POLITECNICO MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

EXECUTIVE SUMMARY OF THE THESIS

Three-dimensional bin packing with vertical support

LAUREA MAGISTRALE IN COMPUTER SCIENCE AND ENGINEERING - INGEGNERIA INFORMATICA

Author: JACOPO LIBÈ

Advisor: PROF. OLA JABALI

Co-advisor: DAVIDE CROCI

Academic year: 2021-2022

1. Introduction

Recent progress in the digitalization of industrial processes led to a rise in studies on the Three-Dimensional Bin Packing Problem (3D-BPP). The problem consists in packing a set of three-dimensional items in the minimum number of bins without any overlap. The 3D-BPP finds many applications in real-world industrial settings, such as loading boxes onto pallets (Pallet Loading Problem) or in containers (Container Loading Problem). Studies in such application areas mentioned above have shown that 3D-BPP can be successfully applied to industrial problems as long as it is extended with new practical constraints. Many works in the literature state that one of the most important practical constraints is vertical support (?), i.e., the fact that each item has sufficient support to keep it from falling. In this thesis, we develop a new extension of the 3D-BPP that ensures vertical support for each item, we call this problem the Three-Dimensional Bin Packing Problem with Vertical Support (3D-BPPVS). Similarly to other works in the literature (??), we model this property as a constraint on the minimum amount of area or vertices that rest on other items.

Our research stems from the case study of a logistics company in northern Italy. The company manages large warehouses where automated lines bring boxes to different packing stations, and then they are loaded onto pallets of standard size. Since the company deals directly with customers' orders, boxes have very different sizes and are usually packed in smaller quantities. Moreover, the assortment of items to pack is strongly heterogeneous. Therefore, using standard approaches based on layers of homogeneous boxes is impractical. During the palletization, levels of already packed items are wrapped to ensure the better overall stability of the pallet. This wrapping procedure requires minimal unused space between items, a property that is measured with a metric called cage ratio. The cage ratio is the ratio between the volume of the packed items inside a bin and the volume of the cuboid which surrounds them, the cage. The cage has the same base as the bin and height equal to the highest packed item inside the bin. The company's current commercial solutions pack items with an average cage ratio of 60%, and a target of 70% was set as a baseline for our work. A visual representation of the cage ratio is shown in ??.

In this thesis we

- formulate a mixed-integer linear programming model for the 3D-BPPVS with a discretized version of the support constraints,

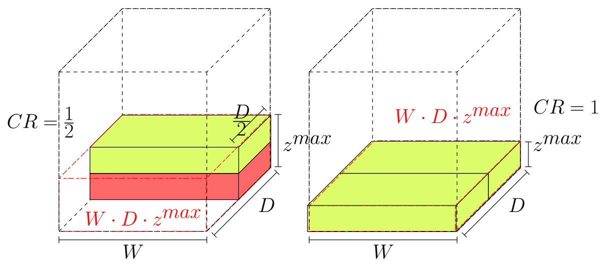

Figure 1: Cage ratio of two different bin configurations

- develop a constructive heuristic that fills bins while guaranteeing vertical support, without explicitly building layered solutions,
- introduce a beam search heuristic that expands the constructive heuristic's solution space by exploring different orders of item palletization,
- validate the proposed heuristic against our model and against the most relevant heuristics from the 3D-BPP literature,
- generate and share a data set based on real-world instances that we use to benchmark our heuristic.

2. Literature Review

The 3D-BPP is the generalization of the one-dimensional bin packing problem, which is NP-Hard (?). Exact methods can only solve small instances of the problem, which means that most solutions proposed in the literature are heuristics. Heuristics for the 3D-BPP are designed to solve the standard problem and do not consider practical constraints. ? provided a set of benchmark instances for 3D-BPP heuristics and an exact method based on a two-level branch-and-bound algorithm. Their method used a staircase placement strategy, where a series of corner points were identified as possible placement points to evaluate. The method was later extended in ? to find new placement niches that were previously ignored, introducing the concept of Extreme Points. ? introduced a biased random-key genetic algorithm (BRKGA) for the 3D-BPP, which is one of the best-performing heuristics on the benchmark instances of ?. Their algorithm was later extended in ? by incorporating a variable neighborhood descent procedure, which improved the number of generations needed to find high-quality results.

The concept of vertical support received most of its contribution from the literature on Container Loading Problems (CLP) and Pallet Loading Problems (PLP). As noted in ?, static stability is usually implicitly enforced as a consequence of load compactness or explicitly guaranteed by using filler material in a post-processing step. Most heuristics for CLPS and PLPs try to build dense layers composed of similar items that they stack, reducing the problem to a one-dimensional bin packing problem. Layers are filtered based on the fill rate, and when they are below a certain threshold, they are discarded (e.g., ??). When no new layer can be built, new bins are opened, simpler placement methods are used to pack the remaining items, or filler material is used to complete the layers.
Our solution to the problem fills the gap in the research by finding solutions to the 3D-BPPVS without explicitly building layers, and without the use of filler material.

3. Proposed Solution

Based on other publications from the literature and on our case study partner's insights, we state that an item is vertically supported if one of the following conditions holds.

Condition 1. the percentage of base area that leans on other items is larger than a predetermined threshold α_s.

Condition 2. the number of vertices resting on other items is greater than or equal to 3, and Condition **??** is met with a smaller threshold α'_s.

We consider two items to be of equal height if their heights do not differ more than a tolerance threshold β_s A visual representation of both support conditions is presented in **??**.

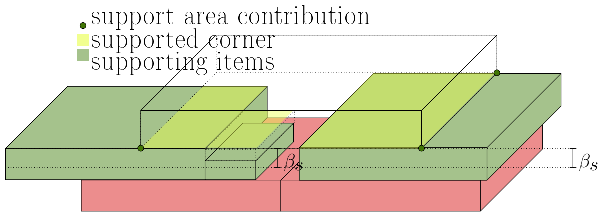

Figure 2: Representation of an item with conditions **??** and **??** of vertical support given $\alpha_s = 0.5, \beta_s$

To solve the 3D-BPPVS, we propose a heuristic that combines a constructive heuristic with a beam-search algorithm. The main idea of the heuristic is to build solutions to the 3D-BPPVS without explicitly building layers and without the use of filling material. The constructive heuristic is designed to solve a single-bin packing problem with vertical support, while the beam-search expands the heuristic's solutions by exploring different sequences of item placements. A conceptual representation of the heuristic can be seen in **??**.

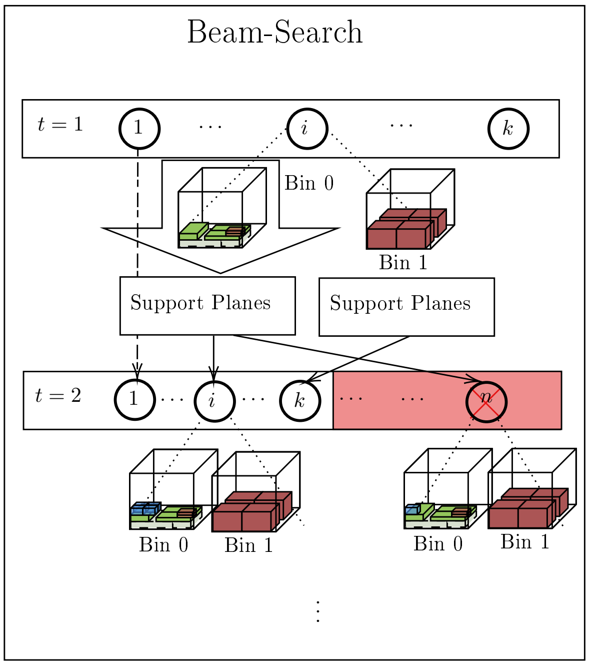

Figure 3: Conceptual representation of the proposed heuristic

We developed a new constructive heuristic for the single-bin 3D-BPP called Support Planes (SP). This heuristic sequentially inserts items within the bin, placing them on top of other items already placed. In this way, it is always possible to guarantee vertical support for the inserted items. Given a set of already positioned items, we define the two-dimensional plane that rests on the top face of those items as a support plane. Each bin will have multiple support planes since each item is placed at different elevations and has different heights. Each support plane is identified by its coordinate on the vertical axis (z), has regions where there is no support from items below, and regions where it is not possible to place items without overlapping with other placed items.

The insertion of items into a bin is performed by solving a two-dimensional bin packing problem (2D-BPP) for each support plane. Such 2D-BPP takes into account regions without vertical support and regions with obstacles. For each support plane, we test the insertion of each item. The plane is selected with a first-fit strategy starting from the lowest z, while the items to be inserted are selected with a best-fit strategy until the selected plane is filled. The best-fit strategy is based on selecting items that can generate well-supported planes along with the already placed items. In addition, placements are also intended to avoid over-satisfying the support constraint, which could lead to unbalanced bins as reported in **?**.
To reduce the number of placements to consider, we propose two SP configurations: in the first, 2D-BPP is solved for single item placement (PS), while in the second, 2D-BPP is solved for placement of groups of items with the same dimensions (PM). We solve the aforementioned 2D-BPP with a new modified version of the Extreme Points algorithm proposed in **?**. In our implementation, we use several tricks to improve performance, such as the use of axis-align bounding box trees for fast collision checks (**?**). Our studies have shown that choosing placements with a best-fit strategy does not achieve satisfactory solutions. For this reason, we extend our constructive heuristics with a beam-search algorithm. We define a solution as a set of open bins with their packed items. We say that a solution is partial if there are items that still need to be packed. Our beam-search algorithm starts from the empty solution and iteratively explores k new partial solutions. These partial solutions are generated from the solutions of the previous iteration by applying SP. Whenever SP is not able to place items in the opened bins of a partial solution, a new bin is opened. Since each new partial solution is obtained by inserting a positive amount of items in another partial solution, the algorithm will always converge to a set of solutions where no item is unpacked. To limit the number of partial solutions considered at each iteration, we define a ranking between them, using metrics related to the objective function of our MILP model. We rank these solutions considering the number of opened bins, the amount of packed volume,

and the average cage ratio among opened bins. Since different sequences of placements can lead to the same partial solution, we also developed a removal procedure that filters duplicate solutions based on a hashing function.

4. Conclusions

In this thesis, we presented a heuristic for the Three-Dimensional Bin Packing Problem with Vertical Support. We modified the two-dimensional Extreme Points algorithm of **?** to consider vertical support. We then used this modified algorithm in a constructive heuristic which builds solutions to the single bin three-dimensional bin packing problem by filling planes generated based on the previously inserted items. Finally, we introduced a beam-search algorithm that evaluates different sequences of item placements by using our proposed constructive heuristic and removes duplicate solutions at each iteration.
We tested a relaxed version of our heuristic without the support constraint and the rotation of items on the benchmark instances for the 3D-BPP proposed by **?**. Our heuristic achieved an average gap of 5.37% against the best solutions in the literature; however, we were able to solve the same instances in a fraction of their computational time. We consider this a great result since it states that our algorithm is also competitive in the realm of 3D-BPP without support. We generated a data set of problem instances based on real-world products from our case study, and we used them to evaluate our heuristic. Our solutions exceeded the target metric of 70% cage ratio in most configurations, with some of them having a negligible execution time. In **??** we report the average execution time (TT) in milliseconds, number of opened bins (B), and cage ratio (CR) across all our case study instances. Each row of the table reports the aggregated results on all instances with different value of k. Different placement modes are marked as PS (single placement) and PM (multiple items per placement). In **??** we analyze the trade-off between running times and cage ratio obtained by each configuration of the heuristic. We list the difference between the average running time of each configuration (TT) and the best average running time (TT*), together with the difference between the best average cage ratio (CR*) and the average cage ratio obtained by each configuration (CR). We discover that the most promising configuration for our industrial partner consists in using the PM placement mode with $k \in 20, 50$. We show some of the results of the case study experiments in **????** where the heuristic was configured in PM mode with $k = 200$.
Further research could introduce new practical constraints such as family groupings, load-bearing, and compatibility between items. Improvement heuristics could also be adapted to account for vertical support, such as the space defragmentation techniques introduced by **?**.

Table 1: Summary of case study tests

	PS			**PM**		
k	***TT (ms)***	***B***	***CR (%)***	***TT (ms)***	***B***	***CR (%)***
1	423.87	1.37	65.87	65.18	1.31	**70.70**
5	1,597.54	1.34	69.19	185.22	1.29	**73.08**
10	2,627.52	1.32	70.35	344.90	1.27	**73.56**
20	5,373.79	1.34	70.78	620.95	1.27	**74.57**
50	14,203.10	1.31	72.11	1,279.96	1.29	**74.61**
100	26,934.21	1.31	73.23	2,340.37	1.26	**75.36**
200	48,944.90	1.30	73.89	4,465.78	1.25	**76.39**

Table 2: Case study experiments trade off between average execution times and average cage ratio

k	***PS***		***PM***	
	CR* – CR (%)	***TT – TT* (ms)***	***CR* – CR (%)***	***TT – TT* (ms)***
1	10.56	358.69	5.73	0.00
5	7.24	1,532.36	3.35	120.04
10	6.08	2,562.34	2.87	279.72
20	5.65	5,308.61	**1.85**	**555.77**
50	4.32	14,137.92	**1.82**	**1,214.78**
100	3.20	26,869.03	1.07	2,275.19
200	2.54	48,879.72	0.04	4,400.60

5. Acknowledgements

My deepest gratitude goes to my supervisors, Prof. Ola Jabali and Dr. Davide Croci for their insights and support. This work wouldn't have been possible without them.

Thanks to the team at ERMES-X s.r.l who provided me with the case study and the data used for our experiments.

Many thanks to my family, my mom Patrizia, my father Severino, and my sister Giulia for always supporting me during this journey. Last but not least, thanks to my colleagues Alberto, Edoardo, Mirko, and Kevin, that brightened every day at the university.

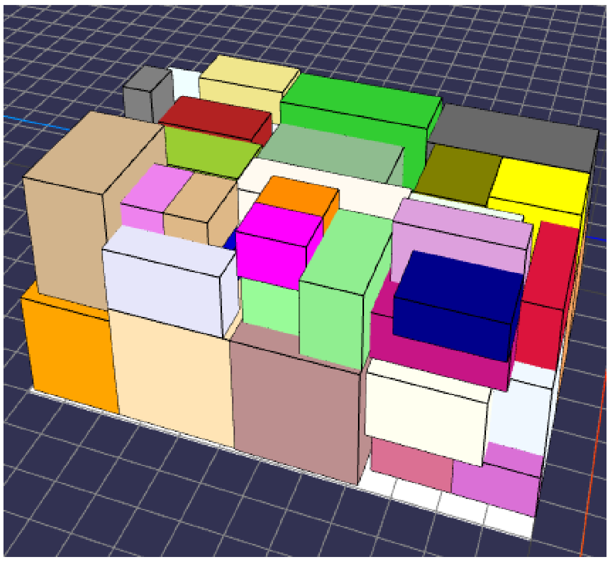

Figure 4: Solution to Instance 82 from case study tests

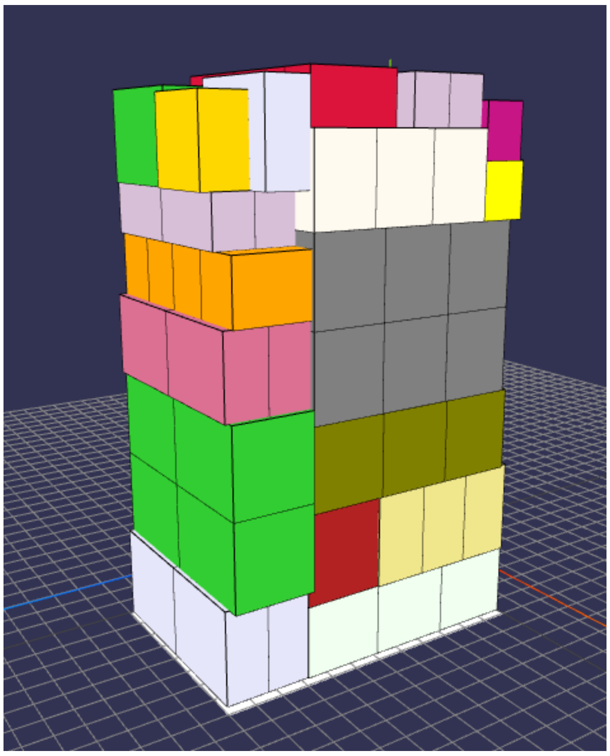

Figure 5: Solution to Instance 66, Bin 1 from case study tests