCB 1	5.100.01	MKNNAPCB 4	10.100.01
	5.100.02		10.100.02
	5.100.03		10.100.03
	5.100.04		10.100.04
	5.100.05		10.100.05
	5.100.06		10.100.06
	5.100.07		10.100.07
	5.100.08		10.100.08
	5.100.09		10.100.09
	5.100.10		10.100.10

6.1 Experimental Design

The computational experiments are designed to study the influence of integrating the crossover component to the landmark operator in the Modified-BPIO, it is worthy of mentioning that three varying values crossover parameter are employed in the experiments. The parameter settings were chosen carefully based on our preliminary experiments over multidimensional knapsack problems. The settings of these parameters should not be considered as the optimal set of values but a generalised set of values since the performance of the proposed algorithm is fairly well over the test problems. The detailed parameter settings for the Modified-BPIO such as using three convergence scenario is provided in Table 4. Similarly, other values of the remaining parameters are adopted from [7]. For instance, the population size (P_N), map and compass operator rate (MC_R) and the maximum iterations number (MIN) are fixed at 250, 0.5 and 1500 respectively.

Table 4: Modified BPIO Parameter Settings

Case	Population Size (P_N)	Map and Compass Rate (MC_R)	Crossover rate (C_R)
Case 1	250	0.5	0.1
Case 2	250	0.5	0.2
Case 3	250	0.5	0.3

6.2 Experimental results

In this section, experimental results of the Modified-BPIO using the three experimental scenarios are provided in Table 5, 6 and 7, where the values in the fourth to sixth columns of these tables represent the fitness values of 10 runs (highest is best). For each experimental scenario, the best results and mean of each problem instance over 10 runs are provided. The best results obtained by the proposed methods for each instance of the three datasets are highlighted in bold. As shown in Table 4, the three experimental scenarios of the Modified-BPIO achieved equal results in virtually all the instances of the smallest dataset except in instance 7, where Case 3 obtained the best results.

Similarly, as presented in Table 6 and 7, the performance of the proposed method for each experimental scenarios for the two hard datasets show that Case 2 with a crossover rate of 0.2 achieved best results in six instances of the MKNAPCB 1 dataset while Case 3 with crossover rate 0.3 came 2 by obtained best results in the remaining 4 instance. Finally, on MKNAPCB 4 dataset, Case 2 of the Modified-BPIO achieved best results in 5 instances of the dataset, while Case 3 and 1 obtained best results in 4 and 2 instances of the dataset respectively.

Table 5: Experimental results on effect of Crossover Rate on MKNAP 1 instances

Instance	Instance		Case 1	Case 2	Case 3
	1	Best	**3,800**	**3,800**	**3,800**
		Mean	3,800	3,800	3,800
	2	Best	**8,706.1**	**8,706.1**	**8,706.1**
		Mean	8,706.1	8,706.1	8,706.1
	3	Best	**4,015**	**4,015**	**4,015**
MKNAP 1		Mean	4,015	4,015	4,015
	4	Best	**6,120**	**6,120**	**6,120**
		Mean	6,120	6,120	6,120
	5	Best	**12,400**	**12,400**	**12,400**
		Mean	12,388	12,398	12,390
	6	Best	**10,604**	**10,604**	**10,604**
		Mean	10,580.4	10,594.4	10,559.2
	7	Best	16,508	16,508	**16,518**
		Mean	16,462.4	16,429	16,432.4

Table 6: Experimental results on effect of Crossover Rate on MKNAPCP 4 instances

Dataset	Instance		Case 1	Case 2	Case 3
MKNAPCB 1	5.100.01	Best	23,530	23,624	23,551
		Mean	23,198	23,441.2	23,391.2
	5.100.02	Best	23,322	23,642	23,317
		Mean	23,112	23,431.4	23,046.8
	5.100.03	Best	22,603	23,128	22,987
		Mean	22,388.4	22,690.2	22,627.4
	5.100.04	Best	22,718	22,951	23,312
		Mean	22,491.8	22,698.2	22,701.2
MKNAPCB 1	5.100.05	Best	23,332	23,512	23,207
		Mean	23,050.4	23,135.8	22,873.2
	5.100.06	Best	23,570	24,026	23,797
		Mean	23,367.4	23,630.6	23,544.8
	5.100.07	Best	24,804	24,796	25,016
		Mean	24,470.2	24,628.8	24,551.4
	5.100.08	Best	22,703	22,552	22,915
		Mean	22,448.2	22,307	22,485.2
	5.100.09	Best	23,614	23,648	23,590
		Mean	23,209.8	23,279.4	23,412.8
	5.100.10	Best	23,887	23,806	24,031
		Mean	23,631	23,524.4	23,594.2

Table 7: Experimental results on effect of Crossover Rate on MKNAPCP 4 instances

Dataset	Instance		Case 1	Case 2	Case 3
	10.100.01	Best	22,611	22,728	22,430
		Mean	22,105.8	22,301.8	22,157
	10.100.02	Best	21,593	22,040	21,754
		Mean	21,346.8	21,644.8	21,620.8
	10.100.03	Best	21,436	21,436	21,339
		Mean	21,035.6	21,309	21,041.8
	10.100.04	Best	22,226	22,313	22,325
		Mean	21,886.6	22,133.6	21,741.2
MKNAPCP 4	10.100.05	Best	22,113	21,840	21,973
		Mean	21,757.8	21,513.4	21,811.4
	10.100.06	Best	21,733	22,046	21,734
		Mean	21,608.8	21,877	21,432.6
	10.100.07	Best	21,353	21,465	20,918
		Mean	20,887.2	20,823.6	20,589.4
	10.100.08	Best	21,666	21,734	21,915
		Mean	21,496.8	21,450.6	21,582.8
	10.100.09	Best	22,239	21,737	21,897
		Mean	21,625.6	21,288.6	21,235.6
	10.100.10	Best	21,672	21,775	21,806
		Mean	21,382.2	21,647.4	21,567.4

7 Comparative Analysis with Existing Techniques

The results produced by the proposed Modified-BPIO is compared with other existing techniques that worked on the same MKP benchmark, which are Binary cuckoo search algorithm BCS [13], Standard binary particle swarm optimization with penalty function PSO-P [19], quantum inspired cuckoo search QICSA [22] and a binary pigeon-inspired optimization algorithm Binary-PIO [7]. The performance of the Modified-BPIO is comparable to the other existing techniques in the small instance of the MKP benchmark. Virtually, all the methods obtained optimal or near-optimal values for all the instances of the MKNAP 1 as provided in Table 8.

Table 8: The best results achieved by the Modified-BPIO and other comparative methods

Dataset	Instance	Best Known	BCS	PSO-P	QICSA	BPIO	Modified- BPIO
	1	3,800	3,800	3,800	3,800	3,800	3,800
	2	8,706.1	8,706.1	8,706.1	8,706.1	8,706.1	8,706.1
	3	4,015	4,015	4,015	4,015	4,015	4,015
MKNAP 1	4	6,120	6,120	6,120	6,120	6,120	6,120
	5	12,400	12,400	12,400	12,400	12,400	12,400
	6	10,618	10,618	10,618	10,618	10,604	10,604
	7	16,537	16,537	16,537	16,537	16,508	16,518

Furthermore, Tables 9 shows the results of the proposed Modified-BPIO in comparison with these techniques from the literature. As shown in Table 9, the best results are highlighted in bold. Apparently, the Modified-PIO is able to obtain feasible solutions for all problem instances of both datasets. Furthermore, the proposed technique obtained best results in nine out of ten instances of the MKNPACB 1 dataset when compared with previous methods, however, the BPIO achieved the best result in the remaining one instance (i.e. in MKNAPCB 1 5.100.10). Similarly, the proposed Modified-BPIO achieved the first rank in seven out of 10 instances of the MKNAPCB 4 dataset while the BPIO came second in the remaining problem instances. This shows that modification of the landmark component of the BPIO aided the algorithm to navigate the solution space rigorously to obtain a good solution.

Table 9: The best results achieved by the Modified-BPIO and other comparative methods

Dataset	Instance	Best Known	BCS	PSO-P	QICSA	BPIO	Modified BPIO
	5.100.01	24,381	23,510	22,525	23,416	23,494	23,624
	5.100.02	24,274	22,938	22,244	22,880	23,227	23,642
	5.100.03	23,551	22,518	21,822	22,525	22,942	23,128
	5.100.04	23,534	22,677	22,057	22,727	22,895	23,312
MKNAPCP 1	5.100.05	23,991	23,232	22,167	22,854	23,502	23,512
	5.100.06	24,613	-	-	-	23,725	24,026
	5.100.07	25,591	-	-	-	24,746	25,016
	5.100.08	23,410	-	-	-	22,717	22,915
	5.100.09	24,216	-	-	-	23,566	23,648
	5.100.10	24,411	-	-	-	24,082	24,031
	10.100.01	23,064	21,841	20,895	21,796	22,237	22,728
	10.100.02	22,801	21,708	20,663	21,348	22,203	22,040
	10.100.03	22,131	20,945	20,058	20,961	21,614	21,436
	10.100.04	22,772	21,395	20,908	21,377	22,236	22,325
MKNAPCP 4	10.100.05	22,751	21,453	20,488	21,251	22,157	21,973
	10.100.06	22,777	-	-	-	21,304	22,046
	10.100.07	21,875	-	-	-	21,813	21,465
	10.100.08	22,635	-	-	-	21,644	21,915
	10.100.09	22,511	-	-	-	22,061	22,239
	10.100.10	22,702	-	-	-	21,806	22,125

To show the performance of the proposed Modified-BPIO against the BPIO, a Wilcoxon Signed Rank is presented in Table 10 where the R+, R-, and p-value are computed for all the pairwise comparisons concerning Modified-BPIO. As shown in Table 10, the Modified-BPIO shows a significant improvement over BPIO, with a level of significance less than $\alpha = 0.01$ on MKNAPCP 1 instance while their performance is relatively the same on MKNAPCP 4. Note that the two algorithms (i.e. Modified-BPIO and BPIO) used the same number of instances thus makes the Wilcoxon Signed Rank easier to test.

Table 10: Wilcoxon Signed Ranks Test Results.

Dataset	Algorithm	R+	R-	p-value
MKNAPCP 1	Modified-BPIO vs BPIO	53	2	0.009
MKNAPCP 4	Modified-BPIO vs BPIO	37	18	0.359

8 Conclusion

This paper presents a new modified swarm-intelligence method based on pigeon-inspired optimization algorithm for solving the multidimensional knapsack problem (MKP). The PIO is a class of swarm intelligence population-based algorithm that is proposed for continuous optimization problem and recently adopted to cope with the nature of 0 - 1 multidimensional knapsack problem. Experimentally, the BPIO lost the diversity of the solution space quickly during the search operations. In a bid to alleviate this problem, the crossover operator is integrated after the landmark operator of the BPIO. The performance of the Modified-BPIO is evaluated on some MKP benchmarks taken from OR-Library. The results of the experiments proved that the Modified-BPIO is better than existing algorithms that worked on the problem. Note that the proposed Modified-BPIO is tailored for solving the MKP only, in order to justify its performance, further investigation on its performance to other discrete optimization problems like traveling thief problem, timetabling and scheduling problem is very necessary as one of the future research. More so, the performance of the Modified-BPIO algorithm could be investigated for the ambient assisted living application and finally, the algorithm could be hybridized with expectation-maximization and k-means algorithm for clustering problems. Finally, the usage of late acceptance strategy in the map and compass operator of the Modified-BPIO can be investigated for its advantages in escaping local optima.

Acknowledgement: The authors will like to appreciate the anonymous reviewers and editor for their insightful comments in enhancing the readability of the paper.

References

[1] Mohamed Abdel-Basset and Yongquan Zhou, An elite opposition-flower pollination algorithm for a 0-1 knapsack problem, *International Journal of Bio-Inspired Computation* **11** (2018), 46–53.

[2] Hazem Ahmed and Janice Glasgow, Swarm intelligence: concepts, models and applications, *School Of Computing, Queens University Technical Report* (2012).

[3] Leanderson André and Rafael Stubs Parpinelli, *A Binary Differential Evolution with Adaptive Parameters Applied to the Multiple Knapsack Problem*, Nature-Inspired Computation and Machine Learning, Springer, 2014, pp. 61–71.

[4] Jagdish Chand Bansal and Kusum Deep, A modified binary particle swarm optimization for knapsack problems, *Applied Mathematics and Computation* **218** (2012), 11042–11061.

[5] Asaju La'aro Bolaji, Aminu Ali Ahmad and Peter Bamidele Shola, Training of neural network for pattern classification using fireworks algorithm, *International Journal of System Assurance Engineering and Management* **9** (2018), 208–215.

[6] Asaju La'aro Bolaji, Mohammed Azmi Al-Betar, Mohammed A Awadallah, Ahamad Tajudin Khader and Laith Mohammad Abualigah, A comprehensive review: Krill Herd algorithm (KH) and its applications, *Applied Soft Computing* **49** (2016), 437–446.

[7] Asaju La'aro Bolaji, Balogun Sulaiman Babatunde and Peter Bamidele Shola, *Adaptation of Binary Pigeon-Inspired Algorithm for Solving Multidimensional Knapsack Problem*, Soft Computing: Theories and Applications, Springer, 2018, pp. 743–751.

[8] Mingchang Chih, Self-adaptive check and repair operator-based particle swarm optimization for the multidimensional knapsack problem, *Applied Soft Computing* **26** (2015), 378–389.

[9] Mingchang Chih, Three pseudo-utility ratio-inspired particle swarm optimization with local search for multidimensional knapsack problem, *Swarm and evolutionary computation* **39** (2018), 279–296.

[10] Benny Chor and Ronald L Rivest, A knapsack-type public key cryptosystem based on arithmetic in finite fields, *Information Theory, IEEE Transactions on* **34** (1988), 901–909.

[11] Bianca De Almeida Dantas and Edson Norberto Cáceres, A Parallelization of a Simulated Annealing Approach for 0-1 Multidimensional Knapsack Problem Using GPGPU, in: *Computer Architecture and High Performance Computing (SBAC-PAD), 2016 28th International Symposium on*, IEEE, pp. 134–140, 2016.

[12] Haibin Duan and Peixin Qiao, Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning, *International Journal of Intelligent Computing and Cybernetics* **7** (2014), 24–37.

[13] Amira Gherboudj, Abdesslem Layeb and Salim Chikhi, Solving 0-1 knapsack problems by a discrete binary version of cuckoo search algorithm, *International Journal of Bio-Inspired Computation* **4** (2012), 229–236.

[14] Tim Guilford, Stephen Roberts, Dora Biro and Iead Rezek, Positional entropy during pigeon homing II: navigational interpretation of Bayesian latent state models, *Journal of theoretical biology* **227** (2004), 25–38.

[15] Boukthir Haddar, Mahdi Khemakhem, Saïd Hanafi and Christophe Wilbaut, A hybrid quantum particle swarm optimization for the multidimensional knapsack problem, *Engineering Applications of Artificial Intelligence* **55** (2016), 1–13.

[16] Ran Hao, Delin Luo and Haibin Duan, Multiple UAVs mission assignment based on modified Pigeon-inspired optimization algorithm, in: *Guidance, Navigation and Control Conference (CGNCC), 2014 IEEE Chinese*, IEEE, pp. 2692–2697, 2014.

[17] Chaitr S Hiremath and Raymond R Hill, First-level tabu search approach for solving the multiple-choice multidimensional knapsack problem, *International Journal of Metaheuristics* **2** (2013), 174–199.

[18] Liangjun Ke, Zuren Feng, Zhigang Ren and Xiaoliang Wei, An ant colony optimization approach for the multidimensional knapsack problem, *Journal of Heuristics* **16** (2010), 65–83.

[19] Min Kong and Peng Tian, Apply the particle swarm optimization to the multidimensional knapsack problem, in: *International Conference on Artificial Intelligence and Soft Computing*, Springer, pp. 1140–1149, 2006.

[20] Xiangyong Kong, Liqun Gao, Haibin Ouyang and Steven Li, A simplified binary harmony search algorithm for large scale 0–1 knapsack problems, *Expert Systems with Applications* **42** (2015), 5337–5355.

[21] Chi-Sung Laih, Jau-Yien Lee, Lein Harn and Yan-Kuin Su, Linearly shift knapsack public-key cryptosystem, *Selected Areas in Communications, IEEE Journal on* **7** (1989), 534–539.

[22] Abdesslem Layeb, A novel quantum inspired cuckoo search for knapsack problems, *International Journal of Bio-Inspired Computation* **3** (2011), 297–305.

[23] Silvano Martello and Paolo Toth, *Knapsack problems: algorithms and computer implementations*, John Wiley & Sons, Inc., 1990.

[24] AJ McAuley, A New Trapdoor Knapsack Public Key Cryptosystem., in: *Advances in Cryptology: Proceedings of EUROCRYPT 84. A Workshop on the Theory and Application of Cryptographic Techniques-Paris, France, April 9-11, 1984*, 209, Springer, p. 150, 2007.

[25] Cordula V Mora, Michael Davison, J Martin Wild and Michael M Walker, Magnetoreception and its trigeminal mediation in the homing pigeon, *Nature* **432** (2004), 508–511.

[26] Wen-Tsao Pan, A new fruit fly optimization algorithm: taking the financial distress model as an example, *Knowledge-Based Systems* **26** (2012), 69–74.

[27] HuaXin Qiu and HaiBin Duan, Multi-objective pigeon-inspired optimization for brushless direct current motor parameter design, *Science China Technological Sciences* **58** (2015), 1915–1923.

[28] Günther R Raidl and Jens Gottlieb, Empirical analysis of locality, heritability and heuristic bias in evolutionary algorithms: A case study for the multidimensional knapsack problem, *Evolutionary Computation* **13** (2005), 441–475.

[29] Rizk M Rizk-Allah, Ragab A El-Sehiemy and Gai-Ge Wang, A novel parallel hurricane optimization algorithm for secure emission/economic load dispatch solution, *Applied Soft Computing* **63** (2018), 206–222.

[30] Wei Shih, A branch and bound method for the multiconstraint zero-one knapsack problem, *Journal of the Operational Research Society* **30** (1979), 4.

[31] Shyam Sundar, Alok Singh and André Rossi, *An artificial bee colony algorithm for the 0–1 multidimensional knapsack problem*, Contemporary Computing, Springer, 2010, pp. 141–151.

[32] M Fatih Tasgetiren, Quan-Ke Pan, Damla Kizilay and Gursel Suer, A differential evolution algorithm with variable neighborhood search for multidimensional knapsack problem, in: *Evolutionary Computation (CEC), 2015 IEEE Congress on*, IEEE, pp. 2797–2804, 2015.

[33] Paolo Toth, Dynamic programming algorithms for the zero-one knapsack problem, *Computing* **25** (1980), 29–45.

[34] Gai-Ge Wang, Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems, *Memetic Computing* **10** (2018), 151–164.

[35] Gai-Ge Wang, Suash Deb and Leandro dos Santos Coelho, Earthworm optimisation algorithm: a bio-inspired metaheuristic algorithm for global optimisation problems., *IJBIC* **12** (2018), 1–22.

[36] Gai-Ge Wang, Suash Deb, Xiao-Zhi Gao and Leandro Dos Santos Coelho, A new metaheuristic optimisation algorithm motivated by elephant herding behaviour, *International Journal of Bio-Inspired Computation* **8** (2016), 394–409.

[37] Gai-Ge Wang, Amir H Gandomi, Amir H Alavi and Dunwei Gong, A comprehensive review of krill herd algorithm: variants, hybrids and applications, *Artificial Intelligence Review* **51** (2019), 119–148.

[38] Haizhou Wu, Yongquan Zhou and Qifang Luo, Hybrid symbiotic organisms search algorithm for solving 0-1 knapsack problem, *International Journal of Bio-Inspired Computation* **12** (2018), 23–53.

[39] Zhen Yang, Guoqing Wang and Feng Chu, An effective GRASP and tabu search for the 0–1 quadratic knapsack problem, *Computers & Operations Research* **40** (2013), 1176–1185.

[40] Bo Zhang and Haibin Duan, *Predator-prey pigeon-inspired optimization for UAV three-dimensional path planning*, Advances in Swarm Intelligence, Springer, 2014, pp. 96–105.

[41] Shujian Zhang and Haibin Duan, Gaussian pigeon-inspired optimization approach to orbital spacecraft formation reconfiguration, *Chinese Journal of Aeronautics* **28** (2015), 200–205.

[42] Shujian Zhang and Haibin Duan, Multiple UCAVs Target Assignment via Bloch Quantum-Behaved Pigeon-Inspired Optimization, in: *Control Conference (CCC), 2015 34th Chinese*, IEEE, pp. 6936–6941, 2015.

[43] Yongquan Zhou, Zongfan Bao, Qifang Luo and Sen Zhang, A complex-valued encoding wind driven optimization for the 0-1 knapsack problem, *Applied Intelligence* **46** (2017), 684–702.

[44] Yongquan Zhou, Xin Chen and Guo Zhou, An improved monkey algorithm for a 0-1 knapsack problem, *Applied Soft Computing* **38** (2016), 817–830.
[45] Yongquan Zhou, Liangliang Li and Mingzhi Ma, A complex-valued encoding bat algorithm for solving 0–1 knapsack problem, *Neural Processing Letters* **44** (2016), 407–430.