



Research Question: How effectively does an adapted three-dimensional First-Fit Decreasing heuristic algorithm pack rectangular boxes into fixed-size bins, as measured by packing density over 60 simulated trials, and what mathematical limitations of this approach can be identified?

Page count: 19

Introduction:

The bin packing problem is about fitting items of different sizes into as few boxes (of the same volume) as possible. Bin packing optimization is a well-known problem that emerged in the 1970s as part of optimization studies in major industries such as logistics. Modern-day companies like Amazon, FedEx, or DHL all face the task of fitting items into containers, delivery vans, or warehouse shelves. Using fewer containers reduces costs, saves fuel, and reduces environmental waste; therefore, there has been a prevailing motivation to find solutions to this problem (Romero et al., 2023). As this mathematical problem emerged, researchers proved that finding the optimal solution is very difficult; however, they discovered that the problem can be approximated accurately with minimal computational resources (Martello and Toth 1; Coffman et al. 187). It is essential to note that the bin packing problem is not only relevant in logistics but also in other industries, such as computer science, where applied solutions are utilized to organize computer memory (Martello and Toth 2).

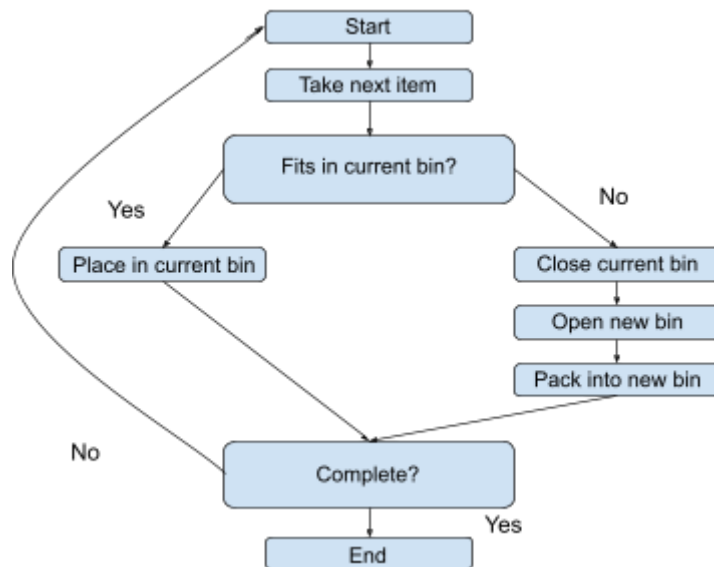
I became interested in this problem through personal experience. When I worked in a warehouse, one of my main jobs was packing products for shipment. Very often, I noticed how difficult it was to choose which items could fit together in a particular-sized box, as shapes made it impossible to close the lid, generally making the process of selecting a box for the items harder. This made me think about whether algorithms could make the process faster and more efficient by telling an employee in a warehouse exactly what to do to reduce the time taken to make an accurate packing decision. Moreover, if algorithms can be adapted to be faster and more accurate at arranging products for shipping than workers, firms can save on high packing material and shipping costs.

Existing algorithms:

Many algorithms exist for this problem, often considered to be in the heuristic algorithm category, because they do not find optimal solutions but use approximations and predefined rules to produce quick solutions. 2 of the most well-known and commonly taught are the so-called Next-Fit and First-Fit (decreasing) algorithms; these are, in fact, covered in the Edexcel A-level mathematics syllabus.

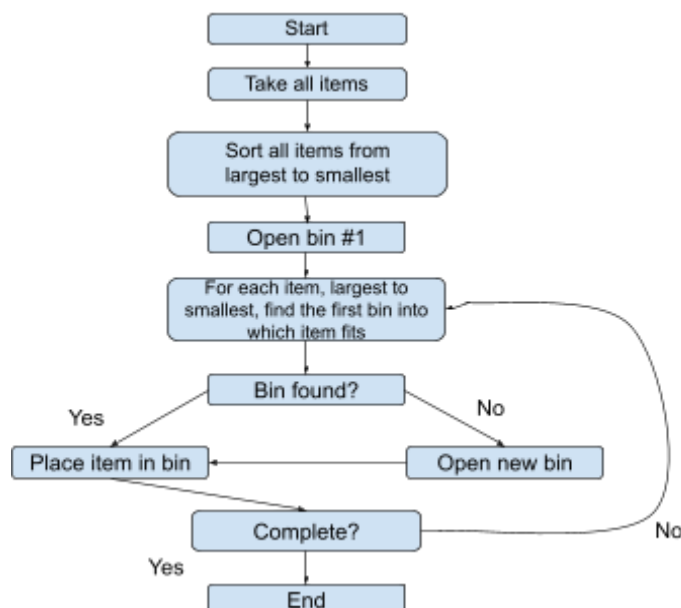
Firstly, the next fit involves, as the name implies, one open bin (a large box used for shipping, for example) into which items are simply added in order. If an item does not fit, the bin is closed, and the item is placed into a new one. Due to its simplicity, this algorithm is one of the fastest; however, it comes at the cost of efficiency, using up to twice as many boxes as is ideal (Coffman et al. 189). It is also important to note that this measure of efficiency is calculated under the condition that the item that follows the one being placed is unknown and cannot be placed or rearranged within pre-existing bins. A visual representation of the algorithm is shown in Figure 1.

Figure 1: Diagram explaining the pathway through which packing is achieved in the Next Fit algorithm, produced by the author.



The next algorithm is the first-fit decreasing (FFD). Theoretically, it works better. Items are sorted from largest to smallest, then placed in the first box they fit; the logic is visually represented in Figure 2. This algorithm uses no more than 11% extra boxes than the optimal solution as found by the work of Coffman et al. and Johnson et al.; therefore, it is much more efficient than next fit. However, both next fit and first fit decreasing algorithms can not be fully applied to the real world, this is because they work on the assumption that items are defined one or two dimensionally, that is, only defined by their volume or, for example, width and height, ignoring real-world constraints of shape (Martello and Toth 15).

Figure 2: Diagram explaining the pathway through which packing is achieved in the First Fit decreasing algorithm, produced by the author.



It is important to note that in logistics and warehousing, specialised three-dimensional bin-packing algorithms are used to achieve full spatial utilisation of packing boxes. Specialized algorithms are necessary, as the 2 aforementioned examples ignore real-life constraints and cannot sort real items for packing unless adapted specially to do so. One common type of 3 dimensional algorithms is “extreme point” heuristics, which treat corners and edges inside a box as “candidate positions” (simply a spot where the algorithm can place an item) and always try to place the next item on one of these points, for example, in the deepest-bottom-left corner, then stacking and layering items on top and beside each other. Studies show that well-tuned 3D heuristics of this type can achieve packing densities of 80–90% in realistic test cases, which are often the most common in real-life situations (Crainic et al., 2008).

Aim and rationale:

Research into the bin-packing problem motivated me to design my own three-dimensional bin-packing heuristic, inspired by methods taught in the Edexcel A-level syllabus. In this investigation, a simple one-dimensional First Fit Decreasing (FFD) algorithm will be adapted to a more realistic warehouse-style scenario, where items have three independent dimensions and must satisfy practical geometric constraints. The goal is to examine whether this adapted heuristic can achieve consistently useful packing densities while remaining simple enough to implement in an automated setting. (Edexcel, 2017).

Research question:

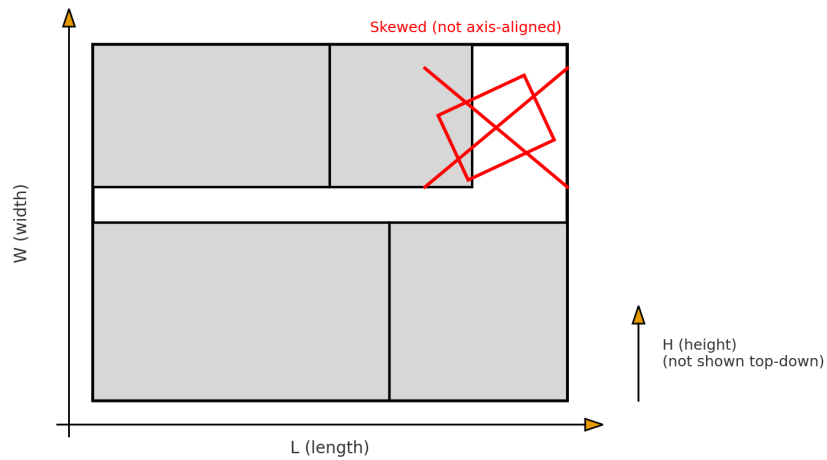
How effectively does an adapted three-dimensional First-Fit Decreasing heuristic algorithm pack rectangular boxes into fixed-size bins, as measured by packing density over 60 simulated trials, and what mathematical limitations of this approach can be identified?

Scope and assumptions:

Before applying the First Fit Decreasing algorithm, assumptions must be defined because the original heuristic does not apply to the full complexity of the three-dimensional bin packing problem. In this exploration, the following simplifying assumptions are used:

- All items and bins are modelled as rectangular prisms whose faces are aligned with the axes of the larger bin.
- Items are indivisible: each item must be packed as a whole and cannot overlap any other item.
- The primary constraint is volumetric capacity, described using the length, width, and height of both items and bins.
- Rotations are restricted to right angles, so each item has six possible orientations when placed; in other words, items must be axis-aligned, see figure 3 for example.
- Real-world factors such as irregular shapes, fragility, weight distribution, and stacking stability are ignored to keep the mathematics controlled and transparent. (Edexcel, 2017; Ali et al., 2022).

Figure 3: Diagram depicting alignment of axes, grey are aligned whilst red is not, generated by the author.



Bin Packing in Edexcel A-Level Mathematics

As mentioned, the Edexcel A-Level syllabus teaches three simple algorithms for the 1-dimensional version of the problem. I will be mainly exploring the first-fit decreasing (FFD) algorithm for my investigation. In Edexcel's FFD, the algorithm is 1-dimensional: every item is given a single size (usually a number), and bins have a maximum capacity that is also 1-dimensional (a numerical value). The aim is to fill the bins without exceeding this limit (Edexcel, 2017).

Why the 1-D Algorithms Do Not Fit the Problem

In the warehouse, boxes had three dimensions: length, width, and height. An algorithm that considers only one number, such as volume, does not capture this reality. For example, a long, thin box and a cube may have the same volume, but they cannot always be packed together because of their shapes.

Research in logistics also highlights this issue. Studies in three-dimensional bin packing show that items may be feasible in terms of volume but impossible to fit due to length or orientation (Zuo et al., 2022). The 1-dimensional models are useful for teaching and basic estimation, but they are too simple to represent real situations.

Adapting a Bin Packing Algorithm

Since the 1-D versions are not enough, an adaptation is needed. A practical approach is to take the First Fit Decreasing algorithm (in volume) and add a dimension gate, i.e., check if items fit via length, width, and height.

The adapted algorithm works as follows:

1. Compute each item's volume and sort them in decreasing order (much like FFD).
2. For each item, allow 90-degree rotations and pick an orientation that passes the following check:

$$l_i \leq L, w_i \leq W, h_i \leq H$$

3. Single-shelf rule: in a bin, items share the shelf only if their placed lengths sum to the bin's corresponding dimension (Width and height are checked per item; no multi-row stacking, items are always side by side). This single-shelf rule deliberately simplifies the three-dimensional problem but also introduces a clear efficiency trade-off. By forbidding stacking on multiple shelves, the algorithm leaves large regions of vertical space unused whenever long items quickly exhaust the 40 cm shelf space. While special conditions could have been implemented, it would stray away from the original first-fit decreasing heuristic, which would make the algorithm much more complex, beyond the scope of this internal assessment.
4. If an item does not fit the current bin's remaining shelf length, open a new bin.

This small change makes the algorithm more realistic for real boxes, while keeping the simple structure taught in Edexcel A-level Mathematics. Research in logistics supports similar ideas, adding dimension-based checks and orientation to improve feasibility (Romero et al., 2023; Ali et al., 2022).

Mathematical formulation:

To mathematically describe the bin packing problem, the following mathematical notations can be used:

1-dimensional bin packing problem:

Each bin has a capacity of C in cm^3 , while items are labeled $x_1, x_2, \dots, x_n$. Essentially, the key constraint in 1D packing is that the sum of all item sizes x_i in the bin J cannot exceed the capacity of the bin, or C :

$$\sum_{i \in B_j} x_i \leq C$$

For example, if the bin capacity is $C = 10 \text{ cm}^3$, and the bin J contains items of sizes 6 and 3, then $6 \text{ cm}^3 + 3 \text{ cm}^3 = 9 \text{ cm}^3 \leq 10 \text{ cm}^3$, the arrangement of items in a bin is valid.

3-dimensional check:

For each item i , its length, width, and height must be smaller than or equal to the length, width, or height of the bin that it is being placed in, represented by L, W, H . This dimension check added to the algorithm prevents items from being unable to fit into bins, even when they would fit by volume. A good way to visualise this is to imagine placing a broom into a shoe box; while the broom may have a smaller overall volume than the shoe box, length-wise, it could never fit.

$$l_i \leq L, w_i \leq W, h_i \leq H$$

Take, for example, a bin with the following dimensions: $40 \times 30 \times 30 \text{ cm}$, and an item with the dimensions of: $45 \times 20 \times 10 \text{ cm}$. While the volume of the bin is larger by almost a factor of 3, the algorithm proposed would see that the arrangement is invalid because the statement $l_i \leq L$ is not true.

After this point, the algorithm would try a new arrangement. This dimension check would need to be done for each item dimension and compared to the available leftover space in a bin.

Packing density:

Packing density and the calculation of wasted space are also highly important for understanding and evaluating the packing algorithm's efficiency. Let v_i be the volume of the item i in cubic centimeters, for $i = 1, 2, \dots, n$. The total item volume is:

$$\sum_{i=1}^n v_i$$

Packing density is therefore the total item volume divided by the number of bins multiplied by the volume of one bin, where N is the number of bins used and V_{bin} is the volume of one bin in cm^3 :

$$Density = \frac{\sum_{i=1}^n v_i}{N \times V_{bin}}$$

The density will always be represented by a value between 0 and 1; simply put, the larger the density, the more efficient the packing is. For instance, let's take the following scenario: total item volume is 50000 cm^3 , number of bins is 4, bin volume = 20000 cm^3 . If we enter these values into our density equation, we will get 0.625. This would mean that 62.5% of the total bin space is used.

1. Worked Mini-Example of the Adapted Algorithm:

Suppose we have bins of dimensions $40 \times 30 \times 30$ cm (length, width, height in cm).

We want to pack four boxes (items):

Box A: $35 \times 20 \times 10$ cm

Box B: $30 \times 15 \times 15$ cm

Box C: $25 \times 25 \times 12$ cm

Box D: $20 \times 18 \times 18$ cm

Step 1 – Calculate Volumes

$$A: 35 \text{ cm} \times 20 \text{ cm} \times 10 \text{ cm} = 7000 \text{ cm}^3$$

$$B: 30 \text{ cm} \times 15 \text{ cm} \times 15 \text{ cm} = 6750 \text{ cm}^3$$

$$C: 25 \text{ cm} \times 25 \text{ cm} \times 12 \text{ cm} = 7500 \text{ cm}^3$$

$$D: 20 \text{ cm} \times 18 \text{ cm} \times 18 \text{ cm} = 6480 \text{ cm}^3$$

Step 2 – Sort in decreasing volume order

C (7500 cm^3), A (7000 cm^3), B (6750 cm^3), D (6480 cm^3).

Step 3 – Place items using FFD + Dimension Gate

C (7500 cm^3): Start Bin 1. Fits (length $25 \text{ cm} \leq 40 \text{ cm}$, width $25 \text{ cm} \leq 30 \text{ cm}$, height $12 \text{ cm} \leq 30 \text{ cm}$).

A (7000 cm^3): Try Bin 1. By volume, there is space, but check the dimensions: length $\leq 40 \text{ cm}$ and width $\leq 30 \text{ cm}$. Yes, fits in Bin 1, but beside C does not (since $25 + 35 \text{ cm} > 40 \text{ cm}$, they cannot be on the same row). Therefore needs a new bin. Open Bin 2.

B (6750 cm^3): Try Bin 1. Fits dimensions ($30 \times 15 \times 15 \text{ cm}$) in leftover space with C (25 cm used in length, 15 cm left; B length = 30 cm - doesn't fit). Try Bin 2: A already takes 35 cm length, leaving only 5 cm. B is too long. So open Bin 3.

D (6480 cm^3): Try Bin 1 (space left: length 15 cm, width 30 cm, height 30 cm). D's length = 20 cm, too long. Try Bin 2 (space left: length 5 cm, width 30 cm, height 30 cm), too long again. Try Bin 3 (B uses 30 cm, leaving 10 cm). Too small. So, must open Bin 4.

Result:

Bins used: 4

$$\text{Total volume used} = 7500 \text{ cm}^3 + 7000 \text{ cm}^3 + 6750 \text{ cm}^3 + 6480 \text{ cm}^3 = 27,730 \text{ cm}^3$$

$$\text{Total bin volume} = 4 \times 36000 \text{ cm}^3 = 144000 \text{ cm}^3$$

$$\text{Packing density} = \frac{27,730 \text{ cm}^3}{144000 \text{ cm}^3} = 0.1925 \approx 0.193$$

$$\text{Wasted space} = 1 - 0.193 = 0.807$$

By volume alone, one might think these could all fit into 2 bins. But the dimension gate shows that their shapes stop this.

This illustrates the problem with 1D versions of the bin packing problem: volume says “yes,” but dimension check says “no.”

2. Worked Mini-Example of the Adapted Algorithm:

Suppose we have bins of dimensions $40 \times 30 \times 30$ cm (length, width, height in cm).

We want to pack four boxes (items):

P: $25 \times 30 \times 30$ cm

Q: $15 \times 30 \times 30$ cm

R: $20 \times 30 \times 30$ cm

S: $20 \times 30 \times 30$ cm

Step 1 – Calculate Volumes

$$P: 25 \text{ cm} \times 30 \text{ cm} \times 30 \text{ cm} = 22,500 \text{ cm}^3$$

$$Q: 15 \text{ cm} \times 30 \text{ cm} \times 30 \text{ cm} = 13,500 \text{ cm}^3$$

$$R: 20 \text{ cm} \times 30 \text{ cm} \times 30 \text{ cm} = 18,000 \text{ cm}^3$$

$$S: 20 \text{ cm} \times 30 \text{ cm} \times 30 \text{ cm} = 18,000 \text{ cm}^3$$

Step 2 – Sort in decreasing volume order

$$P(22,500 \text{ cm}^3), R(18,000 \text{ cm}^3), S(18,000 \text{ cm}^3), Q(13,500 \text{ cm}^3)$$

Step 3 – Place items using FFD + Dimension Gate

P(22,500) Start Bin 1. Fits length $25 \text{ cm} \leq 40 \text{ cm}$, width $30 \text{ cm} \leq 30 \text{ cm}$, height $30 \text{ cm} \leq 30 \text{ cm}$.

Q(13,500) Try Bin 1. By volume, there is space. Check dimensions: length $\leq 40 \text{ cm}$ and width $\leq 30 \text{ cm}$. Yes, fits in Bin 1, and beside P: $25 \text{ cm} + 15 \text{ cm} = 40 \text{ cm}$, length fits exactly on the same row. Place in Bin 1.

R(18,000) Try Bin 1. Bin 1 is now full ($25 \text{ cm} + 15 \text{ cm} = 40 \text{ cm}$ length used). No remaining shelf length. Open Bin 2. Fits dimensions: length $\leq 40 \text{ cm}$, width $\leq 30 \text{ cm}$, height $\leq 30 \text{ cm}$. Place in Bin 2.

S(18,000) Try Bin 1. No space. Try Bin 2. By volume, there is space. Check dimensions: length $\leq 40 \text{ cm}$ and width $\leq 30 \text{ cm}$. Yes, fits in Bin 2, and beside R: $20 \text{ cm} + 20 \text{ cm} = 40 \text{ cm}$, length fits exactly on the same row. Place in Bin 2.

Result:

Bins used: 2

$$\text{Total volume used} = 22500 \text{ cm}^3 + 18000 \text{ cm}^3 + 18000 \text{ cm}^3 + 13500 \text{ cm}^3 = 72,000 \text{ cm}^3$$

$$\text{Total bin volume} = 2 \times 36000 \text{ cm}^3 = 72,000 \text{ cm}^3$$

$$\text{Packing density} = \frac{72000 \text{ cm}^3}{72000 \text{ cm}^3} = 1.00$$

$$\text{Wasted space} = 1 - 1.00 = 0.00$$

This illustrates the advantage of favourable item dimensions in the adapted first-fit decreasing algorithm. When shapes align with bin constraints, the algorithm performs optimally.

Table 1: A comparison of worked examples 1 and 2.

Set	Total volume (cm ³)	Bins	Density	Wasted
Example 1 (A–D)	27,730	4	0.193	0.807
Example 2 (P–S)	72,000	2	1.000	0.000

The two examples of my heuristic algorithm illustrate 2 cases: one where it is efficient and one where it is not. In the second example, the items are of such size that they can perfectly fill the bins (25 cm + 15 cm, 20 cm + 20 cm). Because of the favourable item sizes, the algorithm can reach a density of 1, meaning 100% of the bin space is filled. The first example shows a case in which several items are too long to fit in a single bin, totaling more than 40. This forces the algorithm to open 4 different bins for 4 items, resulting in a density of 0.193.

The data shows that side-length conflicts pose the greatest challenge when applying this algorithm. While in a warehouse setting, this model may be efficient for placing items into boxes, the robotic intuition would still ignore factors such as weight distribution, fragility, and the ability to pack items across multiple shelves. The flaws in this model highlight the difficulties of applying solutions to the bin packing problem for real-life applications. However, it is important to note that the examples were chosen specifically to depict best- and worst-case scenarios; therefore, a randomly generated simulation may provide a more realistic picture of the algorithm's efficacy.

Methodology:

Modelling assumptions and constraints:

This section explains the modelling assumptions, describes the adapted First Fit Decreasing heuristic in detail, and outlines the simulation and statistical analysis. The focus is on making each step of the algorithm explicit so that the resulting bin counts and packing densities can be reproduced and interpreted in relation to the research question.

Experimental design:

The investigation will systematically analyze the bin-packing efficiency of the previously described, adapted First Fit decreasing algorithm under realistic geometric constraints. Bin packing will

be simulated using custom Python code, and 60 independent trials will be generated, with 18 items of random dimensions placed into bins of fixed size.

In each trial, n items are created. For each item, integer sizes were chosen that fit the bin: length 6–35 cm, width 6–30 cm, height 6–30 cm. I allow 90° rotations, so an item can be turned to meet the bin limits ($\leq 40, \leq 30, \leq 30$). I run N trials (e.g., $N = 60, n = 18$) and use a fixed random seed to ensure reproducibility. Both the lower and upper bounds of the items are chosen so that they always fit within the bins' dimensions, ensuring that all items can be packed.

Independent variables and success metrics:

To evaluate the adapted heuristic, the following variables are recorded for each trial:

1. Number of fixed-size bins used. This counts the number of fixed-size bins opened by the algorithm. Fewer bins indicate a more efficient packing strategy in practical warehouse terms.
2. Packing density is defined as the ratio of total item volume to total available bin volume across all bins in a trial. Values closer to 1 indicate that the algorithm uses space more efficiently:

$$\frac{\Sigma (\text{item volumes})}{\# \text{ of bins} \times \text{bin volume}} = \text{packing density}$$

3. Wasted space, representing the proportion of bin volume that remains empty. Larger values reveal configurations where the heuristic fails to exploit the available three-dimensional space.

$$1 - \text{packing density} = \text{wasted space}$$

Reproducibility and randomization:

For the simulation, a pseudorandom seed was used (`random.seed(42)`) to ensure results are replicable and can be regenerated.

Algorithmic procedure:

1. Item Generation: The dimensions are chosen as above, and each item's volume is then calculated.
2. Sorting: Items for each trial are sorted in descending order of volume, as FFD prescribes.
3. Dimension Gate with Rotations: For each item, all six right-angle orientations are checked. The first orientation that passes all three-dimensional checks is selected for placement; this means the optimal orientation will not be selected, as it is outside the scope of the first-fit-decreasing algorithm.
4. Placement Rule: Attempt to place the item into the first bin with available length along the shelf, observing the single-shelf rule. If it doesn't fit, open a new bin.
5. Completion: Process continues until all items are packed.

6. Data collection: For each trial, the number of bins used, packing density, and wasted space will be recorded. Formulas used are as defined in the independent variables section.

Statistical analysis methods:

- 1) Descriptive statistics: Mean, median, and standard deviation, and bins used. Mean, median, standard deviation, and interquartile range will be calculated for the number of bins used across all trials, packing density, and wasted space.
- 2) Normality testing: The Shapiro–Wilk test was used to assess whether the packing density values follow approximately a normal distribution. While it is beyond the scope of the mathematics AI SL syllabus, A.Ghasemi and S. Zahediasl (2012) state in their research that visual assessment of a box plot may be inaccurate; therefore, to ensure scientific rigor, this method was chosen. The following hypotheses were derived:

- a) Null hypothesis H_0 : The packing density values come from a normal distribution $X \sim N(\mu, \sigma^2)$.
- b) Alternate hypothesis H_1 : The packing density values do not form a normal distribution, i.e. X is not normally distributed.

The test, implemented in Python's `scipy.stats`, produced $W = 0.9903$ and $p = 0.916$. Since $p > 0.05$, there is no evidence to reject H_0 , so treating packing density as approximately normal is reasonable for summarising it with a mean and standard deviation.

- 3) Correlation: Pearson's r for bins used and packing density will be used. Exploring the relationship between the number of bins used and packing density helps understand how the algorithm output (bins) relates to efficiency (density). Pearson's correlation coefficient between bins used and packing density was computed using spreadsheet software, which applies the standard formula to all 60 pairs:

$$r = \frac{\Sigma(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\Sigma(x_i - \bar{x})^2 \Sigma(y_i - \bar{y})^2}}$$

The following hypotheses were used for correlation testing:

- a. Null Hypothesis H_0 : there is no linear correlation between bins used and packing density in the population, i.e. $\rho = 0$.
- b. Alternate Hypothesis H_1 : There is a linear correlation between bins used and packing density in the population, i.e. $\rho \neq 0$.

Using spreadsheet software, the sample correlation was $r = -0.1186$, with a two-tailed p-value of $p \approx 0.3692$. Since $p > 0.05$, there is no statistically significant linear relationship

between the number of bins used and packing density. In practical terms, the effect size is negligible, so knowing how many bins the algorithm opens explains none of the variation in packing density.

4. Theoretical comparison: Results are contrasted with the literature's Next Fit and FFD bounds. The comparison between these algorithms, described in the introduction, will allow for an evaluation of the effectiveness of the adapted algorithm.

Controls and assumptions:

1. All generated items can fit in a bin in at least one orientation.
2. No item overlap; all items/bins are perfect rectangular prisms.
3. Rotations are limited to right angles (six possible).
4. Physical constraints (fragility, weight) and non-rectangularity are excluded.

Limitations:

1. The single-shelf rule and the "first valid orientation" heuristic may underperform other algorithms, potentially yielding suboptimal outcomes.
2. Potential improvements include multi-shelf packing or optimizing item placement to minimize wasted space, particularly regarding item orientation.
3. Only geometric feasibility is modeled, not logistics complexities such as weight distribution, fragility, etc.

Ethical and other considerations:

1. Trial Python code is available upon request for verification. The pseudocode for the algorithm is provided in Appendix A.
2. The code used to generate the trial data was developed with assistance from an AI large language model (ChatGPT 5.0), mainly to fix errors or for corrections regarding technical details for the code. All code was reviewed and adapted by the author to ensure correctness and relevance to the research question. Data was graphed personally; however, PerplexityAI was used to create clearer and more visually appealing designs, adding extra notations such as mean for reference, as can be seen in Figures 5 and 6. All graphs were carefully checked to ensure mathematical accuracy.

Results:

All 60 trials of the adapted first-fit-decreasing algorithm produced a complete set of raw data: trial number, bins used, and total bin volume, packing density, and wasted space for each run. The full dataset is presented in Appendix B so that every calculation and conclusion in this section can be independently verified from the original values. After inspection of the raw computer-generated values, I recognize that item dimensions and volumes are widely distributed; therefore, a very varied data set was

used to calculate the efficiency of the algorithm at hand, simulating the complex real-life conditions of warehouse packing work in a realistic yet manageable way.

Table 2 summarises the central tendency and spread of both packing density and the number of bins used. The mean packing density of 39.38% and a standard deviation of 5.22% suggest a relatively consistent but systematically low packing efficiency. Moreover, the number of bins used has a mean of 8.68, a median of 9, and a standard deviation of 1.02; this suggests that most runs use around 9 bins, packing items so that approximately 39.38% of the available volume is used.

Table 2: summary statistics for bin usage and packing density over 60 randomly simulated trials.

Metric	Packing Density (%)	Bins used
Mean	39.38	8.68
Median	39.22	9
Std. Deviation	5.22	1.02
Minimum	26.83	6
Maximum	52.75	11
Range	25.92	5
IQR	5.81	1

Worked examples:

Consider the trial in Appendix B, where the total item volume is $96\,004\text{ cm}^3$, and 8 total bins are used.

As stated previously, each bin has volume $40 \times 30 \times 30 = 36\,000\text{ cm}^3$, therefore total bin volume is:

$$8 \times 36000 = 288000\text{ cm}^3$$

The packing density for this trial is:

$$\text{density} = \frac{96004}{288000} \approx 0.3333 = 33.33\%$$

To compute the overall mean packing density, the same calculation was performed for all 60 trials, and the 60 percentage values were then averaged using the following formula:

$$\bar{x} = \frac{1}{60} \sum_{i=1}^{60} x_i$$

$$\bar{x} = 39.38\%$$

The standard deviation of 5.22% was obtained from the 60 density values using the regular standard deviation formula:

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$$

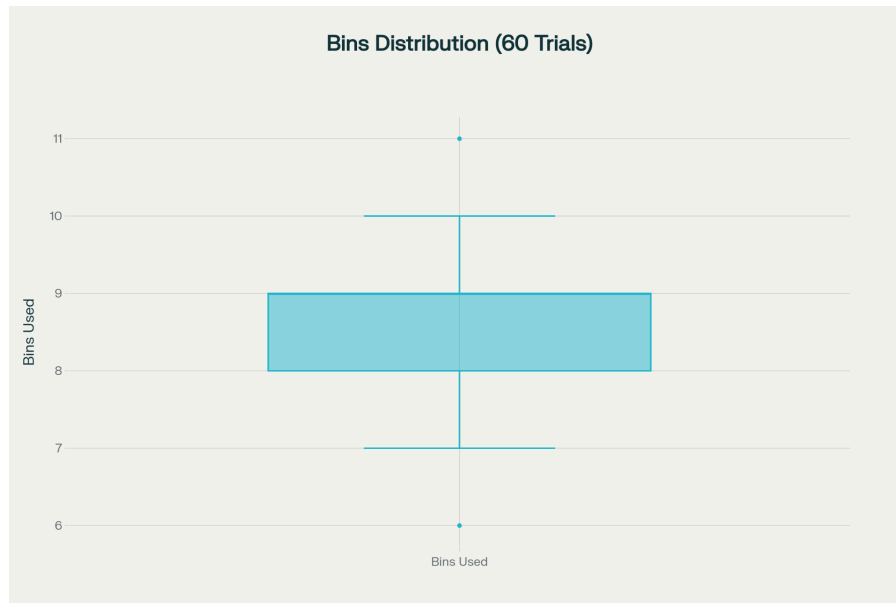
$$s = 5.22 \%$$

Analysis of bins used:

The box plot of bins used (Figure 4) shows that half of all trials lie between 8 and 9 bins, with a median of 9 and an interquartile range of only 1 bin. The minimum of 6 and maximum of 11 bins appear as isolated points, indicating best- and worst-case scenarios rather than typical performance, i.e., outliers in the set. The number of bins appeared to be roughly around 8-9 bins in most of the trials (IQR represents 50 percent of values), this is reflective of my warehouse work experience, where over multiple days of work, all items looked very different yet boxes of similar size would be used, with an intermediate size being the most common (or a common number of boxes was commonly observed).

From a mathematical perspective, the small standard deviation of the Bins used data set, and the narrow IQR of 1 (Figure 4) indicate low variability in the algorithm's output; changing the set of items usually results in a change of 1 bin in the bin count. However, the presence of several runs at 11 bins shows that there are particular combinations of item dimensions for which the first-fit-decreasing algorithm performs significantly worse, which is important when interpreting and comparing packing efficiency with other algorithms.

Figure 4: Box plot depicting the distribution of bins used over 60 trials, generated by the author with the help of PerplexityAI (only for graphic design).



Analysis of packing density:

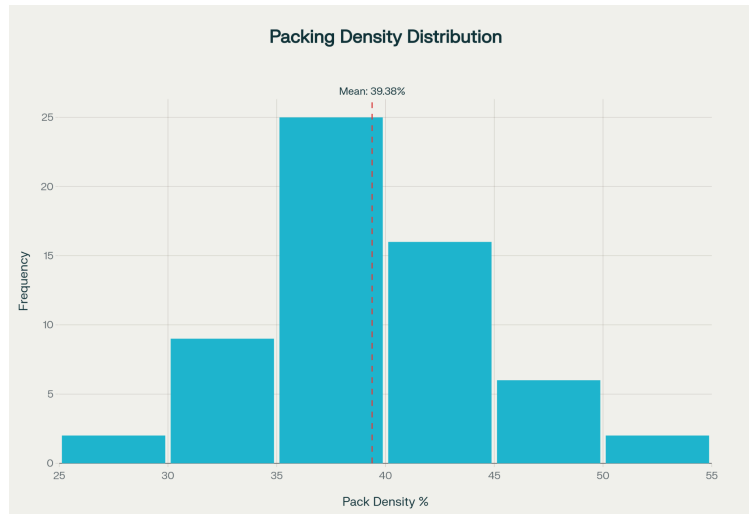
The histogram of packing density (Figure 5) shows that most trials fall between 35% and 45%, with a mean of 39.38% marked by the vertical dashed line. The shape is roughly symmetrical and unimodal (i.e., it has one peak), which also matches the Shapiro-Wilk test result (Table 3), which found no evidence to reject the claim that the distribution is normal ($W = 0.9903$, $p > 0.05$). The practical impact is that the mean and standard deviation are sensible but not particularly powerful summaries of performance, given the observed normality of the results. The Pearson's correlation coefficient of $r = -0.1186$ with $p = 0.369$, which is both statistically non-significant and negligible in effect size, indicating that changes in the number of bins used explain almost none of the variation in packing density. This reinforces the conclusion that geometric factors such as item shape and single-shelf constraint, rather than bin count alone, are the dominant drivers of efficiency in this heuristic.

Table 3: Statistical tests for packing density and bins used (60 trials).

Test/Metric	Variables Analysed	Value	Interpretation
Shapiro–Wilk W	Packing density (%)	0.9903 (W)	$p = 0.916 > 0.05$; data is approximately normally distributed.
Pearson r	Bins used vs. packing density (%)	-0.1186 (r)	$p = 0.369 > 0.05$. Very weak negative linear relationship, statistically insignificant at 5% significance level.

The histogram indicates the algorithm's efficiency is consistently low rather than highly variable across 60 trials, as evidenced by a relatively small standard deviation of 5.22% and a coefficient of variation of about 13.25%. In essence, the adapted first-fit decreasing algorithm occasionally discovers quite efficient arrangements; however, this does not compensate for inefficiencies, as the heuristic typically wastes 60% of the bin volume. This highlights the algorithm's flawed methodology, something also evident in the worked examples and, informally, while packing real boxes in a warehouse, with similar inefficiencies common.

Figure 5: Histogram depicting packing density (%) over 60 trials, generated by the author with the help of PerplexityAI (only for graphic design).

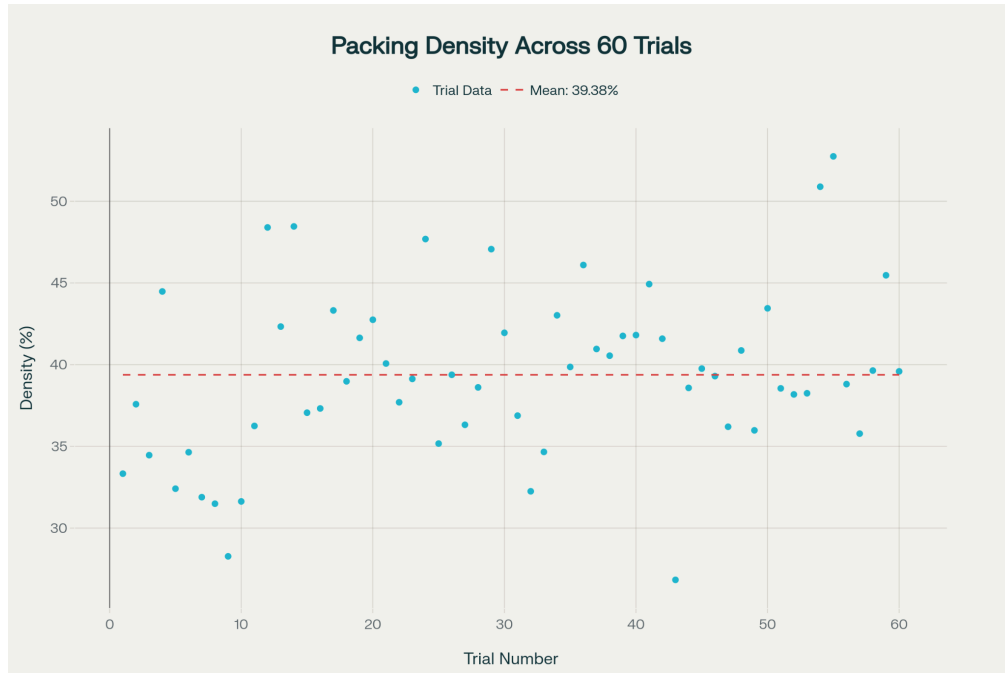


Behavior across trials:

As shown in Figure 6, packing density is plotted against trial number, with a horizontal dashed line representing the mean of 39.38%. The points fluctuate around this line without any visible upward or downward trend, which, while not conclusive, simply shows that the simulation did not generate obviously consistent “easier” or “harder” arrangements (batches of items). This is again supported by the Shapiro-Wilk test described earlier and by the approximately constant spread of points across the full range of trial numbers. This suggests that each trial can be treated independently, and therefore descriptive statistics such as mean and standard deviation can be used to evaluate the effectiveness of the first-fit-decreasing 3-dimensional adaptation (Ghasemi and Zahediasl, 2012).

When packing density is plotted against the number of bins used (not shown but analyzed numerically), the Pearson correlation coefficient (Table 3) is -0.1186, indicating a weak negative correlation between the variables. In other words, different from intuitive assumptions, a larger number of bins does not reliably or significantly change the packing density. Instead, efficiency is primarily driven by the convenience of the individual items and their relative fit to the bin's dimensions, as well as to the dimensions of other items already present in the bin. This both aligns with personal experience and highlights the main flaw of the simulated algorithm. Often, good packing can be ruined by large, inconveniently shaped items, while a human tackling the problem would likely re-shuffle previously placed items to fit better (or consider alternate combinations of items). This is, however, beyond the abilities of the heuristic.

Figure 6: Scatter plot of density (%) across individual trials, with horizontal line representing mean, generated by the author with the help of PerplexityAI (only for graphic design).



Comparison with Theoretical Bounds:

To assess the effectiveness of the algorithm and the empirical results, the simulated model must be compared with benchmarks from other algorithms in the literature on bin-packing heuristics. Previously 1 another model of the bin packing problem was discussed, called the next fit algorithm. Theory related to the next-fit model guarantees that the solution found will never use more than twice as many bins as the optimal solution, corresponding to an essentially worst-case packing density of 50% (Coffman et al. 189). The typical one-dimensional model of the FFD algorithm has a proven efficiency bound (lower) of approximately 81.8%.

The 3D-FFD heuristic used in this investigation underperformed both algorithms, achieving a mean packing density of 39.38% and a maximum of 52.75%. This essentially means that, in the best case, the algorithm used for this investigation performs similarly to the worst-case behavior of the Next fit algorithm. Although it must also be noted that 50% the density may not mean 50% packing density, as optimal may not be exactly 100%. The underperformance of the 3-dimensional FFD algorithm compared to the 1-dimensional version is not due to coding errors or chance, but rather to the geometric constraints imposed to mimic real-world conditions. Constraints include the dimension gate, the single-shelf rule, and the use of the first valid orientation, which together remove many of the packings that classical theory silently assumes are available.

To compare these results within practical 3D bin packing applications, it is logical to consider the Extreme Points heuristic, an industry-leading method for three-dimensional packing problems (Crainic et al.). This heuristic systematically evaluates candidate positions, i.e., specific corners or edges inside a box

where items can feasibly be placed, such as the deepest-bottom-left corner, and exploits these points to optimize spatial utilization through stacking and layering. Such heuristics consistently achieve packing densities of 80%-90% in realistic scenarios, significantly outperforming the adapted FFD algorithm tested in this investigation. The gap highlights how the algorithm explored here sacrifices some real-world packing complexity

Interpretation and Reflection:

Overall, statistical analysis of the results of the simulation show that the adapted FFD heuristic used in this investigation is reliably inconsistent: the normal, tightly-clustered distribution around 39-40% in Figure 5, the narrow box around 9 bins in Figure 4, and the lack of any trend in Figure 6, all point to a stable but poor level of performance rather than unpredictable behaviour. While this is disappointing, it also reflects my real-life packing frustrations with the difficulties and intricacies of the bin-packing problem. Consistently low packing density scores attest to the complexity of this mathematical problem.

Statistical analysis implies that the main bottleneck of this algorithmic approach discussed is long items, which quickly consume the 40 cm shelf length. This, combined with the single-shelf assumption, prevents the algorithm from exploiting the full 3-dimensional space.

Some possible improvements beyond the scope of this IA include adding rules to allow shelves to be added (a common addition to heuristic algorithms for the 3D version of the bin packing problem). Allowing layers to accumulate height could also be a helpful addition to this algorithm, possibly increasing its effectiveness.

Conclusion:

This investigation has demonstrated that simple one-dimensional heuristic algorithms, like the first-fit decreasing algorithm from the Edexcel A-level syllabus, can be meaningfully adapted to the three-dimensional bin packing problem by incorporating dimension and orientation constraints. Through statistical analysis of 60 randomized trials, the investigation revealed that the dimension gate and single-shelf assumptions significantly constrain bin usage efficiency. The adapted algorithm exhibits consistent but limited packing efficiency, with an average utilization of around 39%, reflecting real-world geometric challenges such as item length and non-stackability.

The exploration highlights the inherent complexity of three-dimensional bin packing, emphasizing that effective solutions require balancing mathematical rigor with practical feasibility. Future research could enhance the adapted heuristic by allowing multi-shelf packing, optimizing rotational placements, and incorporating item weight and fragility considerations. The complexity of finding a solution to this problem is highlighted by its importance in logistics, helping firms save millions while reducing fuel use and environmental waste.

Bibliography:

Ali, S., Li, Z., & Zhang, Y. (2022). On-line three-dimensional packing problems: A review of models and algorithms. *Computers & Industrial Engineering*, 171, 108495. Available at: <https://doi.org/10.1016/j.cie.2022.108495>

Coffman, Edward G., Jr., et al. "Approximation Algorithms for Bin Packing: A Survey." *Handbook of Approximation Algorithms and Metaheuristics*, edited by Teofilo F. Gonzalez, Chapman & Hall/CRC, 2007, pp. 187–202.

Crainic, Teodora, et al. "Extreme Point-Based Heuristics for Three-Dimensional Bin Packing." *INFORMS Journal on Computing*, vol. 20, no. 2, 2008, pp. 1–15.

Edexcel (2017). *Pearson Edexcel AS and A level Mathematics Decision Mathematics 1 Student Book*. Pearson Education Limited.

Ghasemi, Asghar, and Saleh Zahediasl. "Normality tests for statistical analysis: a guide for non-statisticians." *International Journal of Endocrinology and Metabolism*, vol. 10, no. 2 (2012): 486–489. doi:10.5812/ijem.3505

Johnson, David S., et al. "On the Approximation Algorithms for Bin Packing." *Journal of the Association for Computing Machinery*, vol. 28, no. 1, 1981, pp. 1–14.

Martello, Silvano, and Paolo Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, 1990.

Romero, A., Panadero, J., & Ruiz, R. (2023). Solving logistic-oriented bin packing problems through matheuristics. *arXiv preprint arXiv:2308.02787*. Available at: <https://arxiv.org/pdf/2308.02787>

Saska, C. (2020) 'Exploring the Bin Packing Problem', *The Startup* (Medium), 11 May. Available at: <https://medium.com/swlh/exploring-the-bin-packing-problem-f54a93ebdbe5> (Accessed: 25 September 2025)

Zuo, Y., Wu, J., & Lim, A. (2022). A constructive heuristic algorithm for 3D bin packing of irregular shaped items. *arXiv preprint arXiv:2206.15116*. Available at: <https://arxiv.org/abs/2206.15116>

Appendices:

Appendix A: Pseudocode for the investigation section, written by the author with the help of OpenAI's LLM ChatGPT 5.0:

```
FUNCTION AdaptedFFD(items, bin_dimensions)
  INPUT: items = list of (length, width, height) for each item
         bin_dimensions = (L, W, H) for each bin
  OUTPUT: bins_used, packing_density

  // Step 1: Calculate volume and sort
  FOR each item i:
    volume[i] = length[i] × width[i] × height[i]

  Sort items in DECREASING order of volume

  // Step 2: Initialize bins and tracking
  bins = empty list
  current_bin_index = 0
  bins[0] = new empty bin
  bins[0].remaining_length = L
  bins[0].remaining_width = W
  bins[0].remaining_height = H

  // Step 3: Try to place each item
  FOR each sorted item i:
    placed = FALSE

    // Try existing bins first
    FOR each existing bin b:
      // Try all 6 possible orientations
      FOR each orientation (l, w, h) of item i:

        // Check dimension gate
        IF ( $l \leq L$  AND  $w \leq W$  AND  $h \leq H$ ):

          // Check single-shelf rule: can it fit alongside?
          IF ( $l + (L - \text{bin}[b].\text{remaining\_length}) \leq L$  AND
               $w \leq \text{bin}[b].\text{remaining\_width}$  AND
               $h \leq \text{bin}[b].\text{remaining\_height}$ ):

            // Place item in this bin
            bin[b].remaining_length -= l
```

```

        bin[b].items.append(item i with orientation (l,w,h))
        placed = TRUE
        break

    IF placed == TRUE:
        break

    // If not placed in any existing bin, create new bin
    IF placed == FALSE:
        current_bin_index += 1
        bins[current_bin_index] = new empty bin
        // Place item in new bin
        (l, w, h) = first valid orientation of item i
        bins[current_bin_index].remaining_length = L - l
        bins[current_bin_index].items.append(item i)

    // Step 4: Calculate metrics
    total_bins_used = current_bin_index + 1
    total_item_volume = sum of all item volumes
    total_bin_volume = total_bins_used × (L × W × H)
    packing_density = total_item_volume / total_bin_volume
    wasted_space = 1 - packing_density

    RETURN (total_bins_used, packing_density, wasted_space)

```

Appendix B: Raw data collected from 60 trials of the simulated First-Fit-Decreasing heuristic algorithm, using random seed 42.

Trial	Bins Used	Total Item Volume (cm ³)	Total Bin Volume (cm ³)	Packing Density	Wasted Space	Packing Density (%)
1	8	96004	288000	0.3333	0.6667	33.33
2	9	121744	324000	0.3758	0.6242	37.58
3	9	111642	324000	0.3446	0.6554	34.46
4	10	160128	360000	0.4448	0.5552	44.48
5	9	104993	324000	0.3241	0.6759	32.41
6	8	99774	288000	0.3464	0.6536	34.64
7	9	103339	324000	0.3189	0.6811	31.89
8	9	102041	324000	0.3149	0.6851	31.49
9	10	101761	360000	0.2827	0.7173	28.27
10	8	91103	288000	0.3163	0.6837	31.63
11	8	104406	288000	0.3625	0.6375	36.25
12	9	156830	324000	0.484	0.516	48.4
13	9	137158	324000	0.4233	0.5767	42.33
14	9	157010	324000	0.4846	0.5154	48.46
15	9	120088	324000	0.3706	0.6294	37.06
16	9	120922	324000	0.3732	0.6268	37.32
17	7	109161	252000	0.4332	0.5668	43.32
18	10	140324	360000	0.3898	0.6102	38.98
19	9	134898	324000	0.4164	0.5836	41.64
20	8	123128	288000	0.4275	0.5725	42.75
21	8	115406	288000	0.4007	0.5993	40.07
22	8	108581	288000	0.377	0.623	37.7
23	8	112698	288000	0.3913	0.6087	39.13
24	8	137352	288000	0.4769	0.5231	47.69
25	10	126602	360000	0.3517	0.6483	35.17
26	9	127597	324000	0.3938	0.6062	39.38
27	10	130755	360000	0.3632	0.6368	36.32
28	9	125095	324000	0.3861	0.6139	38.61
29	9	152504	324000	0.4707	0.5293	47.07
30	9	135909	324000	0.4195	0.5805	41.95

31	10	132753	360000	0.3688	0.6312	36.88
32	6	69653	216000	0.3225	0.6775	32.25
33	9	112294	324000	0.3466	0.6534	34.66
34	9	139376	324000	0.4302	0.5698	43.02
35	11	157846	396000	0.3986	0.6014	39.86
36	7	116183	252000	0.461	0.539	46.1
37	6	88483	216000	0.4096	0.5904	40.96
38	9	131396	324000	0.4055	0.5945	40.55
39	9	135307	324000	0.4176	0.5824	41.76
40	9	135461	324000	0.4181	0.5819	41.81
41	9	145565	324000	0.4493	0.5507	44.93
42	9	134743	324000	0.4159	0.5841	41.59
43	11	106248	396000	0.2683	0.7317	26.83
44	8	111107	288000	0.3858	0.6142	38.58
45	9	128822	324000	0.3976	0.6024	39.76
46	8	113181	288000	0.393	0.607	39.3
47	9	117297	324000	0.362	0.638	36.2
48	8	117706	288000	0.4087	0.5913	40.87
49	8	103614	288000	0.3598	0.6402	35.98
50	8	125150	288000	0.4345	0.5655	43.45
51	9	124912	324000	0.3855	0.6145	38.55
52	8	109959	288000	0.3818	0.6182	38.18
53	8	110153	288000	0.3825	0.6175	38.25
54	9	164892	324000	0.5089	0.4911	50.89
55	9	170901	324000	0.5275	0.4725	52.75
56	9	125743	324000	0.3881	0.6119	38.81
57	10	128796	360000	0.3578	0.6422	35.78
58	8	114167	288000	0.3964	0.6036	39.64
59	9	147315	324000	0.4547	0.5453	45.47
60	6	85525	216000	0.3959	0.6041	39.59