ERASMUS UNIVERSITY ROTTERDAM
Erasmus School of Economics

Bachelorscriptie Econometrie en Operationele Research
Bachelor Thesis Econometrics and Operations Research

A Tabu Search approach to the Three-dimensional Bin Packing Problem with Practical Extensions

---

Author:
Thomas ten Damme
539638

Supervisor: Correia Duarte, PJ
Second Assessor: Wagelmans, APM

the Netherlands
3rd of July 2022

Abstract

The Tabu Search algorithm for the three-dimensional bin packing problem as proposed by Lodi et al., 2002 has shown great merit. In practice, the algorithm often misses out on considerations that are crucial to bin packing allocations. The three practical extensions to the algorithm that are implemented and discussed in this paper are: a weight limit to bins, constraining the extremity of imbalance of the bins and the minimization of fragile bins containing at least one fragile item. The proposed extended Tabu Search algorithms are tested on a variety of sample sets and evaluated accordingly. A simple but effective extension is found for the weight limit and fragility problems. The attempted algorithm for ensuring balance clashed with the inherent imbalance of the original algorithm.

Contents

1 Introduction

In the US alone, logistics costs were estimated at 1494.7 billion USD in 2017, of which 965.5 billion USD were transportation costs (Kearney, 2018). It is not uncommon for transport costs to account for 10% of the total cost of a product and empirical evidence shows that decreasing transport costs by 10%, increases trade volumes by more than 20% (Rodrigue, 2020). Among others, these statistics made that efficient cargo handling has become a widely discussed topic. Improvements in the sector would benefit many transport businesses, as well as consumer prices and improve trade in general.

Among the three-dimensional packing problems, which nearly all apply to cargo transport in some way and seek to improve efficiency, the three-dimensional bin packing problem (3D-BPP) is prominently discussed. The bin packing problem is an optimization problem, in which items of different sizes must be packed into a finite number of bins or containers, each of a fixed given capacity, in a way that minimizes the number of bins used. The three-dimensional BPP concerns three-dimensional items and bins. Finding the optimum to this problem is NP-hard. This results in a necessity for heuristics in solving larger problems in realistic time, which is very often the case in practice.

The standard form of the problem, though, does not consider practical constraints like weight. It only considers the volume constraint. This often makes the proposed algorithms infeasible for businesses and only significant in the academic world. This paper attempts to bridge the gap between the academic theory and the practical use by taking a promising approach to the 3D-BPP using a Tabu Search framework as proposed by Lodi et al. (2002) and adjust it to include certain practical constraints.

Often, volume is not the only constraint when considering cargo for a vehicle. Weight might be a limiting factor when considering cargo; air freight rates are calculated by weight. Allianz Global Corporate Specialty (AGCS) analysed almost 15 000 marine liability insurance claims between 2011 and 2016 and found that human error was the primary factor in 75% of the value of all claims analyzed - more than 1.5 billion euros of losses (AGCS, 2019). Decreasing the amount of incidents caused by human error could be achieved by aiming to make bins easier to handle and less likely to topple over. Lastly, taking into account the fragility of items and therefore of bins, the value of cargo damaged by human error could be decreased. These three practical constraints are studied in this paper, a limit to the maximum weight of a bin (equal for all bins), weight distribution and the fragility of items.

It was found that the implementations of a weight limit and fragility consideration are effective. General performance was not significantly damaged and in the case of fragility, the number of fragile bins was decreased drastically. The extension involving the stabilization of bins proved fruitless, likely because the forced relocation of the centre of mass clashes with the nature of the existing algorithm forcing items into a corner.

In Section 2, previous research on the subject is shortly discussed. In Section 3, the Tabu Search algorithm is discussed as proposed by Lodi et al. (2002). Section 4 contains the methods used to adjust the algorithm and to gather results. In Section 5, the found results are discussed and in Section 6, conclusions are drawn from these results. Lastly in Section 7, the shortcomings of this paper and future research are discussed.

2 Literature Review

Previous literature on the 3D-BPP and concerning a weight limit is not uncommon. Bortfeldt and Wäscher (2013) found that 14.1% of their 163 reviewed container loading papers (3D-BPP is a version of a container loading problem) considered a bin weight limit. In all of these the weight limit was addressed as a hard constraint. Of these container loading papers, 26.1% (so 3.7% of total) were about the 3D-BPP. None of these, and none since, consider a bin weight limit for the 3D-BPP in combination with a Tabu Search framework.

Weight distribution is also relatively well-discussed, with 11.7% of the papers reviewed by Bortfeldt and Wäscher (2013) considering weight distribution. All consider the placement of the centre of mass in relation to the middle of the bins in some way. Some try to keep it near the middle of the bin floor (Davies and Bischoff, 1999; Gehring et al., 1990) where others try keeping it as low as possible (Scheithauer and Terno, 1996; Sommerweiß, 1996).

The only class of constraints considering fragility are stacking constraints (or load-bearing constraints). These restrict how items can be stacked inside bins and they arise from the limited load-bearing strength, or fragility, of item cases and their materials. From this strength, one can determine the amount of weight that can be placed upon the items (Bischoff and Ratcliff, 1995). 15.3% of the papers reviewed by Bortfeldt and Wäscher (2013) include stacking constraints. Many different ways of including fragility in the model are covered. It can be assumed that nothing can be placed upon fragile items (Bortfeldt and Gehring, 1999). Some papers assume fragile items can be stacked upon all items while non-fragile items can only be stacked upon other non-fragile items (Gendreau et al., 2006). Another way is giving all items some fragility index and only allowing the placement of an item on top of another if the top item has a lower or equal index (Scheithauer and Terno, 1996).

None of the fragility constraints however, take into account the fragility of the resulting bins. When considering shipping containers, this might be justified, but with other (smaller and weaker) bins it might be necessary to mark bins that contain fragile items. If so, it might also be desirable to minimize this number of fragile bins. There are currently no papers discussing this extension of any container-loading problems though. This new extension is also explored in this paper.

There are numerous papers from the last 20 years that cover bin-packing algorithms with a Tabu Search metaheuristic. None of these however, use the framework proposed by Lodi et al., 2002 (search algorithm and metaheuristic implementation) and modify it to include the previously discussed practical constraints. This paper attempts the combination of this Tabu Search framework with the previously mentioned practical constraints to see if they can be effectively implemented.

3 Problem Description

In this section the original problem is first described. Then, the three different extensions and how they change the problem are covered.

3.1 Original Problem

The 3D-BPP considers n three-dimensional rectangular items with a width w_j, a height h_j and a depth d_j ($j \in N = \{1, ..., n\}$), and an unlimited amount of bins with width W, height H and depth D to store these items in. Without loss of generality, we assume that $w_j \leq W$, $h_j \leq H$ and $d_j \leq D$ $\forall j \in N$. The problem is to fit all these n items in as few bins as possible. Computationally, the problem is NP-hard and the deciding if certain items can fit into a specific number of bins is NP-complete.

3.2 Weight Limit Extension

To include a hard weight limit M (for mass to prevent confusion with w for width) to the bins, the problem should now consider n items with a weight m_j ($j \in N = \{1, ..., n\}$) apart from the existing width, height and depth. As with the dimensions, we assume that, without loss of generality, $m_j \leq M$ $\forall j \in N$ (otherwise items with a weight bigger than the limit could never be placed in a bin). Now, apart from the items fitting in the bin, the sum of the weights of the items in the bin should also be less than the weight limit. Still, the objective of the problem is minimizing the amount of bins.

3.3 Weight Distribution Extension

For the weight distribution, the weights w_j ($j \in N = \{1, ..., n\}$) of items are again necessary. To ensure stability, it is desired to have a centre of mass (CoM) for every bin that lies near the centre of the bin floor. The centre of mass of a bin is calculated by taking the weighted sum of the centres of mass of the items in the bin. The centre of mass of the items is assumed to be in the middle of the item in all dimensions. The calculation is done for the width and depth dimensions separately. To include this extension to the original problem, constraints are added ensuring the centre of mass has a maximum distance from the middle of the bin floor in the width and depth dimensions. This distance could also be added as an objective to be minimized, but this would not ensure stability.

3.4 Fragility Extension

To include the fragility of bins, the problem will be extended to include fragile items. Every item gets a fragility f_j ($j \in N = \{1, ..., n\}$) which is equal to 1 if the item is considered fragile and equal to 0 otherwise. This item characteristic is then used to determine if a bin should be considered fragile. If one or more items in a bin are fragile, the bin is considered fragile. Now the problem has two objectives - to fit all items in as few bins as possible and to have as few fragile bins as possible amongst those.

4 Methodology

In this section, the existing algorithm proposed by Lodi et al. (2002) is explained first, then the modifications for each extension are covered and lastly the methods of evaluating these extensions are discussed.

4.1 Lodi et al.

The Tabu Search algorithm proposed in Lodi et al., 2002 is a metaheuristic, meaning that it requires a solution and uses this to explore the neighbourhood of this solution by changing this solution slightly. This means that Tabu Search needs a search algorithm to find an initial solution and to explore the neighbourhood. The heuristic search algorithm proposed in Lodi et al., 2002 is called *Height first - Area second* (HA) and it finds a solution for any feasible problem. We will first discuss this search algorithm.

4.1.1 Height first - Area second

In the first phase, the items are sorted by non-increasing height (height first) and clustered based on height, depending on a parameter β ($\beta \in [0, 1]$). The tallest item j defines the first cluster with height h_j. All items with a height h_k satisfying $h_k \geq \beta h_j$ are included in this cluster. The tallest item s for which $h_s < \beta h_j$ defines the next cluster with height h_s, and so on. The items in the clusters are sorted by non-increasing $w_j d_j$ value. Now the items are packed in layers as follows (tallest cluster first and item with biggest base $w_j d_j$ first in cluster), starting without initialized layers:

- For every normal position p - a position where the item can not move leftwards or backwards - in an initialized layer with a height $H_\ell \geq h_j$, compute score function $S(j, \ell, p)$ (see Equation 1 below). If the maximum found score function value is bigger than zero, pack item j in this layer ℓ at position p.

- If item j has not been packed yet, compute $S(j, \ell, p)$ for all normal positions in the remaining initialized layers ($H_\ell < h_j$) and pack item j in the location that corresponds to the maximum score value if this is bigger than zero.

- If item j is still not packed, initialize a new layer and pack the item at $(0, 0)$ - the only normal position.

$$S(j, \ell, p) = \rho \frac{P(j, \ell, p)}{2w_j + 2d_j} + \mu \frac{\sum_{k \in J_\ell} w_k d_k}{WD} - (1 - \rho - \mu) \frac{|H_\ell - h_j|}{H_\ell}, \tag{1}$$

where ρ and μ are prefixed real valued parameters such that $\rho, \mu \in [0, 1]$ and $\rho + \mu \leq 1$ and J_ℓ is the set of items already in layer ℓ. If item j does not physically fit at location p with the current packing of ℓ, we set $S(j, \ell, p) = 0$. The three terms take into account respectively:

1. The fraction of the base perimeter of item j touching the edges of ℓ or other items (more is better).

2. The already packed portion of the base of ℓ (more is better).

3. The relative height difference between item j and layer ℓ (less is better).

This procedure is called PACK. After executing it, we have a number of layers with heights $H_1, ..., H_g$. These are then put into bins with height H by solving the one-dimensional bin packing problem (1D-BPP) with heights $H_1, ..., H_g$ and capacity H with exact algorithm MTP (Martello and Toth, 1990). This results in the combination of the layers into the minimum number of bins.

Procedure PACK is executed again in a second phase, but now with all items resorted by non-increasing $w_j d_j$ value (area second) and empty layers with heights $H_1, ..., H_g$. If this results in empty initiated bins, these are removed.

The two phases can also be executed on synchronously rotated items and bins, resulting in layers in the width and depth direction. HA selects the best of these six configurations as a result.

4.1.2 Tabu Search

The Tabu Search metaheuristic is initiated by executing algorithm HA once. This result is saved as the incumbent (best-found) solution and z denotes the number of bins used in the incumbent solution. Then the current solution is initiated by naively packing each every item in a separate bin. Let z^c be the number of bins used in the current solution.

Tabu Search is initiated with an empty Tabu List for every neighbourhood size k with $k \in \{1, ..., k_{\max}\}$ of length $\tau_k, k \in \{1, ..., k_{\max}\}$. $k_{\max}$ and τ_k are set parameters. k is initially set to 1.

During the Tabu Search, a *target bin* will be selected by minimizing over all the bins i, the *filling* function

$$\phi(S_i) = \alpha \frac{\sum_{j \in S_i} w_j h_j d_j}{WHD} - \frac{|S_i|}{n}, \tag{2}$$

where S_i is the set of items currently packed in bin i and α is a specified positive weight. This bin should be easiest to empty. A subset S is created to include the items from k different bins and one item, j, from the target bin. Algorithm HA is performed on this subset and if this results in k or less bins, item j has been removed from the target bin, otherwise S is changed by selecting a different set of k bins or (if all subsets S have been attempted) a different item j from the target bin. When a move results in a current solution with less than z bins, the incumbent solution is updated. Removing item j from the target bin after finding a solution to HA with $\leq k$ bins is called a move, and it might be prevented from being performed if this specific move is in the Tabu List of k. The sum of the filling functions of the $k+1$ bins involved is calculated to check if that value is in the Tabu List. If not, the move is performed and this value is added to the front of the Tabu List of k, removing the last value.

The current search is halted when no feasible move is found after looking through all k bin configurations and all items j or when ℓ moves have been attempted without improving the incumbent solution (with ℓ a set parameter). In that case, k is increased by one if it can ($k < k_{\max}$), otherwise a diversification action is performed.

There are two diversification actions that happen alternatively. A 'soft' diversification happens the first time, and this simply means that the algorithm will now choose the bin with the second smallest filling function as the target bin. The 'hard' diversification consists of replacing the items in the $\lfloor z^c/2 \rfloor$ bins with the smallest filling function values into separate bins. Tabu Search terminates when reaching a set time limit or when some lower bound is reached (Lodi et al., 2004).

Lodi et al., 2002 set the following parameters after doing preliminary experiments: $\alpha = 1.5, \beta = 0.75, k_{\max} = 3, \tau_k = 20$ for all $k, \ell = 100 - 25(k-1)$, $\rho = 0.3$ for phase 1, $\rho = 0.2$ for phase 2, $\mu = 0.7$ for phase 1 and $\mu = 0.3$ for phase 2. These parameters will be used unless mentioned differently.

4.2 Extensions

To implement each of the extensions, the score function will need to be edited and the way the different layers are combined (1D-BPP) adjusted. This combination of layers is done in Lodi et al., 2002 with the exact branch-and-bound algorithm MTP (Martello and Toth, 1990). To be more easily adapted for a specific extension, the combining of the layers is described as a linear programming problem (Martello and Toth, 1990):

$$
\begin{aligned}
&\text{minimize} && z = \sum_{i=1}^{n} y_i && && \text{(3a)}\\
&\text{subject to} && \sum_{j=1}^{n} h_j x_{ij} \leq H y_i, && i \in N = \{1, ..., n\}, && \text{(3b)}\\
& && \sum_{i=1}^{n} x_{ij} = 1, && j \in N, && \text{(3c)}\\
& && y_i \ \in \{0, 1\}, && i \in N, && \text{(3d)}\\
& && x_{ij} \in \{0, 1\}, && i \in N, j \in N, && \text{(3e)}
\end{aligned}
$$

where $y_i = 1$ if bin i is in use and 0 otherwise and $x_{ij} = 1$ if layer j is in bin i and 0 otherwise. h_j is the height of layer j and H is the height of the bins. A bin is created for every layer (n layers), that is why both the layer index j and the bin index i are taken from $N = \{1, ..., n\}$. Equation 3b ensures that the combined layers fit in a bin and equation 3c ensures that every layer is placed in exactly one bin.

The lower bound that is used to determine if the Tabu Search can be stopped is lower bound L2 as described in Martello et al., 2000. It uses another proposed lower bound L1, which produces a tight lower bound when items are relatively big by considering the items that can only be stacked in one dimension and using a one-dimensional lower bound on those. L1 can be performed in all three dimensions. Lower bound L2 combines this

L1 with continuous lower bound L0, which sums the volumes of all items to compare to the volume of a bin. L0 works well with many small items. The resulting combination L2 dominates L0 and L1.

In Sections 4.2.1, 4.2.2 and 4.2.3 the weight limit, weight distribution and fragility extensions respectively are discussed. Note that index j is generally used for the object to be placed in the bigger container. This might be the actual item j to be placed into a layer, or this might be layer j to be placed in a bin.

When estimating the parameters in the next three sections, Class 5 (see Section 4.3) was chosen as it consists of items from Types 1-5 while not favouring one dimension (like Classes 1-3 do) and contains the relatively smallest items (Class 4 contains bigger items) so the bins are relatively well-filled. The estimations are first run on $n = 50$ for the first extension but then on $n = 150$ for the other two since it is decided that in bigger samples the potential slowness of extensions is better emphasized.

4.2.1 Weight Limit Extension

To implement a weight limit, the score function receives a new term that encourages the placement of items which bring the weight-height ratio of the layer towards that of a bin weighing exactly the weight limit. The resulting score function is given below:

$$\begin{aligned} S(j,\ell,p) = \rho\,\frac{P(j,\ell,p)}{2w_j + 2d_j} &+ \mu\frac{\sum_{k\in J_\ell} w_k d_k}{WD} \\ &+ \gamma\frac{1}{\left|\frac{m_{(J_\ell\cup j)}}{h_{(J_\ell\cup j)}} - \frac{M}{H}\right|} - (1-\rho-\mu-\gamma)\frac{|H_\ell - h_j|}{H_\ell}, \end{aligned} \tag{4}$$

where $\gamma \in [0,1]$ is a parameter such that $\rho + \mu + \gamma \leq 1$. Also, $m_{(J_\ell\cup j)}$ and $h_{(J_\ell\cup j)}$ are the weight and height of layer ℓ after adding item j respectively, M is the weight limit of the bin and H is the height of the bin.

All that remains is adjusting the original formulation of the 1D-BPP to include the weight limit. All that needs to happen is to replace Equations 3b by the following:

$$\sum_{j=1}^{n} h_j x_{ij} \leq H y_i, \qquad i \in N = \{1,...,n\}, \tag{5a}$$

$$\sum_{j=1}^{n} m_j x_{ij} \leq M y_i, \qquad i \in N = \{1,...,n\}, \tag{5b}$$

where h_j and m_j are the height and the weight of item j respectively and H and M are the height and weight capacities of the bins respectively. Equation 5b is a simple replication of Equation 3b ensuring the weight of a bin does not exceed the weight limit.

Due to limited time, the value of γ is estimated on a small instance of Class 5 (see Section 4.3) with $n = 50$ and 10 repetitions, choosing the value where the results had the lowest number of total bins. The optimal value was found to be $\gamma = 0.45$. See the following figure:

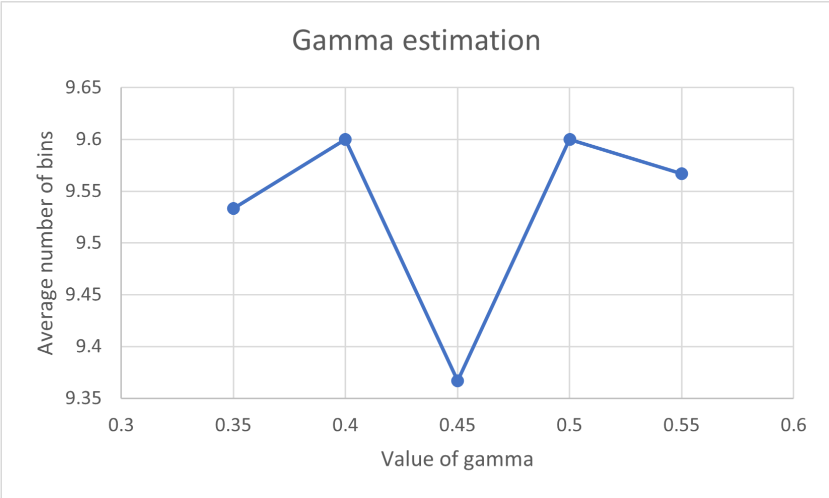

Figure 1: Graph of estimation of γ

For all data see Table 6.

4.2.2 Weight Distribution Extension

To ensure stability of bins, the centre of mass of each bin should be as centered as possible. This is attempted by encouraging the placement of items in layers that bring the centre of mass towards the middle of the layer. This is how the centre of mass of a layer is calculated:

$$\mathrm{CoM}_{w,\ell} = \frac{\sum_{j \in L} \mathrm{CoM}_{w,j} m_j}{\sum_{j \in L} m_j}, \tag{6}$$

where $\mathrm{CoM}_{w,\ell}$ is the centre of mass of layer ℓ in the width direction. $\mathrm{CoM}_{w,j}$ is the centre of mass of item j in the width direction, which is assumed to be in the centre of the item. L is the set of items in layer ℓ and m_j is the weight of item j. Equation 6 works the same in the depth direction by interchanging $\mathrm{CoM}_{w,\ell}$ and $\mathrm{CoM}_{w,j}$ with $\mathrm{CoM}_{d,\ell}$ and $\mathrm{CoM}_{d,j}$.

The score function is adjusted to look as follows:

$$\begin{aligned} S(j,\ell,p) = \rho \frac{P(j,\ell,p)}{2w_j + 2d_j} + \mu \frac{\sum_{k \in J_\ell} w_k d_k}{WD} \\ + \delta \left(1 - \left|\frac{\mathrm{CoM}_{w,\ell,j}}{W} - 1/2\right| - \left|\frac{\mathrm{CoM}_{d,\ell,j}}{D} - 1/2\right|\right) - (1 - \rho - \mu - \delta) \frac{|H_\ell - h_j|}{H_\ell}, \end{aligned} \tag{7}$$

where $\delta \in [0,1]$ is a parameter such that $\rho + \mu + \delta \leq 1$. Also, $\mathrm{CoM}_{w,\ell,j}$ and $\mathrm{CoM}_{d,\ell,j}$ are the centres of mass of the layer after adding item j in the width and depth respectively. W is the bin width and D the bin depth. This term motivates the placement of items that move the centre of mass of the layer to the middle of the bin.

The formulation of the 1D-BPP is also changed to include a weight distribution constraint. The following constraints are added to the original formulation (Equations 3a to 3e):

$$\frac{\sum_{j \in N} \mathrm{CoM}_{w,j} m_j x_{ij}}{\sum_{j \in N} m_j x_{ij}} \geq \frac{1 - p_{\mathrm{CoM}}}{2} W, \qquad i \in N, \tag{8a}$$

$$\frac{\sum_{j \in N} \mathrm{CoM}_{w,j} m_j x_{ij}}{\sum_{j \in N} m_j x_{ij}} \leq \frac{1 + p_{\mathrm{CoM}}}{2} W, \qquad i \in N, \tag{8b}$$

$$\frac{\sum_{j \in N} \mathrm{CoM}_{d,j} m_j x_{ij}}{\sum_{j \in N} m_j x_{ij}} \geq \frac{1 - p_{\mathrm{CoM}}}{2} D, \qquad i \in N, \tag{8c}$$

$$\frac{\sum_{j \in N} \mathrm{CoM}_{d,j} m_j x_{ij}}{\sum_{j \in N} m_j x_{ij}} \leq \frac{1 + p_{\mathrm{CoM}}}{2} D, \qquad i \in N, \tag{8d}$$

where p_{CoM} is the proportion of the bin's width and depth around the bin floor centre that should contain the resulting centre of mass. Equations 8a and 8b ensure that the centre of mass is near the center in the width dimension. Equations 8c and 8d do the same for the depth dimension.

Similar to how γ is estimated, the value of δ is estimated on a small instance of Class 5 (see Section 4.3) with n = 150 and 3 repetitions due to limited time. In this case also weighing in the different proportion values p_{CoM}. The value with the biggest improvement in replacing the centre of mass towards the middle while maintaining a similar amount of total bins used is chosen. $\delta = 0.2$ was the only value where there was any improvement in the centre of mass (on average). See the following figure:

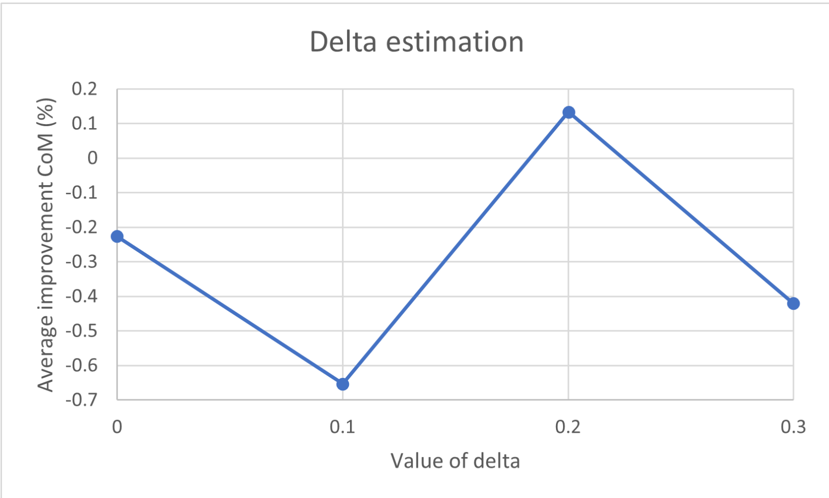

Figure 2: Graph of estimation of δ

For all data see Tables 7 and 8.

When performing this extension, the HA algorithm does not perform for the rotated items and bins (see Section 4.1.1) because this would require a very different approach to calculating the centre of mass.

4.2.3 Fragility Extension

Any bin containing at least one fragile item should be considered fragile. Therefore it should be discouraged to place a fragile item in a non-fragile layer (making it fragile), especially if it is reasonably well-filled already. This is done by adding a term to the score function to look like this:

$$\begin{aligned} S(j,\ell,p) = \rho\,\frac{P(j,\ell,p)}{2w_j+2d_j} &+ \mu\frac{\sum_{k\in J_\ell} w_k d_k}{WD} \\ &- \varepsilon\frac{\sum_{k\in J_\ell} w_k d_k}{WD}(f_j - F_\ell)f_j - (1-\rho-\mu-\varepsilon)\frac{|H_\ell - h_j|}{H_\ell}, \end{aligned} \tag{9}$$

where $\varepsilon \in [0,1]$ is a parameter such that $\rho+\mu+\varepsilon \leq 1$. Also, $f_j = 1$ if item j is fragile and 0 otherwise, and $F_\ell = 1$ if layer ℓ is already fragile and 0 otherwise. The new term penalizes proportional to the area of the layer that has already been filled, because making a layer fragile while it was almost full should be strongly discouraged.

The formulation of the 1D-BPP is changed to the following:

$$\begin{aligned}
&\text{minimize} && z = (1-p_f)\sum_{i=1}^{n} y_i + p_f\sum_{i=1}^{n} f_i && && \text{(10a)}\\
&\text{subject to} && \sum_{j=1}^{n} h_j x_{ij} \leq H y_i, && i \in N = \{1,...,n\}, && \text{(10b)}\\
& && \sum_{i=1}^{n} x_{ij} = 1, && j \in N, && \text{(10c)}\\
& && f_i \geq x_{ij}, && j \in F, && \text{(10d)}\\
& && y_i \in \{0,1\}, && i \in N, && \text{(10e)}\\
& && x_{ij} \in \{0,1\}, && i \in N, j \in N, && \text{(10f)}
\end{aligned}$$

where $f_i = 1$ if the ith bin is fragile (at least one fragile layer inside) and 0 otherwise. Also, p_f is the parameter indicating the importance of minimizing the number of fragile bins with respect to minimizing the total number of bins. F is the set containing the indices of all fragile layers.

The objective function is modified to include the minimization of the number of fragile bins (Equation 10a). Equation 10d ensures that every bin containing at least one fragile layer is considered fragile.

Again, due to limited time, ε is estimated based on a small instance of Class 5 (see Section 4.3) with $n = 150$ and 5 repetitions, choosing the value that decreases the number of fragile bins most while keeping the total number of bins somewhat the same. This time, also the value for the fragility importance p_f is estimated. Table 9 shows that there was virtually no increase in the total number of bins for any value for ε or the fragility importance. There was, however, a definite difference in the number of fragile bins. With

$\varepsilon = 0.5$ and $p_f = 0.7$, there was the biggest decrease in the number of fragile bins, see the following figures:

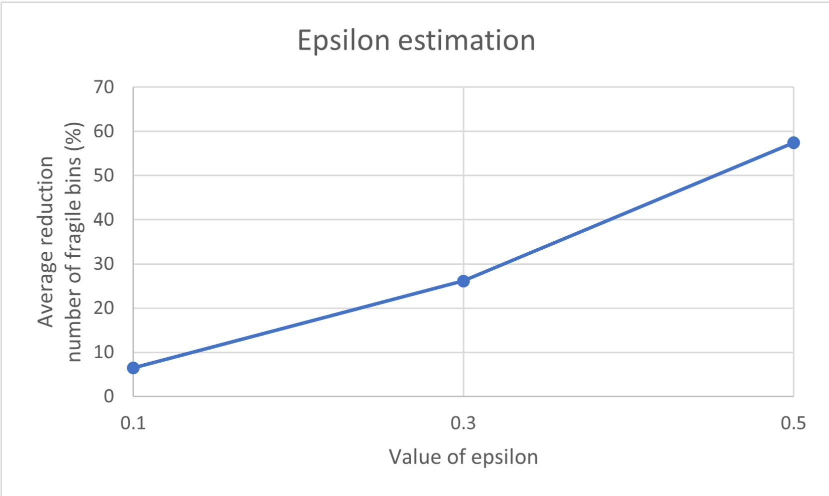

Figure 3: Graph of estimation of ε

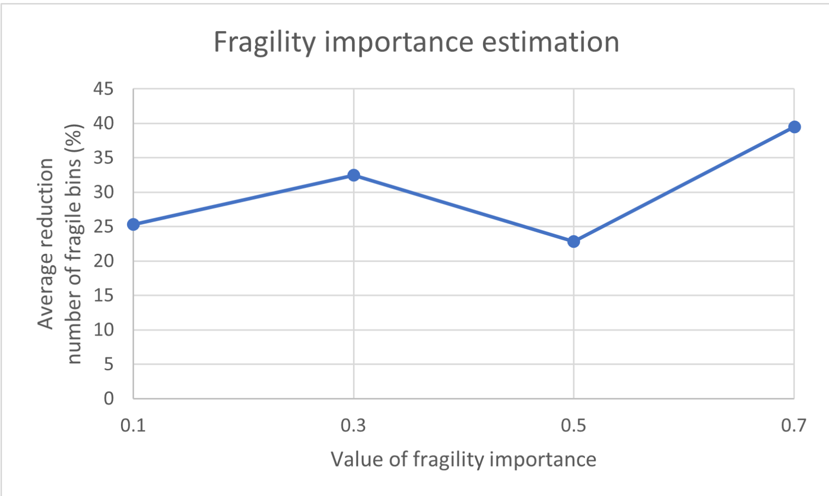

Figure 4: Graph of estimation of fragility importance

See Table 10 for all estimation results.

4.3 Evaluating Extension Performance

To evaluate performance of the three extensions, they are tested with samples from 8 different classes, as proposed by Faroe et al. (2003). For Classes 1-5, the bin size is $W = H = D = 100$ and there are 5 types of items:

Type 1: item width, height and depth uniformly random in $[1, \frac{1}{2}W]$, $[\frac{2}{3}H, H]$ and $[\frac{2}{3}D, D]$ respectively;

Type 2: item width, height and depth uniformly random in $[\frac{2}{3}W, W]$, $[1, \frac{1}{2}H]$ and $[\frac{2}{3}D, D]$ respectively;

Type 3: item width, height and depth uniformly random in $[\frac{2}{3}W, W]$, $[\frac{2}{3}H, H]$ and $[1, \frac{1}{3}D]$ respectively;

Type 4: item width, height and depth uniformly random in $[\frac{1}{2}W, W]$, $[\frac{1}{2}H, H]$ and $[\frac{1}{2}D, D]$ respectively;

Type 5: item width, height and depth uniformly random in $[1, \frac{1}{2}W]$, $[1, \frac{1}{2}H]$ and $[1, \frac{1}{2}D]$ respectively;

Classes k with $k = 1, ..., 5$ consist of items that have a 60% chance to be of type k and a 10% chance to be of each of the other types. Classes 6 are as follows:

Class 6: bin size $W = H = D = 10$ and all item dimensions are uniformly random in $[1, 10]$;

Class 7: bin size $W = H = D = 40$ and all item dimensions are uniformly random in $[1, 35]$;

Class 8: bin size $W = H = D = 100$ and all item dimensions are uniformly random in $[1, 100]$;

The original Tabu Search algorithm and the extended algorithms are run on all 8 of these classes for $n \in \{50, 100, 150, 200\}$, a time limit of 60 seconds and with 10 repetitions. Since the time limit is the same, similar results to the original algorithm, while including other constraints or objectives, means they were implemented efficiently. How the three extensions are evaluated exactly and how the other item parameters are set is discussed in the next three sections. Note that the original TS algorithm and the extended algorithms were performed on the same generated samples.

4.3.1 Weight Limit

Firstly, to implement a weight limit, the items need a weight and the bins need a weight limit. The weight of an item is assumed to be proportional to its volume with a random dispersion between 10% lighter and 10% heavier. So the weight is set as follows:

$$m_j = v_j(1 + d_m), \qquad d_m \in [-0.1, 0.1], \tag{11}$$

where m_j is the weight of item j, v_j is the volume of item j and d_m is a random dispersion factor. For the results of the first extension, the number of bins in the optimal solution is considered when setting the weight limit at three different values; $M \in \{0.5V_{\text{bin}}, 0.7V_{\text{bin}}, 0.9V_{\text{bin}}\}$, with V_{bin} the volume of a bin, i.e. $V_{\text{bin}} = WHD$.

To evaluate the performance of this extension, the number of bins in the results with the various weight limits are reported, as well as the number of bins in the result of the original algorithm (without a weight limit). The proportion of the bins in the original algorithm's result that would satisfy the three different weight limits is also reported to determine how strict the weight limits actually were.

If the weight limit extension manages to keep a similar number of bins in the result when the weight limit constraint is non-binding (original TS result has a high proportion of bins satisfying the weight limit) this means that the extended algorithm is efficient. When the weight limit becomes binding (original TS result has a low proportion of bins satisfying the weight limit) it is possible that the number of bins in the result increased significantly while the extension is still successful.

4.3.2 Weight Distribution

For the weight distribution, the weights of items is set the same as with the weight limit (see Section 4.3.1). The performance of the weight distribution extension is evaluated by comparing the following values of the algorithm's resulting configuration; the number of bins used, the average center of mass over all bins, the number of invalid results. The layers that are given to the 1D-BPP might result in an infeasible problem if the centre of mass constraint is too strict. If that is the case, the algorithm cannot come to a result and it is considered invalid. These values are reported and compared for $p_{\text{CoM}} \in \{0.5, 0.7, 0.9\}$ (see Equations 8a - 8d). Both the absolute centres of mass and the relative centres of mass are reported, but the relative centres of mass are considered in the evaluation, to allow for a fair comparison between the bin sizes.

If the centres of mass have moved towards the middle while not increasing the total number of used bins and getting few invalid results, the weight distribution extension is considered a successful extension. Note that this extension is compared to the original TS algorithm where the HA algorithm is not executed on synchronously rotated items and bins (see Section 4.1.1).

4.3.3 Fragility

Lastly, for the fragility extension, it is assumed that any item has a 20% chance of being fragile. Here, the performance is evaluated by comparing the total number of bins used and the number of fragile bins in the found optimal solution, with those in the result of the original TS algorithm.

If the total number of used bins is not much greater than the total number of used bins in the original TS algorithm's result, while the number of fragile bins has significantly decreased, this extension is considered successful.

5 Results

The algorithm is coded and run in Java IDE Eclipse on a Windows 10 computer with an AMD Ryzen 7 5800X 8-Core 3.80GHz CPU. IBM ILOG CPLEX Optimization Studio is the optimization software package used to solve the linear programming problems (1D-BPP).

5.1 Weight Limit

The average increase in the resulting number of bins when introducing the weight limit is given in Table 1:

Table 1: Average increase in number of bins - weight limit extension

	Weight Limit		
	$\mathrm{WL} = 0.5V_{\text{bin}}$	$\mathrm{WL} = 0.7V_{\text{bin}}$	$\mathrm{WL} = 0.9V_{\text{bin}}$
average increase in number of bins (vs original TS)	41.88%	7.08%	0.26%

Also, the average proportion of the bins in the original TS solution that would have been under the weight limit are given in Table 2:

Table 2: Average proportion of original solution under weight limit

	Weight Limit		
	$\mathrm{WL} = 0.5V_{\text{bin}}$	$\mathrm{WL} = 0.7V_{\text{bin}}$	$\mathrm{WL} = 0.9V_{\text{bin}}$
average percentage of bins in original TS solution meeting limit	12.59%	50.82%	97.10%

All results from comparing the number of bins used by the extended algorithm with the original Tabu Search algorithm can be found in Section A.4.

Table 1 shows that with a weight limit that is a very loose constraint ($\mathrm{WL} = 0.9V_{\text{bin}}$), the algorithm performs similar to the original Tabu Search algorithm and finds solutions with nearly the same amount of bins (0.26% more bins on average). This indicates that the consideration of weight does not hurt the performance of the algorithm itself much. This minimal increase can be explained by the 2.9% (see Table 2) of bins in the original solutions that would not have met the weight limit.

When the weight limit is more strict, the number of bins naturally grows. This is mostly the case when $\mathrm{WL} = 0.5V_{\text{bin}}$. At $\mathrm{WL} = 0.7V_{\text{bin}}$, the results are still relatively close to the optimal values found by the original algorithm (7.08% higher). This, while only about half of the bins in the result of the original algorithm would have met this stricter weight limit. When $\mathrm{WL} = 0.5V_{\text{bin}}$, the 41.88% increase in number of used bins is easily explained by the fact that 87.41% of the bins in the original algorithm's solution would have been over the limit. To get those bins under the limit, items in those bins would have to be placed in other bins, but with only 12.59% of the bins under the limit, new bins definitely need to be used to satisfy that constraint.

Considering the ease of implementation and the average performance, this extension could be very useful when dealing with a weight limit apart from a volume constraint in a three-dimensional bin packing scenario. The weight here could also be replaced for some other measure that is added between items and must be limited per bin.

5.2 Weight Distribution

A summary of the results when introducing the weight distribution constraint is given in Table 3:

Table 3: Summary statistics - weight distribution extension

	weight distribution proportion		
	0.5	0.7	0.9
average increase in number of bins (vs original TS in %)	0.09	0.20	0.01
average increase in CoM towards middle (vs original TS in %)	0.01	0.02	0.02
average proportion of invalid results (in %)	10	2.19	0

All results of the simulation experiment are given in Section A.5, where Table 12 reports the resulting centre of mass as an absolute position in the bin and Table 13 reports the relative centre of mass with respect to the dimensions of the bin. The columns denoted "invalid" report the proportion of simulations resulting in an invalid result due to an impossible 1D BPP with centre of mass constraints.

None of the three weight distribution proportion constraints had a significant effect on moving the average centre of mass towards the middle of the bins, with at most 0.02% movement. The average number of used bins has also increased insignificantly, with at most 0.20%, though this increase was generally bigger than the movement of the CoM. Lastly, a stricter weight distribution constraint has a definite effect on the number of experiments resulting in invalid results. With the centre of mass being only allowed in the middle half of the bin, $1/10$ of the experiments leads to an invalid result.

The improvement in the centre of mass of the bins is too insignificant to be used in practice, especially with the significant possibility for invalid results. The average centres of mass of any sample lie at most 10% away from the middle in the results of the original TS algorithm (values for CoM_w and CoM_d in Table 13 for the original TS algorithm are at least 0.40%). This together indicates that relatively little could be gained in the weight distribution but mostly that implementing a weight distribution constraint to the Lodi et al. (2002) Tabu Search framework as done in this paper is not the approach to do so. The method of placing items in the corners of layers and only in normal positions inherently clashes with the desire to move the centre of mass towards the middle of the bin.

5.3 Fragility

The results of the simulation experiments with fragile items and bins are given in Section A.6. The average increase in the number of used bins and average decrease in the number of used fragile bins for the different sample sizes and different Classes is given in the following two tables:

Table 4: Relative number of bins and fragile bins in Fragility TS vs original TS per sample size

n	relative n_{bin} (%)	relative n_{frag} (%)
50	100.98	82.81
100	101.09	75.59
150	101.19	76.51
200	100.51	75.33
Average	100.94	78.56

Table 5: Relative number of bins and fragile bins in Fragility TS vs original TS per class

Class	relative n_{bin} (%)	relative n_{frag} (%)
1	100.46	77.28
2	100.81	90.21
3	100.78	82.50
4	100.04	81.60
5	101.39	64.13
6	100.17	82.42
7	102.19	74.01
8	101.68	76.34

From Table 4 it is evident that generally the number of fragile bins in the resulting solution has decreased drastically (78.56%) while the total number of bins in the resulting solution has remained close to the same (100.94% on average). The extended algorithm seems to have a harder time finding solutions with less fragile bins in smaller samples (82.81% on average for $n = 50$). This could be explained by the fact that with a smaller sample it is more likely the original algorithm has found a near-optimal solution, so decreasing the number of fragile bins while keeping a similar number of total bins might be more difficult.

Table 5 shows that the extended algorithm performs generally well on all classes. The algorithm finds more improvement to the number of fragile bins in Class 5 ($W = H = D = 100$ and items from types 1-5 with an item having 60% of being of type 5). This is probably because the items are relatively small in these samples. Therefore a lot of items fit in a bin which in turn produces a lot of fragile bins in the original algorithm that does not consider fragility. Indeed, the proportion of bins in the original TS solution for Class 5 that are considered fragile is 70.77%, where that proportion for the other Classes is 57.54% (see Table 14). This means that an algorithm considering fragility had a lot to improve here.

6 Conclusions

Between the three explored expansions, using the proposed Tabu Search framework by Lodi et al., 2002 proved effective for the introduction of a weight limit and the minimization of fragile bins to the three-dimensional bin packing problem. Aiming to centre the weight distribution by setting constraints on the location of the centre of mass of the bins proved unsuccessful.

The addition of a loose weight limit resulted in allocations nearly as good as in instances without a weight limit. This shows that the consideration of weight and a weight limit does not hurt general performance of the algorithm and that the increased number of bins in the allocation with more stringent weight constraints is a direct consequence of the low weight limit. The ease of implementation and effectiveness of the algorithm make this a very appealing practical extension to the existing Tabu Search algorithm.

Moving the centre of mass of bins towards the physical centre of the bins by motivating specific item placements and constraining the allowed centres of mass for bins proved a bad fit with the proposed Tabu Search algorithm. When the constraints were easily met, the centres of mass hardly moved from those in the result of the original algorithm and when the constraints were more strict, it resulted in numerous invalid results because of impossible combination of layers. All this while getting resulting configurations with slightly more bins.

The consideration of fragile items and bins proved very effective in combination with the Tabu Search framework. The resulting allocations of this extended algorithm used barely any more bins while drastically reducing the number of fragile bins compared to the original algorithm that does not consider fragility. With greater amounts of items and relatively smaller items the extension was especially effective.

7 Discussions

Due to limited time, less attention could be paid to the estimation of parameters used in the evaluation $(\gamma, \delta, \epsilon, p_f)$ and as a result, these estimations are relatively unreliably. A better approach would be to do this estimation on the same samples and sample sizes as the final results to ensure the best values were chosen. Now the samples and sample sizes were less carefully chosen. Also, more and more exhaustive values for the parameters should have been considered. Now, only three or four values were considered per parameter.

To further evaluate the performance of the weight limit extension, one might consider more diverse weight limits and diverse weights of items. The weight of items as they were currently determined might be unrealistic with extreme weights. Very light and big items and very heavy and small items exist, including these might throw off the performance of the extension.

Adding the weight distribution to the objective could also have been explored as a potentially more reliable and better performing way of enforcing stability.

In the case of fragility, 20% of items being fragile could prove to be unrealistic. The fragility of items might be correlated to the volume or weight, implementing this could improve the reliability of the evaluation.

Further research might attempt to find a viable way of ensuring stable bin configurations, like ensuring the different layers can practically be stacked and will not just fall next to items on the layer below. Attempting to ensure a centered centre of mass clashes with the nature of the algorithm where items are forced into a corner. This makes it difficult to incorporate in the algorithm. It might be better as an algorithm performed on a resulting allocation that checks and recombines layers.

Since the weight limit and fragility extensions have proved effective, further research might consider their joint performance to see if this is as effective.

References

AGCS. (2019). Shipping safety - human error comes in many forms. *AGCS*. https://www.agcs.allianz.com/news-and-insights/expert-risk-articles/human-error-shipping-safety.html

Bischoff, E. E. & Ratcliff, M. (1995). Issues in the development of approaches to container loading. *Omega*, *23*(4), 377–390.

Bortfeldt, A. & Gehring, H. (1999). Zur behandlung von restriktionen bei der stauraumoptimierung am beispiel eines genetischen algorithmus für das containerbeladeproblem. *Logistik management* (pp. 83–100). Springer.

Bortfeldt, A. & Wäscher, G. (2013). Constraints in container loading–a state-of-the-art review. *European Journal of Operational Research*, *229*(1), 1–20.

Davies, A. P. & Bischoff, E. E. (1999). Weight distribution considerations in container loading. *European Journal of Operational Research*, *114*(3), 509–527.

Faroe, O., Pisinger, D. & Zachariasen, M. (2003). Guided local search for the three-dimensional bin-packing problem. *Informs journal on computing*, *15*(3), 267–283.

Gehring, H., Menschner, K. & Meyer, M. (1990). A computer-based heuristic for packing pooled shipment containers. *European Journal of Operational Research*, *44*(2), 277–288.

Gendreau, M., Iori, M., Laporte, G. & Martello, S. (2006). A tabu search algorithm for a routing and container loading problem. *Transportation Science*, *40*(3), 342–350.

Kearney. (2018). Accelerating into uncertainty. *CSCMP's Annual State of Logistics Report*.

Lodi, A., Martello, S. & Vigo, D. (2002). Heuristic algorithms for the three-dimensional bin packing problem. *European Journal of Operational Research*, *141*(2), 410–420.

Lodi, A., Martello, S. & Vigo, D. (2004). Tspack: A unified tabu search code for multi-dimensional bin packing problems. *Annals of Operations Research*, *131*(1), 203–213.

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

Martello, S. & Toth, P. (1990). *Knapsack problems: Algorithms and computer implementations*. John Wiley & Sons, Inc.

Rodrigue, J.-P. (2020). *The geography of transport systems the geography of transport systems* (5th ed.). Routledge.

Scheithauer, G. & Terno, J. (1996). A heuristic approach for solving the multi-pallet packing problem. *Decision Making under Conditions of Uncertainty (Cutting–Packing Problems); Mukhacheva, EA, Ed*, 140–154.

Sommerweiß, U. (1996). Modeling of practical requirements of the distributers packing problem. *Operations research proceedings 1995* (pp. 427–432). Springer.

A Tables

A.1 γ estimation

Table 6: γ parameter estimation (Class 5, $n = 50$, 10 repetitions)

γ	Weight Limit (of volume of bin)			Average
	$0.5V_{\text{bin}}$	$0.7V_{\text{bin}}$	$0.9V_{\text{bin}}$	
0.35	11.2	9.1	8.3	9.533333
0.4	10.7	9.5	8.6	9.6
0.45	10.1	9.7	8.3	**9.366667**
0.5	11.3	9.1	8.4	9.6
0.55	10.9	8.9	8.9	9.566667

A.2 δ estimation

Table 7: δ parameter estimation - number of bins (Class 5, $n = 150$, 3 repetitions)

δ	Original TS	Weight Distribution TS			Average increase (%)
		CoM proportion			
		0.7	0.8	0.9	
0	23.27	26.7	23.25	23.8	5.66
0.1	22.07	22	22	22.8	**0.91**
0.2	22.53	22.25	24.6	22.4	2.44
0.3	21.20	23.2	20.8	20.6	1.57

Table 8: δ parameter estimation - CoM improvement (Class 5, $n = 150$, 3 repetitions)

δ	Relative CoM improvement x (%)			Relative CoM improvement y (%)			Average improvement (%)
	CoM proportion			CoM proportion			
	0.7	0.8	0.9	0.7	0.8	0.9	
0	0.13	0.20	-1.47	-0.50	0.95	-0.67	-0.23
0.1	0.11	0.27	-0.35	-1.41	-1.27	-1.26	-0.65
0.2	0.63	0.46	-0.68	-0.46	0.87	-0.02	**0.13**
0.3	0.38	-2.17	-0.04	0.00	-0.80	0.11	-0.42

A.3 ε estimation

Table 9: ε parameter estimation - number of bins (Class 5, $n = 150$, 5 repetitions)

ε	Fragility TS				Average performance (vs original TS in %)
	Fragility Importance				
	0.1	0.3	0.5	0.7	
0.1	24.4	22.2	23.2	22.2	100
0.3	21.6	22.4	21.6	24.2	100
0.5	22	23	20.2	20.4	100.23
Average performance (vs original TS in %)	100.3	100	100	100	

Table 10: ε parameter estimation - number of fragile bins (Class 5, $n = 150$, 5 repetitions)

ε	Fragility TS				Average decrease (vs original TS in %)
	Fragility Importance				
	0.1	0.3	0.5	0.7	
0.1	16.8	13	10	13.4	7.44
0.3	6.6	9.2	9.8	3	26.17
0.5	7.6	6.4	2.6	9.8	**57.46**
Average decrease (vs original TS in %)	25.32	32.45	22.82	**39.51**	

A.4 Weight Limit Extension Results

Table 11: Results Weight Limit Extension Simulation (time limit = 60s, 10 repetitions)

Class	n	Original TS	Proportion of bins under weight limit			Weight Limit TS (WL = weight limit)		
			WL = $0.5V_{\text{bin}}$	WL = $0.7V_{\text{bin}}$	WL = $0.9V_{\text{bin}}$	WL = $0.5V_{\text{bin}}$	WL = $0.7V_{\text{bin}}$	WL = $0.9V_{\text{bin}}$
		n_{bins}	%	%	%	n_{bins}	n_{bins}	n_{bins}
1	50	14.6	11.7	53.4	100	20.6	15.4	14.7
	100	26	9.3	51.6	98.1	37.6	27.9	26
	150	40.5	5.4	42.5	98.5	58.4	42.9	40.5
	200	50.6	3.9	45.5	97	73.8	53.8	50.5
2	50	14.2	4.6	60.9	100	20	15	14.2
	100	26.5	4.2	51.7	98.9	37.9	28.1	26.6
	150	39	2.6	50.7	99	56.4	41.1	38.9
	200	50.8	2.3	46.9	98.3	73.6	54.3	50.9
3	50	14.2	9.4	67	100	19.7	14.8	14.2
	100	26.7	7.1	52.1	99.6	38	27.9	26.8
	150	38.8	4.7	43.5	97.9	56.6	41.3	38.8
	200	50.6	4.9	46.8	97.8	73	53.8	50.7
4	50	29.2	60.3	85.2	100	33.7	29.4	29.2
	100	56	55.6	86.8	100	62.2	56.5	56
	150	90.1	64.8	87.3	99.9	99.4	91	90.1
	200	115.9	60.8	85.7	98.9	128.5	116.9	116
5	50	8	12.9	69.7	100	10.7	8.2	8
	100	15.7	7.5	52.8	99.4	22.6	16.7	15.7
	150	21.3	3.8	43.7	97.7	31.4	22.9	21.3
	200	27.4	2.1	34.4	97.3	41.4	30.1	27.4
6	50	12.1	10.6	51.5	94.5	17.4	13.1	11.9
	100	20.1	2.9	23.3	87	32.5	23.8	20.3
	150	27.5	2.5	14.7	80.4	45.6	33.2	27.9
	200	37.6	1.3	13.4	74.4	63.4	46.2	38.3
7	50	7.7	12.6	69	100	10.5	7.8	7.9
	100	13	7.7	52.4	100	18.3	13.8	13.1
	150	19.3	2.4	31.3	100	28.8	21.1	19.3
	200	23.9	3.1	26.5	97.3	36.1	26.2	23.9
8	50	10.1	12.5	67.6	100	13.8	10.5	10.2
	100	18.5	3.8	48.8	99.1	27.1	20	18.5
	150	26.8	3.9	34.8	98.8	39.1	29	26.8
	200	34.3	1.6	34.9	97.5	51.3	37.6	34.3

A.5 Weight Distribution Extension Results

Table 12: Results Weight Distribution Extension Simulation (time limit = 60s, 10 repetitions)

Class	n	Original TS			Weight Distribution TS (WD = allowed centre of mass proportion)											
					WD = 0.5				WD = 0.7				WD = 0.9			
		n_{bins}	CoM$_w$	CoM$_d$	n_{bins}	CoM$_w$	CoM$_d$	invalid	n_{bins}	CoM$_w$	CoM$_d$	invalid	n_{bins}	CoM$_w$	CoM$_d$	invalid
1	50	14.7	46.77	42.42	14.5	46.87	43.05	0.4	14.8	46.89	42.3	0.1	14.8	46.8	42.31	0
	100	26.1	47.64	43.06	25.6	47.68	43.19	0.2	26.1	47.6	43.02	0	26.1	47.6	43.02	0
	150	40.7	47.65	43.39	40.8	48.02	43.38	0	40.8	47.98	43.35	0	40.8	47.98	43.35	0
	200	50.6	48.09	43.52	50.7	48.11	43.59	0.1	50.6	48.05	43.57	0	50.6	48.05	43.57	0
2	50	14.5	42.53	42.48	14.4	42.01	42.4	0.1	14.6	42.41	42.44	0	14.5	42.56	42.5	0
	100	26.7	42.89	42.97	27	42.9	43.03	0.1	26.8	42.79	43.01	0	26.7	42.88	42.97	0
	150	39.3	43.25	43.22	39.3	43.24	43.25	0	39.3	43.23	43.24	0	39.3	43.23	43.24	0
	200	51.3	43.23	43.16	51.3	43.32	42.95	0.2	51.4	43.33	43.04	0	51.4	43.33	43.04	0
3	50	14.5	42.59	46.42	14.2	42.76	46.55	0.1	14.3	42.93	46.38	0	14.4	42.67	46.24	0
	100	26.9	43.46	47.18	26	42.66	47.07	0.6	26.4	43.1	47.44	0.1	26.9	43.41	47.14	0
	150	38.9	43.75	47.81	38.8	43.72	47.76	0	38.9	43.74	47.85	0	38.9	43.74	47.85	0
	200	50.8	43.43	48.26	50.8	43.46	48.31	0.2	50.8	43.46	48.23	0	50.8	43.46	48.23	0
4	50	29.2	40.98	39.87	29.2	40.97	39.93	0	29.2	40.97	39.92	0	29.2	40.97	39.92	0
	100	56.2	40.41	40	56.1	40.42	40.18	0.1	56.2	40.37	39.97	0	56.2	40.37	39.97	0
	150	90.2	40.59	39.9	90.2	40.56	39.96	0	90.2	40.56	39.96	0	90.2	40.56	39.96	0
	200	116.3	39.99	40.1	116.1	39.98	40.26	0	116.1	39.98	40.26	0	116.1	39.98	40.26	0
5	50	8.3	43.7	42.88	8.5	43.69	42.78	0.2	8.4	43.89	42.97	0	8.3	43.7	43.04	0
	100	15.9	45.11	45.01	15.8	45.12	45.07	0	15.8	45.1	45.04	0	15.8	45.1	45.04	0
	150	21.6	45.38	45.59	22.1	45.43	45.34	0.2	22.2	45.32	45.44	0.1	21.7	45.31	45.57	0
	200	27.7	45.94	45.91	27.9	45.88	45.85	0	27.9	45.88	45.85	0	27.9	45.88	45.85	0
6	50	12.1	4.61	4.51	12.3	4.62	4.52	0	12.2	4.62	4.51	0	12.1	4.61	4.52	0
	100	20.3	4.67	4.69	20.3	4.67	4.69	0	20.3	4.67	4.69	0	20.2	4.67	4.69	0
	150	28	4.77	4.79	27.7	4.78	4.78	0.1	27.9	4.77	4.78	0	27.9	4.77	4.78	0
	200	37.8	4.82	4.83	37.6	4.82	4.83	0.2	37.7	4.81	4.82	0.1	37.8	4.82	4.82	0
7	50	8	17.29	17.45	8.3	17.15	17.38	0.1	8.2	17.4	17.35	0.1	7.9	17.21	17.57	0
	100	13.2	17.62	17.82	13.1	17.6	17.85	0	13.2	17.61	17.88	0	13.3	17.62	17.9	0
	150	19.6	18.18	18.1	19.7	18.19	18.1	0	19.7	18.18	18.13	0	19.7	18.19	18.13	0
	200	24.1	18.3	18.49	24.3	18.28	18.46	0	24.3	18.28	18.47	0	24.3	18.28	18.47	0
8	50	10.4	43.9	43.42	10.4	43.97	43.6	0	10.2	44.26	43.49	0	10.3	44.27	43.16	0
	100	19.1	44.4	45.19	19.1	44.43	45.14	0.1	19.3	44.31	45.15	0.2	19.1	44.43	45.24	0
	150	27.1	45.53	44.97	27.2	45.59	45.02	0	27.2	45.6	45.03	0	27.2	45.6	45.03	0
	200	34.5	45.88	46.09	34.8	45.68	46.11	0.2	34.5	45.82	46.12	0	34.5	45.82	46.12	0

Table 13: Results Weight Distribution Extension Simulation with relative centre of mass (time limit = 60s, 10 repetitions)

Class	n	Original TS			Weight Distribution TS (WD = allowed centre of mass proportion)											
					WD = 0.5				WD = 0.7				WD = 0.9			
		n_{bins}	CoM$_w$	CoM$_d$	n_{bins}	CoM$_w$	CoM$_d$	invalid	n_{bins}	CoM$_w$	CoM$_d$	invalid	n_{bins}	CoM$_w$	CoM$_d$	invalid
1	50	14.7	0.47	0.42	14.5	0.47	0.43	0.4	14.8	0.47	0.42	0.1	14.8	0.47	0.42	0
	100	26.1	0.48	0.43	25.6	0.48	0.43	0.2	26.1	0.48	0.43	0	26.1	0.48	0.43	0
	150	40.7	0.48	0.43	40.8	0.48	0.43	0	40.8	0.48	0.43	0	40.8	0.48	0.43	0
	200	50.6	0.48	0.44	50.7	0.48	0.44	0.1	50.6	0.48	0.44	0	50.6	0.48	0.44	0
2	50	14.5	0.43	0.42	14.4	0.42	0.42	0.1	14.6	0.42	0.42	0	14.5	0.43	0.43	0
	100	26.7	0.43	0.43	27	0.43	0.43	0.1	26.8	0.43	0.43	0	26.7	0.43	0.43	0
	150	39.3	0.43	0.43	39.3	0.43	0.43	0	39.3	0.43	0.43	0	39.3	0.43	0.43	0
	200	51.3	0.43	0.43	51.3	0.43	0.43	0.2	51.4	0.43	0.43	0	51.4	0.43	0.43	0
3	50	14.5	0.43	0.46	14.2	0.43	0.47	0.1	14.3	0.43	0.46	0	14.4	0.43	0.46	0
	100	26.9	0.43	0.47	26	0.43	0.47	0.6	26.4	0.43	0.47	0.1	26.9	0.43	0.47	0
	150	38.9	0.44	0.48	38.8	0.44	0.48	0	38.9	0.44	0.48	0	38.9	0.44	0.48	0
	200	50.8	0.43	0.48	50.8	0.43	0.48	0.2	50.8	0.43	0.48	0	50.8	0.43	0.48	0
4	50	29.2	0.41	0.4	29.2	0.41	0.4	0	29.2	0.41	0.4	0	29.2	0.41	0.4	0
	100	56.2	0.4	0.4	56.1	0.4	0.4	0.1	56.2	0.4	0.4	0	56.2	0.4	0.4	0
	150	90.2	0.41	0.4	90.2	0.41	0.4	0	90.2	0.41	0.4	0	90.2	0.41	0.4	0
	200	116.3	0.4	0.4	116.1	0.4	0.4	0	116.1	0.4	0.4	0	116.1	0.4	0.4	0
5	50	8.3	0.44	0.43	8.5	0.44	0.43	0.2	8.4	0.44	0.43	0	8.3	0.44	0.43	0
	100	15.9	0.45	0.45	15.8	0.45	0.45	0	15.8	0.45	0.45	0	15.8	0.45	0.45	0
	150	21.6	0.45	0.46	22.1	0.45	0.45	0.2	22.2	0.45	0.45	0.1	21.7	0.45	0.46	0
	200	27.7	0.46	0.46	27.9	0.46	0.46	0	27.9	0.46	0.46	0	27.9	0.46	0.46	0
6	50	12.1	0.46	0.45	12.3	0.46	0.45	0	12.2	0.46	0.45	0	12.1	0.46	0.45	0
	100	20.3	0.47	0.47	20.3	0.47	0.47	0	20.3	0.47	0.47	0	20.2	0.47	0.47	0
	150	28	0.48	0.48	27.7	0.48	0.48	0.1	27.9	0.48	0.48	0	27.9	0.48	0.48	0
	200	37.8	0.48	0.48	37.6	0.48	0.48	0.2	37.7	0.48	0.48	0.1	37.8	0.48	0.48	0
7	50	8	0.43	0.44	8.3	0.43	0.43	0.1	8.2	0.44	0.43	0.1	7.9	0.43	0.44	0
	100	13.2	0.44	0.45	13.1	0.44	0.45	0	13.2	0.44	0.45	0	13.3	0.44	0.45	0
	150	19.6	0.45	0.45	19.7	0.45	0.45	0	19.7	0.45	0.45	0	19.7	0.45	0.45	0
	200	24.1	0.46	0.46	24.3	0.46	0.46	0	24.3	0.46	0.46	0	24.3	0.46	0.46	0
8	50	10.4	0.44	0.43	10.4	0.44	0.44	0	10.2	0.44	0.43	0	10.3	0.44	0.43	0
	100	19.1	0.44	0.45	19.1	0.44	0.45	0.1	19.3	0.44	0.45	0.2	19.1	0.44	0.45	0
	150	27.1	0.46	0.45	27.2	0.46	0.45	0	27.2	0.46	0.45	0	27.2	0.46	0.45	0
	200	34.5	0.46	0.46	34.8	0.46	0.46	0.2	34.5	0.46	0.46	0	34.5	0.46	0.46	0

A.6 Fragility Extension Results

Table 14: Results Fragility Extension Simulation (time limit = 60s, 10 repetitions)

Class	n	Original TS		Fragility TS	
		n_{bins}	n_{frag}	n_{bins}	n_{frag}
1	50	14.6	7.9	14.6	6.5
	100	26	14.8	26.3	11
	150	40.5	21.9	40.7	16.3
	200	50.6	29.2	50.7	22.8
2	50	14.2	7.1	14.4	6.9
	100	26.5	14.7	26.6	13.2
	150	39	22.8	39.5	20.3
	200	50.8	27	50.9	22.9
3	50	14.2	7	14.4	6.4
	100	26.7	14.9	26.9	12.4
	150	38.8	21.8	39.1	17
	200	50.6	28.3	50.7	21.9
4	50	29.2	9.3	29.2	7.3
	100	56	15.9	56	13.9
	150	90.1	24.2	90.1	19.9
	200	115.9	35.4	116.1	27.7
5	50	8	4.8	8.1	3.2
	100	15.7	12	16.1	7.6
	150	21.3	15.3	21.6	10.1
	200	27.4	20.5	27.5	12.4
6	50	12.1	7	12	5.6
	100	20.1	11.6	20.1	9.9
	150	27.5	18.5	27.7	15.5
	200	37.6	25.2	37.9	20.3
7	50	7.7	5.8	7.9	4.6
	100	13	10.6	13.3	8.3
	150	19.3	15.9	19.8	10.7
	200	23.9	20.1	24.2	14.3
8	50	10.1	6.2	10.3	5.4
	100	18.5	12	18.8	9
	150	26.8	17.8	27.4	12.7
	200	34.3	24.2	34.6	17.4

A.7 Variable or abbreviation explanations

Table 15: Variables and abbreviations and their explanations

Variable name or abbreviation	**Explanation**
TS	Tabu Search
N	Set of all n item indices: $\{1, ..., n\}$
w_j	Width of item j
h_j	Height of item j
d_j	Depth of item j
m_j	Weight of item j
f_j	Fragility of item j
v_j	Volume of item j $(= w_j h_j d_j)$
d_m	Random dispersion factor for weight of items ($d_m \in [-0.1, 0.1]$)
W	Width of bins
H	Height of bins
D	Depth of bins
M	Weight limit of bins
H_ℓ	Height of layer ℓ
F_ℓ	Fragility of layer ℓ
J_ℓ	The set of items currently packed in layer ℓ
V_{bin}	Bin volume $(= WHD)$
WL	Weight limit
$\text{CoM}_{w,\ell}$	The centre of mass in the width of layer ℓ
$\text{CoM}_{d,\ell}$	The centre of mass in the depth of layer ℓ
$\text{CoM}_{w,j}$	The centre of mass in the width of item j
$\text{CoM}_{d,j}$	The centre of mass in the depth of item j
β	HA algorithm height clustering parameter
α	Tabu Search algorithm filling function parameter
ρ	Score function perimeter parameter
μ	Score function area parameter
γ	Score function weight limit parameter
δ	Score function weight distribution parameter
ε	Score function fragility parameter
k	Neighbourhood size Tabu Search
k_{max}	Maximum neighbourhood size Tabu Search
τ_k	Tabu List length
ℓ	Maximum number of Tabu Search moves before diversification
x_{ij}	LPP variable indicating placement of layer j in bin i
y_i	LPP variable indicating usage of bin i
p_{CoM}	The proportion of the bin around the middle of the bin in which every centre of mass should fall