

Automated Cargo Planning Software

by Ali Khoramshahi

Background

Logistics is included of the delivery and store management of the products from the producer to the customer, it can be broken into external and internal logistics, nowadays all different kind of carriers, from ships and airplanes, there are always a special container for the carrier.

Managing the space that each carrier can carry has always been a problem that previously it would get solved by humans, but as the industry for computer and computing machines grows, lots of special algorithm to make the best cargo plan has been suggested.

The huge growing of producing different type of products make this matter impossible to handle this planning by human resource, so from almost a decade ago developers has started to suggest different solutions, but still this problem seems to be an important part of external logistic and each year more optimized solution are suggesting.

Cargo Loading Solution

For this software we are intend to suggest an optimized solution for automated cargo planning so the space inside of each carrier could be used with the most efficient usage, we are also looking for solutions to combine cargo planning with the path optimization so the software can decide the best type of the carrier and the path that it should take.

the main area of the customers for this software are contain from shipping company and the massing producers who spend huge amount of money each year for their shipping, the solution that this software can bring could reduce the amount of costs for the shipping and delivering as well as providing their resources for producing their goods.

The most important customers seems to be the massive producer, as they are already using such a software there is no need to convince the customers, and also the result of the optimization of the software would be much easier to compare with other software.

In the next level of customers the small and medium business can be found it would change their whole way of treating with such a problem therefore there some fundamental changes seems essential.

Because of the complexity of the equation and the processes of this software C++ has been suggested as the main programming language for this software so the power of developing a powerful user-interface can be combine with high speed of making a solution.

Feature and Functionality

The required feature for this software is to make it seems as a real world solution as much as we can so it seems essential to follow these limits, as they are in real life world:

- Each cargo weight.
- The maximum amount of weight that each carrier can take.
- The limitation of putting cargo two sides for each cargo.
- Easy accessibility for loading and unloading a carrier.
- The amount of weight that each cargo can take on top.
- Division of the whole cargoes weight to the whole surface of the carrier.
- The cost of shipping for each carrier.

for the future of this application it seems to be a good idea that this application can produce the result in such a way that can be seen in all sort of portable devices so it could help the loading procedure.

Scope and structure

This application is mainly contains from two part:

1. Core algorithm
2. User interface

each part should be implemented in such a way that the whole power of the software can be used to reach to the best solution.

Outcomes

As the outcomes of this software could be a plane the suggest the customer to go with the special kind of planning this software also should have the ability to get the feed back from the user and to get the user-define modification.

Project Plan

Design

This application will be design as the HND project for me, Ali Khoramshahi, so the main developer for the following application would be me and this will be evaluated by Prague College in the following year.

Tasting

Tasting procedure should start after each certain point of development to speed up the whole procedure of debugging and also because this procedure is of course a Research and Develop project this can prevent all the possible mistake in algorithm and miss-design situation. the procedure will start after finishing the first step of building the algorithm and in case of any possible bugs or mistakes it going to get fixed during the time of development.

Time Table

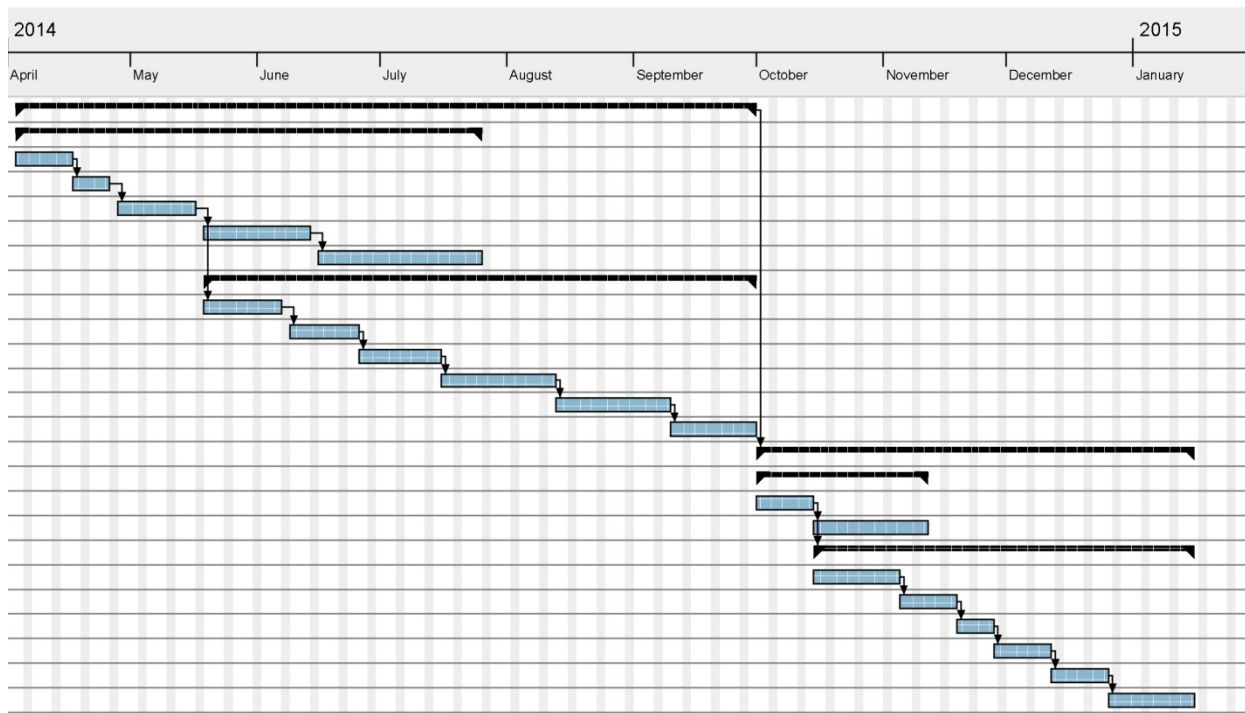
Project Stages

In the following chart different stage for this project will be explained:

1. **Automated Planing (a)**
 1. **Algorithm design (a.1)**
 1. Analyzing available algorithm (a.1_1)
 2. implementation of possible exiting library (a.1_2)
 3. Testing the implementations (a.1_3)
 4. Design an algorithm based on efficiency (a.1_4)
 5. Analyzing the designed algorithm (a.1_5)
 2. **Interface Implementation (a.2)**
 1. Turn the designed algorithm into C++ code (a.2_1)
 2. Providing data structure needed (a.2_2)
 3. Implementing simple interface to test the implemented code (a.2_3)
 4. Extend the interface (a.2_4)
 5. Testing (a.2_5)
 6. Optimization of the code (a.2_6)
2. **Path Optimizer (b)**
 1. **Algorithm design (b.1)**
 1. Design the algorithm accordingly (b.1_1)
 2. Analyzing the designed algorithm (b.1_2)
 2. **Interface Implementation (b.2)**
 1. Turn the designed algorithm into C++ code (b.2_1)
 2. Providing data structure needed (b.2_2)
 3. Implementing simple interface to test the implemented code (b.2_3)
 4. Extend the interface (b.2_4)
 5. Testing (b.2_5)
 6. Optimization of the code (b.2_6)

Time Table Charts

| Name | Begin date | End date |
|-------|------------|----------|
| a | 4/3/14 | 9/30/14 |
| a.1 | 4/3/14 | 7/25/14 |
| a.1_1 | 4/3/14 | 4/16/14 |
| a.1_2 | 4/17/14 | 4/25/14 |
| a.1_3 | 4/28/14 | 5/16/14 |
| a.1_4 | 5/19/14 | 6/13/14 |
| a.1_5 | 6/16/14 | 7/25/14 |
| a.2 | 5/19/14 | 9/30/14 |
| a.2_1 | 5/19/14 | 6/6/14 |
| a.2_2 | 6/9/14 | 6/25/14 |
| a.2_3 | 6/26/14 | 7/15/14 |
| a.2_4 | 7/16/14 | 8/12/14 |
| a.2_5 | 8/13/14 | 9/9/14 |
| a.2_6 | 9/10/14 | 9/30/14 |
| b | 10/1/14 | 1/15/15 |
| b.1 | 10/1/14 | 11/11/14 |
| b.1_1 | 10/1/14 | 10/14/14 |
| b.1_2 | 10/15/14 | 11/11/14 |
| b.2 | 10/15/14 | 1/15/15 |
| b.2_1 | 10/15/14 | 11/4/14 |
| b.2_2 | 11/5/14 | 11/18/14 |
| b.2_3 | 11/19/14 | 11/27/14 |
| b.2_4 | 11/28/14 | 12/11/14 |
| b.2_5 | 12/12/14 | 12/25/14 |
| b.2_6 | 12/26/14 | 1/15/15 |



Released Steps

1. Analyzing available algorithm (a.1_1)

There are mainly two suggested algorithms regarding to this problem:

1. First Fit Algorithm:

This is a Heuristic algorithm, based on the ordering of the packages arbitrarily and try to fit them into the first bin if possible. After putting each packet the rest of the space in the bin will be defined as a new bin appropriately.

This algorithm can be optimized by ordering the packets by their size, from the biggest to smallest, so after putting the first there for biggest packet into the bin we could have the smallest portion of space as our next bin therefore more space would be saved.

Implementation of the following algorithm in one dimensional environment seems to be quite straightforward but for using the following algorithm in three dimensional environments as expected requires some changes.

Although in the main algorithm there is no limitation as the weight for pallets and maximum payload for each pallet it could be implemented in such a way.

2. Tabu Search Solution

This algorithm works based on the computed penalties for each bin in the space, so with the calculation of the penalty for each package depends on how well it is fitted to the given space, and find the packet with the minimum penalty for every space it can get the decision of putting the package in that space.

As the main point of this algorithm is based on the given penalties to every individual bins and package to fit through them it seems to be easier to implement this algorithm based on real life limitation for shipment.

2. Implementation of possible existing library (a.1_2)

In the following procedure several libraries have been found but the main problem with most is the implementation on two dimensional environment also as the main language for the development has been chosen as C++ due to the mentioned reasons it can't be possible to use libraries implemented in python, mat-lab and so on.

Although the 3d-bin-pack which is a GIT-HUB based project is an example of three dimensional packing written in C.

The following code still has the lack of weight for each package and total payload for the vehicle and package that located underneath. This library is an implementation of heuristic algorithm and is not so efficient on its time level.

Resources

1. http://www.micsymposium.org/mics_2004/Sweep.pdf
2. <http://www.math.tu-dresden.de/~scheith/PREPRINTS/91-3BPP.pdf>
3. <https://github.com/wknechtel/3d-bin-pack>
4. <http://amsacta.unibo.it/2117/1/TSpack.pdf>
5. http://en.wikipedia.org/wiki/Bin_packing_problem