Data Paper

The International Journal of Robotics Research
2023, Vol. 0(0) 1–8

DOI: 10.1177/02783649231193048
journals.sagepub.com/home/ijr
Sage

BED-BPP: Benchmarking dataset for robotic bin packing problems

Florian Kagerer[1,2], Maximilian Beinhofer[1], Stefan Stricker[1] and Andreas Nüchter[2]

Abstract

Many algorithms that were developed for solving three-dimensional bin packing problems use generic data for either experiments or evaluation. However, none of these datasets became accepted for benchmarking 3D bin packing algorithms throughout the community. To close this gap, this paper presents the benchmarking dataset for robotic bin packing problems (BED-BPP), which is based on realistic data. We show the variety of the dataset by elaborating an n-gram analysis. Besides, we propose an evaluation function, which contains a stability check that uses rigid body simulation. We demonstrated the application of our dataset on four different approaches, which we integrated in our software environment.

Keywords

Three-dimensional bin packing, palletizing, competition, benchmarking, dataset

1. Introduction

Three-dimensional bin packing problems (3D-BPPs) are a research area with an industrial application in warehouse logistics. One task in warehouse logistics is to stack a set of cartons onto a pallet in an optimal way (see Figure 1). Depending on the use case, optimality is defined as minimizing the utilized space of the packed items while ensuring that the pile does not collapse.

Since 3D-BPP is NP-hard (Martello et al., 2000), many scientists conduct research on variants of this problem (Hu et al., 2017; Zhao et al., 2021). In its variant *online three-dimensional bin packing (O3DBP)*, which is interpreted as a three-dimensional version of the video game Tetris, a fixed sequence of items has to be placed in a bin. While we focus on online variants, in contrast, there is also offline 3D bin packing, where the sequence is not fixed, that is, one orders the items arbitrarily.

Due to the problem complexity, especially, heuristics or approximation methods were developed that tackle the problem. Since neural networks showed impressive results in solving complex problems (Mnih et al., 2015; Silver et al., 2016), deep reinforcement learning was soon applied to solve 3D-BPPs (Hu et al., 2017).

In order to compare the performance of different algorithms, benchmark datasets were provided not only for machine learning (Deng et al., 2009; Lin et al., 2014) but also for research in robotics (Kümmerle et al., 2009; Geiger et al., 2013; Leung et al., 2017). However, to our knowledge, a benchmarking dataset for three-dimensional bin packing problems is missing.

To this end, we present a novel dataset that is used to evaluate the performance of developed solvers for different variants of the three-dimensional bin packing problem (cf. Figure 4). We demonstrate our dataset's scope of application on three variants of 3D-BPP. Another novelty of our data is that it only contains realistic data of boxes with grocery items in it. Besides the article id, the dimensions, and the weight of an item, we also provide a product group this item belongs. For this reason, our data is also applicable to problems with stacking constraints that are based on article information.

In order to meet our demands to provide a benchmarking dataset, we operate an evaluation website on https://floriankagerer.github.io/leaderboard/that stores the current benchmark, that is, the algorithm that has the highest rating. Furthermore, with the release of our development environment, which is based on the standard API gym for reinforcement learning, we intend to revive a virtual manufacturing automation competition.

[1]TGW Logistics Group GmbH, Artificial Intelligence Competences, Marchtrenk, Austria
[2]Julius-Maximilians-University, Informatics XVII: Robotics, Würzburg, Germany

Corresponding author:
Florian Kagerer, TGW Logistics Group GmbH, Ludwig Szinicz Straße 3, Marchtrenk 4614, Austria.
Email: florian.research@icloud.com

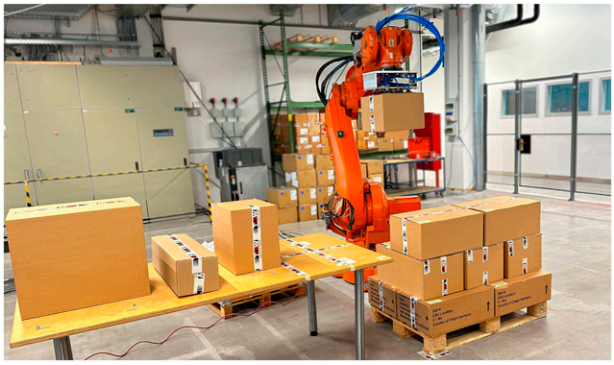

Figure 1. Robot test setup for online 3D bin packing.

2. Related work

Research on three-dimensional bin packing focuses mainly on the development of solvers. Even though benchmarking data is missing, some contributions overcame this issue by providing instance generators. These instances are interpreted as problem configuration, that is, they define a set of items that have to be packed in a particular target. Martello et al. (2000) generated instances based on nine classes that differ in the target bin size and in the range from which the item sizes are sampled. On the one hand, sampling the item dimension from a range increases the variety of the items with respect to the size. On the other hand, if the upper bound of the range is close to the bin dimensions, the instance complexity decreases, since the items per instance decrease.

In contrast, the variety of different item sizes in our dataset may be less than in Martello et al. (2000). However, the variety in our dataset depends not only on the item size but also on the different articles and their sequential arrangement. In addition, our instances contain 43 items in average, which we believe makes our instances quite challenging.

With the aim of creating realistic instances, Elhedhli et al. (2019) derived the distribution of the item characteristics from industry data. Thus, instance complexity increased and the use of the generator, which was presented in Elhedhli et al. (2019), improved the applicability to industrial applications. To further close the gap to reality, we enrich item characteristics in BED-BPP with article information, which consequently extends our dataset's scope of application to problems with stacking constraints that depend on article information.

Designed for the evaluation of their deep reinforcement learning approach, Zhao et al. (2021) generated 2000 sequences each, for three different strategies. Due to the determination of the target bin size and network architecture in that publication, the authors had a set of 64 items they sampled from. That dataset was reused for evaluating the developed approach in Zhao et al. (2022) and Puche and Lee (2022), where the latter were missing a standardized benchmark to compare different algorithms for variants of the 3D bin packing problem.

We postulate that our dataset is challenging enough and hence has the potential to become accepted for standardized benchmarking. We believe that when using our data, one sees how algorithms would perform in non-laboratory conditions.

3. Data processing

The benchmarking dataset BED-BPP, which is described in this paper, can be downloaded from https://floriankagerer.github.io/dataset/as JSON file. It consists of orders, that is, instances, that are used for solving three-dimensional bin packing problems. This section describes how we collected orders for this dataset, the format and the use of the data, and the way packing results are evaluated.

3.1. Procedure

In general, we divide the generation of the benchmarking dataset into three steps: on the basis of archival orders, we (1) prepare the item master data, (2) extract relevant orders for the dataset, and (3) analyze the selected orders. Our data is based on different profiles of companies in the grocery sector.

The main effort of the first step was the removal of non-essential information. This included that we made the article data anonymous, for example, we removed the manufacturer name. For each given article, we tried to find generic terms, for example, different types of bread were summarized as *bread*. Then, we gathered these items in product groups, for example, bread belongs to the group *bakery products*. This information is useful for applications in industry whenever product groups have to be separated on different targets. Finally, we set the units of length, width, and height to millimeters and the unit of weight to kilogram.

Since the underlying data was a collection of orders that had been made in the past, we had to identify suitable ones. For this reason, we defined two commonly used targets for our bin packing tasks: (1) a Euro-pallet with a base area of 1200 mm × 800 mm and (2) a rollcontainer with a base area of 800 mm × 700 mm. With these definitions, we describe the volume of an order as average height with respect to a target's base area.

In addition to the extent of an order, we also took the amount of different item footprints, that is, base areas, into account. To represent size tolerances like in reality, we used the item's side lengths in 2 cm steps for determining its footprint. Consequently, we obtained orders by using the following three filter criteria: (a) The target is a Euro-pallet with at least five different item footprints and an average height of [150 cm, 190 cm]. (b) The target is a Euro-pallet but with at least nine different item footprints and an average height of [120 cm, 150 cm). (c) The target is a rollcontainer

Table 1. Characteristics of BED-BPP.

Property	Value (mean ± standard deviation)
Length	10 003 orders
Different articles	2621 articles
Different item sizes	974 sizes
Items per order	43 items ± 14 items
Length of longest n-gram	8 items ± 3 items
Different item sizes per order	22 sizes ± 8 sizes

with at least five different item footprints and an average height of [150 cm, 180 cm].

Finally, we evaluated the selected orders. Table 1 summarizes the characteristics of the dataset. Besides statistical characteristics, an *n-gram* analysis was performed that shows a variety of our orders (Cavnar and Trenkle, 1994). Since we designed the dataset especially for online three-dimensional bin packing, for all orders we searched the longest item subsequence that occurs in one another, that is, the biggest n.

For this reason, the n-gram of an order was composed of substrings, where each substring contains the dimensions of an item. Similarly like determining the footprints above, we used a size tolerance of 2 cm, which means that we rounded the dimensions to a multiple of 20 mm. As a result, the description of an item's size in centimeters reduced not only the computing time but also gave a lower bound of the dataset's variety. Note that algorithms must take the size tolerances and the accuracy of the palletizing system into consideration to avoid box collisions.

The strings of the items are separated with a blank space. For example, the string "341828 281836" represents the sequence of items with sizes (a) 340 mm × 180 mm × 280 mm and (b) 280 mm × 180 mm × 360 mm.

A key result of the n-gram analysis on our dataset is that the longest item subsequence of an order that occurs identically in at least one other order is eight on average. This circumstance is explained by the fact that in our real-life data, in many cases not only one item is ordered per article but also in a bigger amount.

3.2. Structure

On the basis of Figure 2, we use the terminology of Python to describe our data. After parsing the JSON file in Python, a dictionary is obtained. The keys in the dictionary are unique order identification strings, and the values contain the corresponding order. The order itself is a dictionary with the keys "item_sequence" and "properties."

An element in the item sequence represents an object that has to be packed. It is defined by its dimensions, its weight, an article name with the product group it belongs, an id, and the position in the sequence. Besides the packing target, the properties contain additional information about an order.

3.3. Evaluation of packing plans

For a standardized evaluation of the results, we define the structure as in Figure 3(a) and denote the output of a solver as *packing plan*. Similar to the structure of BED-BPP, a packing plan is a dictionary with the unique order identification strings as keys and its values are interpreted as *packing lists*. The latter are composed of *actions*, which are defined by (a) the placed item, (b) its orientation, and (c) its front-left-bottom (FLB) coordinate.

We define the coordinate system on a target as depicted in Figure 3(b). In general, the origin of the coordinate system is in a corner on the top of the target. The x- and y-axes are parallel to the longer and shorter edge of the target, respectively. Therefore, the *FLB corner* of an item is always the corner that is closest to the origin of the coordinate system.

Another assumption we made is that the longer edge of an item with *orientation* 0 is parallel to the longer edge of the target. In contrast, a value one of the orientation means that the item is rotated by 90°.

3.3.1. Key performance indicators (KPIs). For the purpose of evaluation, we consider the amount of unpalletized items per packing list n_u, the volume utilization η_{util}, the maximum stacking height on the target h_n in meters, the mean of the support areas of all items on a target $\overline{A}_{supp}$, the interlocking ratio r_{interl}, and the physical stability of a pile that is created with the actions of a packing list.

We define the volume utilization η_{util} as the ratio of the sum of the volume of all items on a target divided by the volume of the circumscribed cuboid (cf. Figure 3(c)). The value of this dimensionless indicator is in (0, 1] and measures which percentage of the volume is filled with items.

When we speak about the support area of an item, we mean the ratio of the item's base area in which the item has direct support with a surface below. Thus, items that are directly placed on the target have a support area of one and $\overline{A}_{supp} \in (0, 1]$.

For determining the interlocking ratio r_{interl}, we only consider items without direct target contact. If all items touch the target, we set the value to NaN. Otherwise, r_{interl} is the quotient of the amount of items that are placed on either a single item with different orientation or more than one item regardless of the orientation, divided by the amount of

items on the target. This indicator is dimensionless and has values in [0, 1].

In order to simplify the definition of metrics, we summarize the KPIs in $\boldsymbol{x}_{\text{kpi}} \in \mathbb{R}^6$ with

$$\boldsymbol{x}_{\text{kpi}} := \left(n_{\text{u}}/N, \eta_{\text{util}}, h_n, \overline{A}_{\text{supp}}, r_{\text{interl}}, \mathbb{1}_{\text{stable}}\right)^T \quad (1)$$

where N is the total amount of items in an order and $\mathbb{1}_{\text{stable}}$ is the indicator function that decides whether a pile is stable.

3.3.2. Stability check. The decision whether a pile that is defined by a packing list is stable is based on a rigid body simulation. For this, we use the physics engine of *Blender*,[1] which is an open source 3D creation suite, and create a scene.

Since this simulation is resource intensive, we decided to check for stability after all items are placed. According to the packing list, we assert an item's geometry and mass to a cuboid object, move it to the determined coordinates, activate the rigid body property, and set the friction value to 0.5.[2]

Then, the simulation runs for 240 frames, which represent 10 s. If for any item the deviation of final to initial position exceeds a predefined threshold, the stability check fails and the indicator function $\mathbb{1}_{\text{stable}}$ returns 0. Otherwise, the check is successful and $\mathbb{1}_{\text{stable}} = 1$. Figure 3(d) visualizes two frames of a stability check.

3.3.3. Metrics. To make the results of two packing lists that belong to the same order comparable, we define scores. Due to the definition of the KPI vector (1), we easily obtain *weighted score* (2) by multiplying $\boldsymbol{x}_{\text{kpi}}$ with a weight vector $\boldsymbol{w} \in \mathbb{R}^6$, which reads as follows

$$sc_{\boldsymbol{w}} := (w_1, w_2, w_3, w_4, w_5, w_6)\boldsymbol{x}_{\text{kpi}} \quad (2)$$

For the special choice of $\boldsymbol{w} = (-1, 1, H^{-1}, 1, 1, 1)$, we obtain the dimensionless KPI score (3)

$$sc_1 := -\frac{n_{\text{u}}}{N} + \eta_{\text{util}} + \frac{h_n}{H} + \overline{A}_{\text{supp}} + r_{\text{interl}} + \mathbb{1}_{\text{stable}} \quad (3)$$

where $H > 0$ is a suitable, maximum palletizing height in meters that limits the ratio h_n/H by 1. Consequently, algorithms should aim to maximize score (3) of a packing list.

The previously defined scores describe how well items are placed on a target. For a future, virtual palletizing competition, we require an evaluation procedure for algorithms. We are interested to develop an algorithm that performs well on a wide range of orders, that is, produces a big amount of stable packing lists with high scores. To this end, we introduce our rating scheme.


```
{
  "order_i": {
    "item_sequence": {
      "1": {
        "article": "art-id1",
        "id": "id1",
        "product_group": "pg_1",
        "length/mm": 300,
        "width/mm": 200,
        "height/mm": 100,
        "weight/kg": 3.14,
        "sequence": 1
        },
        ...
    },
    "properties": {
      "id": "0123456",
      "order_nr": "01234567",
      "type": "chilled frozen grocery",
      "target": "euro-pallet"
    }
  },
  ...
}
```


Figure 2. Example of the format of an order within the dataset.

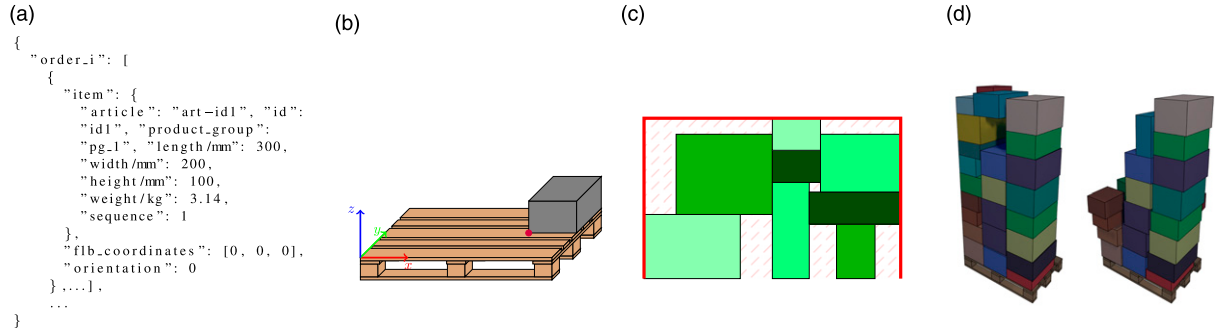

Figure 3. Depictions that support the understanding of the evaluation of packing plans. (a) Example of structure of a packing plan, (b) definition of coordinate system on a target and FLB corner of an item, (c) detailed front view of upper part of a target: η_{util} records how much air is included within a packing plan (shaded area), and (d) start and end frames of a stability check in Blender with $\mathbb{1}_{\text{stable}} = 0$.

3.3.4. Benchmark score of an algorithm. Finally, to compare different methods, we rate an algorithm based on its packing plan evaluation results. Since it is a non-trivial task to get a meaningful score by setting appropriate weights in (2), we rate an algorithm by summing up the percentage of stable palletized items per packing list within a certain height level and denote the value as Σ_{algo}.

In other words, for each list in the packing plan we (a) check whether the resulting pile is stable. (b) If it is stable, we count the amount of items $\tilde{n}_u$ that are unpalletized and add the amount of items whose top exceeds a pile height of 2 m. If the stability check fails, $\tilde{n}_u$ equals the amount of items of the order the packing list belongs to. (c) Finally, we sum up the ratio of the amount of palletized items related to the amount of items in BED-BPP, denoted as $N_{\text{BED-BPP}}$. The resulting score is $\mathbb{1}_{\text{stable}}(1 - \tilde{n}_u / N_{\text{BED-BPP}})$.

4. Experiments with BED-BPP

With regard to previous virtual manufacturing automation competitions, this section describes experiments, which we think are worth investigating. In addition to the dataset, we share our Python environment on https://github.com/floriankagerer/bed-bpp-env, which we used to develop algorithms and to conduct the suggested experiments.

4.1. Tasks for experiments

In Figure 4, four variants of O3DBP are visualized, for which we propose to use our data. Due to the fact that three-dimensional bin packing is NP-hard, it is difficult to develop algorithms that provide optimal results. To improve the performance of "online" algorithms, one adds additional information to the solver, for example, the size of upcoming items. But one should be aware that adding more and more information about upcoming items means that the solver tends to be an "offline" solver.

4.1.1. Online 3D bin packing with preview and selection. Since this task provides most information for a solver, we expect that the results of this variant surpass those of other online tasks. Online 3D bin packing with preview p and selection s (O3DBP-p-s) is interpreted as follows: The picking robot selects one out of s items that is packed next. In its decision, it takes into account its knowledge about the upcoming p items and thus improves the outcome of the decision.

In general, $p, s \in \mathbb{N} : p \geq s$ holds. For our experiments, we define that $p \in [2, 5]$ and $s \in [2, p]$. Note that for having preview and selection, that is, to select the next item from a pool of items and still have a preview, $p > s$ must hold.

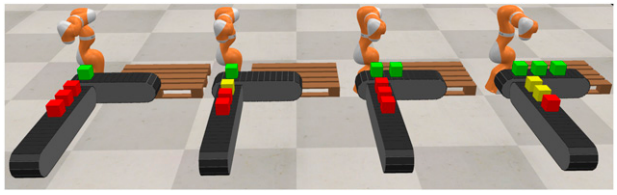

Figure 4. Variants of the three-dimensional bin packing problem. Green cuboids are selectable for the next palletizing step, for the yellow ones only the item information is known, and red objects are completely unknown.

4.1.2. Online 3D bin packing with selection. The special case where $p = s$ is named online 3D bin packing with selection. This task represents the situation in which a picking robot knows the next s items and selects one for palletization.

Since it knows the upcoming s items, but not any further items, the value of the preview p is identical to the amount of the items to select. With the previously defined upper bound of s for our experiments, we get $s \in [2, 5]$.

4.1.3. Online 3D bin packing with preview. When only one item is selectable for the next placement, but information about p upcoming items is known, we speak about online 3D bin packing with preview (O3DBP-p).

This task represents the scenario, where the picking robot takes only the next item but knows the upcoming p items when taking the current item. Based on the limits mentioned above, the value of $p \in [2, 5]$.

4.1.4. Online 3D bin packing. In contrast to O3DBP-p-s, in online 3D bin packing algorithms neither have an item preview nor a possibility to select the next item. They must place the item that is obtained. Since in this case, the solver obtains little information, we believe that this is the most challenging task.

The previously presented experiments all have an application in industry. Although offline 3D bin packing algorithms are the state-of-the-art in warehouse logistics, they come with the disadvantage that the pre-calculated item sequence has to be guaranteed in order execution, which leads to performance losses in the automated storage and retrieval systems of modern warehouses. The use of online solvers (O3DBP) overcomes this disadvantage but typically results in worse packing densities. A way to overcome this disadvantage is to increase the information an algorithm obtains (O3DBP-p-s and O3DBP-p). Each of these tasks is feasible in industrial applications due to the structure of a warehouse.

5. Applications of dataset

In order to show the scope of applicability, we present four approaches that address different tasks and use the benchmarking dataset. For details on the implementation, we refer to the previously mentioned

Table 2. Results of algorithms on BED-BPP.

		KPIs for "00100408"					
Algorithm	Σ_{algo}	n_{u}/N	η_{util}	h_n	$\overline{A}_{\text{supp}}$	r_{interl}	$\mathbb{1}_{\text{stable}}$
Online-3D-BPP-PCT	0.197204	0	0.614	2.105	0.814	0.458	1
heuristic O3DBP-3-2	0.020356	0	0.443	2.915	0.959	0.261	1
sisyphus	0.724553	0	0.385	3.360	0.893	0.545	1
xflp	0.010930	0	0.648	1.995	0.822	0.087	0

repository. The main effort in the integration of the software was the conversion of the data. The results are summarized in Table 2.

5.1. Online 3D bin packing

To demonstrate the task O3DBP, we selected the novel approach of Zhao et al. (2022). The authors introduced a packing configuration tree to represent the state and action space of packing and developed a deep reinforcement learning model.

We used the default parameters to train two Online-3D-BPP-PCT[3] solvers, one for Euro-pallets as target and one for rollcontainers as target. Furthermore, we changed the data that is used to train the solvers such that it matches with the item sizes in the benchmarking dataset.

In Figure 5(a), we visualized the pile that results of the packing list of Online-3D-BPP-PCT for the order "00100408". The KPIs for this packing list were $\boldsymbol{x}_{\text{kpi}} = (0, 0.614, 2.105, 0.814, 0.458, 1)$. This algorithm was able to create 28.661% stable piles for BED-BPP, which led to a final score of 0.197204.

5.2. Online 3D bin packing: Preview and selection

We set preview $p = 3$ and selection $s = 2$, hence, we developed a heuristic for O3DBP-3-2. Besides the item orientation and the item position, derived from corner points, cf. Martello et al. (2000), in this task the heuristic has to decide which item to palletize next. To determine the next action, the heuristic uses information about the upcoming items, that is, it virtually places all known items one after another. After placing, the heuristic only investigates the 25 actions with the highest ratings as starting point for the next placing.

The rating of a single action is a weighted sum of values of the (i) estimated support area, (ii) normalized estimated height in this location, (iii) item orientation, and (iv) normalized distance to origin in $\mathbb{R}^3$. Since the heuristic aims to maximize the score of an action, the used weights are (i) 1.3, (ii) -2.0, (iii) -1.00001, and (iv) -1.2. Finally, the heuristic chooses the next action that virtually leads to the combination with the highest sum of action ratings. For implementation details, see https://github.com/floriankagerer/bed-bpp-env

Figure 5(b) depicts the resulting pile for order "00100408" of our heuristic. The KPIs had the value $\boldsymbol{x}_{\text{kpi}} = (0, 0.443, 2.925, 0.959, 0.261, 1)$. Our heuristic was able to create 6.418% stable piles for the orders in the dataset, which led to a final score of 0.020356.

5.3. Offline 3D bin packing

Even though we proposed to use BED-BPP for variants of O3DBP, it is applicable to offline three-dimensional bin packing problems as well. To demonstrate this, we integrated the solver sisyphus[4] that won the IEEE ICRA Virtual Manufacturing Automation Competition (VMAC) in 2012 (Demisse et al., 2012). This approach tries to build layers out of articles with the same height. Whenever it is impossible to group enough items for a layer, they are placed in the center on the top of the target (see Figure 5(c)). The actions of sisyphus accomplished the KPIs $\boldsymbol{x}_{\text{kpi}} = (0, 0.385, 3.360, 0.893, 0.545, 1)$ for order "00100408". This algorithm was able to create 94.212% stable piles for BED-BPP, which led to a final score of 0.724553.

Furthermore, we integrated the heuristic solver xflp[5] that was developed for real-world container-loading problems in 3D and was inspired by Martello et al. (2000); de Castro Silva et al. (2003); and Crainic et al. (2012). We configured the solver for single-bin packing with rotation of items. We slightly had to change the sequence of the items this algorithm produced. In order to be able to pack the items in reality, we sorted the items ascending with respect to the z-coordinate of the FLB corner. The pile of the packing list that is visualized in Figure 5(d) had the KPIs $\boldsymbol{x}_{\text{kpi}} = (0, 0.648, 1.995, 0.822, 0.087, 0)$. This algorithm was able to create 2.669% stable

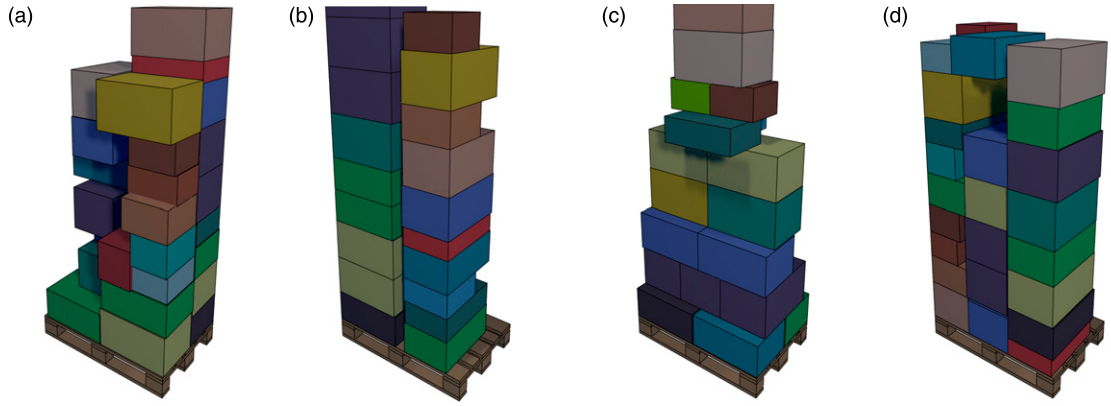

Figure 5. Comparison of final packing plan of different approaches for the same order. (a) Online-3D-BPP-PCT, (b) heuristic O3DBP-3-2, (c) sisyphus, and (d) xflp.

packing plans for BED-BPP, which led to a final score of 0.010930.

6. Conclusion

In this paper, we presented BED-BPP, a novel dataset for three-dimensional bin packing that contains beside item sizes also article information and information about the packing target. Thereby, the orders were not randomly sampled but extracted from real industry data of grocery companies. To determine the variety of the dataset, we developed an n-gram analysis on the item sequence and compared the results of all orders.

We demonstrated that our dataset is usable for benchmarking different algorithms, not only for offline three-dimensional bin packing but also for online variants of 3D bin packing.

In future work, we would like to extend BED-BPP: include data from other industry sectors and incorporate stacking rules that depend on item information. Furthermore, to close the gap between simulation and reality, the stability check needs to incorporate the placement by a robot and continuously check for stability. Regarding palletizing algorithms, we think that machine learning approaches have the potential to provide good results for online 3D bin packing. Our next step is to develop such an algorithm that satisfies the requirements for industrial use.

Acknowledgments

The authors like to thank the colleagues of Informatics XVII from the University of Würzburg and from TGW for the many helpful discussions.

Declaration of conflicting interests

The authors declared no potential conflicts of interest with respect to research, authorship and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Österreichische Forschungsförderungsgesellschaft (Austrian Research Promotion Agency [FFG]) (grant number 896048).

ORCID iD

Florian Kagerer https://orcid.org/0000-0002-4433-9060
Maximilian Beinhofer, https://orcid.org/0000-0002-8464-9952
Stefan Stricker, https://orcid.org/0000-0001-8782-1079
Andreas Nüchter, https://orcid.org/0000-0003-3870-783X

Notes

1. https://docs.blender.org/manual/en/latest/physics/rigid_body/introduction.html
2. https://wiki.unece.org/display/TransportSustainableCTUCode/Appendix+2.%09Friction+factors
3. https://github.com/alexfrom0815/Online-3D-BPP-PCT
4. https://github.com/josch/sisyphus
5. https://github.com/hschneid/xflp

References

Cavnar WB and Trenkle JM (1994) N-gram-based text categorization. *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*. Las Vegas, NV: University of Nevada, 161–175. Available at: https://www.osti.gov/biblio/10184077.

Crainic TG, Perboli G and Tadei R (2012) Recent advances in multi-dimensional packing problems. In: C Volosencu (ed) *New Technologies*. London: IntechOpen. DOI: 10.5772/33302. ISBN 978-953-51-0480-3.

de Castro Silva JL, Soma N and Maculan N (2003) A greedy search for the three-dimensional bin packing problem: the packing static stability case. *International Transactions in Operational Research* 10: 141–153. DOI: 10.1111/1475-3995.00400.

Demisse G, Mihalyi R, Okal B, et al. (2012) Mixed palletizing and task completion for virtual warehouses. *Virtual*

Manufacturing Automation (VMAC '12) Workshop at IEEE International Conference Robotics and Automation. Saint Paul, MN: ICRA.

Deng J, Dong W, Socher R, et al. (2009) Imagenet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, Miami, FL, 20–25 June 2009.

Elhedhli S, Gzara F and Yildiz B (2019) Three-dimensional bin packing and mixed-case palletization. *INFORMS Journal on Optimization* 1(4): 323–352. DOI: 10.1287/ijoo.2019.0013.

Geiger A, Lenz P, Stiller C, et al. (2013) Vision meets robotics: the kitti dataset. *The International Journal of Robotics Research* 32: 1231–1237.

Hu H, Zhang X, Yan X, et al. (2017) Solving a new 3d bin packing problem with deep reinforcement learning method. arXiv:1708.05930. DOI: 10.48550/ARXIV.1708.05930.

Kümmerle R, Steder B, Dornhege C, et al. (2009) On measuring the accuracy of slam algorithms. *Autonomous Robots* 27: 387–407. DOI: 10.1007/s10514-009-9155-6.

Leung KYK, Lühr D, Houshiar H, et al. (2017) Chilean underground mine dataset. *The International Journal of Robotics Research* 36(1): 16–23.

Lin T, Maire M, Belongie SJ, et al. (2014) Microsoft COCO: common objects in context: Cham, Springer International Publishing. *CoRR* abs/1405.0312. Available at: https://link.springer.com/chapter/10.1007/978-3-319-10602-1_48.

Martello S, Pisinger D and Vigo D (2000) The three-dimensional bin packing problem. *Operations Research* 48(2): 256–267.

Mnih V, Kavukcuoglu K, Silver D, et al. (2015) Human-level control through deep reinforcement learning. *Nature* 518: 529–533.

Puche AV and Lee S (2022) Online 3d bin packing reinforcement learning solution with buffer. arXiv:2208.07123. DOI:10.48550/ARXIV.2208.07123.

Silver D, Huang A, Maddison CJ, et al. (2016) Mastering the game of go with deep neural networks and tree search. *Nature* 529(7587): 484–489.

Zhao H, She Q, Zhu C, et al. (2021) Online 3d bin packing with constrained deep reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence* 35(1): 741–749. DOI: 10.1609/aaai.v35i1.16155.

Zhao H, Yu Y and Xu K (2022) Learning efficient online 3d bin packing on packing configuration trees. In: *International conference on learning representations*, Virtual Event, April 25-29, 2022, OpenReview.net, https://openreview.net/forum?id=bfuGjlCwAq.