Sajini Anand P S[1]

[1]Affiliation not available

May 4, 2018

Abstract

The objective of the current document is to align the model to include the goals of the user stories in backlog for AMI-15 Project (Discussion with Ross and Team on 6 April 2018). The Mathematical model presented here has a framework of Linear and Mixed Integer Programming methods, and it relies on optimization strategies of Operational Research.

Assume that we have a list of Purchase Orders (each associated with a volume, weight, estimated date of delivery) and a list of Containers (each associated with a cost, capacity, weight constraint, Journey details- Starting Journey date, Duration in Transit and ending Journey Date). The following scenario defines the global optimization problem. Given n Purchase Orders (POs) and m available Containers, the objective is to create a Load Plan to allocate a set of containers to accommodate all the POs by optimizing the Containers such that all the POs reach the destination on or before their expected date of delivery. Optimization strategy requires minimizing the cost of the containers (which itself can be a variable based on available vessels and their timings), minimizing the number of containers needed for effective packing, and minimizing the unused volume of the containers, and selection of cheapest vessel for ocean freight for maximizing the profit.

Note: References for older versions of this document are available on the following links.

First draft (8 March 2018)

Second Draft (17 March 2018)

Third Draft (29 March 2018)

Introduction

The methodology used here is to split the larger problem into smaller sub problems and find solutions of the problems via combinatorial,LP,IP optimization algorithms. The three major elements of the global optimization problem - Decision variables, objective functions and constraints are listed in this section.

Maximum and minimum Fill Rates of Containers

Let V_i represents the allowed volume of i^{th} container, which is in general less than the actual container volume AV_i. Containers in general are not allowed to fill up to its the maximum allowed volume. Hence

$V_i = \alpha.AV_i$, and the multiplying constant α in range [0 ,1] represents the maximum fill–rate (the percentage of allowed volume) of a particular container. When $\alpha = 1$, $V_i = AV_i$ and 100% of the container volume is allowed. In general, $\alpha = 0.8$ indicating only 80% of the container volume as allowed volume, but α can vary depending upon the type and requirements of purchase orders. Similarly we can define a minimum fill rate for the container $\beta = 0.2$; as the container is not shipped unless it is filled above minimum fill rate $V_i = \beta.AV_i$.

Select a set of m different containers (each with a allowed volume V_i, cost per volume Q_i, and a weight constraint W_i where $i = 1, \dots m$) to hold given set of n purchase orders (each associated with a volume v_i, weight w_i and where $i = 1, \dots n$).

(1) allocate the set of containers to hold the total volume C_p of n purchase orders ($C_p = \sum_{i=1}^{n} v_i$) and

(2) find the optimal allocation matrix for the purchase orders ($s_{i,j}$ a binary variable which is equal to 1 if purchase order number i is placed in container j; otherwise it is equal to 0)

- **Decision variables** (those that describe our choices that are under our control)
  1. Container variables : volume V_i, profit P_i , cost Q_i, Weight constraint W_i where $i = 1, \dots m$
  2. Purchase order variables: volume v_i, weight w_i where $i = 1, \dots n$
- **Objective function**: describes a criterion that one wants to minimize (e.g., cost) or maximize (e.g., profit) in order to compare alternative solutions. Following objective functions (a) maximizes profit, (b) minimizes cost and (c) minimize unused volume after packing.
  1. maximize Profit $\Rightarrow$ maximize $\sum_{i=1}^{m} P_i x_i$
  2. minimize Cost $\Rightarrow$ minimize $\sum_{i=1}^{m} Q_i x_i$
  3. minimize Unused Volume in containers after packing $\Rightarrow$ minimize $\sum_{i=1}^{m} V_i x_i - \sum_{i=1}^{n} v_i x_i$
- **Constraints**: describe the limitations that restrict our choices for decision variables. Constraints can be the following form for the objective functions listed above.
  1. $\sum_{i=1}^{n} V_i x_i \geq C_p$

     $0 \leq x_i \leq d_j$ (since Containers are not unique and can be repeated as many as available)

     Total volume of the containers is greater than or equal to the total volume of purchase orders (The number of required containers in each category are available within a bounded limit of d_j, where d_j is the available containers in each category of the containers)
  2. $\sum_{i=1}^{n} v_i x_i \leq V_j$ for each $j = 1, 2, \dots m$

where v_i is the volume of purchase orders and V_j indicate volume of containers.

and

$x_i = \{0, 1\}$ (parcels are unique and cannot be repeated)

3. $\sum_{i=1}^{n} w_i x_i \leq W_j$ for each $j = 1, 2, \ldots m$

   where w_i is the weight of purchase orders and W_j is the weight constraint on containers.

   and

   $x_i = \{0, 1\}$ (parcels are unique and cannot be repeated)

Container Selection

Goal is to select an optimal number of containers to pack a given set of purchase orders. Both containers and purchase can be different in their sizes. The primary objective is to minimize cost of the container and the unused space of the container (thus maximizing the profit and hence minimizing the cost). It is assumed that the volume and weight constraint of each container types and the volume and weight of each purchase orders are given.

The parameters and the variables used in the model are given below:

Q_j	Cost associated with a particular type of container
V_j	Allowed volume associated with a particular type of container
n	Total number of purchase orders to be stacked in the containers
C_p	The total volume of purchase orders to fill in the containers
n	Total number of purchase orders to be stacked in the containers
C_p	The total volume of purchase orders to fill in the containers
d_j	An upper bound on the number of containers of a particular type j
m	Type of containers available

Algebraic formulation

Minimise the cost of the containers $\Rightarrow$ Minimize $\sum_{i=1}^{n} Q_i x_i$

Subject to the constraints

- $\sum_{i=1}^{n} V_i x_i = C_p$
- $0 \leq x_i \leq d_j$ (since Containers are not unique and can be repeated as many as available)

  Total volume of the containers is greater than or equal to the total volume of purchase orders (The number of required containers in each category are available within a bounded limit of d_j, where d_j is the available containers in each category of the containers)

Section 2. Optimized Allocation of purchase orders into selected Containers

Once the set of m containers are selected (each with a volume V_i and profit P_i, weight constraint W_i where $i = 1, \ldots m$) goal is to allocate all the purchase orders (of total volume C_p of n purchase orders) optimally into the selected containers. The primary objective is to minimize the unused space (thus maximizing the profit and hence minimizing the cost). It is assumed that the volume and weight constraint of each container types and the volume and weight of each purchase orders are given.

Then the parameters are and the variables used in the model are defined as below:

- n Total number of purchase orders to be stacked in the containers.
- C_p is total volume of purchase orders to fill in the containers
- d_j number of containers of a particular type j
- m Total number of type of containers available.
- v_j for $j = 1, \ldots n$ volume associated with each purchase order
- w_j for $j = 1, \ldots n$ weight associated with each purchase order
- W_j for $j = 1, \ldots m$ weight constraint of a particular type of container

then,

2.0 Algebraic formulation of the problem by using volume constraints of container

Purchase Order Volume $v_i \leq$ Container Volume V_j

maximize

$$\sum_{i=1}^{n} v_i x_i$$

Subject to the constraint

$$\sum_{i=1}^{n} v_i x_i \leq V_j$$

and

$x_i = \{0, 1\}$ (parcels are unique and cannot be repeated)

2.1 (Case1): Algebraic formulation of the problem by using volume constraints of container– Purchase Order Volume $v_i \geq$ Container Volume V_j (volumes of sub–units of the Purchase Order are known)

AMI(257) Fractional PO assignment into containers

If the volumes of the sub–units(carton volumes) of purchase order volume v_i is known, optimize the packing using the knapsack algorithm mentioned in 2.0 for sub–units/ cartons. Let u_i be the volume of the sub–units that makes the entire purchase order v_i. In this case individual u_j's will be less than the Container Volume

V_j, hence the problem is reduced to the scenario discussed in section 2.0. Algorithm for 2.0 can be used for the following algebraic formulation,

maximize

$$\sum_{i=1}^{n} u_i x_i$$

Subject to the constraint

$$\sum_{i=1}^{n} u_i x_i \leq V_j$$

and

$x_i = \{0, 1\}$ (parcels are unique and cannot be repeated)

2.1 (Case2) : Algebraic formulation of the problem by using volume constraints of container–Purchase Order Volume $v_i \geq$ Container Volume V_j, but volumes of sub–units of the purchase order are unknown

In this case we can do a fractional allocation of the purchase order to a number of available containers One can skip or refine the container allocation output in section 1.1, using teh following linear program formulation,

Minimize the Profit of containers $\Rightarrow$ Minimize $\sum_{i=1}^{n} P_i x_i$

Subject to the constraint

- $\sum_{i=1}^{n} V_i x_i \geq v_i$
- $0 \leq x_i \leq d_j$ (since Containers are not unique and can be repeated as many as available)

2.2 Algebraic formulation of the problem by using weight and volume constraints of container

AMI(178) Optimisation using weight and volume constraints

maximize $\sum_{i=1}^{n} v_i x_i$

Subject to the constraints

$$\sum_{i=1}^{n} w_i x_i \leq W_j \text{ for each } j = 1, 2, \ldots m$$

$$\sum_{i=1}^{n} v_i x_i \leq V_j \text{ for each } j = 1, 2, \ldots m$$

where Q_i is the volume of purchase orders and W_j, V_j indicate weight, volume constraint of containers.

and

$x_i = \{0, 1\}$ (parcels are unique and cannot be repeated)

2.3 Algebraic formulation of the problem by using weight constraints of container

maximize $\sum_{i=1}^{n} v_i x_i$

Subject to the constraints

$\sum_{i=1}^{n} w_i x_i \leq W_j$ for each $j = 1, 2, \ldots m$

where v_i is the volume of purchase orders and W_j indicate weight constraint of containers.

and

$x_i = \{0, 1\}$ (parcels are unique and cannot be repeated)

Selection of Algorithm for set of equations in 2.3

Integer Programming model– Zero-One Knapsack combinatorial optimization.

Algebraic formulation in 2.3 has a single constraint and it can be solved by a simple integer program. Zero-One Knapsack combinatorial optimization is ideal as it is a simple integer program that is easy to implement. Given a set of items, each with a volume and a weight, Knapsack determine the items to be filled in the container so that the total weight is less than or equal to a weight constraint of the container and the total volume is as large as possible.

Notes :

Data specific requirements- the reality about parcel weights and volume while filling in containers might give a clarity whether volume or weight has a priority among the constraints, hence will be essential to determine the type of model to be used

2.4 Algebraic formulation of the problem for Bin Allocation and Simultaneous filling of Containers

Algebraic formulation in 2.3 has volume of purchase orders (v_i) in both the maximization equation and the constraint. One alternative is to use a bin packing algorithm that can simultaneously assign all items to all selected containers. Since the containers have variable sizes (33,67,76,86 CBMs each), split the total volume of selected containers into a multiple of the smallest volume (33, 2*33, 2*33,2*33) with additional spare volume $\Delta = (0, 1, 10, 20)$ units each). Then allocate the parcels simultaneously into all bins of equal size V (V being the smallest container volume 33).

Given a set of bins S_1, S_2... with the same size V and a list of n items with sizes $a_1, \ldots, a_n$ to pack in the bins,

(i) find an integer number of bins β and

(ii) a β – partition $S_1 \cup \cdots \cup S_B$ of the set $a_1, \ldots, a_n$ such that,

$\sum_{i \in S_k} a_i \leq V$ for all $k = 1, \ldots, \beta$

minimize $B = \sum_{i=1}^{n} y_i$

subject to :

- $B \geq 1,$
- $\sum_{j=1}^{n} a_j x_{ij} \leq V y_i$

  $\forall i \in \{1, \ldots, n\}$
- $\sum_{i=1}^{n} x_{ij} = 1,$

  $\forall j \in \{1, \ldots, n\}$
- $y_i \in \{0, 1\},$

  $\forall i \in \{1, \ldots, n\}$ ($y_i = 1$ if i'th Bin is used.)
- $x_{ij} \in \{0, 1\},$

  $\forall i \in \{1, \ldots, n\} \, \forall j \in \{1, \ldots, n\}$ ($x_{ij} = 1$ if item j is put into Bin i.)

Selection of Algorithm for set of equations in 2.4

A solution is optimal if the model has a minimal β. The algebraic formulation in section 2.4 is in standard format for Integer Linear Programming. Use any standard Bin- Allocation algorithm for the implementation of the model.

2.5 Algebraic formulation for maximizing volume by using ETD as a constraint for Purchase Orders

Algebraic formulation of section 2.2 can be modified as follows to maximize the priorities of the purchase orders (include a proxy of estimated delivery date ETD) by using volume of container as a constraint.

maximize

$\sum_{i=1}^{n} d_i x_i$, where d_i can be a proxy–variable for selecting ETD (eg. Find a function to map ETD Delivery date to d_i, that is purchase orders that need to be sent urgently get a high value compared to the orders that can be sent much later)

Subject to the constraints

$\sum_{i=1}^{n} v_i x_i = V_j$ for each $j = 1, 2, \ldots m$

where v_i is the volume of purchase orders and V_j indicate volume of containers.

and

$x_i = \{0, 1\}$ (Parcels are unique and cannot be repeated)

Selection of Algorithm for set of equations in 2.5

Algebraic formulation of the problem in 2.5 has only one constraint and can be implemented using Zero-One Knapsack Optimisation. The Model maximizes the priority of purchase orders in each selected containers (one by one) using volume of the of the container as a constraint.

---

3. Data Specific requirements and assumptions in the Optimisation

The Real Flow scenario is as follows. The cargoes are being filled into containers which are taken by shipping vessels for an ocean freight between a pair of ports - Port of Loading (POL) and a Port of Destination (PoD). Models described in section 1 and 2 aims for a Load Plan to allocate a set of containers to accommodate a required set of purchase orders by optimizing (i) number of required Containers to carry the purchase orders by minimizing cost and volume such that all the purchase orders reaches the destination on or before their expected date of delivery.

Assumptions

1. Models for container selection in section 1 consider the cost / profit associated with each of the containers as a constant. But cost of the container in fact is a function of Vessel Carrier Cost depending upon the availability of the vessels and Containers which is a variable.

   Having prior data, gives a possibility of estimating and forecasting the container cost over years based on mathematical models. The seasonal variation of container cost in different locations can give a better understanding of the forecasting model and it can be optimized specific to locations / ports/ type of cargoes/ type of ocean freight further.

2. Models for allocation of purchase orders from section 2.1 to 2.4 assumes that the purchase orders are already classified based on ETD (it is a simple sorting problem with least computational complexity) and then use one of the combinatorial algorithm (knapsack or bin sack) for optimization based on various constraints. Model for allocation of purchase orders in section 2.5 is using a proxy of ETD to maximize priority of purchase order to fill in the containers.

Further Suggestions for Dataspecific Optimisation

Specific details of data can be used for further optimization of allocation of Parcels such as,

1. Classification of the purchase order based on ETD to sort the purchase order based on priority
2. Cluster of ETD Vs i for $i = 1, 2, \ldots n$ to organize and cluster the purchase orders according to ETD
3. difference between PO–line upload rate and ETD Vs i for $i = 1, 2, \ldots n$ to organize the customers/ consignees based on the patterns of their Purchase orders update request and delivery date
4. container cost variation (i) global (with respect of cost of fuel, economy) and (ii) seasonal (with respect to different seasons due to market scarcity and demand of purchase orders)
5. Distribution Profile of volume of the purchase order v_i Vs i for $i = 1, 2, \ldots n$ to further optimize the bin packing algorithm to allocate the purchase orders

.