A dynamic programming algorithm and a compact MILP formulation for the urban freight transport by rail

Onur OZTURK

Université Paris-Est, ESIEE Paris, Département Ingénierie des Systèmes, 2, boulevard Blaise Pascal Cité DESCARTES BP 99 93162 Noisy le Grand CEDEX

onur.ozturk@esiee.fr

Mots-clés : *urban freight transport, dynamic programming, mixed integer linear programing.*

1 Introduction

As a rising trend in some countries, urban rail transit systems are used for the freight transport through cities ([1]). We present here a decision support framework for the problem of urban freight movement by rail and present mathematical methods for the optimal distribution of goods. The problem we consider has a single rail line on which some stations can be used as loading/unloading platforms for goods. Demand is known in advance and each client desires a different time for the delivery.

2 Dynamic programming procedure for the problems having a fixed number of demand sizes

We model the problem as a parallel batch scheduling problem in the presence of different job/demand sizes. The case we tackle here has two stations : first one dedicated to loading demands, second for the final destination for all demands. Goods have equal release dates and equal due dates. In addition, the number of different demand sizes, i.e., demand volumes, is considered to be constant and equal to K. There are T trains (or train trips) employed and it is known in advance the departure time of trip t. The objective is the minimization of total waiting time of goods at the departure station. This objective is equivalent to minimizing $\sum C_j$ where C_j is the processing ending time of job j in a scheduling problem, i.e., delivery time of demand j. Note that if K is arbitrary, the problem is $NP-hard$ ([2]). When a constant number of demand volumes is considered, the problem becomes equivalent to deciding how many objects of volume (or size) $size_1$, $size_2$,..., $size_K$ should be loaded on train t. Define $f[q_1, q_2, ...q_K, t]$ to be the minimum $\sum C_j$ value for a partial schedule completed at time train t arrives to the final destination containing the first q_k demands of size $size_k$, $k = 1, ..., K$. Initially, set $f[0, 0, ...0, 0] = 0$ and all other values to infinity. The optimal $\sum C_j$ value will be the smallest value of the form

$$minf[n_1, n_2, ...n_K, T]$$

where T is the last train and $n_1, n_2, ...n_K$ are the total numbers of demands having size $size_k$, $k = 1, ..., K$.

The function values can be computed using the following recursive relation :

$$f[q_1, ...q_K, t] = min_{q'_k \in \{0,...,q_k\}} f[q'_1, ..., q'_K, t-1] + \sum_{k=1}^{K} (q_k - q'_k) * (arrT_t - depT_t)$$

where $\sum_{k=1}^{K} (q_k - q'_k) * size_k \leq Capacity$, $arrT_t$ and $depT_t$ are arrival and departure times of train t at the delivery and departure stations, respectively. The complexity of the procedure is $O(n^{2K} * T))$.

3 A compact MILP formulation to minimize total tardiness in case of arbitrary demand sizes and multiple stations

We start by generating a matrix M which contains the tardiness values of each job j assigned to train t. The elements M_{jt} of matrix M is calculated with the following formula :

$$M_{jt} = \begin{cases} max(0, A_{t,arr_j} - d_j), & \text{if } r_j \leq A_{t,dep_j} \\ \infty, & \text{otherwise} \end{cases}$$

where r_j is the ready time and d_j the due time of demand j, A_{ts} is the arrival time of train t at station s, dep_j and arr_j are the departure and arrival stations, respectively, of demand j. Let x_{jt} be the binary variable that equals to 1 if demand j is assigned to train t and 0 otherwise. Then, tardiness value of demand j assigned to train t is determined with the following sum : $\sum_{t=1}^{T} x_{jt} * M_{jt}$.

Minimize $\sum_{\forall j} \sum_{\forall t} x_{jt} * M_{jt}$
s.t.

$$\sum_{\forall t} x_{jt} = 1 \quad j = 1, .., N \tag{1}$$

$$\sum_{\forall j'} x_{j't} * size_{j'} \leq Cap \quad t = 1, .., T \text{ and } j' \in J'_s \tag{2}$$

$$\sum_{\forall j''} x_{j''t} * time_{j''} \leq wait_{max} \quad t = 1, .., T \text{ and } j'' \in J''_s \tag{3}$$

Constraint set (1) assigns each demand to a single train. Constraint set (2) allows for each train t at each station s a maximum capacity use equal to Cap. Moreover for each t at each station, set J'_s contains demands j such that $dep_j \leq s < arr_j$ and $r_j \leq A_{ts}$, i.e., station s is greater or equal to the departure station and smaller than the arrival station for demand j, and the release date of demand j is smaller than the arrival time of train t at the departure station of that demand. This way, we calculate the capacity use of trains taking into account demands that can be present in train t at station s. Similarly, constraint set (3) computes the total loading and unloading time of items for each train at each station. For each train t and station s, set J''_s contains demands j if station s is the departure or arrival station for demand j respecting the feasibility of demand release dates and train arrival times at departure stations.

4 Conclusions

In this paper, we proposed an optimization framework for the use of urban/suburban rail transit for freight transport. First, we studied the case of two stations in the presence of fixed demand sizes and developed pseudo-polynomial dynamic programming algorithm. Then we continued with a further generalization considering any demand size in the presence of multiple departure and arrival stations. Numerical results showed that the dynamic programming procedure is faster in terms of solution time than the MILP model when the number of different demand sizes is smaller or equal to 5. Other than that, the MILP model converges quickly to the optimal solution such that the optimality gap is less than 1% at the end of one minute of computation for instances containing 100 demands per day, 30 to 40 trains, and 2 to 5 stations.

Références

[1] Marin Marinov, Federico Giubilei, Mareike Gerhardt, Tolgahan Özkan, Evgenia Stergiou, Mihaela Papadopol, and Luis Cabecinha. Urban freight movement by rail. *Journal of Transport Literature*, 7(3) :87–116, 2013.

[2] R. Uzsoy. Scheduling a single batch processing machine with non-identical job sizes. *International Journal of Production Research*, 32(7) :1615–1635, 1994.