Scheduling multi-commodity ows in a two-level transportation network with handling capacity constraints

Xavier Libeaut, Eric Pinson, Jorge E. Mendoza

Scheduling multi-commodity flows in a two-level transportation network with handling capacity constraints

Xavier Libeaut
Eric Pinson
Jorge E. Mendoza
LUNAM Université - LISA (EA 4094 CNRS)
Université Catholique de l'Ouest, Angers, France
Email: {xavier.libeaut,eric.pinson,jorge.mendoza}@uco.fr

1 Introduction

Load plan design aims to determine how a set of commodities should be routed from origin to destination through a transportation network to minimize the operational cost of the system (Wieberneit, 2008). In industries such as less-than-a-truckload transportation (Jarrah et al., 2009; Erera et al., 2012) and express package delivery (Barnhart et al., 2002) two-level transportation networks with origin/destination terminals and consolidation hubs allow for significant cost reductions due to higher resource utilization rates. In this context, load plan designing shall determine not only the routes the commodities follow through the network but also when and where to hold commodities to improve consolidation (Erera et al., 2012). In this research we focus on the problem of deciding when and how to move the commodities through a two-level network in order to minimize the cost of a load plan. In the considered network, both terminals and hubs have a limited handling capacity and the commodities have release and due dates that must be met to warrant customer service levels. To tackle the problem, we propose a large neighborhood search approach with MIP-based neighborhoods.

2 Problem definition and formulation

The transportation network considered in this work involves two types of terminals: a set $\mathcal{C}$ of inbound/outbout treatment centers (TCs) and a set $\mathcal{H}$ of consolidation hubs (Hs). A set $\mathcal{P}$ of commodities has to be moved through the network along a planning horizon Θ which is assumed to be partitioned into τ equal periods. Each site $s \in \mathcal{C} \cup \mathcal{H}$ has a given handling capacity per period η_s (which is assumed to be constant over the planning horizon). Each commodity $p \in \mathcal{P}$ is characterized by a 7-uplet $[s(p), t(p), qt(p), es(p), lc(p), h_o(p), h_d(p)]$, where $s(p)$ and $d(p)$ denote its origin and destination TCs; $qt(p)$ denotes the amount of commodity to be transported from $s(p)$ to $d(p)$; $es(p)$ denotes the earliest time at which the $qt(p)$ units of the commodity are available at $s(p)$; $lc(p)$ denotes the latest time (deadline) at which the $qt(p)$ units of the commodity must be handled at $t(p)$; finally, $h_o(p)$ and $h_d(p)$ denote the hubs that define the path $s(p) \rightarrow h_o(p) \rightarrow h_d(p) \rightarrow t(p)$ that commodity p must follow to go from its origin to its destination. It is worth noting that for some commodities $h_o(p) = h_d(p)$, meaning that their path transits only one consolidation hub. Every commodity $p \in \mathcal{P}$ has a constant handling time per unit of commodity at both hubs and treatment centers. To move commodities across the network, an unlimited fleet of vehicles is available on each echelon (CT→H, H→H, or H→CT). The vehicle's attributes (e.g., capacity, cost) vary according to the link type. The problem is to determine a minimal-cost plan to move the commodities through the network while meeting their deadline constraints and the capacity constraints of the treatment centers and hubs.

Let us define a transport operation (TO) as the assignment of a vehicle to a network link (CT→H, H→H, or H→CT) at a given time slot. Thus, a TO ω is characterized by a 6-uplet $[o(\omega), d(\omega), st(\omega), ct(\omega), cp(\omega), cf(\omega)]$, where $o(\omega)$ and $d(\omega)$ denote its start and destination site, $st(\omega)$ denotes its starting time at $o(\omega)$ and $ct(\omega)$ its arrival time at $d(\omega)$, $qt(\omega)$ denotes its capacity, and $cf(\omega)$ its fix operating cost. We denote by Ω the set of all the potentially eligible TO according to the network structure. A route is then defined as an ordered and chronologically consistent sequence of TOs linking two TCs. Hereafter, we refer to $\mathcal{R}$ as the set of all the routes that can be built from Ω. For all $\sigma \in \mathcal{R}$ and $\omega \in \Omega$, parameter a_ω^σ takes the value of 1 iff the route σ includes TO ω, and 0 otherwise. For each site $s \in \mathcal{C} \cup \mathcal{H}$, let us denote by $\mathcal{P}_s$ the subset of commodities transiting through site s. Let us also define $\mathcal{R}^p \subset \mathcal{R}$ as the set of routes that can be used to move commodity p according to its predefined path and release and due dates. Finally, let us define the decision variable y_ω as the number of times that TO $\omega \in \Omega$ is selected in an optimal solution, and x_p^σ as the amount of commodity $p \in \mathcal{P}$ moved through route $\sigma \in \mathcal{R}$. A mix-integer programming formulation of the problem at hand is:

$$Min \sum_{\omega \in \Omega} cf(\omega) y_\omega \tag{1}$$

s.t.

$$\sum_{\sigma \in \mathcal{R}^p} x_p^\sigma = qt(p) \quad \forall p \in \mathcal{P} \tag{2}$$

$$\sum_{p \in \mathcal{P}} \sum_{\sigma \in \mathcal{R}} a_\omega^\sigma x_p^\sigma \leq cp(\omega) y_\omega \quad \forall \omega \in \Omega \tag{3}$$

$$\sum_{p\in\mathcal{P}_s}\sum_{t\le\theta}\sum_{\sigma\in\mathcal{R}_s^t} x_p^\sigma \ge \sum_{p\in\mathcal{P}} qt(p) - t\eta_s \quad \forall s\in\mathcal{C}, \theta\in\Theta \tag{4}$$

$$\sum_{p\in\mathcal{P}_h}\sum_{\sigma\in\mathcal{R}_h(\theta^-,\theta^+)} x_p^p \le (\theta^+-\theta^-)\eta_h \quad \forall h\in\mathcal{H}, (\theta^-,\theta^+)\in\Theta^2, \theta^-<\theta^+ \tag{5}$$

$$y_\omega \in \mathbf{Z}^+ \quad \forall\omega\in\Omega \tag{6}$$

$$x_p^\sigma \ge 0 \quad \forall p\in\mathcal{P}, \sigma\in\mathcal{R}^p \tag{7}$$

were $\mathcal{R}_h(\theta^-,\theta^+) = \{\sigma\in\mathcal{R}_h | (\exists\omega\in\Omega_h^-, a_\omega^\sigma = 1 \wedge ct(\omega)\ge\theta^-) \wedge (\exists\omega\in\Omega_h^+, a_\omega^\sigma = 1 \wedge st(\omega)\ge\theta^+)\}$

Constraints (2) ensure that each commodity is fully routed in the network. The coupling constraints (3) make sure that every TO has enough accumulated capacity to cover the requirements of all the routes that include the TO. Constraints (4) and (5) model the handling capacity of TCs and Hs, respectively. Constraints (6) and (7) define the nature of the decision variables. $\mathcal{R}_h(\theta^-,\theta^+)$ simply denotes the subset of routes that include a TO arriving at hub h after θ^- and a TO leaving h before θ^+.

3 A large neighborhood search approach

Depending on the size of the transportation network (number of CTs and Hs), the number of products, and the number of epochs τ in the planning horizon, MIP (1)-(7) (henceforth referred to simply as MIP) can rapidly become intractable for commercial solvers. To overcome this difficulty, we propose a large neighborhood search (Shaw, 1998) approach that uses a partial version of MIP as a neighborhood scheme (hereafter LNS). At each iteration the algorithm improves its search solution by solving MIP considering only the x and y variables induced by a reduced set of TOs $\bar{\Omega}\subset\Omega$ and the set of routes $\mathcal{R}(\bar{\Omega})\subset\mathcal{R}$ formed by all the routes that can be built using TOs in $\bar{\Omega}$. To generate $\bar{\Omega}$ for the first iteration our algorithm uses a greedy heuristic. Sweeping the planning horizon from the first to the last epoch, the heuristic checks which commodities are available for shipment to the next stop on their path and orders, or delays until the next epoch, their shipment depending on a pre-define vehicle utilization rate. The TOs used in the shipments ordered by the heuristic are then used by the LNS to solve the partial MIP and find the starting solution. On each subsequent iteration, the LNS modifies $\bar{\Omega}$, and thus $\mathcal{R}(\bar{\Omega})$, by applying a *maintenance operator*. The operator first eliminates from $\bar{\Omega}$ all the transport operations that are not used in the current solution, i.e., $y_\omega = 0 | \omega\in\bar{\Omega}$. Then the operator randomly selects a commodity p' and adds to $\bar{\Omega}$ all the TOs associated with the links forming the commodity's path from its origin to its destination. Finally, the operator updates $\mathcal{R}(\bar{\Omega})$.

Next, our approach uses a destruction operator to partially destroy the current solution. The operator simply frees the values of the x and y variables associated with the links forming the path of commodity p' while blocking the values of the remaining decision variables. To re-construct the solution, our algorithm solves the partial MIP over the current sets $\bar{\Omega}$ and $\mathcal{R}(\bar{\Omega})$. The algorithm repeats the same process until a given number of iterations or a maximum execution time is reached.

4 Computational experiments

We implemented our approach in Java and used CPLEX 12.1 to solve the MIP. To the best of our knowledge no benchmarks for our problem are publicly available. Therefore, we generate a set of 30 instances with 10, 30, and 50 CTs and 2, 4, and 6 hubs, respectively. For every instance we considered that there are commodities flowing between every pair of CTs and that the planning horizon is partitioned in 48 epochs. For each instance we also generated 4 different capacity vectors for CTs and hubs. We ran our LNS on the 120 instances setting the maximum execution time to 2h and the maximum number of iterations to $|\mathcal{P}|$. To assess the effectiveness of our algorithm we developed two different lower bounds (LBs). Our algorithm unveiled 5 optimal solutions. For the remaining instances, the algorithm delivered solutions with gaps with respect to the LBs ranging from 0.4% to 10.80% with an average of 5.60%.

References

C. Barnhart, N. Krishnan, D. Kim, and K. Ware. Network design for express shipment delivery. *Computational Optimization and Applications*, 21(3):239–262, 2002.

A. Erera, M. Hewitt, M. Savelsbergh, and Y. Zhang. Improved load plan design through integer programming based local search. *Transportation Science*, In press, 2012.

A. Jarrah, E. Johnson, and L. C. Neubert. Large-scale, less-than-truckload service network design. *Operations Research*, 57(3):609–625, 2009.

P. Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In *Principles and Practice of Constraint Programming*, volume 1520 of *Lecture Notes in Computer Science*, pages 417–431. Springer, 1998.

N. Wieberneit. Service network design for freight transportation: a review. *OR Spectrum*, 30:77–112, 2008.