Pickup & Delivery Problem with Mulitiple Time Windows

Preben Bucher-Johannessen

November 14, 2018

Contents

1 Introduction

TODO..

2 Problem Formulation

Our problem is typically for a vehicle manufacturing company that has a certain set of factories where the vehicles are produced, each with a set of demanded parts for production. The parts, or orders (typically named transport orders but we refer to orders here), can be delivered from a set of suppliers. At a given point in time (could be delivery for one day or a week) we consider a planning problem with a demand for orders that should be satisfied with the given suppliers using vehicles provided by different logistics companies.

The vehicles have different capacities, cost structures, incompatabilities and start at the first pickup location at the first pickup time (ie. costs to get from start to first pickup are not considered neither is time). The cost structure can be multidimentional taking into account total distance and volume in different formats (variable costs, fix costs etc).

When a delivery is assigned to a vehicle, the vehicle must load the delivery from the supplier, and deliver the delivery at the factory dock. The docks in each factory can differ according to which order is being delivered and there is a limit to how many docks each vehicle can visit at each factory.

Each delivery/pickup can have several time windows, lapsing sometimes over several days. If a car arrives before a time window it has to wait.

The mathematical formulation of the problem will now follow where the set of vehicles used is denoted by V and capacity of each vehicle $v \in V$ is denoted by K_v. We let n be the number of orders in the problem, so if i is a specific pickup-node then $i + n$ corresponds to the delivery node for the same order. Here it follows that for each vehicle we have a set O_v of orders that vehicle $v \in V$ can transport based on the incompatabilities. The set of pickup Nodes (supplier docks) we denote using N^P and each delivery node (Factory dock) is denoted by N^D. All nodes are therefore represented by $N = N^P \cup N^D$ Each Factory, $f \in F$, also has a set of Nodes belonging to the same factory wich we denote N_f. The delivery nodes that belong to one factory can then be represented as $N_f^D = N_f \subset N^D$. Each vehicle has a set of Nodes it can travel to (corresponding to orders O_v) represented by N_v. This set also includes an origin node, $o(v)$ and a destination node $d(v)$ corresponding to the first pickup and the last delivery of vehicle v. The set of Edges that each vehicle can traverse is represented by E_v. The factory docking limit is denoted by D_f. Each vehicle has a given amount of docks per factory it wants to deliver to represented by d_f^v.

Each node has a set of h_i time windows represented by $[\underline{T_{ip}}, \overline{T_{ip}}] \in [0, T]$ where $p \in \{0, 1, ..., h_i\}$ and all nodes should be picked up and delivered after time 0 and before time T. Each node has a current time based on when its being served, denoted by t_i and a current load, denoted by l_i, where $i \in 2n$ and $v \in V$. The distance from node i to node j is denoted by d_{ij}. The cost of vehicle v driving from node i to j is denoted by $C_{ij}^{vq(v)w(v)}$ and the corresponding time between the nodes T_{ij}^v. l_i^v is the load of vehicle v leaving node i and x_{ij}^v is a binary vairable indicating if vehicle v is travelling between i and j node. The cost of not transporting an order will be the same for each order (should be set relatively high to avoid dummy transports) so we denote this with C^N and each order, i, not transported is represented by a binary variable y_i.

$$min \sum_{v \epsilon V} \sum_{s \epsilon S} C_{vs} d_{vs} + \sum_{i \in N^P} C_i y_i \tag{1}$$

subject to:

$$\sum_{s \epsilon S} x_{ijvs} = x_{ijv} \leq 1, \qquad v \in V, (i, j) \in E_v \tag{2}$$

$$\sum_{v \epsilon V} \sum_{j \epsilon N_v} x_{ijv} + y_i >= 1, \qquad i \in N^P \tag{3}$$

$$\sum_{j \epsilon N_v} x_{ijv} - \sum_{j \epsilon N_v} x_{jiv} = 0, \qquad v \in V, i \in N_v \backslash \{o(v), d(v)\} \tag{4}$$

$$\sum_{j \epsilon N_v} x_{o(v)jv} = 1, \qquad v \in V \tag{5}$$

$$\sum_{j \epsilon N_v} x_{jd(v)v} = 1, \qquad v \in V \tag{6}$$

$$\sum_{j \epsilon N_v} x_{ijv} - \sum_{j \epsilon N_v} x_{(i+n)jv} = 0, \qquad v \in V, i \in N_v^P \tag{7}$$

$$l_{iv} + Q_j - l_{jv} \leq K_v(1 - x_{ijv}), \qquad v \in V, j \in N_v^P, (i, j) \in E_v \tag{8}$$

$$l_{iv} - Q_j - l_{jv} \leq K_v(1 - x_{i(j+n)v}), \qquad v \in V, j \in N_v^P, (i, n + j) \in E_v \tag{9}$$

$$0 \leq l_{iv} \leq K_v, \qquad v \in V, i \in N_v^P \tag{10}$$

$$h_j - 1 - h_i \leq H_f(1 - x_{ijv}), \qquad v \in V, (i,j) \in N_f^D, f \in F \tag{11}$$

$$h_i \leq H_f x_{ijv}, \qquad v \in V, j \in E_v \backslash N_f^D, i \in N_f^D, f \in F \tag{12}$$

$$h_i = x_{ijv}, \qquad v \in V, i \in N_f^D, j \in E_v \backslash N_f^D, f \in F \tag{13}$$

$$0 \leq h_i, \qquad i \in N \tag{14}$$

TODO: possible that 14 is redundant? Should be covered by 13 right?

$$\sum_{p=1}^{h_i} u_{ip} = 1, \qquad i \in N \tag{15}$$

$$\sum_{p=1}^{h_i} u_{ip} \underline{T_{ip}} \leq t_i, \qquad i \in N \tag{16}$$

$$\sum_{p=1}^{h_i} u_{ip} \overline{T_{ip}} \geq t_i, \qquad i \in N \tag{17}$$

$$t_{iv} + T_{ijv} - t_{jv} \leq (\overline{T_{ik}} + T_{ij}^v)(1 - x_{ij}^v), \qquad v \in V, (i,j) \in A_v \tag{18}$$

$$t_i^v + T_{i(i+n)}^v - t_{(i+n)}^v \leq 0, \qquad v \in V, i \in N_v^P \tag{19}$$

,

$$\sum_{a \epsilon A} u_{ia} \leq 1, \qquad v \in V, i \in E_v \tag{20}$$

TODO: not final but need something like this

$$u_{ia} \underline{T_{ia}} \leq t_{iv} \leq \overline{T_{ia}} u_{ia}, \qquad v \in V, i \in N_v, \tag{21}$$

TODO: The following should work but I also need to find a way to integrate total weight in the equation

$$\sum_{s \in S} d_{vs} = \sum_{v \in V} \sum_{(i,j) \in E_v} x_{ijv} D_{ijv}, \qquad v \in V \tag{22}$$

$$B_{(s-1)} w_{vs} \leq d_{vs} \leq B_s w_{vs} \tag{23}$$

$$\sum_{s \in S} w_{sv} = 1, \qquad v \in V \tag{24}$$

$$y_i \in \{0,1\} \quad i \in N_v^P \tag{25}$$

$$x_{ij}^v \in \{0,1\} \quad v \in V, (i,j) \in A_v \tag{26}$$

The objective function (1) sums up to the cost of the spot cars and the aim is to minimize these costs.

List of Variables and parameters:
$v \in V$ - Vehicles
n - number of orders
$s \in S$ - size of cost structure regarding distance
i - orders
K_v - Vehicle Capacity
O_v - Set of orders i that vehicle v can transport
N^P - pickup nodes
N^D - delivery Nodes
N_f^D - Delivery nodes for factory f
E_v - Edges/Arcs the vehicle v can traverse
$f \in F$ - Factory
H_f - Factory docking limit
h_i - factory visits at node i
C_{ijvs} - Cost of transport from i to j with vehicle v
T_{ia} - Node i has a sets of a time windows in tuples $[\underline{T_{ia}}, \overline{T_{ia}}]$ where $a \in A$
T_{ijv} - Travel time from i to j using v
u_{ia} - binary variable for timewindow a at node i
l_{iv} - Load at node i with vehicle v
Q_j - weight of order at node j
x_{ijvs} - binary variable indicating travel from i to j with v
x_{ijv} - sum of x_{ijvs} for all $s \in S$
y_i - binary indicating order not picked up
C_i - cost of not transporting an order at node i
B_s - cost structure based on distance, $B_1 = 0$

3 Solution in AMPL

TODO.. I wrote something again and again.. hey it worked!

4 References

Lu 2012
https://www.sciencedirect.com/science/article/pii/S0305054813002165
https://www.sciencedirect.com/science/article/pii/S0305048304001550
https://link.springer.com/article/10.1007/s10732-005-5432-5
https://pubsonline.informs.org/doi/abs/10.1287/trsc.1050.0135
M. Christiansen & K. Fagerholt (2002) - Robust Ship Scheduling with Multiple Time Windows -