A SEMANTIC QUESTION

ANSWERING FRAMEWORK

***Mid-Semester Report of***

***6th Semester Mini Project***

*for the degree Of*

**BachElOr OF TECHNOLOGY**

*In*

**INFORMATION TECHNOLOGY**



*BY*

Anupam Jaiswal (IIT2014038)

Mohneesh Khaneja (IIT2014041)

Yash Jain (IIT2014043)

Harsh Shah (IIT2014071)

Anujraaj Goyal (IIM2014002)

*Under the supervision of*

Prof. U. S. Tiwari

**INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, ALLAHABAD**

March 2017

**CANDIDATES’ DECLARATION**

We hereby declare that the work presented in this project report entitled “**A SEMANTIC QUESTION ANSWERING FRAMEWORK**”, submitted mid-semester report of 6th Semester report of B.Tech. (IT) at Indian Institute of Information Technology, Allahabad, is an authenticated record of our original work carried out from January to March 2017 under the guidance of **Prof. U. S. Tiwari**. Due acknowledgements have been made in the text to all other material used. The project was done in full compliance with the requirements and constraints of the prescribed curriculum.

Place: Allahabad

Date:

Anupam Jaiswal (IIT2014038)

Mohneesh Khaneja (IIT2014041)

Yash Jain (IIT2014043)

Harsh Shah (IIT2014071)

Anujraaj Goyal (IIM2014002)

**CERTIFICATE**

This is to certify that the above statement made by candidates is correct to the best of my knowledge.

Date: Prof. U. S. Tiwari

Place: Allahabad IIIT-Allahabad

Table of Contents

1.Introduction…………………………………………………………………………………….1

1.1 Purpose & Motivation……………………………………………………………………..1

1.2 Application………………………………………………………………………………...1

1.3 Problem Solved……………………………………………………………………………1

2. Problem Description...…………………………………………………………………………2

2.1 Problem Statement………………………………………………………………................2

2.2 Assumptions……………………………………………………………………………….2

2.3 Challenges…………………………………………………………………………………………..2

3. Literature Survey…………………………………………………………………………........4

3.1 Constraints……….………………………………………………………………………..4

3.2 Comparative Table………………………………………………………………………..5

3.3 Base Paper………………………………………………………………………………...6

4. Algorithm Design……………………………………………………………………………...7

4.1 Algorithm Flow chart……………………………………………………………………..7

4.2 Mathematical Modeling…………………………………………………………………..8

5. Languages and Tools used………………………………………………………………….10

6. Work Flow Details………………………………………………………………………….11

6.1 Activity Chart Diagram…………………………………………………………………………11

6.2 Work done till mid-sem…………………………………………………………………………..13

6.3 Work to be done after mid sem…………………………………………………………………..13

7. Visualization of container………………………………………………………………….14

8. Scope……………………………………………………………………………………...…19

9. References………………………………………………………………………………….…21

1. **INTRODUCTION**

1.1 Purpose and Motivation:

The purpose of our project is to solve the Container Loading Problem (CLP) with Multi-drop constraint. Several industries produce goods and those need to be transported to different places on a daily basis. To minimize the transportation cost the industry needs to pack their produced goods in a very efficient manner. This was the motivation behind our project. Main motive is to maximize the Volume used with minimal space wastage in the Container taking care of multi-drop.

1.2 Application:

These are the practical applications of the CLP:

* In transportation facilities of heavy industries.
* General goods transportation of E-commerce companies.
* Cargo Shipping.

1.3 Problems Solved:

These are the problems solved by CLP:

* Economical: Efficient Packing results reduction in Transportation Cost.
* Environmental: Lesser the number of trucks, lesser will be the air & sound pollution.
* Accidents & Traffic Jam: Overloaded Trucks causes Accidents.

**2. PROBLEM DESCRIPTION**

2.1 Problem Statement

We have a container of dimension (*L, B, H*) that has to be filled with a set of *n* boxes, *Bi,*

*i* = 1, 2,.*, . . . n*. Each box has dimensions (l*i,* b*i,* h*i*) in cm and a destination or drop point number *di .*

* Maximize volume utilization of the given container to be loaded with multi drop constraint & Minimize number of leftover boxes.
* At a particular drop point, all the consignments to be dropped at that point must be visible.
* Minimum disturbance to other boxes during unloading at any drop point.
* Required orientation of the boxes must be maintained.
* Each consignment should be placed either on the floor of the container or on the top of other boxes.
* Only orthogonal packing would be considered. i.e., each consignment is placed parallel to the edges of the container.
* Consignments can be rotated spatially, i.e., each consignment has six possible orientations but some of these orientations may be not permitted. In our code, we have considered only two orientations (X-Y).

2.2 Assumptions

* Order of all the drop points are known.
* All the details about consignments and container, i.e. length, breadth, height are assumed to be integers and known in beginning.
* No Articles can added in the middle of loading of the articles on the container.
* We have assumed the container as 3 Dimensional space and the inner-most left bottom corner as origin.

2.3 Challenges:

* Choosing between the best fit and best volume approach as which one will be more appropriate for packing.
* Generation of new spaces depending upon the position of box placed and the existing spaces adjacent to it.
* Merging of spaces was cumbersome due to handling of adjacent parallelepipeds.
* Implementing visibility criteria.
* Packing a box in to multiple disjoint spaces

**3. LITERATURE SURVEY**

3.1 Constraints:

Some Practical Constraints that need to be taken care of while loading the Container:

1. Multi drop Constraint
2. Container related constraints
   1. Weight limits
   2. Weight distribution
   3. Centre of gravity
3. Item related constraints
   1. Loading priorities
   2. Orientation constraints
   3. Stacking and load bearing constraints
4. Cargo related constraints
   1. Complete-shipment constraints
   2. Allocation constraints
5. Positioning constraints
   1. Visibility & Reachability
6. Load related constraints
   1. Stability constraints
   2. Complexity constraints

3.2 Comparative Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.N.** | **Title/Year** | **Conf./Journal** | **Constraints** | **Approach** | **Assumptions** | **Dataset** |
| **1.** | **Constraints in container loading – A state-of-the-art**  **review - 2012 [1]** | **Journal** | **1,2,3,4,5,6** | **Literature Research** | **NA** | **NA** |
| **2.** | Hybrid Genetic Algorithm for container packing in 3D [2] | Journal | 2,3,5 | Hybrid Genetic Algorithm | All Boxes have same sizes | 200 Benchmark Data Set |
| **3.** | An Efficient Metaheuristic for  Multi-Dimensional MultiContainer Packing – 2011 [3] | Conference | 2,4,6 | Branch and Bound  & Shelf | NA | Or.deis.unibo.it |
| **4.** | Mathematical Models for  Multi Container Loading Problems [4] | Conference | 2a,2b,2c,3b | Integer Linear Programming | Known Layer Composition | ORTEC Company |
| **5.** | **A Grasp Algorithm for the container loading problem with multidrop constraints 2015 [5]** | **Journal** | **1,2,3,4,5,6a,** | **Greedy randomized adaptive search procedure** | **Origin is bottom left front corner** | **ESICUP Company** |
| **6.** | A New Grouping Genetic  Algorithm for the Multiple Knapsack Problem-2008 [6] | Conference | 2,3,5 | Similar kind of knapsack | NA | Martello & Toth Knapsack Book |
| **7.** | **Container loading with multidrop constraints-2009 [7]** | **Journal** | **1,2** | **Tree Search Framework** | **All Destinations and**  **Boxes to be delivered already known** | **OR Library** |
| **8.** | **MIP-based approaches for the container loading problem-2011 [8]** | **Journal** | **1,2** | **Mixed Integer Linear Programming** | **All Destinations and**  **Boxes to be delivered already known** | **Danish**  **Distribution Company** |
| **9.** | Parallelization of the Multi-  Objective Container Loading Problem – 2012 [9] | Conference | 2a,2b | Evolutionary  Algorithm  (MOEA) | Stability not considered(filler material used) | METCO |
| **10.** | **Integral Optimization of CLP with Multi drop Constraints-2011 [10]** | **Journal** | **1,2a,2b,3b** | **MIP Heuristics Approach** | **All the boxes are cubical & drop points are already known** | **SMAA-O Ordinal Data** |
| **11.** | **Multi drop Container Loading Using Evolutionary Algorithm 2013 [11]** | **Conference** | **1,2,3b** | **Evolutionary Algorithm** | **Stacking and Load Bearing Not Considered** | **MOCL Company** |
| **12.** | **Heuristics for CLP with Multi drop -2013 [12]** | **Journal** | **1,2,4,5** | **Heuristics Algorithm** | **Weight Distribution not Considered** | **Diku / pisinger** |

**Table** 1**:** Comprises of all the paper details

**Table 1** gives us the overview of the literature available and research work done in the field of optimization problems dedicated to Container Loading.

3.3 Base Paper

Paper Number 5 ‘**A Grasp Algorithm for the container loading problem with multi drop constraints 2015’** is our base research paper.Taking ideas (the way of creating new spaces, merging, strategy of placing boxes [min-x,min-z, min-y]) from this paper, we attempt to target the problem using a Greedy Heuristic approach that has been explained in subsequent slides.

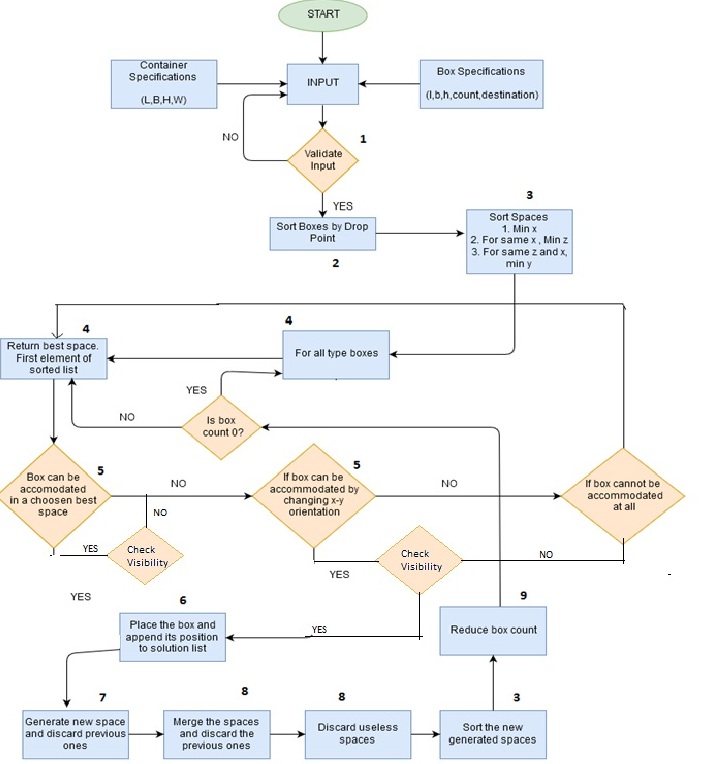
WHAT IS GRASP?

GRASP (Greedy Randomized Adaptive Search Procedure) is an iterative procedure combining a constructive phase and an improvement phase. In the constructive phase, a solution is built step by step, adding elements to a partial solution. The constructive phase is iterative, greedy, randomized, and adaptive.

In the constructive phase, a best space characterized by (min-x, max-z) is selected, and packing is done. Thereafter, the task involves creation of new spaces, merging of spaces & discarding unusable ones. Finally, it involves improvement phase where we remove k% of the solution and redo the algorithm. Adaptive function guides the greedy algorithm based on the element selected for the solution we are constructing.

**4. ALGORITHM DESIGN**

4.1 Flow Chart



10

10

Figure 1: Flow Chart of proposed algorithm

**4.2 MATHEMATICAL MODELLING**

This problem is an optimization problem, where our main objective is to maximize the volume occupied by the boxes. Mathematically, we can represent this by:-

**F**: <L, B, H, SB> → **R**

Where:-

L: Length of the container

B: Breadth of the container

H: Height of the container

SB:Set of Boxes (according to their different types)

1) SB = {(li, bi, hi, counti, dropi, remainingi) | i and n = no. of

boxes}

Where:-

li: Length of the box

bi: Breadth of the box

hi: Height of the box

subject to the constraint:-

2) SB’= SB  such that Bi = (li, bi, hi, counti, dropi, remainingi) and for i ≤ j,

(li\*bi \*hi\*counti) ≥ (lj \*bj\*hj\*countj) and desti ≥ destj

This mathematically defines a sorted list of boxes, sorted with respect to destination and for same destinations, according to volume. This is to be noted that list is sorted in descending order.

3) Ss = { ((li, bi, hi, xi, yi, zi)| i {1, 2,…., ||Ss||) } where Ss is the unordered set of spaces.

SS ‘ = (li,, bi, hi, xi, yi, zi) for i < j , xi < xj or (xi = xj , zi < zj) or (xi = xj, zi = zj, yi < yj)

where Ss is the ordered set of spaces. This mathematically defines a sorted list of spaces, sorted with respect to the innermost, uppermost & leftmost spaces kept in the list first.

B , Sbest = So where So is top most space in sorted Ss or S’S.

For all boxes, get the best space from the sorted list. It is the topmost element from the list.

5) Bspace allocated = So , (li\* bi\*hi)B (lo\* bo\*ho)**s** ,

(li lo, bi bo, hi ho), : Box kept normally

or (bi lo, li bo, hi ho) : Box kept after rotating spatially

First we try to accommodate the box normally, then after rotation about the z-axis.

6) Bpos = (xso,yso,zso, Q, ID) where Q is orientation, and ID identifies the type of box.

Solution = Solution **U** Bpos

Initialize the coordinates of the box. (The place where it is kept). Append to the solution list.

7) 3 new spaces created Stop, Sside, Sfront

3 new spaces are generated when a box is kept into a space. One on the top of the box, one in the front, and the last one on the side.

8) Ss = Ss U {Stop, Sside, Sfront}

Ss = Ss – {Si} where Si is previous space and non-usable spaces.

Include the new spaces generated into the list of spaces. Remove the old space and all other non-usable spaces, where a non-usable space is a space Si Ss such that, B SB , B can’t be accommodated within Si

9) Bremainng = Bremaining - 1

Reduce the count of that particular box. And recur for the rest.

10) Bspace allocated = Bi and && xj >= xi+li, Dropj > Dropi , yj <= yi , yj + wj >= yi + wi , zj + hj >= zi + hi

The efficiency of our packing strategy can be calculated as:-

Efficiency = …………. (1)

**Our Algorithm terminates when all the boxes are evaluated.**

**5. Language, Libraries, Tools and Dataset to be used**

**Languages Used:**

* Java
* HTML5
* CSS
* JavaScript
* PHP
* MYSQL

**Libraries Used:**

* JQuery
* Three.js

**Softwares / Tools Used:**

* Netbeans
* Xampp(Apache server + MYSQL server)

**Data Set used:**

The dataset has been taken from the research paper: Container loading with multi-drop constraints. Informatics and Mathematical Modelling, Technical University of Denmark [13]

**6**. **WORK FLOW DETAILS:**

6.1 Activity Chart Diagram

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S.No*.*** | **Task** | ***1- 15***  ***(*Aug*)*** | ***15-30***  ***(Aug)*** | ***1-5***  **(Sept)** | ***5-20***  **(Sept)** | ***Mid-Sem***  **Eval** | **1-10**  **(Oct)** | **10-30**  **(Oct)** | **1-15**  **(Nov)** | **15-25**  **(Nov)** | **25-30**  **(Nov)** |
| 1. | **Literature Survey** |  |  |  |  |  |  |  |  |  |  |
| 2. | **Algorithm Design** |  |  |  |  |  |  |  |  |  |  |
| 3. | **Preliminary Documentation** |  |  |  |  |  |  |  |  |  |  |
| 4. | **Algorithm Implementation (Phase 1)** |  |  |  |  |  |  |  |  |  |  |
| 5. | **Mid Sem Exams and Vacation** |  |  |  |  |  |  |  |  |  |  |
| 6. | **Algorithm Implementation (Phase 2)** |  |  |  |  |  |  |  |  |  |  |
| 7. | **GUI Designing** |  |  |  |  |  |  |  |  |  |  |
| 8. | **Modular Integration** |  |  |  |  |  |  |  |  |  |  |
| 9. | **Final Submission** |  |  |  |  |  |  |  |  |  |  |

**Table** 2**:** Activity Chart Diagram

D:\Firefox Downloads\Untitled Diagram.png

**Figure** 2 Work Flow Diagram

6.2 WORK DONE TILL MID SEM

1. Literature Survey
2. Algorithm Design
3. Preliminary Documentation
4. Algorithm Implementation (Phase 1):

The software to be built follows design paradigms as specified in the MVC (Model View Controller) architecture.

Designed a bare bone implementation of the entire project including various classes involved and their coupling in the business logic part of the MVC architecture.

Designed an algorithm taking into account multi drop, orientation and visibility constraints hence achieving an efficiency of 80% in the standard data set of the given in the following research paper:-

Container loading with multi-drop constraints. Informatics and Mathematical Modelling, Technical University of Denmark [13]

Output is in the form of coordinates and orientation of each box in the container.

6.3 WORK TO BE DONE AFTER MID SEM

1. Algorithm Implementation (Phase 2)

* We will be mainly focusing on Stacking constraints and will try to improve efficiency taking into account all of the constraints.
* We will try to figure out solution to deal with the problem of placing a box in multiple disjoint spaces generated previously.

1. GUI Designing
2. Modular Integration
3. The MVC pattern to be implemented will involve the following modules:-

Model is the business logic doing the backend computation in Java.

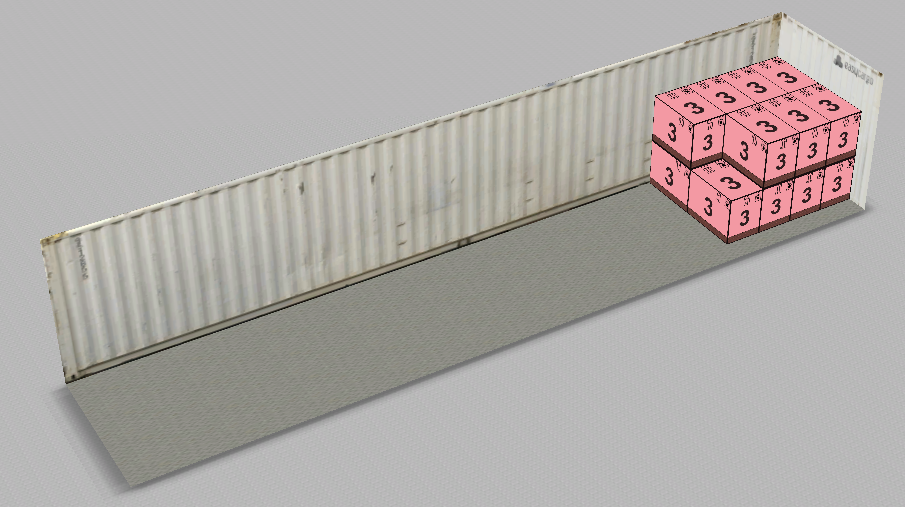
View module includes the GUI where the boxes placed are displayed.

Controller is the mediator which is responsible for delivering output from

Model to View module.

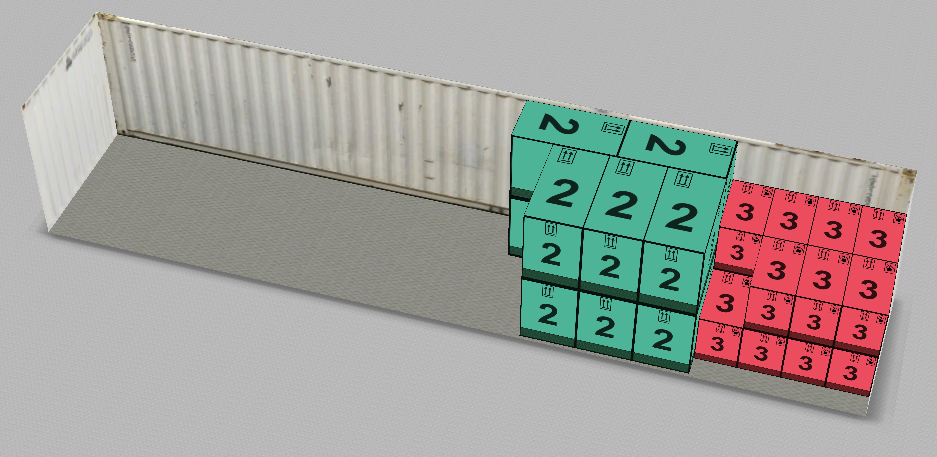
**7. Visualization of the way containers are to be loaded**

The figures 3,4,5,6,7,8 and 9 given below give an idea of how the output after implementation of GUI will look like.



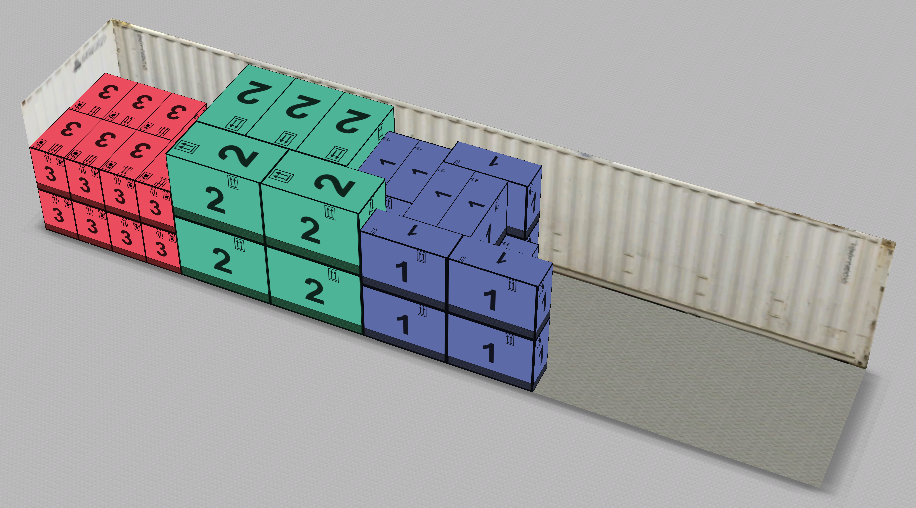
**Figure** 3: First Consignment Filled

**Figure 3:** As shown above, the boxes are filled in the spaces according to the criterion (min x, then min z, then min y).



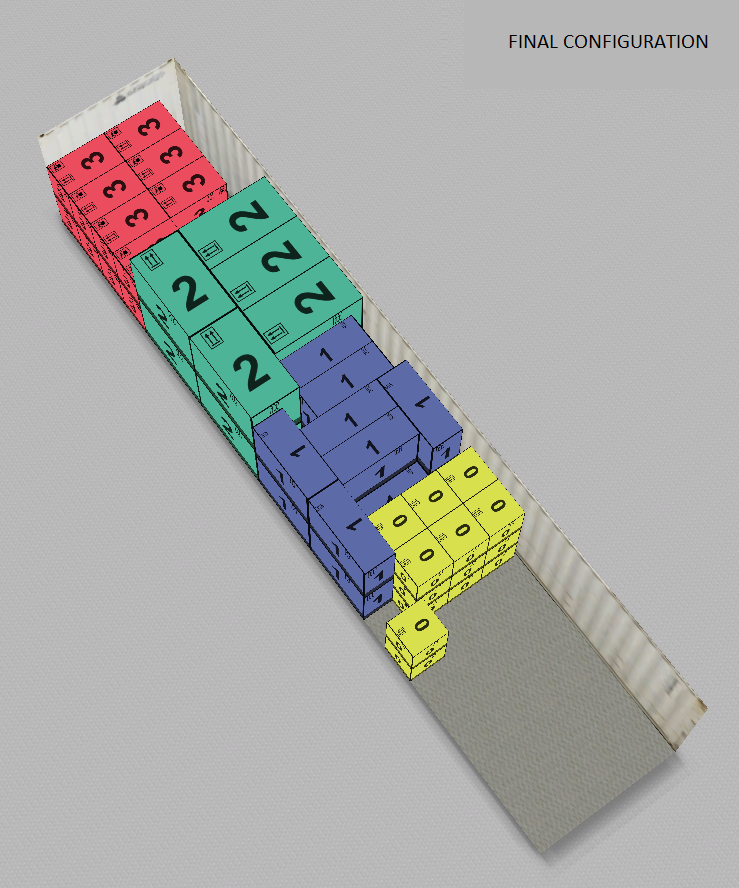
**Figure** 4: Second Consignments Filled

**Figure 4:** The consignments of the proceeding drop point are put in front of the previous ones, taking care that the visibility criteria is not violated. Also, we can see from the images that some boxes has been rotated spatially to accommodate better. We can also observe the fact that some spaces on the side of the red consignment is wasted as they are non-usable spaces. (Spaces where no box can be accommodated even after spatial rotation)



**Figure** 5**:** Third Consignments Filled

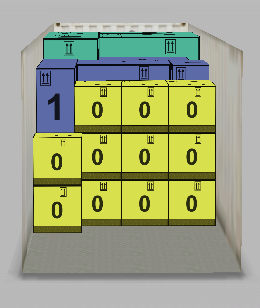
**Figure 5:** Violet consignments placed affront due to visibility constraint instead of placing on top of Red consignments. Due to visibility criteria, the violet consignments haven’t been put above the red ones, although it could fit there perfectly, because of visibility criteria.



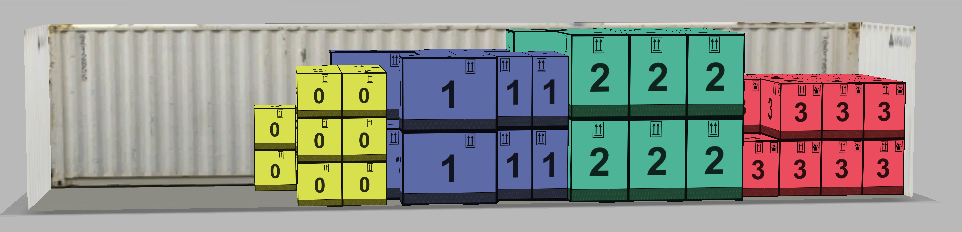
**Figure** 6: Fourth Consignment Filled

**Figure 6:** We can observe that some space got wasted for the yellow consignments as the front two boxes have a width greater than the width of the space left. Even after spatial rotation, it couldn’t fit.

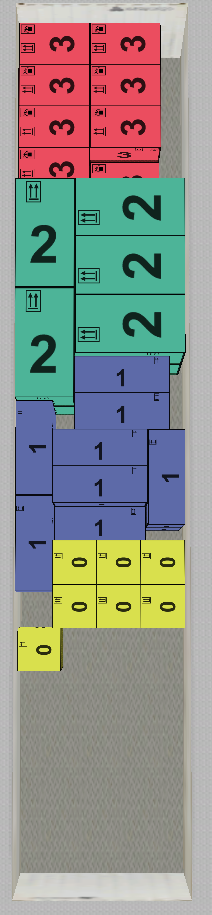
FRONT VIEW (through the door)



**Figure** 7: Front View of the Container

SIDE VIEW

**Figure** 8: Side View



**Figure** 9: Top View of the container

**­**TOP VIEW OF THE CONTAINER

**9. REFERENCES**

[1] Andreas Bortfeldt, Gerhard Wäscher. *“Constraints in container loading – A state-of-the-art review”*. 2013 European Journal of Operational Research

[2] Jin-Ling Lin, Bobbie F;oote\*, Simin Pulat',Chir-Ho Chang, andJohn **Y.** Cheung “*Hybrid Genetic Algorithm for Container Packing in three dimensions”*. 1996 University of Oklahoma

[3] Guido Perboli, Teodor Gabriel Crainic and Roberto Tadei. “*An Efficient Metaheuristic for Multi-Dimensional Multi-Container Packing* 2011” IEEE International Conference on Automation Science and Engineering Trieste, Italy

[4] M.T.Alonso z ,R.Alvarez-Valdes y ,M. Iori x,F.Parre~no z , J.M.Tamarit. “*Mathematical Models for Multi Container Loading Problems”*. University of Valencia

[5] D. Alvarez Mart´ınez1, R. Alvarez-Valdes2\*\* and F. Parre˜no3. “*A GRASP Algorithm for container loading problem with multi-drop constraints”.* 2015 Brazilian Operations Research Society

[6] Alex S Fukunaga. “*A New Grouping Genetic Algorithm for the Multiple Knapsack Problem”* 2008 IEEE Congress on Evolutionary Computation

[7] Søren Gram Christensen and David Magid Rousøe. *“Container loading with multi-drop constraints”* 2008 International transactions in operational research

[8] Leonardo Junqueira · Reinaldo Morabito Denise Sato Yamashita*. “MIP-based approaches for the container loading problem with multi-drop constraints. Springer Science Business Media”*, LLC 2011

[9] Jesica de Armas, Yanira Gonzalez, Gara Miranda, and Coromoto Le´on *“Parallelization of the Multi-Objective Container Loading Problem”*. 2012 IEEE World Congress on Computational Intelligence

[10] Leonardo Junqueira, M.T.Alonso “*Integral optimization of CLP with Multi-drop constraints*” 2011

[11] M.T.Alonso David Magid Rousøe “*Multi drop Container Loading Using Evolutionary Algorithm”* 2013

[12] Reinaldo Morabito Alex S Fukunaga. “*Heuristics for CLP with Multi drop”* 2013

[13] Søren Gram Christensen David Magid Rousøe “*Container loading with multi-drop constraints. Informatics and Mathematical Modelling”*, Technical University of Denmark. <http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/5342/pdf/imm5342.pdf>

Suggestions of Board Members