Jacob Wilkens

CSC 490

Professor Blade

28 April 2022

Bin Packing

Computer scientists employ algorithms whenever a problem arises, and these algorithms can either be simple or complex. For this project, a complex algorithm was needed. For example, consider a situation where a packer has only one bin type and needs to pack multiple items in these bins while minimizing how many containers are used for packing. The problem was defined as finding the minimum number of bins to pack variable-sized items. This paper will go through the process I took to try and find a solution to the bin packing problem. I will note the research I employed, the algorithm I tried to employ, my struggles, and the project and class review.

I researched multiple areas to try and understand this problem before attacking it. First, I needed to understand the difference between polynomial and nonpolynomial time relating to algorithms. Techopedia notes, “If a problem is in non-deterministic polynomial time, the non-deterministic Turing machine can first guess at the solution, and then run a verifiable algorithm that will confirm whether or not that guess was correct.” This means specific problems do not have a straightforward algorithm to solve in a reasonable amount of time. Understanding this problem was nonpolynomial, a heuristic would need to be employed to find the number of bins. An NP problem cannot be solved in polynomial time, and it needs to be verified if the solution is correct.

The next area was known heuristics for bin packing. Bin packing is a common problem that other computer scientists have faced, so there are known heuristics for this problem. An article “Bin Packing - The Definitive Guide for 2021” that I found by Derek Mwiti outlined heuristics for two-dimensional bin packing. The two-dimensional heuristics would be able to apply to three-dimensional. Examples of the heuristics listed were next-fit, next-k-fit, first-fit, best-fit, worst-fit, and first-fit-decreasing. Two heuristics that I wanted to explore further were best-fit and first-fit decreasing. Best-fit can be described as placing an item in the bin with the maximum load. First-fit-decreasing sorts the item by descending order and placing the item in the first available bin, and if none exists, the item gets put into a new container.

The last part of research I did was to look at another implementation of bin packing to understand the different aspects involved. A paper that I could follow was written by Erick Dube and Leon Kanavathy which was “Optimizing Three-Dimensional Bin Packing Through Simulation”. An area that derailed my project was referenced in this paper which was the three dimensions of the x, y, and z axis. Dube and Kanavathy highlighted this in their paper by saying, “As previously explained each Item has six rotation types. Consider an item: The six rotation types can be obtained by rotating about the x, y and/or z axis as shown in Table 1: The bins are packed one at a time and the algorithms use a series of pivot points at which to pack the item.” I spent a lot of time trying to understand this which was a struggle.

Visualization is a skill that I am improving upon, and I need to visualize and see the different dimensions and orientations of an object. I looked up some code to use matplotlib. Below are screenshots of all the orientations for an object with dimensions of 5x4x3. Understanding that an object can be rotated, turned, and flipped with all of those orientations complicated this problem. Another aspect I did not understand how to tackle was representing the placement of items in a bin. I understood that each item would have x, y, and z coordinates. Until this problem, I have not worked with the x, y, and z planes. I spent all of the time trying to understand the three-dimensional aspects of representing the position of the item in the box along with the different orientations.

Chart

Description automatically generatedChart

Description automatically generatedChart

Description automatically generated

Chart

Description automatically generatedChart

Description automatically generatedChart

Description automatically generated

On top of the orientations and coordinates of the items in the box, another element that I could not tackle was checking if the item had a valid placement inside the box and a valid placement for existing items in the box. Without a complete understanding of the coordinates and positions of the items inside the bin, no logic would be written for proper placement. Additionally, someone packing the items in the bins would need to know the positions to put each box.

With this being a busy semester, I could not spend enough time on the project. I tried to understand the problem domain and aspects of the problem before doing any programming. As I mentioned, I got stuck on the factors relating to the three dimensions. On my GitHub repo for this project, there are the base classes that I would have employed to implement this algorithm. I wanted to try both first-fit-decreasing and best-fit, but the factors above prevented me from implementing the bin packing problem.

An area that I explored while trying to visualize the three-dimensional aspects was to see if a heuristic could be employed to pack only one type of item in a bin with 12x9x6 dimensions. With a 3x3x3 item, 24 of those items can fully fit into a bin. Those items have the best dimensions to maximize space inside bins. Below is a screenshot of half a bin filled with 12 3x3x3 items.

Chart, surface chart

Description automatically generated

The last part I would like to address in this paper is the class and project. I think the bin packing problem is a good project topic to assign. For seniors, the project was assigned to late in the semester to give enough time to employ a solution. I spent most of the time on the project stumbling through understanding the three-dimensional aspects. Overall I did enjoy most of the content we covered in class. I particularly liked the introduction and review of hash tables and binary trees. The most beneficial topic in this class was the time and space complexity. For some of my job interviews, questions about time complexity were asked. Without taking this class I would have been unaware of what time complexity was. Some suggestions I have for this class would be to assign the project at the beginning of the semester to give students more time. As a senior, I did not have enough time at the end of the semester with job interviews and work from other classes to implement a solution.

Works Cited

Dube, Erick & Kanavathy, Leon & K@i, Leon & Za, Owave. (2006). Optimizing Three-Dimensional Bin Packing Through Simulation.

“3D Voxel / Volumetric Plot¶.” *3D Voxel / Volumetric Plot - Matplotlib 3.1.0 Documentation*, https://matplotlib.org/3.1.0/gallery/mplot3d/voxels.html.

Mwiti, Derrick. “Bin Packing - The Definitive Guide for 2021.” *Cnvrg*, 16 Feb. 2021, https://cnvrg.io/bin-packing/.

Techopedia. “What Is Non-Deterministic Polynomial Time (NP)? - Definition from Techopedia.” *Techopedia.com*, Techopedia, 29 Aug. 2019, https://www.techopedia.com/definition/21028/non-deterministic-polynomial-time-np#:~:text=Polynomial%20time%20emerged%20as%20a,not%20that%20guess%20was%20correct.