Packing Containers in 3D

15-48 EC thesis proposal

For:	Master Students that know how to program and are interested. If you've ever done a course in heuristics, it is a pre.
Load:	15 - 48 EC, depending on blocks (see below)
Period (approx.):	From September 2021 onward
Supervisor:	Daan van den Berg
Contact:	daan@yamasan.nl, kawarimasen0010@gmail.com
Introduction video:	https://bit.ly/3DWGFrG

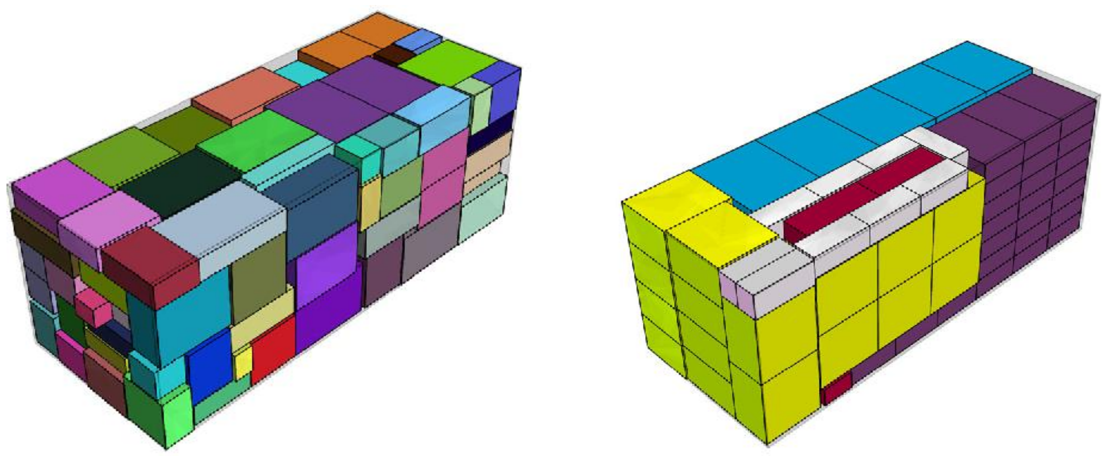

Fig.1: Two typical 3D box packings in a standard container (exact source unknown).

About the project

Recently, a very interesting pattern was found for rectangle packing problems[1]: there is a dependency from the number and the side length of the tiles to the chance of a puzzle being solvable [2]. This dependency is a very strict "phase transition", and it scales quadratically. So there is a possibility to predict the number of solution that a packing puzzle has.

Can we also find such a predictive Data Analytic for 3D packing? That question seems too big for any one MSc-project, but let's make some humble beginnings. We'll have to thread carefully, as we already ran into lots of problems for the 2D packing problem, but let's make some careful beginnings.

Block #1

We'll start very simply, by making some instances. Make *n* rectangular boxes with integer sides between 1 and t_{max}, (for starters, set *n*=6, 7 or 8 and t_{max}=12), add upp their volumes, and see if 3D containers of integer dimensions exist that allow for placement of the blocks in alle directions. This might be quite an experiment in itself already. Initially, the box-set cannot contain duplicate boxes, all sides for a box must be unique for that box. We need a *lot* of instances, and a public repository to store them in.

Block #2

Build a solver that packs boxes in the container. It should be exact, and guaranteed to find a solution if there is one. Programming this algorithm could be so hard that it is likely impossible to pack containers with even just 20 boxes. But let's start and see what we can do. If this block proves too hard, we'll split it up. There might be some suggestions in the work by Silva, Toffolo and Wauters [3]

Block #3

If we can get to packing 16 boxes or so, we can make containers that are somewhat bigger, so that the boxes take up approximately 70% of its volume, and we can see what happens when t_{max} increases. The work by Silva, Toffolo and Wauters suggests something interesting about that: a sudden drop in hardness (see their figure 3). *What on earth* is happening there? We want more details. Again, this could be a very hard block, and is splittable in different blocks if needed.

Block #4

Compare algorithmic performance of different solvers with or without different pruning options.

Block #5

If we have enough solutions, we might be able to predict some solvability characteristics. It is very questionable whether we can reach this block even within a very large masters' project , but any progress in this direction would be great.

[1] https://bit.ly/3xM2QfT

[2] Braam & Van den Berg (2021) How to make (un)solvable Rectangle Puzzles
Contact me for a manuscript; at the time of writing, it is still under revision.

[3] Silva, Everton Fernandes, Túlio Angelo Machado Toffolo, and Tony Wauters. "Exact methods for three-dimensional cutting and packing: A comparative study concerning single container problems." Computers & Operations Research 109 (2019): 12-27.