



An improved nature inspired meta-heuristic algorithm for 1-D bin packing problems

Mohamed Abdel-Basset¹ · Gunasekaran Manogaran² · Laila Abdel-Fatah¹ · Seyedali Mirjalili³

Received: 7 February 2018 / Accepted: 7 March 2018 / Published online: 23 March 2018
© Springer-Verlag London Ltd., part of Springer Nature 2018

Abstract

Bin packing problem (BPP) is a classical combinatorial optimization problem widely used in a wide range of fields. The main aim of this paper is to propose a new variant of whale optimization algorithm named improved Lévy-based whale optimization algorithm (ILWOA). The proposed ILWOA adapts it to search the combinatorial search space of BPP problems. The performance of ILWOA is evaluated through two experiments on benchmarks with varying difficulty and BPP case studies. The experimental results confirm the prosperity of the proposed algorithm in proficiency to find the optimal solution and convergence speed. Further, the obtained results are discussed and analyzed according to the problem size.

Keywords Bin packing problem · Whale optimization algorithm · Best fit algorithm · Meta-heuristic

1 Introduction

Combinatorial optimization [1] is a subfield of discrete mathematics where the decision variables are combinations or permutations that respect a predefined set of constraints. Such optimization problems can be defined as given finite object set φ and an objective function [2]:

$$f : \varphi \rightarrow O \quad (1)$$

where O is an ordered set that associates with a cost or a profit of each object. The objective is to find an object φ with value

(cost or profit) that best minimize or maximize the objective function with respect to some constraints.

One of the most important combinatorial optimization problems is bin packing problem (BPP). The classical bin packing problem consists of a set of items that need to be packed into one bin only. Each item has a weight and each bin has a capacity. The main goal of the problem is to pack all items in the few possible number of bins where each bin does not exceed its capacity. Thereby, BPP involved in each field contains packing problem such as multiprocessor scheduling [3], supply chain [4], logistics [5], telecommunications [6], and cloud computing [7].

The contribution of this paper is to propose a new variant of whale optimization algorithm named improved Lévy-based whale optimization algorithm (ILWOA). According to the no-free-lunch theorem [8, 9] in optimization, there is no algorithm to solve all optimization problems. It has observed that WOA performs well on combinatorial problems, and this motivated our attempts to improve and adapt this algorithm. Therefore, the exploration capabilities of whale optimization algorithm (WOA) [10] is enhanced with deploying Lévy flight for whale movements. Second, WOA is embedded with a new mutation phase to improve convergence speed. Finally, a logistic chaotic map is used for efficient switch between exploration and exploitation phases. In order to map between the continuous search space and the combinatorial one, the largest order value (LOV) [11] technique is applied. By conducting experiments

✉ Mohamed Abdel-Basset
analyst_mohamed@yahoo.com; analyst_mohamed@zu.edu.eg

Gunasekaran Manogaran
gmanogaran@ucdavis.edu

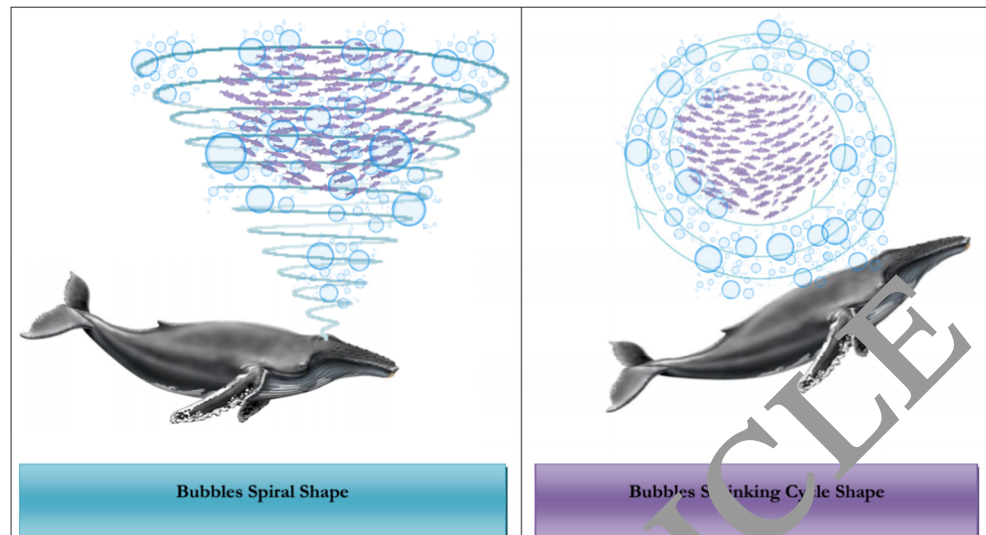
Seyedali Mirjalili
amirjalili@gmail.com

¹ Faculty of Computers and Informatics, Department of Operations Research, Zagazig University, Zagazig, Egypt

² University of California, Davis, CA, USA

³ Institute for Integrated and Intelligent Systems, Griffith University, Nathan Campus, Brisbane, QLD 4111, Australia

Fig. 1 Whale hunting strategy



and statistical analysis, the effectiveness of amendments to improve the performance of WOA and superiority of the improved algorithm comparators to solve the given problem were proved.

The remaining structure of this paper is as follows. An in-depth literature review is given in Sect. 2. Problem formulation is provided in Sect. 3. The WOA and proposed algorithm are discussed in Sects. 4 and 5. The experimental results are shown in Sect. 6. The results analysis is presented in Sect. 7. Finally, the conclusions and future works are provided in Sect. 8.

2 Literature review

BPP belongs to NP-hard problems [12] so that using exact methods (e.g., branch and bound algorithm [13] and branch and price algorithm [14]) or basic heuristic algorithms may become deficient for handling large BPPs. Initially, many heuristic were proposed for solving BPP as next fit [15], first fit [15], worst fit [16], best fit [17], graph-based algorithm [18], etc. Indeed, heuristic algorithms can guarantee a good solution for small-scale instances and they can be defined as “a type of strategy that dramatically limit the search for solutions” [19, 20]. Thus, the new trends are directed towards using meta-heuristic algorithms which are not problem-specific heuristic that can find a high-quality solution through the tradeoff between the exploitation and exploration of search space [21, 22]. For example, the authors in [23, 24] solved BPP with genetic algorithm (GA) [25] which was combined with grouping mechanism for handling the problem. In [26], Layeb and Boussalia combined cuckoo search (CS) [27] algorithm with a binary quantum technique for solving BPP. Also, the authors in [28] applied

CS for solving the problem but they used FF and ranked order value (ROV) [29] for adapting the algorithm. In [30], Levine and Dacatelle proposed ant system (AS) algorithm [31] with local search technique for solving the problem. While in [32], the authors hybridized the previous technique with firefly algorithm (FA) [33]. Sonuc et al. [34] solved BPP with simulated annealing (SA) [35] which combined with a swap function for enhancing the solution.

Roughly speaking, the main challenge of bin packing problem is obtaining a good solution but with less time. In continuous search space, WOA can obtain a high-quality solution in negligible time. Thereby, this paper employs the improved WOA capabilities to solve BPP with more convergence speed.

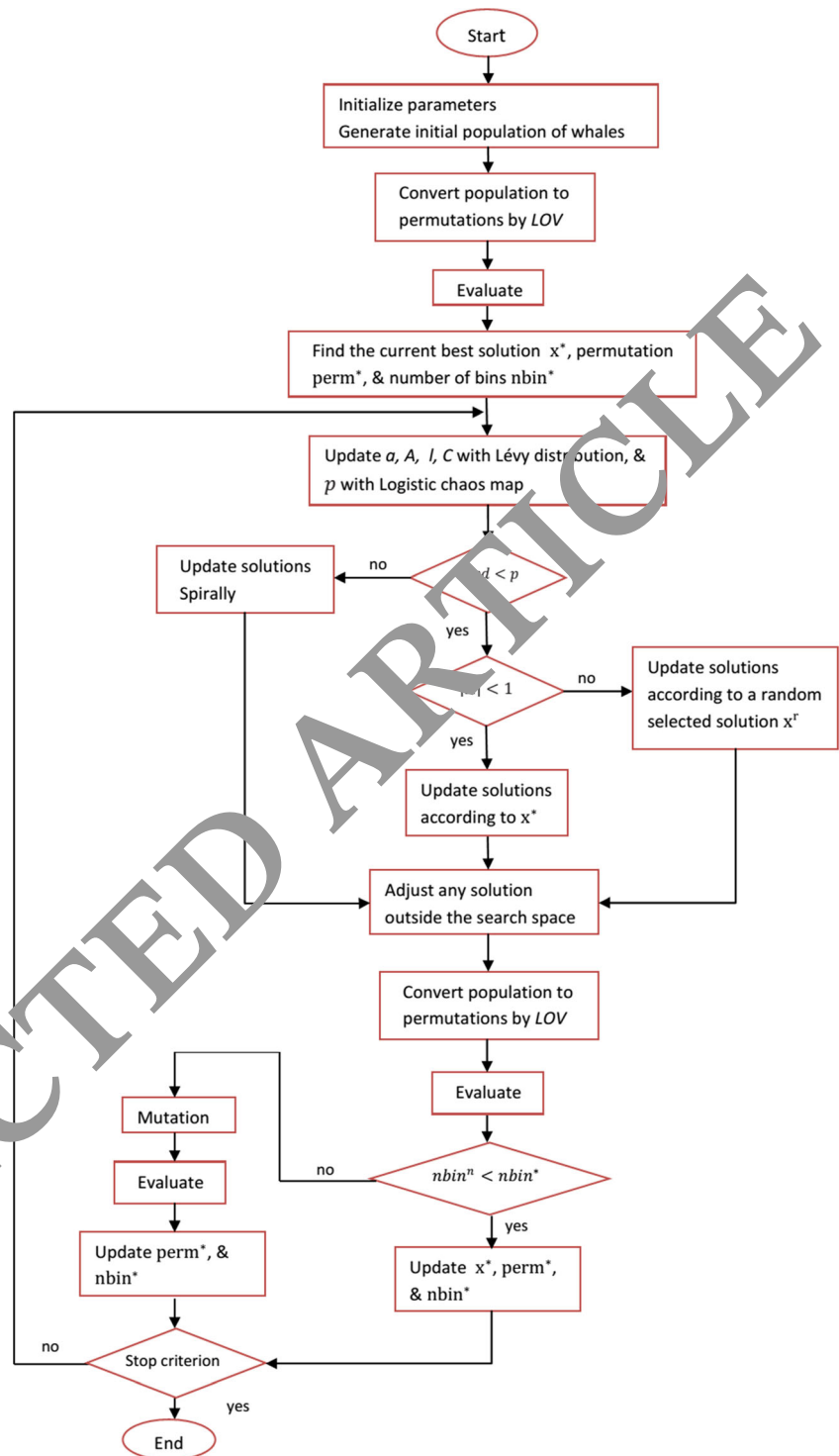
3 Problem formulation

In general, BPP can be defined as follows: Given a set of items with their weights, and a set of fixed-size bins, find the minimum number of bins that can hold all items. Mathematically, the formulation of BPP can be expressed as an integer programming problem as follows [36]:

$$\begin{aligned}
 & \min \sum_{i=1}^n y_i \\
 \text{s.t. } & \sum_{j=1}^n w_j x_{ij} \leq c y_i, i \in N = \{1, \dots, n\}, \\
 & \sum_{i=1}^n x_{ij} = 1, \quad j \in N, \\
 & y_i \in \{0, 1\}, \quad i \in N, \\
 & x_{ij} \in \{0, 1\}, \quad i \in N, j \in N,
 \end{aligned} \tag{2}$$

where y_i is a binary variable that indicates whether bin i contains items, x_{ij} is a binary variable that indicates whether item j is assigned to bin i , w_j is the weight of item j , c is bin capacity, and n is number of available bins. BPP can be

Fig. 2 ILWOA flowchart



extended with no conceptual difficulty to higher dimensions (two or three dimensions) or other variants like knapsack problem [37], bin packing with conflicts [38], colored bin packing [39], etc. Also, BPP can be divided to online and offline problem according to the number of items is fixed or variable.

4 Whale optimization algorithm

4.1 Biological inspiration

Humpback whales are known for their unique acrobatic aerial breaching, the ability to configure a language for mating and

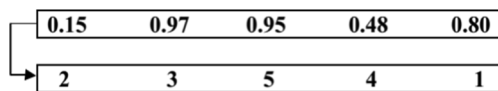


Fig. 3 Mapping between continuous and discrete solution using LOV

hunting through their complex sounds, and the rare hunting “bubble net” strategy [40]. For the feeding strategy, the humpback whales can intelligently cooperate together in order to trap fish into a ring of air by emitting a stream of bubbles. In particular, the emitted bubbles take two shapes: a spiral shape and a shrinking circle shape (as shown in Fig. 1). After that, the trapped fish are swallowed up by all humpback whales [41]. For more information, see [42–44].

4.2 Main procedures of WOA

Whale optimization algorithm (WOA) [10] is a novel population and swarm intelligence-based meta-heuristic that was inspired by the bubble-net feeding strategy previously discussed, i.e., after the humpback whales determine the prey position, they begin to emit a stream of bubbles in two manners (shrinking circle and spiral shape) around their prey. In WOA, the search for the prey represents the exploration phase and the bubble net with shrinking circle and spiral shape represent exploitation phase. Mathematically, the fish positions are randomly initialized which represent the initial population and evaluated in order to find the current best solution. Then, the other candidate solutions are updated according to the current best solution as follows:

$$\vec{D} = |\vec{C} \cdot \vec{x}^*(t) - \vec{x}(t)| \tag{3}$$

$$\vec{x}(t+1) = \vec{x}^*(t) - \vec{A} \cdot \vec{D} \tag{4}$$

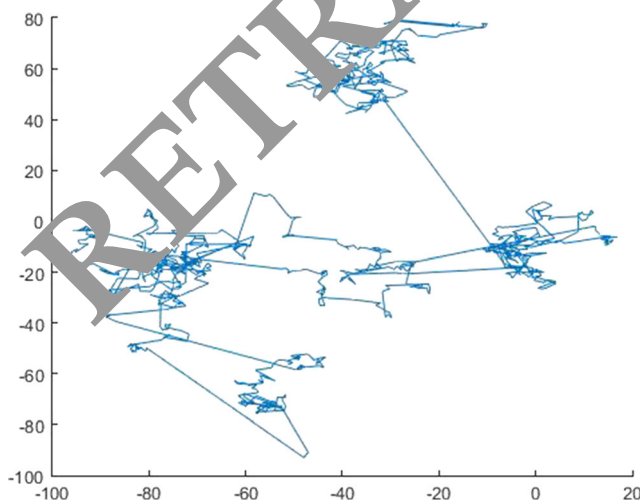


Fig. 4 Isotropic Lévy Distribution

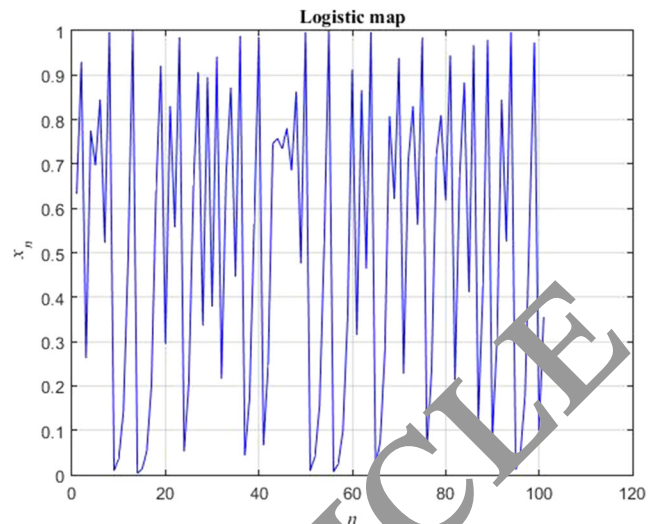


Fig. 5 Logistic chaotic map simulation

where $\vec{x}^*(t)$ is the current best solution at time t , and \vec{C} and \vec{A} are two coefficient vectors that can be calculated as

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r} - \vec{a} \tag{5}$$

$$\vec{C} = 2 \cdot \vec{r} \tag{6}$$

where \vec{a} is linearly decreased from 2 to 0 during iterations which represents the shrinking encircling behavior and \vec{r} is a random number between [0, 1].

For better diversification of the search space, WOA combines two exploration search mechanisms (depending on the current best or random solution) that switched according to the value of \vec{A} . In other words, if $\vec{A} < 1$, then the new solution is generated using Eq. (3). In other cases, the new solution is calculated as the follows:

$$\vec{D} = |\vec{C} \cdot \vec{x}_r(t) - \vec{x}(t)| \tag{7}$$

$$\vec{x}(t+1) = \vec{x}_r(t) - \vec{A} \cdot \vec{D} \tag{8}$$

where $\vec{x}_r(t)$ is a solution selected randomly from the same population.

For simulating the humpback whale spiral movement around their prey (the current best solution), the following formulation is used:

$$\vec{D} = |\vec{x}^*(t) - \vec{x}(t)| \tag{9}$$

$$\vec{x}(t+1) = \vec{D} \cdot e^{bl} \cdot \cos(2\pi l) + \vec{x}^*(t) \tag{10}$$

where b is a predefined constant for defining the shape of the logarithmic spiral and l is a random number between $[-1, 1]$.

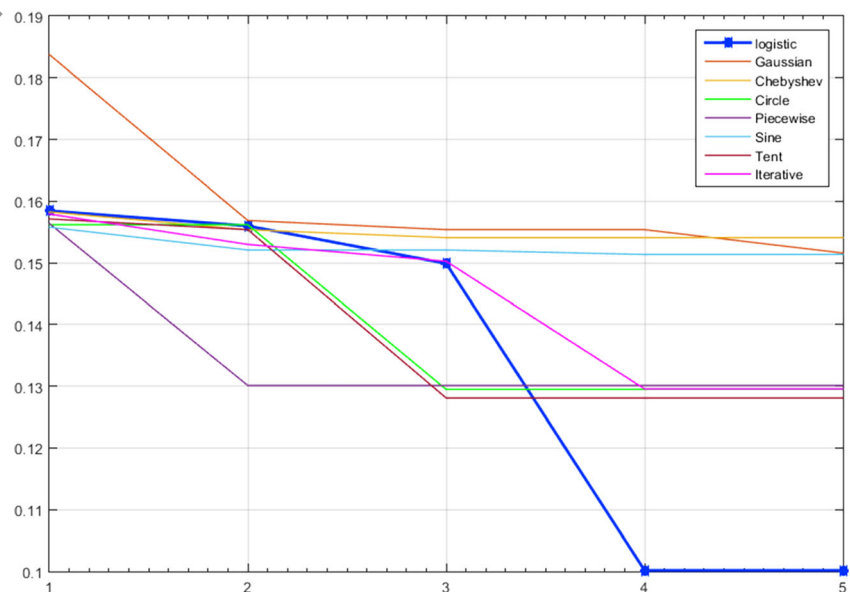
Whale Optimization Algorithm

```

Initialize whales positions
Find the current best position  $\vec{x}^*$ 
while(termination criterion doesn't met )
for each whale position
Update  $a, A, C, l$ , and  $r$ 
if  $r < p$ 
if  $|A| < 1$ 
Calculate  $\vec{D} = |\vec{C} \cdot \vec{x}^*(t) - \vec{x}(t)|$ 
Update position by  $\vec{x}(t + 1) = \vec{x}^*(t) - \vec{A} \cdot \vec{D}$ 
else if  $|A| \geq 1$ 
Select position  $\vec{x}_r(t)$  randomly
Calculate  $\vec{D} = |\vec{C} \cdot \vec{x}_r(t) - \vec{x}(t)|$ 
Update position by  $\vec{x}(t + 1) = \vec{x}_r(t) - \vec{A} \cdot \vec{D}$ 
end
else if  $r \geq p$ 
Calculate  $\vec{D}' = |\vec{x}^*(t) - \vec{x}(t)|$ 
update position spirally by  $\vec{x}(t + 1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi t) + \vec{x}^*(t)$ 
end
end
Adjust position inside search space boundaries
Update the current best position  $\vec{x}^*$ 
end
return  $\vec{x}^*$ 
    
```

Pseudocode 1. WOA.

Fig. 6 Logistic map vs. other maps



During WOA iterations, a probability of 50% is assumed to alter between the searching of food and hunting mechanisms to update the candidate solutions as follows:

$$\vec{x}(t+1) = \begin{cases} \vec{x}(t) - \vec{A} \cdot \vec{D} & , r < p \\ \vec{D} \cdot e^{bl} \cdot \cos(2\pi l) + \vec{x}^*(t) & , r \geq p \end{cases} \quad (11)$$

where $\vec{x}^*(t)$ is the current best solution at time t , p is equal to 0.5, and r is a random number between $[0, 1]$ (see Pseudocode 1).

5 Improved ILWOA

Despite the efficiency of WOA, its convergence rate and performance still need to be improved. For this reason, several enhancements are proposed in this paper in order to improve the performance and the convergence rate of WOA. The pseudocode of the proposed algorithm is shown in Pseudocode 2. Also, the flowchart of ILWOA is presented in Fig. 2.

Improved Lévy-based Whale Optimization Algorithm

Initialize the population of whales
 Map continuous population to combinatorial one by LOV
 Evaluate initial permutations
 Find the current best solution x^*
 Find the current best permutation $perm^*$
 Find the current best number of bins $nbin^*$
 while(termination criterion doesn't met)
 for each whale position
 Update a , A , and l , r , C with Lévy distribution, p with Logistic map
 if $r < p$
 if $|A| < 1$
 Calculate $\vec{D} = |\vec{C} \cdot \vec{x}^*(t) - \vec{x}(t)|$
 Update position by $\vec{x}(t+1) = \vec{x}^*(t) - \vec{A} \cdot \vec{D}$
 else if $|A| \geq 1$
 Select position $\vec{x}_r(t)$ randomly
 Calculate $\vec{D} = |\vec{C} \cdot \vec{x}_r(t) - \vec{x}(t)|$
 Update position by $\vec{x}(t+1) = \vec{x}_r(t) - \vec{A} \cdot \vec{D}$
 end
 else if $r \geq p$
 Calculate $\vec{D}^* = |\vec{x}^*(t) - \vec{x}(t)|$
 update position spirally by $\vec{x}(t+1) = \vec{D}^* \cdot e^{bl} \cdot \cos(2\pi l) + \vec{x}^*(t)$
 end
 end
 Adjust position inside search space boundaries
 Map new solution to combinatorial one by LOV
 Evaluate new combinatorial solution $perm^n$
 if $nbin^n < nbin^*$
 Update $perm^*$, $nbin^*$
 else
 Apply mutation for $perm^n$
 end if
 Evaluate new permutations
 Update $perm^*$, $nbin^*$
 end while
 return $nbin^*$ and $perm^*$

Pseudocode 2. ILWOA

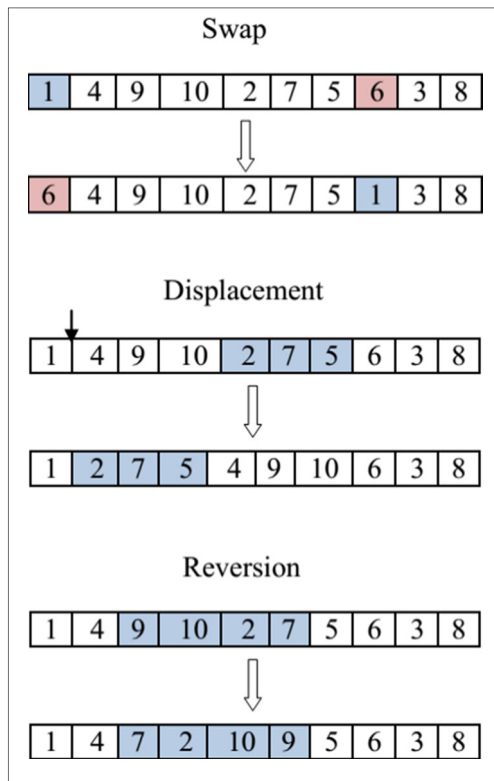


Fig. 7 Mutation operators

5.1 The application of ILWOA to BPP

5.1.1 Discretization of search space

The basic version of WOA was proposed for solving optimization problems in continuous search space. For applying the proposed algorithm in combinatorial search space, LOV [11] is used. LOV can be used to convert the continuous solution to a discrete one through descending order of values as shown in Fig. 3. For BPP, the discrete solution means the order of item allocation to bins which are filled using BF algorithm. In BF, each item is placed in a bin which will leave the least leftover space. If the item does not fit in any bin, a new bin is opened.

5.1.2 Objective function

Roughly speaking, the use of the bin number as a fitness function may lead to the algorithm stagnation because there can be many permutations that have the same number of bins. So, it is more efficient to use bin occupancy as a mean of the solution evaluation. The following formulation was introduced by Gade et al. [45]:

$$min \sigma = 1 - \left(\frac{\sum_{i=1}^n (ocup_i/c)^k}{n} \right) \tag{12}$$

where n is the number of used bins, $ocup_i$ is the occupancy of each bin i , c is the bin capacity, and k is a constant (usually $k = 2$).

Table 1 Experimental results of easy class

Instance	Number	C	Best known	QICS	ACS	FCO	FA	AS	ILWOA
N1C1W1_	50	100	25	25	25	25	26	25	25
N1C1W1_D	50	100	28	28	28	28	28	28	28
N1C1W1_	50	100	25	25	25	25	26	25	25
N1C1W1_B	50	100	31	31	31	31	31	31	31
N1C1W1_E	50	100	26	26	26	26	27	26	26
N1C1W1_F	50	100	27	27	27	27	27	27	27
N1C1W1_I	50	100	25	25	25	25	25	25	25
N2C1W2_P	100	100	68	68	68	68	68	68	68
N2C1W2_N	100	100	64	64	64	64	65	64	64
N2C1W2_O	100	100	64	64	64	64	66	64	64
N2C1W2_R	100	100	67	67	67	67	68	67	67
N4C1W2_T	500	100	323	323	323	323	328	323	323
N4C1W4_C	500	100	365	365	365	365	368	365	365
N4C1W4_A	500	100	368	368	368	368	373	368	368
N4C1W4_D	500	100	359	359	359	359	364	359	359
N4C1W4_B	500	100	349	349	349	349	356	349	349

Table 2 Descriptive statistics of easy class

Algorithm	Number	Mean	Std. deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
FCO	16	138.38	150.472	25	368	26.25	64.00	342.50
QICS	16	138.38	150.472	25	368	26.25	64.00	342.50
ACS	16	138.38	150.472	25	368	26.25	64.00	342.50
FA	16	140.38	152.555	25	373	27.00	65.50	349.00
AS	16	138.38	150.472	25	368	26.25	64.00	342.50
ILWOA	16	138.38	150.472	25	368	26.25	64.00	342.50

5.2 Lévy flight distribution

Lévy distribution is a mathematical model used to describe anomalous diffusion which has infinite mean and variance that cause much longer movement from its current position (see Fig. 4); thus, it is more efficient in the search space exploration [46]. In the proposed algorithm, the parameter \vec{C} is replaced with a random step drawn from isotropic Lévy distribution as follows:

$$Lévy \sim \frac{\lambda \Gamma(\lambda) \sin\left(\frac{\pi\lambda}{2}\right)}{\pi} \frac{1}{s^{1+\lambda}}, \quad (s \gg s_0 > 0) \quad (13)$$

$$s = \frac{U}{|V|^{\lambda^{-1}}}, \quad U \sim (0, \sigma_u^2), \quad V \sim (0, \sigma_v^2) \quad (14)$$

$$\sigma_u^2 = \left[\frac{\Gamma(1+\lambda)}{\lambda \Gamma((1+\lambda)/2)} \cdot \frac{\sin(\frac{\pi\lambda}{2})}{2^{(\lambda-1)/2}} \right]^{1/\lambda}, \quad \sigma_v^2 = 1 \quad (15)$$

where Lévy is a step size drawn from Lévy distribution, $\Gamma(\lambda)$ is the standard gamma function, with $\lambda > 0$ which is drawn according to Mantegna algorithm [47], U and V are samples drawn from Gaussian normal distribution where mean is equal zero, and σ_u^2 and σ_v^2 are variances [48].

5.3 Chaotic maps

Mathematically, the chaotic maps can be defined as the following: given a set Λ (where $f: \Lambda \rightarrow \Lambda$), f is considered a chaotic map on Λ if [49]:

1. f is unpredictable as it sensitively depends on initial conditions.
2. f cannot be decomposed to subsystem as it is transitive based on a topology.
3. f has an element of regularity because points are periodically dense in Λ .

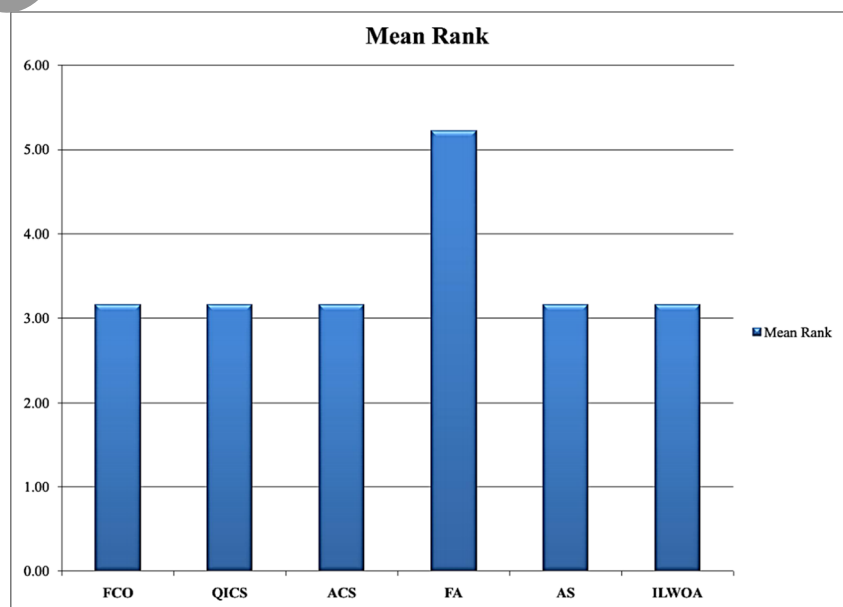
Fig. 8 Friedman test of easy class

Table 3 Experimental results of medium class

Instance	Number	Cap	Best known	QICS	ACS	FCO	FA	AS	ILWOA
N1W1B2R1	50	1000	17	17	17	17	17	18	17
N1W1B1R9	50	1000	17	18	17	17	17	17	17
N1W1B1R2	50	1000	19	20	19	19	20	20	19
N1W1B2R0	50	1000	17	18	17	18	18	18	17
N1W1B2R3	50	1000	16	17	17	17	17	17	17
N2W1B1R0	100	1000	34	36	34	35	35	37	34
N2W1B1R3	100	1000	34	37	35	36	36	36	35
N2W1B1R1	100	1000	34	37	35	36	36	35	35
N2W1B1R4	100	1000	34	37	34	35	35	35	34
N2W3B3R7	100	1000	13	13	13	13	13	13	13
N2W4B1R0	100	1000	12	12	12	12	12	12	12
N4W2B1R0	500	1000	101	109	105	106	106	107	105
N4W2B1R3	500	1000	100	108	104	105	105	106	104
N4W3B3R7	500	1000	74	74	74	74	74	74	74
N4W4B1R0	500	1000	56	58	57	57	57	58	57
N4W4B1R1	500	1000	56	58	57	58	58	58	57

In other words, Chaotic maps are evolutionary functions that produce a deterministic bounded sequence of random numbers depending on initial condition. There are different types of maps like logistic map, Chebyshev map, tent map, etc. After several trials of different chaotic maps, logistic (or quadratic) is selected in the proposed algorithm because it obtained the best results (see Fig. 5). Chaotic map is selected for determining the value of the switching probability p because of being random-like, non-period, and non-converging in parameter adaptation. The logistic map can be formulated as

$$x_{n+1} = ax_n(1-x_n) \quad , x \in (0, 1), 0 < a < 4 \quad (16)$$

where a is a positive constant (sometimes also denoted “biotic potential”) that gives the logistic map. Figure 6 shows a comparative example between objective function values (Eq. (12)) of logistic map and other seven maps including Chebyshev map, Gaussian map, tent map, circle map, piecewise map, iterative map, and sine

map. It is clear that logistic reaches the minimum values of objective function during 5 iterations with 10 search agents.

5.4 Mutation phase

In ILWOA, mutation phase is done before the end of each iteration, i.e., the current best number of bins is checked whether it reached the optimal or not. The optimal solution can be computed as

$$\text{optimal number of bins} = \frac{\sum \text{itemsizes}}{\text{bin capacity}} \quad (17)$$

If the current best number of bins is optimal, the search stops. Otherwise, it is randomly modified through mutation phase, which is divided to three operators: swap, displacement, and reversion. The swap operator selects and swaps two items’ indexes randomly [50]. Displacement operator cuts

Table 4 Descriptive statistics of medium class

Algorithm	Number	Mean	Std. deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
QICS	16	41.81	31.747	12	109	17.25	36.50	58.00
ACS	16	40.44	30.816	12	105	17.00	34.00	57.00
FCO	16	40.94	31.030	12	106	17.00	35.00	57.75
FA	16	41.00	30.984	12	106	17.00	35.00	57.75
AS	16	41.38	31.224	12	107	17.25	35.50	58.00
ILWOA	16	40.44	30.816	12	105	17.00	34.00	57.00

Table 5 Experimental results of hard class

Instance	Number	Cap	Best known	QICS	ACS	FCO	FA	AS	ILWOA
HARD0	200	100,000	56	59	58	59	60	59	58
HARD1	200	100,000	57	60	59	59	60	60	59
HARD2	200	100,000	56	60	59	59	61	60	59
HARD3	200	100,000	55	59	58	59	60	59	58
HARD4	200	100,000	57	60	59	60	61	60	59
HARD5	200	100,000	56	59	58	59	60	59	58
HARD6	200	100,000	57	59	59	59	61	60	59
HARD7	200	100,000	55	59	58	58	59	59	57
HARD8	200	100,000	57	59	59	59	61	60	59
HARD9	200	100,000	56	59	59	59	60	59	59

a random subset of item indexes and inserts it in another random place [51]. Reversion operator cuts a random subset of items' indexes and reverses it [52] (see Fig. 7).

6 Experimental results

In order to test the validity of the proposed algorithm, two experiments are conducted on different benchmarks (all datasets can be downloaded from [53]). All experiments are carried out on a 64-bit operating system with a 2.60 GHz CPU and 6 GB RAM. In addition, the experimental results are analyzed with non-parametric Friedman test [54] to show the differences in performance between ILWOA and the compared algorithm. Friedman test is a non-parametric, rank-based version of one-way ANOVA with repeated measures which can

be performed on more than two independent samples. In other words, Friedman test compares the means of three or more variables measured on the same respondents.

The first experiment is conducted on selected benchmarks from Schott uniformly distributed instances [55] which were divided into three classes according to its difficulty: easy, medium, and hard class. ILWOA is compared with all other population and swarm intelligence based algorithms that have solved BPP, including adaptive cuckoo search (ACS) algorithm [28], firefly colony optimization (FCO) algorithm [32], quantum inspired cuckoo search algorithm (QICS) [26], firefly algorithm (FA) [33], and ant system algorithm (AS) [35]. The results of ACS, QICS, and AS are obtained from [28] while the results of FA are obtained from [32]. For parameter settings, the number of search agents is set to 10 agents and the maximum number

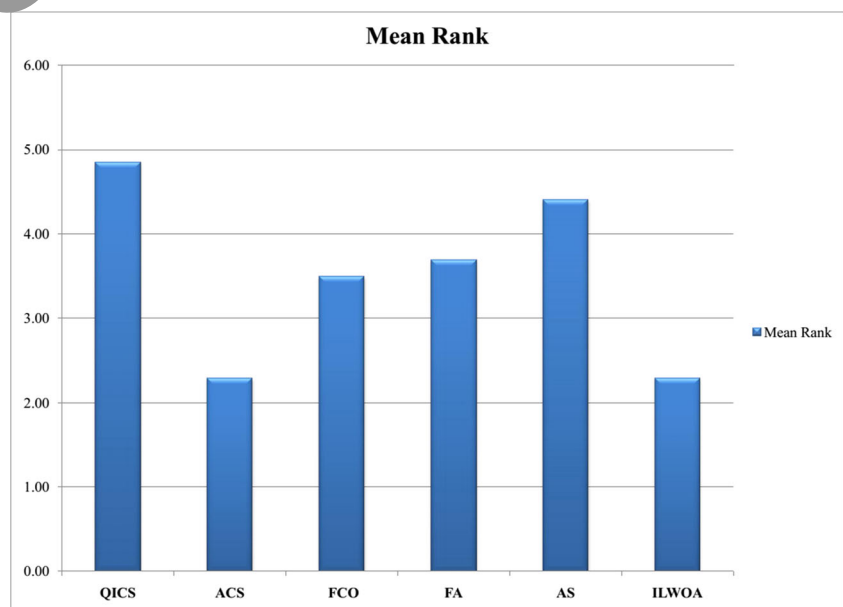
Fig. 9 Friedman test of medium class

Table 6 Descriptive statistics of hard class

Algorithm	Number	Mean	Std. deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
QICS	10	59.30	.483	59	60	59.00	59.00	60.00
ACS	10	58.60	.516	58	59	58.00	59.00	59.00
FCO	10	59.00	.471	58	60	59.00	59.00	59.00
FA	10	60.30	.675	59	61	60.00	60.00	61.00
AS	10	59.50	.527	59	60	59.00	59.50	60.00
ILWOA	10	58.50	.707	57	59	58.00	59.00	59.00

of iterations are set to only 5 iterations, whereas for ACS, it needed 20 search agent and 100 iterations in the literature which indicate the ability of ILWOA to find a good solution with a few number of search agents and iterations. Table 1 shows the experimental results of easy instances. Besides, Table 2 and Fig. 8 present the descriptive statistics of the easy class experiment and the results of Friedman test, respectively. As shown, the proposed algorithm finds the best-known solution like other algorithms (except FA) and their performances are similar. While for the medium and hard classes, the results of the proposed algorithm are superior for many instances (bolded values) as shown in Tables 3 and 5. The descriptive statistics and Friedman test results of the medium class are presented in Table 4 and Fig. 9, respectively (Table 5). The descriptive statistics of the hard class are presented in Table 6 and Friedman test results in Fig. 10. It is clear that the

proposed algorithm has the same descriptive statistics and ranked mean as ACS for the medium class. For the hard class, the result of ILWOA for "HARD7" instance outperforms other algorithms even ACS. In addition, the proposed algorithm scores the minimum mean as shown in Table 6 and the minimum ranked mean as shown in Fig. 10. Although results of ACS and ILWOA are almost similar, the proposed algorithm is significantly faster than ACS as ILWOA can find a good solution with a few number of search agents and iterations.

The second experiment is conducted on selected benchmark from "Sch_Wae2" instances [56] which are very difficult for first fit decreasing heuristic (FFD) [57], best fit decreasing heuristic (BFD) [58], and Martello-Toth packing (MTP) algorithm [36]. So, ILWOA is compared with these algorithms in finding the optimal number of bins. For the parameter settings, a few number of search agents and iterations are set to ILWOA. The number of search agents is set to

Fig. 10 Friedman test of hard class

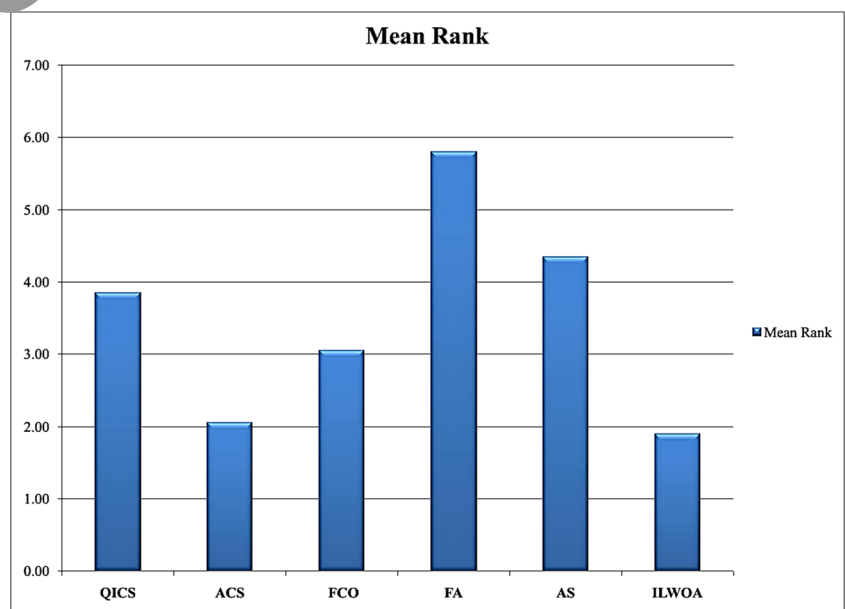


Table 7 Experimental results of *Sch_Wae2* instances

<i>Name</i>	<i>Number</i>	<i>C</i>	<i>Optimal Solution</i>	<i>FFD</i>	<i>BFD</i>	<i>MTP</i>	<i>ILWOA</i>
BPP 1	120	1000	22	23	23	23	22
BPP 2	120	1000	22	23	23	23	22
BPP 3	120	1000	21	23	23	23	22
BPP 4	120	1000	21	23	23	23	22
BPP 5	120	1000	22	23	23	23	22
BPP 6	120	1000	22	23	23	23	22
BPP 8	120	1000	21	23	23	23	22
BPP 9	120	1000	22	23	23	23	22
BPP 11	120	1000	22	23	23	23	22
BPP 13	120	1000	22	23	23	23	22
BPP 14	120	1000	21	23	23	23	22
BPP 15	120	1000	21	23	23	23	22
BPP 16	120	1000	22	23	23	23	22
BPP 17	120	1000	21	23	23	23	22
BPP 18	120	1000	22	23	23	23	22
BPP 19	120	1000	22	23	23	23	22
BPP 21	120	1000	21	23	23	23	22
BPP 22	120	1000	22	23	23	23	22
BPP 23	120	1000	22	23	23	23	22
BPP 24	120	1000	22	23	23	23	22
BPP 25	120	1000	22	23	23	23	22
BPP 26	120	1000	21	23	23	23	22
BPP 27	120	1000	21	23	23	23	22
BPP 28	120	1000	21	23	23	23	22
BPP 29	120	1000	21	23	23	23	22
BPP 30	120	1000	21	23	23	23	22
BPP 31	120	1000	22	23	23	23	22
BPP 32	120	1000	22	23	23	23	22
BPP 33	120	1000	22	23	23	23	22
BPP 34	120	1000	21	23	23	23	22
BPP 35	120	1000	22	23	23	23	22
BPP 36	120	1000	22	23	23	23	22
BPP 37	120	1000	22	23	23	23	22
BPP 38	120	1000	22	23	23	23	22
BPP 39	120	1000	22	23	23	23	22
BPP 40	120	1000	22	23	23	23	22
BPP 41	120	1000	21	23	23	23	22
BPP 42	120	1000	21	23	23	23	22
BPP 43	120	1000	22	23	23	23	22
BPP 44	120	1000	22	23	23	23	22
BPP 45	120	1000	22	23	23	23	22
BPP 55	120	1000	22	23	23	23	22
BPP 56	120	1000	22	23	23	23	22
BPP 72	120	1000	22	23	23	23	22
BPP 75	120	1000	21	23	23	23	22
BPP 83	120	1000	21	23	23	23	22
BPP 89	120	1000	22	23	23	23	22
BPP 92	120	1000	21	23	23	23	22
BPP 96	120	1000	21	23	23	23	22
BPP 100	120	1000	21	23	23	23	22

Table 8 Descriptive statistics of *Sch_Wae2* instances

Algorithm	Number	Mean	Std. Deviation	Minimum	Maximum	Percentiles		
						25th	50th (Median)	75th
FFD	50	23.00	0.000	23	23	23.00	23.00	23.00
BFD	50	23.00	0.000	23	23	23.00	23.00	23.00
MTP	50	23.00	0.000	23	23	23.00	23.00	23.00
ILWOA	50	22.00	0.000	22	22	22.00	22.00	22.00

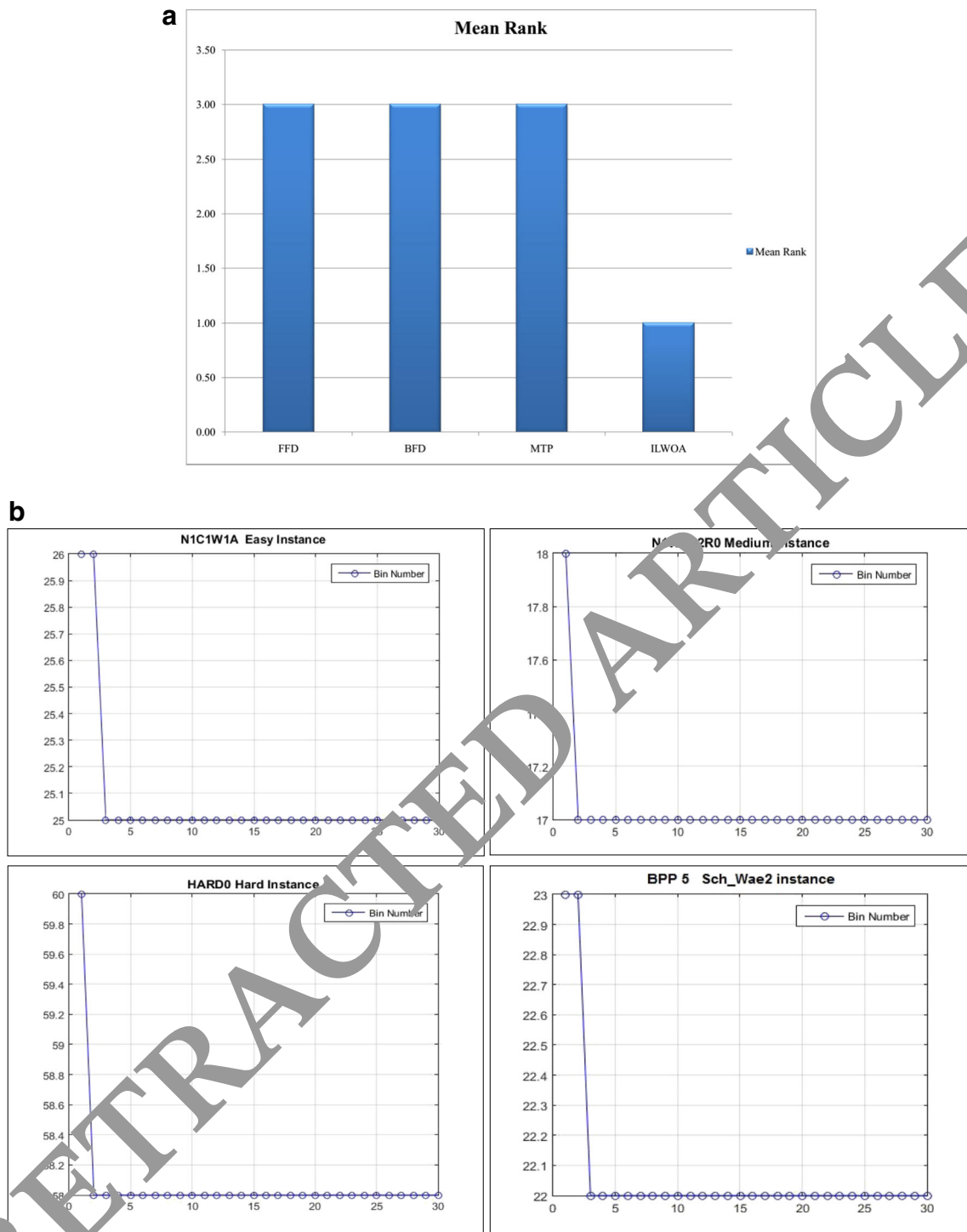


Fig. 11 Friedman test of *Sch_Wae2* instances

10 agents and the maximum number of iterations are set to 30 iterations. The results of FFD and MTP are obtained from [53]. The experimental results of the second test are presented in Table 7. The descriptive statistics of the second experiment and Friedman test results are presented in Table 8 and Fig. 11,

respectively. It is clear that the proposed algorithm is efficiently able to get the optimal solution (bolded values) in most instances compared to the other algorithms. In addition, it scores the lower ranked mean with a significant difference comparing to the other algorithms as shown in Fig. 11.

7 Results of analysis

The values of the bin capacity, items weights, and the number of items mainly affect computational time and solution quality of algorithms [55]. Consequently, the previous issues are taken into consideration when benchmarks are selected in this paper. In the first experiment, different difficulty benchmarks are selected to evaluate the searching capabilities of ILWOA in various cases. For the easy class, the number of items was between 50 and 500 with weights between 1 and 100. The capacity of a bin is between 100 and 150. Also, the number of items is between 50 and 500 in medium class. The capacity of a bin is equal to 1000. The items weights were calculated according to bin capacity that range between bin capacity/9 and bin capacity/3 with a deviation between 20 and 90%. While in the hard class, the number of items is equal to 200. The capacity of a bin is equal to 100,000 and item weights are between 20,000 and 35,000. The second experiment is conducted in Sch_Wae2 instances where the number of items is 120 items with weights ranging between 150 and 200, and the bin capacity is equal to 1000.

The main improvement in ILWOA is mutation phase which significantly increases the convergence speed of WOA. It is clear from the defined number of search agents and iterations. For other improvements, they help in controlling the search in continuous search space which also affects the solution quality.

Obviously, ILWOA obtains rapidly better results in the first experiment instances although the high variation between item weights especially for the hard class characterized by larger values of bin capacity, items weights, and the number of items. This indicates the high stochastic behavior of ILWOA. Also for Sch_Wae2 instances, ILWOA finds the optimal solution in most cases which indicates the good searching performance of ILWOA (see Fig. 12).

8 Conclusion and future works

In this paper, a new variant of WOA is developed and applied for one-dimension BPP. First, the proposed algorithm is adjusted to be applied to solve BPP in combinatorial search

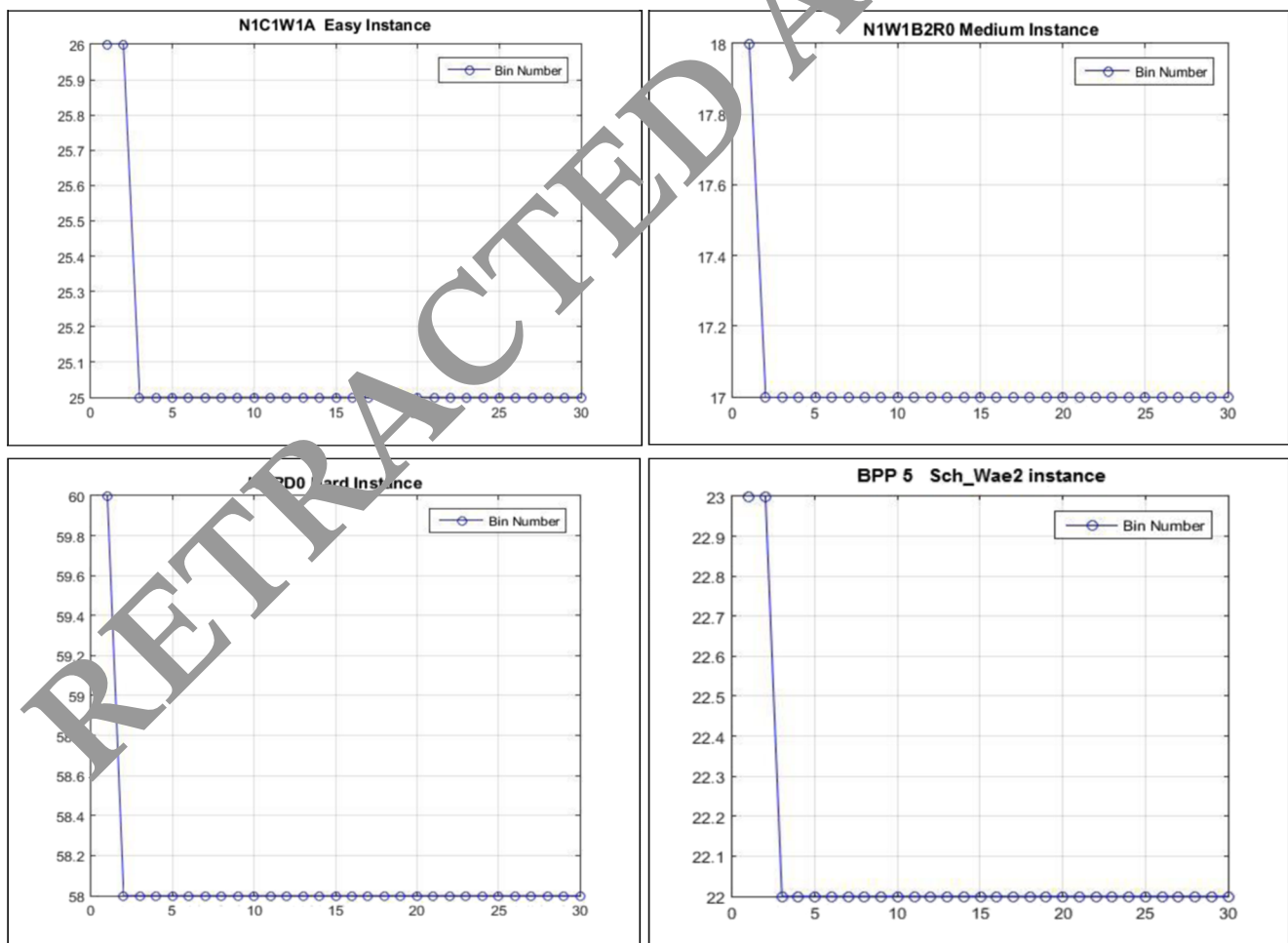


Fig. 12 Examples of ILWOA performance on different problem cases

space. Second, the randomization of the WOA exploration phase is enhanced by utilizing the capabilities of isotropic Lévy flight and chaotic maps. Finally, a mutation strategy is employed for increasing the convergence rate.

Two experiments are conducted and the results are analyzed with non-parametric Friedman test. In the first experiment, ILWOA is compared to QICS, ACS, FCO, FA, and AS. The results of the first experiment prove the superiority of ILWOA in efficiency and rapidity. For the second experiment, the proposed algorithm is compared to FFD, BFD, and MTP. The results show the high efficiency and the ability of the proposed algorithm to find the optimal solution. Besides, the obtained results are discussed and analyzed according to the problem size.

For future work, we suggest applying the proposed algorithm to other variants of BPP. Also, ILWOA can be applied to other combinatorial optimization problems, such as the traveling salesman problem and vehicle scheduling problem. In order to conduct a comprehensive test of ILWOA performance, Internet of Things and cloud computing to manage big data in health services applications [59–63] should be applied in the future.

References

- Papadimitriou CH, Steiglitz K (1998) Combinatorial optimization algorithms and complexity. Prentice-Hall, Inc., Upper Saddle River
- Korte B, Vygen J, Korte B, Vygen J (2012) Combinatorial optimization, vol 2. Springer, Heidelberg
- Coffman EG Jr, Garey MR, Johnson DS (1978) An application of bin-packing to multiprocessor scheduling. *SIAM J Comput* 7(1):1–17
- Perboli G, Gobbato L, Perfetti F (2014) Packing problems in transportation and supply chain: new problems and trends. *Procedia Soc Behav Sci* 111:672–681
- Aggoun A, Rhiat A, Fages F (2016) Perspectives of real-life applications in logistics embedding bin packing optimization algorithms, robotics and cloud computing technologies. In *Logistics Operations Management (GOL) 2016 and International Conference on* (pp. 1–4). IEEE
- Martello S (2014) Two-dimensional packing problems in telecommunication. *Pesqui Operacional* 34(1):31–38
- Song W, Xiat Z, Chen Q, Luo H (2014) Adaptive resource provisioning in the cloud using online bin packing. *IEEE Trans Comput* 63(11):2647–2663
- Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1:67–82
- Holland JC, Pepyne DL (2002) Simple explanation of the no free lunch theorem of optimization. *Cybern Syst Anal* 38(2):292–298
- Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
- Qian B, Wang L, Rong H, Wang WL, Huang DX, Wang X (2008) A hybrid differential evolution method for permutation flow-shop scheduling. *Int J Adv Manuf Technol* 38(7–8):757–777
- Anily S, Bramel J, Simchi-Levi D (1994) Worst-case analysis of heuristics for the bin packing problem with general cost structures. *Oper Res* 42(2):287–298
- Eilon S, Christofides N (1971) The loading problem. *Manag Sci* 17(5):259–268
- Peeters M, Degraeve Z (2006) Branch-and-price algorithms for the dual bin packing and maximum cardinality bin packing problem. *Eur J Oper Res* 170(2):416–439
- Johnson DS, Demers A, Ullman JD, Garey MR, Graham RL (1974) Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM J Comput* 3(4):299–325
- Coffman EG, Galambos G, Martello S, Vigo D (1999) Bin packing approximation algorithms: combinatorial analysis. In *Handbook of combinatorial optimization* (pp. 151–207). Springer US
- Rhee WT, Talagrand M (1993) On line bin packing with items of random size. *Math Oper Res* 18(2):438–445
- Sensarma D, Sarma SS (2014) A novel graph based algorithm for one dimensional bin packing problem. *J Glob Res Comput Sci* 5(8):1–4
- Feigenbaum EA, Feldman J (1995) *Computers and thought*. MIT Press, Menlo Park
- Romanycia MH, Pelletier FL (1985) What is a heuristic? *Comput Intell* 1(1):47–58
- Glover F (1986) Future paths for integer programming and links to artificial intelligence. *Comput Oper Res* 13(5):533–549
- Blum C, Roli A (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Comput Surv (CSUR)* 35(3):268–308
- Falkenauer E, Bouchambre A (1992) A genetic algorithm for bin packing and line balancing. In *Robotics and Automation, 1992. Proceedings. 1992 I.E. International Conference on* (pp. 1186–1192). IEEE
- Falkenauer E (1996) A hybrid grouping genetic algorithm for bin packing. *J Heuristics* 2(1):5–30
- Dempster JR (1976) Adaptation in natural and artificial systems (John H. Holland). *SIAM Rev* 18(3):529–530
- Layeb A, Boussalia SR (2012) A novel quantum inspired cuckoo search algorithm for bin packing problem. *Int J Inf Technol Comput Sci (IJITCS)* 4(5):58–67
- Yang XS, Deb S (2009) Cuckoo search via Lévy flights. In *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on* (pp. 210–214). IEEE
- Zendaoui Z, Layeb A (2016) Adaptive cuckoo search algorithm for the bin packing problem. *Modelling and Implementation of Complex Systems Lecture Notes in Networks and Systems*, 107–120
- Liu B, Wang L, Qian B, Jin Y (2008) Hybrid particle swarm optimization for stochastic flow shop scheduling with no-wait constraint. *IFAC Proceedings Volumes* 41(2): 15855–15860
- Levine J, Ducatelle F (2004) Ant colony optimization and local search for bin packing and cutting stock problems. *J Oper Res Soc* 55(7):705–716
- Dorigo M, Maniezzo V, Colomi A (1996) Ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst Man Cybern Part B (Cybernetics)* 26(1):29–41
- Layeb A, Benayad Z (2014) A novel firefly algorithm based ant colony optimization for solving combinatorial optimization problems. *Int J Comput Sci Appl* 11(2):19–37
- Yang XS (2010) Firefly algorithm, levy flights and global optimization. In *Research and development in intelligent systems XXVI* (pp. 209–218). Springer London
- Sonuc E, Sen B, Bayir S (2017) Solving bin packing problem using simulated annealing. *Proceedings of 65th ISERD international conference, Mecca, Saudi Arabia*
- Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220(4598):671–680
- Martello S, Toth P (1990) *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc

37. Kellerer H, Pferschy U, Pisinger D (2004) Introduction to NP-Completeness of knapsack problems (pp. 483–493). Springer Berlin Heidelberg
38. Epstein L, Levin A (2008) On bin packing with conflicts. *SIAM J Optim* 19(3):1270–1298
39. Twigg A, Xavier EC (2015) Locality-preserving allocations problems and coloured bin packing. *Theor Comput Sci* 596:12–22
40. Hoyt E (2012) Marine protected areas for whales, dolphins and porpoises: a world handbook for cetacean habitat conservation and planning. Earthscan, Washington, DC
41. Whitehead H, Rendell L (2014) The cultural lives of whales and dolphins. University of Chicago Press
42. Hain JW, Carter R, Kraus D, Mayo A, Winni E (1982) Feeding behavior of the humpback whale, *Megapteranovaeangliae*, in the western North Atlantic. *Fish Bull* 80(2):259–268
43. Goldbogen JA, Friedlaender AS, Calambokidis J, Mckenna MF, Simon M, Nowacek DP (2013) Integrative approaches to the study of baleen whale diving behavior, feeding performance, and foraging ecology. *Bioscience* 63(2):90–100
44. Yuan X, Dai X, Zhao J, He Q (2014) On a novel multi-swarm fruit fly optimization algorithm and its application. *Appl Math Comput* 233:260–271
45. Hyde M, Ochoa G, Curtois T, Vázquez-Rodríguez JA (2009) A hyflex module for the one dimensional bin-packing problem. School of Computer Science, University of Nottingham, Tech. Rep
46. Fredriksson L (2010) A brief survey of Lévy walks: with applications to probe diffusion. (Bachelor dissertation). <http://www.divaportal.org/smash/get/diva2:288755/FULLTEXT02.pdf>
47. Mantegna RN (1994) Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes. *Phys Rev E* 49(5):4677–4683
48. Yang XS, Karamanoglu M, He X (2014) Flower pollination algorithm: a novel approach for multiobjective optimization. *Eng Optim* 46(9):1222–1237
49. Devaney R (1989) An introduction to chaotic dynamical systems. Addison-Wesley, Reading, p 13046
50. Banzhaf W (1990) The “molecular” traveling salesman. *Electron J Cybern* 64(1):7–14
51. Michalewicz Z (1996) Genetic algorithms + data structures = evolution programs. Springer, Berlin Heidelberg, New York
52. Grefenstette J, Gopal R, Rosmaita B, Van Vucht D (1985) Genetic algorithms for the traveling salesman problem. In Proceedings of the first International Conference on Genetic Algorithms and their Applications (pp. 160–165)
53. EURO Special Interest Group on Cutting and Packing. (n.d.). Retrieved May 05, 2017, from <https://paginas.fe.up.pt/~esicup/datasets>
54. Gibbons JD, Chakraborti S (2011) Nonparametric statistical inference (pp. 977–979). Springer Berlin Heidelberg, Nonparametric Statistical Inference
55. Scholl A, Klein R, Jürgens C (1997) BISON: a fast hybrid procedure for exactly solving the one-dimensional bin packing problem. *Comput Oper Res* 24(7):627–645
56. Schwerin P, Wäscher G (1997) The bin-packing problem: a problem generator and some numerical experiments with 1-D packing and MTP. *Int Trans Oper Res* 4(5–6):377–399
57. Baker BS (1985) A new proof for the non-fit decreasing bin-packing algorithm. *J Algorithm* 6(1):49–60
58. Simchi-Levi D (1994) New worst-case results for the bin-packing problem. *Nav Res Logist* 41(4):573–585
59. Manogaran G, Varatharajan R, Lopez-Kumar PM, Sundarasekar R, Thota C (2017) Vijayarajam S (ed). In: A new architecture of internet of things and big data system for secured smart healthcare monitoring and alerting. Future generation computer systems. <https://doi.org/10.1016/j.future.2017.10.045>
60. Sundarasekar R, Hsu CH (2017) Machine learning based big data processing in early stage cancer diagnosis using hidden Markov model and GM clustering. *Wireless personal communications*, 1–18. <https://doi.org/10.1007/s11277-017-5044-z>
61. Manogaran G, Varatharajan R, Priyan MK (2017) Hybrid recommendation system for heart disease diagnosis based on multiple kernel learning with adaptive neuro-fuzzy inference system. *Multimedia tools and applications*, 1–21. <https://doi.org/10.1007/s11042-017-5515-y>
62. Varatharajan R, Manogaran G, Priyan MK, Balaş VE, Barna C (2017) Visual analysis of geospatial habitat suitability model based on inverse distance weighting with paired comparison analysis. *Multimedia tools and applications*, 1–21. <https://doi.org/10.1007/s11042-017-4768-9>
63. Gandhi UD, Kumar PM, Varatharajan R, Manogaran G, Sundarasekar R, Kadu S (2018) HIoTPO: Surveillance on IoT devices against recent threats. *Wireless personal communications*, 1–16. <https://doi.org/10.1007/s11277-018-5307-3>