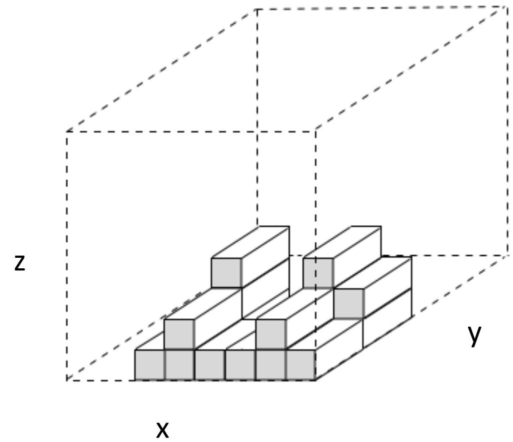
MOOP PROJECT

05Oct17

Imagine a port with a container stack consisting of 400 containers arranged in a 10x10x5 area (see diagram for a visual representation). In a typical day, the crane operating on this stack can move 100 containers – a single move is defined as either moving a container within the stack to another coordinate or removing a container from the stack and delivering it to the port's customers (via a 3rd party carrier, who has a truck at the port). Each container belongs to a specific carrier, and each carrier has their own set of delivery goals related to the number, type and importance of the containers they own. Rules concerning the container stack have been included, see below.



Rules for the container stack:

- Each container resides within a specific x,y,z coordinate
- The maximum x coordinate that containers can be placed in is 10, the maximum y coordinate is 10 and the maximum z coordinate is 5
- In any given column, the container with the highest z coordinate is the only container that can be moved
- Containers can't 'float' in the air, e.g. a container can only be placed on (5,2,5) if a container is already placed on (5,2,4)

Every container in the stack has been assigned a *contract price* between 1 and 50, reflecting the amount of money that the customer has contracted the port for delivery of this container.

Assignment #1

Your first task is to design and write an algorithm that achieves the goal of the port. The port has a single objective function; maximize the total contract price of containers delivered in a day. This algorithm will produce a list of containers that will be delivered for the day (within the '100 moves' constraint). The algorithm also needs to produce a report for each customer, that shows the list of containers that they will have delivered today, along with a value next to each container that represents the z-coordinate (i.e. the approximate cost of a move). The report should also show the list of all other containers owned by that customer, and their z-coordinate value.

Assignment #2

The next step will be to consider a single customer's objective function.

- maximize the total business profit

Each container has a business value in the range 1-20.
Each container has a contract price in the range 1-20.
The business profit = business value – contract price

In order to achieve their goals, the customer can 'promote' a container into the 'included' list provided by the port by expending the number of z-coordinate values recorded against that container, but the customer must also reduce the total number of z-coordinate values in the 'included' list by the same number.

The total contract-price from the included list cannot be reduced below its initial value.
The total z-coordinate value cannot be reduced below its initial value.

E.g.

	Name	Z-coordinate	Contract-price	Business value
Included	Container 1	1	10	14
	Container 2	3	12	14
	Container 3	2	14	14
	SUM	6	36	42
Excluded	Container 4	5	9	18
	Container 5	1	12	11
	Container 6	2	8	19

If the customer were to move Container4 into the included list, it would increase the total z-coordinate value to 11, so Container2 and Container3 would need to be removed from the included list. The total contract price would now be 19, so this would not be an allowed move! Therefore, the Customer would need to increase the contract-price for Container4 to be 17. However, the total business value would now be 46, and the profit would be 10. A better choice would be to swap Container6 with Container3, and pay an additional contract-price of 6, but increase the business value to 47, profit would be 11. Both are an improvement over the current profit of 6.

It is important to note that a customer can only see the values assigned to their containers by the port and the list of containers that will be delivered. They cannot change (or see) values assigned to other customers' containers. The customer starts with full data on their containers, along with the customer report on their containers from the port algorithm in Assignment #1.

Your second assignment is therefore to design and write an algorithm that achieves the goals of a single customer (Carrefour). The algorithm should provide a revised picklist with the sequence of Containers changed, showing updated Contract-Prices, and an improved profit while the sum of z-coordinates and sum of contract-price cannot go below the original values.

Note that you will want to reorder the entire list, not just those in the included section, as this will have a part to play in the third assignment.

Assignment #3

The final step is to take multiple participants into consideration – two customers (Carrefour and Metro), and the port. You can start by duplicating the Carrefour algorithm for Metro, and checking that this performs successfully. Then you will aim to run the two algorithms side by side.

In this third slice of their day, the port wants to maximize throughput, while still maximizing contract-price. Your final assignment is to combine the ports two objectives with the two customers' objective functions into a Multiple Objective Optimization Problem (MOOP) system that provides a Pareto set of solutions *for the port* with regards to container picklist. In this scenario, the port has an additional 50 moves to allocate. It takes the output from the two customer reports above and models the Pareto Front to show the set of moves that maximizes the total contract-price achieved by the port and the number of containers delivered, by splitting the allocation of 50 extra moves across the two customers currently excluded list.

Note: the key challenge of writing an algorithm to solve a MOOP is that there won't be a single optimal solution; a set of Pareto optimal results (known as a Pareto front) will be produced, from which a solution must be chosen. For additional information regarding MOOPs and Pareto fronts, please see Wikipedia here and ask your tutor for additional academic materials (!)