A Constructive Heuristic for Two-Dimensional Bin Packing

Bohan Wang
School of Computer Science and Technology, Huazhong University of Science and technology, Wuhan, 430074, China
Email: Wangbh11@gmail.com

Jiamin Liu
School of Information Science and technology, Shenyang University of technology, Shenyang, 110870,China
Email: Christina.liu48@gmail.com

Yong Yue Malcolm Keech
Faculty of Creative Art, Technologies and Science, University of Bedfordshire, Luton, LU1 3JU, England
Email: yong.yue@beds.ac.uk
keech.malcolm@beds.ac.uk

***Abstract*—Two-dimensional bin packing is encountered in various applications where small rectangular items are packed into a minimum number of large rectangular objects (bins). Aiming at an optimal area utilization, the paper presents an effective constructive heuristic approach to two-dimensional bin packing. The heuristic approach integrates ranking, placement and search strategies along with an effective handling method of the remaining areas during the packing process. In order to obtain an optimal arrangement in a given area, all items are examined for possible positions and orientations using the search strategies. In addition, an effective handling method of the remaining areas is applied through appropriate partitioning and merging to minimize waste areas. Tests with a number of standard test and real world instances have shown that the performance of the proposed approach is superior to that of other approaches published.**

Keywords—heuristic; bin packing; remaining area; utilization

I. INTRODUCTION

The two-dimensional bin packing problem (2DBPP) is to find a set of identical rectangular stocks (normally called bins) to pack a given set of rectangular items where the minimum number of bins is used as the objective. It belongs to cutting and packing problem [1]. There are various application areas involving different constraints and objectives. According to specific applications, several variants can arise, but in most cases the following additional requirements are necessary: (1) *Orientation*: The items may either have a fixed orientation or be rotated by 90°; (2) *Guillotine cuts*: It may be imposed that the items are obtained through a sequence of edge-to-edge cuts parallel to the edges of the items; otherwise the cutting is non-guillotine.

According to a review of 2D bin packing [2-4], several exact methods and lower bounds have been proposed for the problem, e.g. lower bound [5, 6], exhaustive branch-and-bound [7] and constraint programming [8]. However, 2D bin packing is a natural generalisation of the classical 1D problem, and is NP-hard in a strong sense. Most of the published literatures involved in heuristics, e.g. finite bottom-left (FBL) [9], alternate direction [10], touching perimeter (TP) [10], improved level heuristic [11] and an approximation algorithm [12]. Nowadays meta-heuristic approaches are frequently used for the approximate solution of hard combinatorial optimisation problems. Some research exploits meta-heuristic to solve 2D bin packing problems, e.g. tabu search [13], evolutionary particle swarm [14] and generic algorithm [15].

Although some effective procedures for 2D bin packing already exist, the development of methods able to achieve an optimal or near optimal solution remains a challenge. Heuristic approaches are implemented effectively based on heuristic strategies at a low computational cost, but they are local optimal methods. Meta-heuristics can obtain a global optimisation theoretically, but with a longer computational time, and sometimes non-convergence leads to the failure of solution. Whereas exact algorithms achieve optimal results, they are limited by the scale and constraints of the problem. This paper presents an effective approach to 2D bin packing. The approach is an effective constructive heuristic method which combines the heuristic strategies for an optimal arrangement of the items and a placement procedure for the area handling.

II. DEFINITION OF THE PROBLEM

There is a set of n types of rectangular items, where each item is defined by length l_i and width w_i, and its quantity is denoted by m_i, $i = \{1, 2, \dots, n\}$. An unlimited number of identical large rectangular bins B is available, each having length L and width W. It is assumed that a bin is located in a two-dimensional coordinate system with x- and y-axes. Its bottom-left corner is the origin $O(0, 0)$ of the coordinate system. The objective of the problem is to pack all items into a minimum number of bins.

Let (x_i, y_i) be the coordinates of the bottom-left vertex of item i. Item i is described by (x_i, y_i, l_i, w_i). Likewise, a remaining (or waste) area A in a bin, which is an unfilled area, is described by (x_a, y_a, L_a, W_a), where x_a and y_a are the coordinates of the bottom-left vertex of area *A*, and L_a and W_a are the length and width of area *A*, respectively.

The problem addressed in this paper takes into account the following requirements:

- Each item may be packed at any location inside a bin orthogonally. Any two items can touch each other without overlapping.
- Non-guillotine cut: the items cannot be obtained through a sequence of edge-to-edge cuts parallel to the edges of the items.
- It is assumed, without the loss of generality, that the dimensions of items and bins, and the coordinate values of the items in the bin are integers. Any clearance between the edges of two adjacent items can be neglected.

Therefore, the objective of 2D bin packing is to minimise the number of used bins where a given set of items is packed completely without overlapping under the above requirements.

III. CONSTRUCTIVE HEURISTIC

A. Heuristic strategies

Some heuristic approaches published pack items into a level, and each level is then packed into a bin in 1D, which major disadvantage is the values for each decision variable are assigned according to the optimal solution at every step. Consequently, some wasted areas are likely to exist in each level. Although some non-level approaches can pack items along the contour consisting of a mixture of edges of a bin and items in order to reduce the wasted area in levels[9, 10], they often exploit a greedy search strategy to find an optimal solution, which still leaves a considerable room for improvement of area utilisation. This approach, however, applies heuristic strategies for determining the best solution in each given area. The items with the optimal solution are packed into the area generated by the effective handling method for remaining areas. The approach employs a combination of ranking, placement and search strategies to achieve the optimal solution.

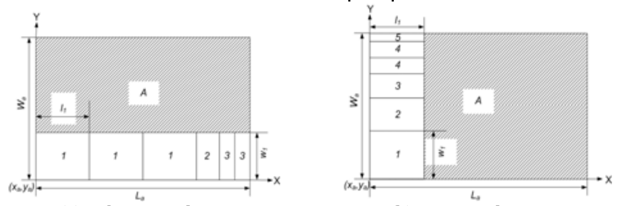
1) Ranking strategies: To avoid items being searched randomly, a ranking strategy is applied. Items are sorted in decreasing order of their lengths. The length of each item is more than or equal to its width. The items with the same length are sorted again in decreasing order of the width. Furthermore, the items are divided into two groups according to their areas to allow the large items to be packed first and therefore to reduce areas wasted. The items are sorted first in decreasing order of their areas, and are then divided according to a given parameter $\alpha, \alpha \in [0.1]$

2) Placement strategies: The sequential packing is that the bins are packed one after another. After the items are packed into the current bin, the remaining items are packed into the next bin until all items are packed completely. The minimum number of used bins is obtained by the maximum area utilisation of each bin. The first item chosen is always packed in the bottom-left corner of the given area.

3) Search strategies: Before the items are packed, all items form two lists of unpacked items according to the ranking strategies. Each item is placed in a given area *A* of a bin, i.e. A = (x_a, y_a, L_a, W_a), based on the placement strategies. All feasible solutions, i.e. the sequences of the packed items in area *A*, are examined by the evaluation criteria of each search strategy. The three search strategies are detailed as below:

Strategy 1 - Complete area packing. This strategy is to find one or more identical items which fill completely area $A, A \in B$, and not produce any remaining area after these items are packed.

Strategy 2 - Complete strip packing. This strategy is to combine identical and different items with the same dimension on one side to form a strip along the horizontal or vertical direction of a given area A (Fig. 1), and a remaining area (shadow area in Fig. 1) is then produced after the horizontal or vertical strip is packed.



(a) A horizontal strip (b) A vertical strip

Figure 1: Complete strip packing.

Strategy 3 – Incomplete vertical strip packing. In many cases, it is impossible to fill a complete strip, and this strategy is used, where identical and different items are combined together and packed along the vertical direction of a given area *A*. Items are packed to form a strip, but not fully filling up the strip, as shown in Fig.2.

To evaluate the solutions efficiently in strategies 2 and 3, the evaluation function *f* is formulized as below:

$$f = l_1 \times W_a - \sum_{i=1}^{r} k_i \times l_i \times w_i \qquad (1)$$

where k_i is the number of packed item type *i* in area *A*; l_1 is the length (or width) of the first chosen item; *r* is the number of the item types which can be packed in area *A*. If *f* is equal to zero, strategy 2 is satisfied. Otherwise, min *f* is considered as an optimal solution.

The combination of the items is considered in search strategies 2 and 3. A backtracking algorithm with bounds is employed to ensure an optimal solution. This procedure is implemented by using a depth-first search algorithm. The first item is chosen from the group of large items when the items are combined by using strategies 2 and 3. This is because the size of each strip is determined by the first item. Thus any optimal arrangement is found in a large area.

B. Handling method for remaining areas

A remaining area (shadow area in Fig.1 or Fig.2) will be produced after identical or different items are packed when strategies 2 and 3 are used. Although it is possible to describe the non-rectangular area in a mathematical representation, it is not easy to manage it for iterative evaluations. Therefore, a suitable handling method for the remaining area needs to be developed to ensure the iteration.

The handling method for the remaining area includes partitioning and merging of the remaining areas.

Before any item is packed into a bin, the area of the initial remaining area is equal to that of the bin. If the combined items can form a horizontal strip, the current packing area will be partitioned into two areas $A_1 = (x_a, y_a, L_a, w_1)$ and $A_2 = (x_a, y_a,+w_1, L_a, W_a-w_1)$ (Fig. 1(a)). Area A_2 is stored as a remaining area. In vertical partitioning, an area is partitioned along the *y*-direction. If the combined items form a vertical strip, the current packing area will be partitioned into two areas $A_1 = (x_a, y_a, l_1, W_a)$ and $A_2 = (x_a+l_1, y_a, L_a-l_1\ W_a)$ (Fig. 1(b)). Once the items are packed to form a non-rectangular remaining area, as shown in Fig. 2, the vertical partition is used in the current packing area (Fig. 3).

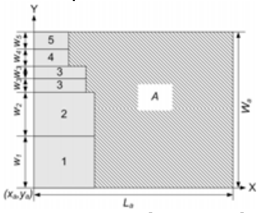


Figure 2. Incomplete vertical strip packing.

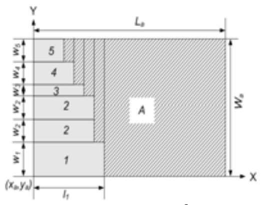


Figure 3. Partition of a remaining area.

During packing, there exist two kinds of remaining areas after partitioning. One is regarded as the waste area which cannot be used to pack any items. The other is the remaining area which can be used to pack items. The more the items are packed, the more waste areas are produced. Moreover, the remaining areas become fragmented with partitioning. Although some remaining areas can be used, they are often not large enough to pack large items or a set of combined small items. Therefore, it is necessary to merge the waste areas with adjacent remaining areas where possible in order to make the waste areas useful once again.

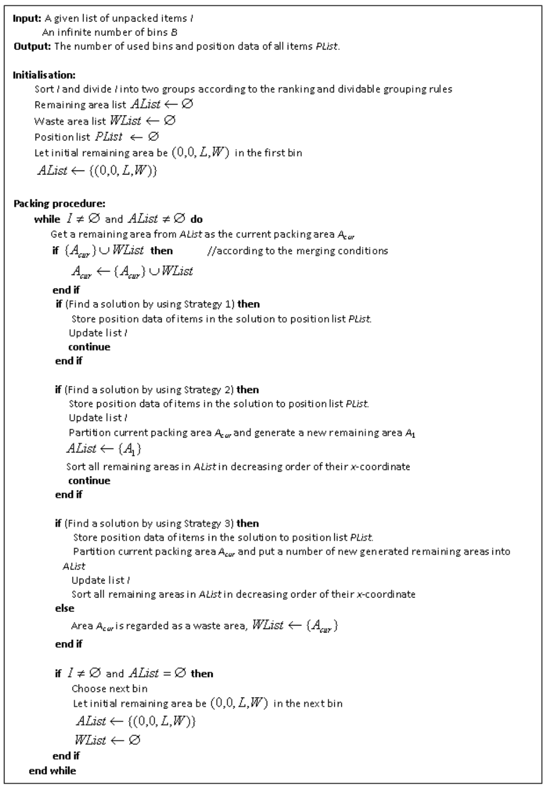
Taking into account the computational time for merging these areas, two situations are considered:

- If there is a given area where an item can be packed, and there exists an adjacent remaining area to its left, they are merged to form a larger area to reduce the number of small areas. In a particular case, a given area is considered to merge with an adjacent remaining area to its right.
- If no item can be packed into a given area, it is considered as a temporary waste area and is then taken into account to merge with other remaining areas during the packing.

C. *The procedure of packing*

First of all, a serial of collections are initialized according to the ranking strategies. Then according to the placement strategies and search strategies introduced above, the packing procedure is processed.

During the packing, the remaining areas are merged according to the merging conditions of different cases. After merging, all remaining areas in list *AList* will be sorted in decreasing order of their *x*-coordinates. When the items chosen by three search strategies are packed according to placement strategy, the coordinate of the bottom-left corner of each item is calculated in terms of the location of the remaining area and the number of chosen item. The remaining areas will be then partitioned after packing. Before each time of packing iteration, the remaining areas are checked for merging. After the iteration, the remaining areas are partitioned once again, and new remaining areas are then generated. The overall procedure of packing can be described by pseudo-code (Fig. 4).



```
Input: A given list of unpacked items I
       An infinite number of bins B
Output: The number of used bins and position data of all items PList.

Initialisation:
    Sort I and divide I into two groups according to the ranking and dividable grouping rules
    Remaining area list $AList \leftarrow \varnothing$
    Waste area list $WList \leftarrow \varnothing$
    Position list $PList \leftarrow \varnothing$
    Let initial remaining area be $(0,0,L,W)$ in the first bin
    $AList \leftarrow \{(0,0,L,W)\}$

Packing procedure:
    while $I \neq \varnothing$ and $AList \neq \varnothing$ do
        Get a remaining area from AList as the current packing area $A_{cur}$
        if $\{A_{cur}\} \cup WList$ then        //according to the merging conditions
            $A_{cur} \leftarrow \{A_{cur}\} \cup WList$
        end if
        if (Find a solution by using Strategy 1) then
            Store position data of items in the solution to position list PList.
            Update list I
            continue
        end if

        if (Find a solution by using Strategy 2) then
            Store position data of items in the solution to position list PList.
            Update list I
            Partition current packing area $A_{cur}$ and generate a new remaining area $A_1$
            $AList \leftarrow \{A_1\}$
            Sort all remaining areas in AList in decreasing order of their x-coordinate
            continue
        end if

        if (Find a solution by using Strategy 3) then
            Store position data of items in the solution to position list PList.
            Partition current packing area $A_{cur}$ and put a number of new generated remaining areas into
          AList
            Update list I
            Sort all remaining areas in AList in decreasing order of their x-coordinate
        else
            Area $A_{cur}$ is regarded as a waste area, $WList \leftarrow \{A_{cur}\}$
        end if

        if $I \neq \varnothing$ and $AList = \varnothing$ then
            Choose next bin
            Let initial remaining area be $(0,0,L,W)$ in the next bin
            $AList \leftarrow \{(0,0,L,W)\}$
            $WList \leftarrow \varnothing$
        end if
    end while
```

Figure 4. The pseudo-code of the overall packing procedure.

IV. COMPUTATIONAL EXPERIMENTS

The heuristic is implemented in Visual C++ under Windows XP. All computational experiments are carried out on a laptop with Intel Centrino Duo CPU 1.66GHz and RAM 1GB.

A. *Comparison with the published*

The heuristic is tested with six classified instances proposed by Berkey and Wang [8]. Each class is characterized by a different size of the bins and by the ranges in which dimensions of the items are randomly generated. For each class and value of *n* (the quantity of item types *n* = 20, 40, 60, 80,100), 10 instances are generated. The 300 instance in the six classes are available at: http://www.or.deis.unibo.it/research_pages/ORinstances/2BP.html.

In order to evaluate the approach, a comparative analysis is made between the current work and other approaches. Fig. 5 shows the performance chart of the six approaches for six classified instances. The other five approaches are respectively B&W-FFF[9], B&W-FBS[9], Lodi-FC[10], Lodi-TP[10] and Lodi-TS[13]. The vertical axis in Fig. 5 is the average ratio of solution and lower bound computed over the fifty instances for each class. The lower bound is a value below the optimal solution. The horizontal coordinate is the six classified instances. The closer to 1 the lines in Fig.5 are, the better the solution is. The test results have shown that the solution obtained by this heuristic significantly improves the published work. The average computational time by this heuristic is 0.494 seconds for 300 instances.

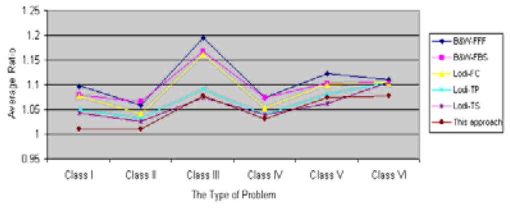
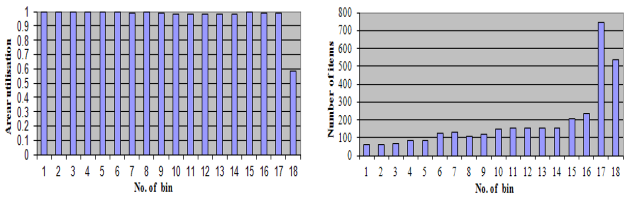


Figure 5. The comparison chart of six approaches for six classes.

B. Experiments for the real world data set

The heusitic is tested further using a real world data set provided by an industry company. The data set has 55 types of items and 3329 numbers of items. All item edges range between [70, 1020] and [50, 185] respectively, and their numbers of each type range between 1 and 250. These items are packed into a number of bins with the length of 3658mm and width of 1220mm. For the practical problem, the objective is to obtain simultaneously both the maximum utilisation in each bin and the minimum number of bins.

The results obtained by the approach in 9.31seconds are shown in Fig. 6. As can be seen, 18 bins are used to pack all items. The area utilisation of each bin is achieved over 98%, but last one (Fig. 6(a)). Although the approach obtains the utilisation of 57.97% in the last bin, the remaining area in the bin will be utilised again once in the next industrial schedule. The sequential approach caters for the practical requirement. Observation from the figures have shown that the large items are first packed into the previous bins because there are less large items to be packed in the previous bins than later those, whilst more small items are packed into the last two bins (Fig. 6(b)).



(a) The distribution of the area utilization for each bin (b) The distribution of the packed item number for each bin

Figure 6. The results by the approach for the real world data.

V. CONCLUSIONS

This paper has presented an effective heuristic approach to 2D bin packing. The heuristic strategies are proposed to obtain an effective solution. To ensure a maximum area utilisation, a flexible handling method for the remaining areas is developed. Through tests with a number of standard test instances, it has been shown that the performance of the approach is superior to that of other approaches published.

The sequential packing strategy of the approach makes it possible for each bin to achieve the maximum area utilisation. It is also feasible to apply the approach to the strip packing problem. Moreover, the approach can be easily improved in simultaneous packing because the handling method for remaining areas is suitable for packing items into many bins at the same time.

ACKNOWLEDGMENTS

The authors would like to thank the participants and the anonymous reviewers for their comments and feedback. This research was supported by the Science & Technology Department of Liaoning Province in China (Grant No. 201102164).

REFERENCES

[1] G. Wӓcher, H. Hauβer and H. Schumann, “An improved typology of cutting and packing problems,” Eur. J. Oper. Res, vol. 183, pp. 1109 -1130, December 2007.

[2] A. Lodi, S. Martello and M. Monaci, “Two-dimensional packing problems: a survey,” Eur. J. Oper. Res, vol. 141, pp. 241-252, September 2002.

[3] E. Hopper and B.C.H. Turton, “A review of the application of meta-heuristic algorithms to 2D strip packing problem,” Artif. Intell. Rev, vol.16, pp. 257-300, 2001.

[4] L.J. Wei,W.C. Oon,W.B. Zhu and A. Lim,”A goal-driven approach to the 2D bin packing and variable-sized bin packing problems,” Eur. J. Oper. Res, vol. 224, pp. 110-121, January 2013

[5] M. Amico, S. Martello, and D. Vigo, “A lower bound for the non-oriented two-dimensional bin packing problem,” Discrete Applied Mathematics, vol.118, no.1-2, pp.13-24, 2002.

[6] Martello and D. Vigo, “Exact solution of the two-dimensional finite bin packing problem,” Manage. Sci, vol.44, no.3, pp.388-399, 1998.

[7] N. Lesh, J. Marks, A. McMahon, and M. Mitzenmacher, “Exhaustive approaches to 2D rectangular perfect packing,” Inf. Process. Lett, vol.90, no.1, pp.7-14, 2004.

[8] D. Pisinger and M. Sigurd., “Using decomposition techniques and constraint programming for solving the two-dimensional bin-packing problem,” INFORMS J. Comput, vol. 19, no. 1, pp. 36-51, 2007.

[9] J.O. Berkey and P.Y. Wang, “Two-dimensional Finite bin-packing algorithms”, J. Oper. Res. Soc, vol. 38, no. 5, pp. 423-429, 1987.

[10] A. Lodi, S. Martello, and D. Vigo, “Heuristic and meta-heuristic approaches for a class of two-dimensional bin packing problems”, INFORMS J. Comput, vol. 11, no. 4, pp. 345-357, 1999.

[11] F. G. Ortmann, N. Ntene and J. H. van Vuuren, “New and improved level heuristics for the rectangular strip packing and variable-sized bin packing problems,” Eur. J. Oper. Res, vol. 203, pp. 306-315, June 2010.

[12] G.C. Zhang, “A 3-approximation approach for two-dimensional bin packing,” Oper. Res. Lett, vol. 33, no. 2, pp. 121-126, 2005.

[13] A. Lodi, S. Martello and D. Vigo, “Recent advances on two-dimensional bin packing problems,” Discrete. Appl. Math, vol. 123, no. 1-3, pp. 379-396, 2002.

[14] D.S. Liu, K.C. Tan, S.Y. Huang, C.K. Goh and W.K. Ho, “On solving multiobjective bin packing problems using evolutionary particle swarm optimization,” Eur. J. Oper. Res, vol. 190, pp. 357–382, October 2008.

[15] A. Soke and Z. Bingul, “Hybrid genetic algorithm and simulated annealing for two-dimensional non-guillotine rectangular packing problems,” Eng. Appl. Artif. Intel, vol. 19, no. 5, pp. 557–567, 2006.