IMPLEMENTATION

The entire problem is solved in three steps,

(i) Developing a port agent
(ii) Developing a client agent
(iii) Applying chosen algorithm to find Pareto front

There are four rules concerning 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 column, the container with the highest z coordinate is the only container that can be moved.
• Containers cannot float in air, for example a container can only be placed on (5,2,5) if and only if container is already placed on (5,2,4).

Two terms promoted and excluded are frequently used in the upcoming sentences. I will explain these words for better understanding. Promoting a container means including a container to the container picklist and excluding means either removing a container from container picklist or not considering a container while making decision among best possible containers available for promotion.

Port agent:

The problem of the port agent is modelled as a linear integer problem (LIP). A variable is assigned to each container, which can be either 0 or 1 and constrained the sum of the variables to be less than 100. The variables of the containers that have containers on top is constrained to be less than the variable of the container on it's top, which in practice means that the container can't be removed if there is a container on it's top.

The objective can also be defined linearly by multiplying each variable by its contract value and adding them all.

Since the problem contains integers, it is solved by using branch and cut method, one of the LIP solving methods. GNU linear programming kit (GLPK) solver is used in order to achieve this. GLPK package is a widely used software package for solving large scale linear programming and mixed integer programming problems.

Client agent:

The best solution for client agent is found procedurally in the following way,

1. Find the best available promotion to make.
2. Make promoted container and exchanged container unavailable.
3. Repeat until the best available promotion is worse than not promoting anything.

By doing so, the problem is reduced to finding the best available promotion at each step. This can be achieved by using following algorithm.

1. For each excluded container we find the best set of available included containers to be removed in place of it.
2. We compute the profit increase in making each promotion.
3. The promotion with the higher profit increase is selected as the best.

By doing so, the problem is further reduced to finding the best set of available included containers to be removed for a given excluded container at each step. This can be achieved in the following method.

1. If the z-value of the container to be promoted is greater than the current selected set of the containers to remove, then find the available included container that would produce the best profit increase, by iterating over all of them.
2. If there is no container that keeps the profit increase positive then mark this container as not to be promoted.
3. Repeat until the z-value of the container to be promoted is lower or equal than the current selected set of containers to remove (or the container is marked as not to be promoted).
4. If marked as not to be promoted return 0, else return the current selected set of containers to remove.

Pareto front:

The final Pareto front solutions are found by NSGA-II (genetic algorithm).

The problem is modelled as finding for each x, y coordinates how many containers to be deliverer to their owner, since the order of delivery isn't taken into account for this problem, except by having to deliver first the ones on top, but having decided how many to deliver from a coordinate the order of delivery is obvious (from top to bottom).

This gives us 100 variables (10x10) that we need to find. Each value is an integer and the range go from 0 to the number of containers that are at its respective x, y coordinate.

For a genetic algorithm to work with this system, we define a function that computes the value of each objective given a list of values for these variables. The objective values that this function returns are: The profit improvement for each of the customers, the sum of the contract prices and the sum of containers delivered. Also, the constraint for this problem is that the number of moves done have to be less than 150.

Having defined the variables, the objectives and the constraint we run the genetic algorithm NSGA-II for 1000000 iterations to find the Pareto front.

Having the Pareto front defined we need to find one of those solutions for the problem. This can be found by calculating an average of all the Pareto front solutions and find the solution closest to the average.

PICTORIAL REPRESENTATION OF THE IMPLEMENTATION: