

# A Hybridized Evolutionary Algorithm for Bi-objective Bi-dimensional Bin-packing Problem

Neeraj Pathak<sup>(✉)</sup>  and Rajeev Kumar 

School of Computer and Systems Sciences, Jawaharlal Nehru University,  
New Delhi 110067, India  
neeraj.bhumcal0@gmail.com, rajeevkumar.cse@gmail.com

**Abstract.** The bin-packing problem is a widely studied combinatorial optimization problem. In classical bin-packing problem, we are given a set of real numbers in the range  $(0,1]$  and the goal is to place them in minimum number of bins so that no bin holds more than one. In this paper we consider a bi-dimensional bin-packing in which we are given a set of rectangular items to be packed into minimum number of fixed size of square bins. Here we consider two objectives applied on a bi-dimensional variant, one is related to minimize number of bins and second is minimize average percentage of wastage/gaps in bins. To solve this problem we incorporate the concept of Pareto's optimality to evolve set of solutions using evolutionary algorithm (EA) tool hybridized with the heuristic operator leading to improve results from existing techniques.

**Keywords:** Bin-packing · Bi-objective · Bi-dimensional · Evolutionary algorithms · Pareto's optimality

## 1 Introduction

The bin-packing problem is a special type of cutting stock NP-hard combinatorial problem [1]. In the classical bin-packing problem (BPP) we are given a list  $L = \{i_1, i_2, i_3, \dots, i_n\}$  of real numbers in the range  $(0,1]$  and the aim is to place them in minimum number of bins. Initially the primary aim of bin-packing problem focused only concern to minimize number of bins. However various other constraints and objectives such as weights, priority items, center of gravity can also be included depending on our requirements. The problem holds various applications as in cargo airplanes [12], loading of tractor trailer trucks, routing [4], scheduling [16] and resource allocation problems.

First non-trivial results on bin-packing problem were proposed by Ullman [15] in 1971. He studied the problem starting from memory allocation problems that involve table formatting, file allocation and prepaging. He also gave asymptotic analysis of two

heuristics: FirstFit (FF) and BestFit (BF). In the later year 1972, Garay et al. [7] explained more detailed analysis of four heuristics: First Fit (FF), Best Fit (BF), First Fit Decreasing (FFD) and Best Fit Decreasing (BFD).

In bi-dimensional geometric bin-packing, we are given collection of bi-dimensional items list and the aim is to pack them in minimum number of bins. There are various criteria of packing items in a bin. First is concerned to item shape, it may be regular or irregular. Here we consider the rectangle (regular) shape. Second is concerned to orientation of items allowed or not because it affects the number of bins used, % of gaps etc.

Unlike existing classical bin-packing aiming to consider minimum bins, bi-objective bi-dimensional variant model for bin-packing problem is formulated in this paper with multiple constraints applied on it. As it is impossible to solve bi-dimensional bin-packing problems in polynomial time, therefore many heuristics and metaheuristics [3, 5, 8, 9, 11, 13, 14] are formulated to solve in efficient way. Depending on requirements bin-packing problem definition is formulated. Here we consider two objectives, minimize no. of bins and minimize percentage of wastage of bins applied on bi dimensional geometric variant.

The remainder of this paper is organized as follows: Sect. 2 describes problem definition. Section 3 involves the existing heuristics applied on rectangle packing problem. Section 4 discusses proposed algorithms based on evolutionary algorithm as metaheuristic. Section 5 shows computational results on algorithm. Finally, conclusions and future work possibilities are discussed in Sect. 6.

## 2 Problem Definition

As there is no specific model in literature to design multi-objective variant of bin-packing problem, depending on our requirements mathematical model bi-objective bi-dimensional bin-packing problem is formulated.

We are given unbounded number of bins having fixed height  $H$ , width  $W$  and  $J$  number of rectangle items with dimensions  $w_j \leq W, h_j \leq H$ , the items are packed into bins such that following objectives are achieved, mathematically formulated as:

Minimize number of bins

$$K = \sum_{i=1}^I \text{ceil} \left( \sum_{j=1}^J \frac{x_{ij} h_j w_j}{H_i W_i} \right) \quad (1)$$

Minimize average percentage of wastage/gaps

$$G_{avg} = \frac{1}{K} \sum_{i=1}^K \frac{a_i * 100}{W_i H_i} \quad (2)$$

With applying constraints:

For each item  $j \in \{1, 2, \dots, J\}$ ;

- $0 < w_j \leq W_i$ ; where  $w_j$  is width of item  $j$ ,  $W_i$  is width of bin  $i$ ;
- $0 < h_j \leq H_i$ ; where  $h_j$  is height of item  $j$ ,  $H_i$  is height of bin  $i$ ;

For all bins  $i \in \{1, 2, \dots, I\}$ ;

- $\sum_{j=1}^J x_{ij} w_j h_j \leq W_i H_i$ ;
- Rotation of any item is allowed through  $90^\circ$ ,  $w_j \leftarrow h_j; h_j \leftarrow w_j$  if  $w_j \leq H_i$  and  $h_j \leq W_i$ .
- $\sum_{i=1}^I \sum_{j=1}^J x_{ij} = J$  (for each item should be assigned to only one bin)

Where

- Ceil ( $x$ ), the smallest integer larger or equal to  $x$
- $x_{ij} \in \{0, 1\}$  for  $i = \{1, 2, \dots, I\}$  and  $j = \{1, 2, \dots, J\}$
- $x_{ij} = 1$  if item  $j$  is assigned to bin  $i$
- $x_{ij} = 0$  else.

In the formulation of bi-objective bi-dimensional bin-packing problem definition, percentage of gaps  $G$  is considered as a second objective to evaluate how balanced or stabilized the bins in packing strategy.

### 3 Existing Heuristics for Rectangle Packing

In this section, we describe various heuristic algorithms to fill rectangle item in bin. Those algorithms consist of two parts: first part denotes the number of permutation for which we have selected some standard strategy whether order the items in decreasing height, width or area. Second part denotes decoding algorithm for permutations. A level algorithm is a decoding algorithm in which items are placed from left to right in rows forming levels. Three most popular level algorithms are the Next Fit (NF), Best Fit (BF) and First Fit (FF) strategies.

Let we are given  $n$  rectangular items and current item  $i \in \{1, 2, \dots, n\}$  have to be placed in a bin. The recently created level is  $s$  starting from level 1 bottom of the bin in beginning of algorithm.

#### 3.1 Next Fit (NF)

In this strategy item  $i$  will be placed at left most feasible position of level  $s$  if possible, otherwise goes to next level by creating  $s := s + 1$  and place it left justified as shown in Fig. 1(a).

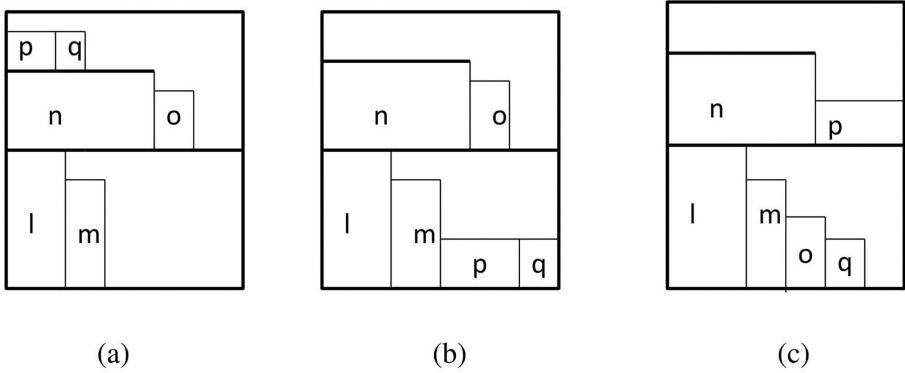
**3.2 Best Fit (BF)**

If there are given multiple unused horizontal area for placing an item  $i$ , place it in that position minimized unused area among all if possible, otherwise follow NF strategy by creating new level as shown in Fig. 1(b).

**3.3 First Fit (FF)**

For an item  $i$  pack through FF check available space starting from level one to the current level  $s$  then place it left justified in first available space if possible, otherwise follow the same procedure as NF by creating next level  $s := s+1$ .

All three procedures are shown in Fig. 1 where we have six rectangle date items of different dimensions  $\{l, m, n, o, p, q\}$ .



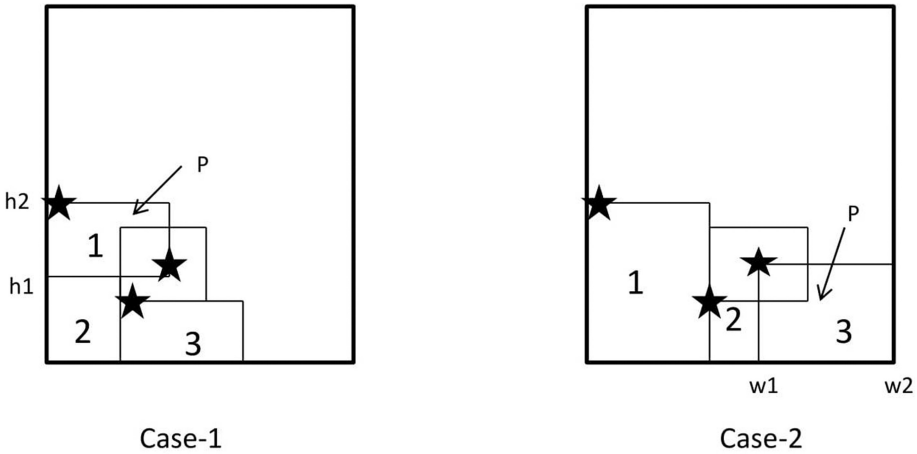
**Fig. 1.** Strip packing problem of (a) Next Fit, (b) Best Fit and (c) First Fit

**3.4 Bottom Left Fill (BLF)**

Bottom left strategy is most documented approach given by Baker et al. [2] in 1980. After that many variants have been proposed in successive few decades, but among all the common characteristics is that items are placed at the bottom left stable position one-by-one. Baker et al. used the concept of Bottom Left Fill (BLF) where item is placed at the left most among lowest possible positions. Jakobs [10] proposed another bottom left (BL) method that holds the item at top right first followed by sliding down then left alternately as long as possible.

**3.5 Overlapping of Items in Bin**

As by definition our aim is to place rectangle items in bin without overlapping fashion. So to avoid this there are two cases as described below:\newline



**Fig. 2.** Overlapping of new item with point P for case-1 and case-2

Case 1: let  $(x_i, y_i)$  is insertion point of incoming item to be placed and  $h_1, h_2$  is the intersection coordinate of item 1 to left side of bin. Therefore mathematically if  $h_1 \leq y_i \leq h_2$  overlapping exist.

Case 2: let  $(x_i, y_i)$  is insertion point of incoming item to be placed and  $w_1, w_2$  is the intersection coordinate of item 3 to bottom side of bin. If  $w_1 \leq x_i \leq w_2$  overlapping exist.

All two cases of overlapping are depicted in Fig. 2 where  $P(x_i, y_i)$  is the point of overlapping and star represents insertion points for new coming rectangle item.

#### 4 Metaheuristics for 2-D Item Packing

In the last three decades, many local search, heuristics and metaheuristics have been proposed on different variants of bin-packing problem. Dowsland [6] gave metaheuristics for 2-D item packing using simulated annealing (SA) with results of feasible and infeasible solutions. Jakobs proposed a metaheuristics for strip packing using genetic algorithm (GA) to find good permutation.

Here we have proposed a hybridized evolutionary algorithm to simulate the bottom left fill (BLF) heuristic with genetic algorithm applied on bi-objective bi-dimensional BPP. BLF heuristic is shown in Algorithm 1.

**Algorithm 1 BLF: (Bottom Left Fill Heuristic)**

**Input:** set of item  $I$ ,  $K$  number of available bins and archive of insertion points.  
**Output:** updated set of insertion points and number of bins  $K'$   
**for** each rectangle item  $i \in \{1, 2, \dots, I\}$  **do**  
  **for** each available bin **do**  
    **for** each insertion point of insertion archive **do**  
      **if** overlapping then  $\rightarrow$  insert new insertion point in the insertion archive  
      **else**  $\rightarrow$  insert item  $i$  at current insertion point, delete current insertion point and add two new insertion point one at top left and one at bottom right  
      **end if**  
    **end for**  
  **end for**  
  **if** no bins to hold item  $i$  then  $\rightarrow$  create new bin with insertion of item  $i$  in it and increment  $K$  as  $K'$   
  **end if**  
**end for**

**4.1 Proposed Hybridized Evolutionary BLF Algorithm**

In this section we have proposed an hybridized evolutionary algorithm of BLF in Algorithm 2 where  $K$  is number of available bins to accommodate  $J$  rectangle items with minimum average % gaps ( $G_{avg}$ ). Here we apply evolutionary algorithm tool as metaheuristic to implement bi-objective bi-dimensional bin-packing problem.

**Algorithm 2 Hybridized Evolutionary BLF**

**Input:** set of items  $J$  packet through BLF,  $K$  number of available bins with average percentage gaps  $G_{avg}$ .  
**Output:** ( $K', G'_{avg}$ ) optimized number of bins with minimum % of gaps.  
**While** it  $\neq$  iteration **do**  
  **Crossover**  $k'_c$  = select bins having (% gaps  $\leq 10$ ) from the set  $k$ , and  $k'' = k - k'$  and apply inter-crossover of single item from the set  $k'$   
  **Mutation**  $k'_m$  = select any bin from the set  $k'$  and apply mutation within bin through rotation or interchange of items  
  **Merging**  $k' = \text{merge}(k'_m, k'_c)$  and apply BLF (select item  $j$  from the set  $k''$  and place it in  $k'$ )  
  Update  $k = k' + k''$   
  it = it + 1  
**end while**

**Selection:** Selection criteria depends on fitness function as specified by problem definition. Initially we are given  $K$  and  $G_{avg}$  as input to selection procedure. As we have taken the percentage of gaps, second objective as fitness function we can break the initial set of  $K$  into two different groups one having gaps percentage  $\leq 10$  as  $K'$  and other having percentage  $> 10$  as  $K''$ . This procedure gives output as  $K'$ .

**Crossover:** For obtained set  $K'$  we apply inter flipping of rectangle item between two bins selected randomly from set  $K'$  to minimize percentage of gaps  $G$ . It gives output  $k'_c$  as new updated set.

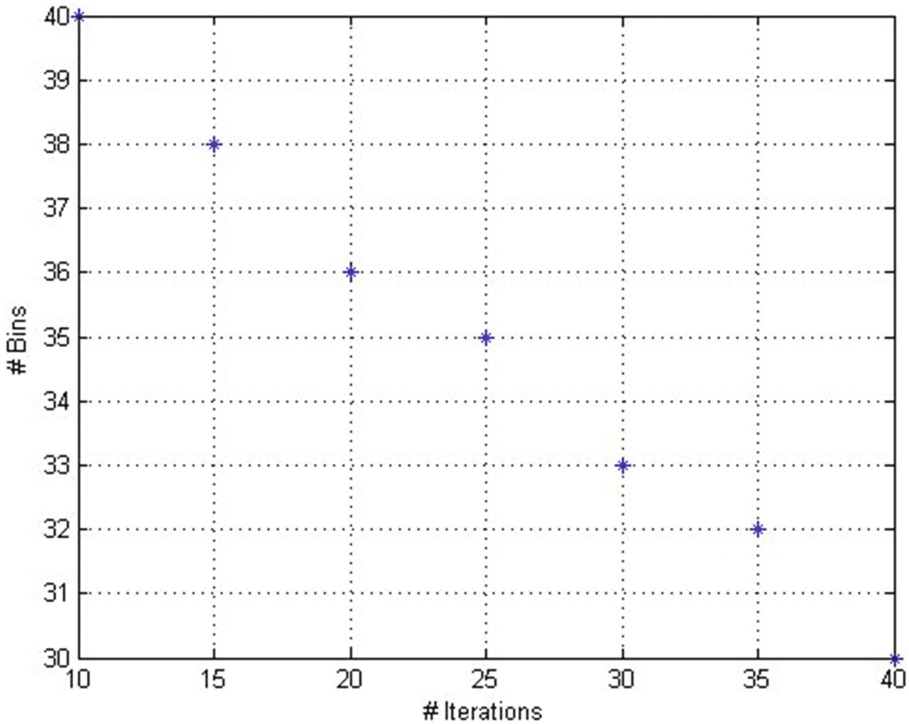
**Mutation:** This involves the intra flipping (within bin) of rectangle item to minimize gaps through rotation or exchanging positions of it. Output of mutation procedure is given as  $k'_m$ .

**Merging and Update:** In this procedure we merge solution set  $k'_m$  and  $k'_c$  to form new  $K'$ . Now to reduce set  $K''$  apply BLF on new  $K'$  using item  $i$  in  $K''$ . Output of this procedure gives updated set  $K$  with more optimized comparing to previous step.

**Termination:** Depending on problem definition we have to terminate the algorithm procedure. Here we considered number of iterations as termination condition that control the completion of evolution method.

## 5 Computational Results

We have implemented the Algorithm 1 for small dataset. Here we have selected random numbers in the range  $[1, 10]$  to initialize dimensions (width and height) of 50 rectangle items. The bin's dimension (width, height) is fixed to 10. We show the results in following graphs. Figure 3 shows number of bins used versus the iterations. In Fig. 4 shows average percentage of gaps versus number of iterations.



**Fig. 3.** Plot 1 of number of iterations versus number of bins.

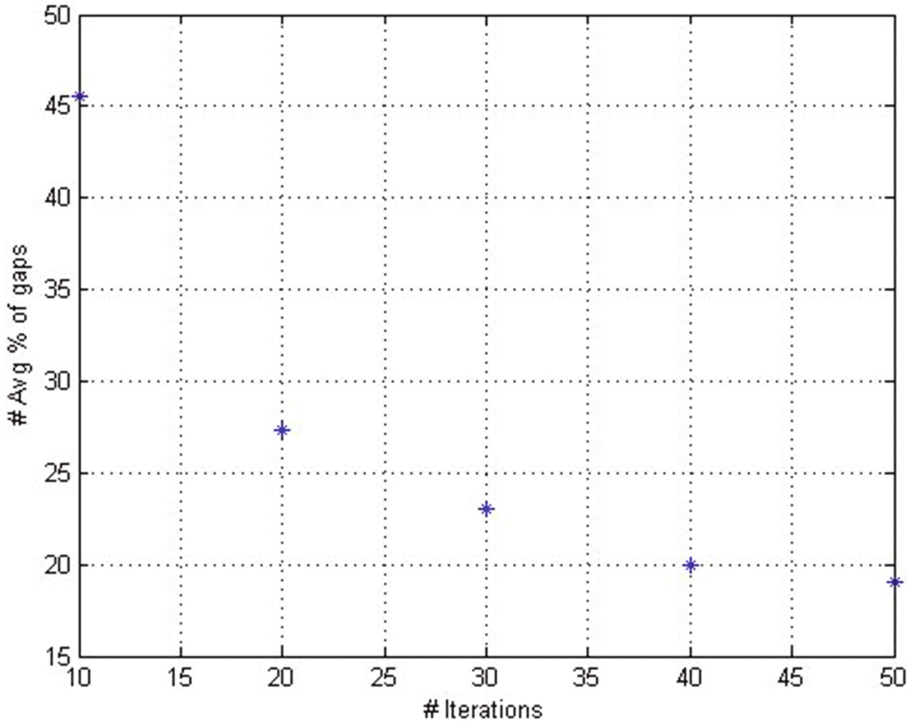


Fig. 4. Plot 2 of number of iterations versus avg % of gaps.

## 6 Conclusions

In this paper, a hybridized model for bi-objective bi-dimensional BPP is presented and the evolutionary algorithm is proposed to solve the problem. In this algorithm, we have considered bottom left fill (BLF) heuristic as operator to evolve better solutions as per definition of problems. We have implemented it on small data set. We are currently working on large data sets with some challenges to verify the final results as per required for proposed algorithm. The problem could be more complex if we consider irregular item in place of rectangle because finding exact position to minimize space wastage is challenging. We are working to formulate mathematical model to find solution criteria in this regard.

## References

1. Anily, S., Bramel, J., Simchi-Levi, D.: Worst-case analysis of heuristics for the bin-packing problem with general cost structures. *Oper. Res.* **42**(2), 287–298 (1994)
2. Baker, B.S., Coffman Jr., E.G., Rivest, R.L.: Orthogonal packings in two dimensions. *SIAM J. Comput.* **9**(4), 846–855 (1980)



3. Bortfeldt, A.: A genetic algorithm for the two-dimensional strip packing problem with rectangular pieces. *Eur. J. Oper. Res.* **172**(3), 814–837 (2006)
4. Chandra, A.K., Hirschberg, D.S., Wong, C.K.: Bin-packing with geometric constraints in computer network design. *Oper. Res.* **26**(5), 760–772 (1978)
5. Chang, Y.-C., et al.: B\*-Trees: a new representation for non-slicing floorplans. In: *Proceedings of the 37th Annual Design Automation Conference*. ACM, New York (2000)
6. Dowland, K.A.: Some experiments with simulated annealing techniques for packing problems. *Eur. J. Oper. Res.* **68**(3), 389–399 (1993)
7. Garey, M.R., Graham, R.L., Ullman, J.D.: Worst-case analysis of memory allocation algorithms. In: *Proceedings of the Fourth Annual ACM Symposium on Theory of Computing*. ACM, New York (1972)
8. Imahori, S., Yagiura, M., Ibaraki, T.: Local search algorithms for the rectangle packing problem with general spatial costs. *Math. Program.* **97**(3), 543–569 (2003)
9. Imahori, S., Yagiura, M., Ibaraki, T.: Improved local search algorithms for the rectangle packing problem with general spatial costs. *Eur. J. Oper. Res.* **167**(1), 48–67 (2005)
10. Jakobs, S.: On genetic algorithms for the packing of polygons. *Eur. J. Oper. Res.* **88**(1), 165–181 (1996)
11. Lesh, N., et al.: New heuristic and interactive approaches to 2D rectangular strip packing. *J. Exp. Algorithmics* **10**, 1–2 (2005)
12. Mongeau, M., Bes, C.: Optimization of aircraft container loading. *IEEE Trans. Aerosp. Electron. Syst.* **39**(1), 140–150 (2003)
13. Murata, H., et al.: VLSI module placement based on rectangle-packing by the sequence-pair. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **15**(12), 1518–1524 (1996)
14. Minorikawa, H., Sawa, S.S.: Current status and future trends of electronic packaging in automotive applications. In: *Proceedings of the International Congress on Transportation Electronics, 1990. Vehicle Electronics in the 90's*. IEEE (1990)
15. Ullman, J.D.: *The Performance of a Memory Allocation Algorithm*. Computer Science Laboratory, Department of Electrical Engineering, Princeton University, Princeton (1971)
16. Van De Vel, H., Shijie, S.: An application of the bin-packing technique to job scheduling on uniform processors. *J. Oper. Res. Soc.* **42**, 169–172 (1991)