3D Bin Packing Problem in Practice

An INFORM Project

RWTH Aachen

June 16, 2020

Team

Akshay Ganesh (M.Sc. DDS)
Clinton Charles (M.Sc. DDS)
Janos Piddubnij (M.Sc. Data Science)
Kalyan Keesara (M.Sc. DDS)
Luca Koczula (M.Sc. Industrial Engineering)
Moritz Dederichs (M.Sc. Computer Science)
Tobias Wagner (M.Sc. Industrial Engineering)
Yordan Manolov (M.Sc. Computer Science)

Outline

Quick Recap

- Task: Optimise item packing
- Current Objective: Minimise used boxes
- Previously: One model to compute an exact solution per order
- Problem: The computational complexity of this approach is to high
- Solution: Reduce computational complexity or use heuristics instead of optimisation.

Heuristics and Models

- Split the exact model to reduce complexity of each sub model
- Extreme Points heuristic
- Genetic Algorithm with Best Match Heuristic
- Peak Filling Slice Push

Split Model, General Idea

- Optimisation problem can be split into two separate optimisation problems
- First: Assign items to a set of boxes according to volume
- Second: Place items within box according to some optimisation criterion

Split Model, Box Assignment

- Minimise the number of boxes needed to fit all items of an order
- Items are considered to fit into a box if the combined volume of the items is less or equal to the volume of the box
- To simplify packing of items we decided to fill boxes only up to a certain volume ratio

Split Model, Packing Items

- Check if items can be placed within box limits
- Compute exact placement coordinates for each item
- Minimize the sum over all z-coordinates to simulate gravity and obtain a proper objective function

Split Model, Additional Runtime Improvements

- Problem: Optimizer alternates between multiple different optimal solutions
- Solution: Allow sub-optimal solutions in packing if the solution is a local minimum of the optimization function
- Additionally: Implemented multiprocessing to process multiple orders simultaneously

Split Model, Packing Instructions

- Define packing layer (all items sharing the same z-coordinates within a box)
- For each layer plot top-down view of all item placements within this layer

Split Model, Comparison of the Implementations

Runtime of optimisation

Case	Old	New
10 items	0 seconds	1 second
13 items	0 seconds	2 seconds
55 items	1511 seconds	36 seconds
59 items	$>$ 3200 seconds	35 seconds

- Currently, the average runtime over 200 random orders is 9 seconds per order
- A solid improvement over single order stats due to multiprocessing

Split Model, Problems

- If items do not fit in the assigned box, the split model returns an empty box
- This happens very rarely (only observed in 1 or 2 orders)

Split Model, What's Next

- Implement an auto-restart if the current assignment is infeasible
- Implement item-by-item packing instructions (e.g. textual description)
- Try to further optimise the runtime of the split model

EP, recap

- (E)xtreme (P)oints-based heuristic [1]
- Simplification of our model: no permutations
- Advantage: (asymptotically) fast, disadvantage: lots of free space
- Implementation differences:
  - no Gurobi dependency
  - data representation (typing, visualization)

EP, Progress

- All functionality implemented
- Mostly working
- Additionally: packing instructions, statistics

```
Place item A959 into box B2
Place item A466 into box B8
Place item A466 into box B5
```

Figure: Sample instructions for human packer

```
2020-06-15 14:11:02,734 Time before processing: 0.008225831001254846
2020-06-15 14:11:02,734 Processed order no 00
                    in 0.0006772319975425489 seconds
                    with 3 boxes
```

Figure: Sample performance measurement

EP, Outlook

- Not consistent with project structure of split-model branch
- Bugs: sorting, box dimensions
- No documentation
- Wish list: module-independent performance measurement
- Packing instructions understandable, but cumbersome
- No (unit) tests yet

Genetic Approach

- Approach is based on a publication by Li et al. (2014) [3]
- Genetic Algorithm (GA): Generations containing a predefined number of chromosomes (100) each
- Each chromosome is made up of two parts
  - Box-Packing-Sequence (BPS): order of items
  - Container-Loading-Sequence (CLS): order of boxes
- Genetic Algorithm generates chromosomes and mutates some of them based on a given probability (15 %)
- Generations are passed to the Best Match Heuristic (BMH) for evaluation

GA w/ BMH: Overview I

- Each chromosome gets evaluated by the BMH
  - Item orientation based on least margin in one dimension
  - Item placing with regards to empty maximal spaces (EMSs)
  - Calculation of 'fitness' (volume ratio) for each generated packing sequence

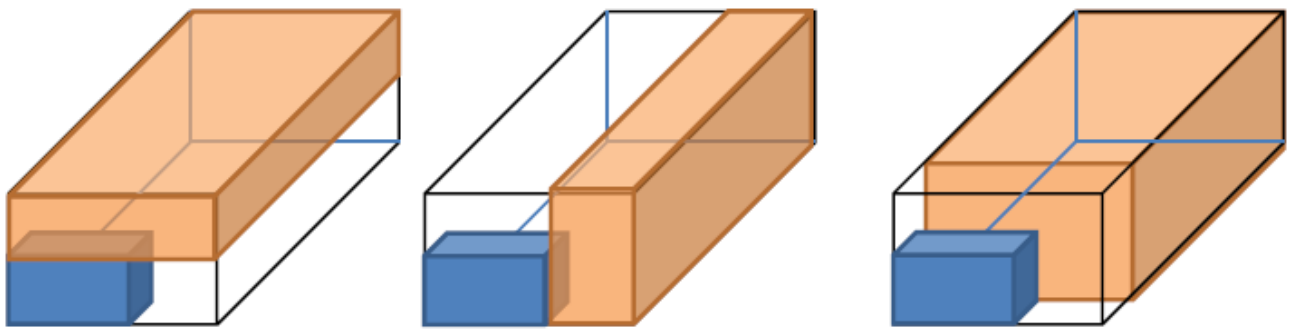

Figure: EMS Calculation[1]

[1] Image Source: "A genetic algorithm for the three-dimensional bin packing problem with heterogeneous bins", p. 4, Li et al. (2014)

GA w/ BMH: Overview II

- Packing Assigment and corresponding fitness value get passed back to the Genetic Algorithm
- Best E (10) chromosomes proceed directly into the next generation
- Rest of the chromosomes have a chance of mutation
  - Mutation: Crossover of two parent chromosomes to create two children/offspring

GA w/ BMH: Progress

- Heuristic is working properly
  - Least margin calculation
  - EMSs calculation and updating
  - packing sequence calculation
  - fitness (volume ratio) calculation
- Genetic Algorithm is working on its own, linking between GA and BMH is not done yet
- Step-by-step packing list generation and evaluation is possible

GA w/ BMH: Results

- Promising results with test instances
- Calculation of single chromosomes is very fast
- Genetic Algorithm works reliably on its own

++++ BEST MATCH HEURISTIC RESULTS: ++++

Input BPS:	Input CLS:	Output Packing Assignment:
('A96', 1)	('B3', 1)	Put item('A96', 1) into box ('B3', 1).
('A96', 2)	('B2', 2)	Put item('A96', 2) into box ('B3', 1).
('A96', 3)	('B5', 3)	Put item('A96', 3) into box ('B3', 1).
('A494', 1)	('B4', 4)	Put item('A494', 1) into box ('B3', 1).
('A494', 2)	('B6', 5)	Put item('A494', 2) into box ('B3', 1).
('A494', 3)	('B2', 6)	Put item('A494', 3) into box ('B2', 2).
('A494', 4)	('B1', 7)	Put item('A494', 4) into box ('B2', 2).
('A494', 5)	('B0', 8)	Put item('A494', 5) into box ('B5', 3).
('A494', 6)	('B7', 9)	Put item('A494', 6) into box ('B5', 3).
('A494', 7)	('B5', 10)	Put item('A494', 7) into box ('B5', 3).
('A494', 8)	('B2', 11)	Put item('A494', 8) into box ('B5', 3).
('A494', 9)	('B6', 12)	Put item('A494', 9) into box ('B5', 3).
('A494', 10)	('B6', 13)	Put item('A494', 10) into box ('B5', 3).
# of Generation: 0, # of Chromosome: 0		Fitness of Packing Sequence: 0.46666667

Figure: Generated Packing Assignment and additional info

GA w/ BMH: Next steps

- Linking of Genetic Algorithm and Heuristic
- Visualization of packing assignment (possibly through exact model)
- Set upper limit volume ratio (fitness)
- Code refactoring and optimization
- Further testing (refine parameters)

Peak Filling Slice Push (PFSP), Recap

The PFSP heuristic is a recursive divide and conquer algorithm. Each time a box is placed a new sub slice is created. The items can be sorted in decreasing order by height, width, lenght to increase efficiency.

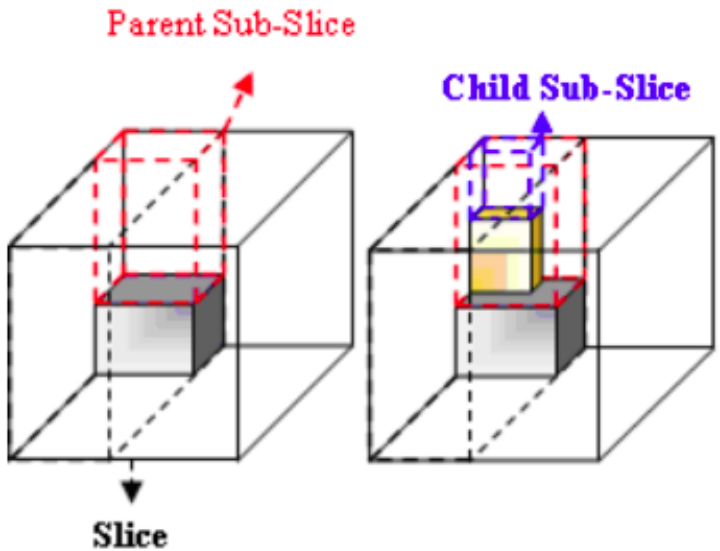

Figure: The slices are added until the container is complete.

Peak Filling Slice Push (PFSP), Insights

- This heuristic makes sure that the biggest box is down.
- Sorting can be done to increase performance
- Can be done for both problems items to box and boxes to pallet

Peak Filling Slice Push (PFSP), Next Steps

- Completion of the Push method
- Integration with the general model
- Performance evaluation with other models

Summary of the Results so far

- 3 heuristics
- 1 (mostly) exact model
- Noticeable improvements in runtime
- All approaches are still work in progress
- Project structure needs consolidation
- Inconsistent performance metrics

References

Teodor Gabriel Crainic, Guido Perboli and Roberto Tadei (2008)

Extreme Point-Based Heuristics for Three-Dimensional Bin Packing

INFORMS Journal on Computing Vol. 20 pages 368–384

Wissam F. Maarouf, Aziz M. Barbar and Machel J. Owayjan (2008)

A New Heuristic Algorithm for the 3D Bin Packing Problem.

In: Elleithy K. (eds) Innovations and Advanced Techniques in Systems, Computing Sciences and Software Engineering. Springer, Dordrecht

Xueping Li, Zhaoxia Zhao and Kaike Zhang (2014)

A genetic algorithm for the three-dimensional bin packing problem with heterogeneous bins

Proceedings of the 2014 Industrial and Systems Engineering Research Conference