DTU Library

Modeling and solving a multimodal transportation problem with flexible-time and scheduled services

Moccia, Luigi; Cordeau, Jean-François; Laport, Gilbert; Røpke, Stefan; Valentini, Maria Pia

Published in:
Networks

Link to article, DOI:
10.1002/net.20383

Publication date:
2011

Document Version
Early version, also known as pre-print

Modeling and solving a multimodal routing problem with timetables and time windows

Luigi Moccia*†, Jean-François Cordeau‡, Gilbert Laporte§, Stefan Ropke§, Maria Pia Valentini¶

January 29, 2008

Abstract

This paper studies a routing problem in a multimodal network with shipment consolidation options. A freight forwarder can use a mix of flexible-time and scheduled transportation services. Time windows are a prominent aspect of the problem. For instance, they are used to model opening hours of the terminals, as well as pickup and delivery time slots. The various features of the problem can be described as elements of a digraph and their integration leads to a holistic graph representation. This allows an origin-destination integer multi-commodity flow formulation with non-convex piecewise linear costs, time windows, and side constraints. Column generation algorithms are designed to compute lower bounds. These column generation algorithms are also embedded within heuristics aimed at finding feasible integer solutions. Computational results with real-life data are presented and show the efficacy of the proposed approach.

Keywords: multicommodity flow problem, time windows, transportation timetable, mutimodal transportation, column generation, non-convex piecewise linear cost function.

*Dipartimento di Elettronica, Informatica e Sistemistica, Università della Calabria, 87036 Rende (CS) - Italy - luigi.moccia@deis.unical.it

†ENEA - Ente per le Nuove Tecnologie, l'Energia e l'Ambiente - C.R. Trisaia, s.s. 106 Jonica km 419+500, 75026 Rotondella (MT) - Italy

‡Canada Research Chair in Logistics and Transportation, HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7 - jean-francois.cordeau@hec.ca

§Canada Research Chair in Distribution Management, HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7 - gilbert@crt.umontreal.ca, sr@ctt.dtu.dk

¶ENEA - Ente per le Nuove Tecnologie, l'Energia e l'Ambiente - C.R. Casaccia, via Anguillarese 301, 00060 Roma (RM) - Italy - mpia.valentini@casaccia.enea.it

1 Introduction

We describe and solve an operational problem faced by a freight forwarder. Given a set of origin-destination transport requests, one must optimally route these requests in a multimodal network. We assume that the freight forwarder does not operate a vehicle fleet, but can access a heterogeneous set of transportation services. These services can be classified according to two main characteristics: type of departure time, and cost function. We differentiate between timetabled services and time-flexible services. Usually, rail and short sea shipping modes are operated with fixed departure times while trucks have flexible departures. Some services allow consolidation of shipments between two terminals. A terminal is where a transfer can take place between modes or between different vehicles of the same mode. Consolidation enables fixed costs sharing. This effect is captured by *piecewise linear* (PL) cost functions that depend upon the total service load. These cost functions are non-convex and, in general, non-concave. Other types of services do not allow consolidation and their cost function is thus that of the single shipment. We therefore distinguish between consolidation and dedicated services. Consolidation services present multiple capacity constraints, e.g. volume, weight, train length, etc. Dedicated services are not viewed as capacitated because they either are feasible or not considered for a given shipment. The pickup of a transport request is done by selecting between multiple time windows, e.g. between 08.00-10.00 AM of day one, or 08.00-10.00 AM of day two of the planning horizon. Similarly there are multiple time windows for the delivery. The chosen route must respect the opening time windows of the terminals. Because of these characteristics, namely multimodal, multiple capacity constraints, and multiple time windows, we denote this problem as the *M++ Routing Problem* (M++RP).

While this paper attempts to set a general framework for multimodal routing, it stems from a real-life application. A freight forwarder operating on the Italian market can integrate rail and road transportation services. The largest customer requires shipments from factories located in Northern Italy to regional distribution warehouses in Central-South Italy. A factory must send many different shipments: they are distinct because there can be different destinations or incompatible pickup and delivery time windows. Block trains can be activated between stations. At the arrival station the shipment can be sent to its final destination by truck. The freight forwarder can consolidate shipments by using trucks between factories and intermediate platforms. These nodes are then connected to the warehouses by dedicated truck services.

However, the dedicated origin-destination shipment by truck is always a possible option. In addition, consolidation by truck can take place between a factory and a loading train station.

The remainder of the paper is organized as follows. Section 2 discusses the relevant literature. Notation and graph construction procedures are introduced in Section 3, while Section 4 describes mathematical models and their properties. Section 5 outlines algorithms to obtain upper and lower bounds, while Section 6 presents computational results. Conclusions are reported in Section 7.

2 Literature Review

We first review freight transportation surveys in Section 2.1 in order to position the problem within the relevant literature. Available methodologies for the representation of multimodal networks are discussed in Section 2.2. The literature dealing with problems similar to ours is assessed in Section 2.3, while Section 2.4 considers the optimization methodologies that are relevant to our approach.

2.1 Freight logistics surveys

Crainic and Laporte (1997) extensively review the optimization models for freight transportation. A main distinction can be established between strategic-tactical and operational models that respectively consider a national or an international multimodal network, such as in the *Service Network Design Problem* (SNDP) (see Crainic (2000)), and the unimodal distribution management models that are variants of the *Vehicle Routing Problem* (VRP) (see Toth and Vigo (2002)). Macharis and Bontekoning (2004) present a freight logistics literature review focused on intermodal transportation. They propose a classification based on two criteria: the type of operator and the length of the problem's time horizon. Four types of operators are distinguished: drayage operators, terminal managers, network planners, and intermodal operators. The time horizon criterion results in the classical differentiation of strategic, tactical, and operational levels. In this classification matrix of twelve categories, the M++RP would correspond to operational problems faced by an intermodal operator, since the problem can be stated as the selection of routes and of services in a multimodal network. This problem category, according to Macharis and Bontekoning (2004) and to our own updated survey (see Section 2.3), is one of the least studied. Bontekoning et al. (2004) review the intermodal literature related to the rail-truck combination. This paper, like the previous one, highlights the need for more research on operational problems faced by intermodal operators, like the optimal route selection. Container based transportation is the key enabler of intermodalism because of various advantages like higher productivity during the transfers, less product damage, etc. Consequently, Crainic and Kim (2006) focus their recent intermodal logistics literature review on the container related aspects of the transportation industry. In particular, empty container repositioning and container terminal management problems are thoroughly discussed.

2.2 Representing multimodal networks

A multimodal network involves services that can operate on the same infrastructure. For instance, trucks of different capacities using the same highway link. Furthermore, important operations, like mode transfers, are not represented on the geographical, or physical, network. These require the creation of a so called "virtual network" where links relate to different operations in the multimodal chain. In multimodal freight transportation models the virtual network definition dates back to the early 1990's (Guélat et al., 1990; Crainic et al., 1990). Jourquin et al. (1999) present a methodology for the automatic generation of virtual networks in a *Geographic Information System* (GIS). Similarly, Southworth and Peterson (2000) deal with the digital representation of a multimodal and international freight transportation network. They extend a commercial GIS to handle intermodal transfers and network access and egress. The above cited methodologies were aimed at tactical and strategic planning. Accordingly, operational aspects, like time synchronization, were not considered. Our contribution, with respect to this literature, consists in procedures for virtual networks with embodied time synchronization.

2.3 Applications related literature

Barnhart and Ratliff (1993) introduce two models to determine a minimum cost route combining truck and rail services. The first model holds when the rail service has a per trailer cost, while in the second model the rail cost is per flatcar with a maximum of two trailers per flatcar. Time related constraints are added for trailer availability and maximum allowed delivery time. Under these hypotheses the minimum cost route for the first model is easily found by means of a shortest path algorithm, while for the second model a non-bipartite weighted matching procedure is applied. Min (1991) studies an international intermodal supply chain. He presents a chance-constrained goal programming model to select the intermodal mix that minimizes costs and risks while satisfying on-time service requirements. Boardman et al. (1997) sketch a decision support system that selects the multimodal route by means of a k-shortest path algorithm. Ziliaskopoulos and Wardell (2000) present an algorithm for computing an optimal path in a multimodal network when travel and transfer times are dynamic. They also consider timetabled services. Their approach is aimed at realistic simulation of large networks both in freight and in passenger transportation. Aldaihani and Dessouky (2003) deal with the problem of mixing timetabled and flexible-time transportation services at the operational planning level in paratransit. They propose a tabu search heuristic as solution approach.

The work of Chang (2007) is the closest to our problem. Chang studies an intercontinental transportation network where there are timetabled transportation services, economies of scale represented as *piecewise linear concave* (PLC) cost functions, and delivery time windows. The main difference with respect to our problem is that he assumes that commodity flows can bifurcate. This assumption is reasonable when considering large shipments at the intercontinental scale. We deal, in contrast, with an inter-regional network where routing a shipment between more than one path is not an acceptable practice. Furthermore, our solution approach is different. Chang applies a Lagrangean relaxation scheme introduced by Amiri and Pirkul (1997) for a multicommodity flow problem with PLC costs. Other differences are that we consider more general cost functions which are not necessarily concave, and multiple time windows. We note that the multiple time windows feature is a relatively recent issue in the vehicle routing literature (Xu et al., 2003; Ibaraki et al., 2005; Caramia et al., 2007).

2.4 Methodology related literature

From a mathematical programming point of view our problem is an extension of capacitated network design models (for a review see, e.g. Gendron et al. (1998)). Since a shipment must use a single path from origin to destination in a capacitated network, it is similar to the *Origin-Destination Integer Multi-Commodity Flow Problem* (ODIMCFP) introduced by Barnhart et al. (2000). In fact, we will formulate the M++RP as an ODIMCFP with time windows, PL cost functions, and resource consumption side constraints. Analogously to the ODIMCFP, the M++RP can be modeled with path variables and solved by column generation. For a survey on column generation see, e.g., Desaulniers et al. (2005). Because of the M++RP special structure, the pricing problem is a *Shortest Path Problem with Time Windows* (SPPTW) studied, e.g., in Desrochers and Soumis (1988), and in Feillet et al. (2004). The literature dealing with piecewise linear costs is also relevant to our problem. Kim and Pardalos (1999) introduce a dynamic slope scaling algorithm to heuristically solve the *Fixed Charge Network Flow Problem* (FCNFP). Kim and Pardalos (2000) applied similar algorithms to the *Concave Piecewise Linear Network flow Problem* (CPLNFP). In fact, through an arc separation procedure, the CPLNFP can be transformed into an FCNFP on an extended graph. A more refined algorithm variant, which employs a trust interval technique, was also presented in the same paper. The dynamic slope scaling concept was exploited by Crainic et al. (2004) to solve the multicommodity version of the FCNFP. The authors propose a heuristic algorithm that combines slope scaling, Lagrangean relaxation, intensification and diversification mechanisms as in metaheuristics. Croxton et al. (2003a) prove that three textbook *mixed-integer linear programming* (MILP) formulations of a generic minimization problem with separable non-convex piecewise linear costs are equivalent. Their LP relaxations approximate the piecewise linear cost function with its lower convex envelope. Independently, Keha et al. (2004, 2006) derived a similar result. Croxton et al. (2007) present valid inequalities based upon variable disaggregation for network flow problems with piecewise linear costs. Croxton et al. (2003b) study an application, the merge-in-transit problem, where the above mentioned technique shows its efficacy.

3 Virtual Network Representation

In this section we introduce the notation and the constructive procedures to represent the M++RP on a digraph. Sections 3.1 to 3.5 describe the procedures for the problem's main features, and Section 3.6 summarizes the digraph notation.

3.1 Pickup and delivery time windows

Let K be the set of shipments, or commodities, to be routed during the planning horizon. Each shipment $k \in K$ is characterized by a set $\Omega(k) = \{1, ..., |\Omega(k)|\}$ of pickup time windows, and a set $\Gamma(k) = \{1, ..., |\Gamma(k)|\}$ of delivery time windows. The penalized delivery time windows are grouped in the set $\Gamma^p(k) \subset \Gamma(k)$ and there is a cost $c^\gamma, \gamma \in \Gamma^p(k)$, associated to each of them, i.e. we assume a staircase penalty cost function upon the arrival time. With this notation we can construct a first part of the directed graph $G = (N, A)$ on which the problem is defined. We denote by O the set of origin nodes, and by D the set of destination nodes. The origin node of a shipment $k \in K$ is denoted by $o^k \in O$ while the destination node is $d^k \in D$. We then add $|\Omega(k)|$

nodes representing the pickup time windows and $|\Gamma(k)|$ nodes for the delivery time windows. We maintain the previous notation and the sets $\Omega(k)$ and $\Gamma(k)$ indicate also the pickup and delivery time windows *node* sets. An origin node o^k is linked to the $\Omega(k)$ nodes by arcs with zero traveling time and zero cost. The notation is similar for the destination node, but the arcs between the $\Gamma^p(k)$ nodes and d^k have a cost $c^\gamma, \gamma \in \Gamma^p(k)$. The resulting portion of the digraph is illustrated in Figure 1, where a time window interval is denoted as $[a^i, b^i]$ for a generic node $i \in \Omega(k) \cup \Gamma(k)$.

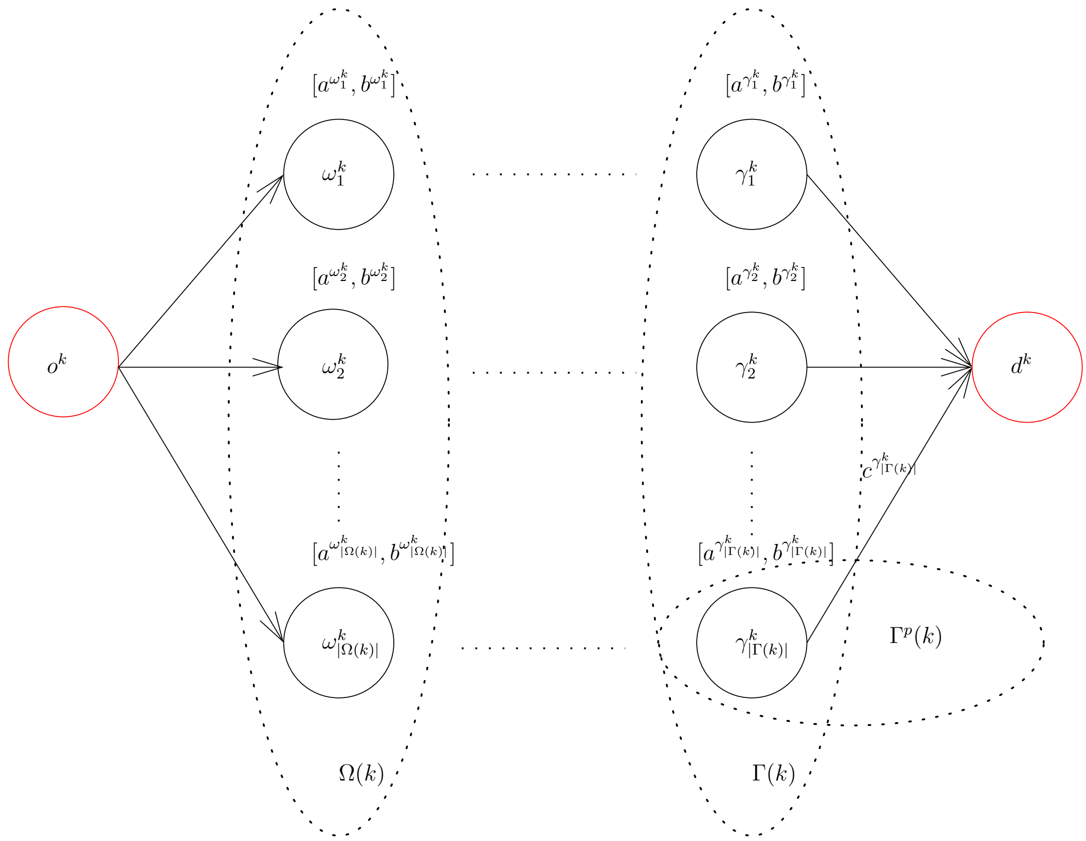

Figure 1: Portion of the digraph representing the pickup and delivery time windows for a shipment

3.2 Accessing a terminal and timetabled consolidation services

The previously described procedure results in a digraph which is still separable by shipment. However, the representation of consolidation services introduces nodes and arcs that can be used by more than one shipment. Let us construct, as an example, the access to a terminal i and the selection of a *Timetabled Consolidation Service* (TCS) from terminal i to terminal j. The physical network could be synthetically represented by two nodes, i and j, and one linking arc. The virtual network must take into account many operational characteristics, like:

- opening hours of the terminals;
- transfer times and costs that depend upon the entering mode or vehicle type, the departing mode, and the shipment requests;
- timetables of the available services.

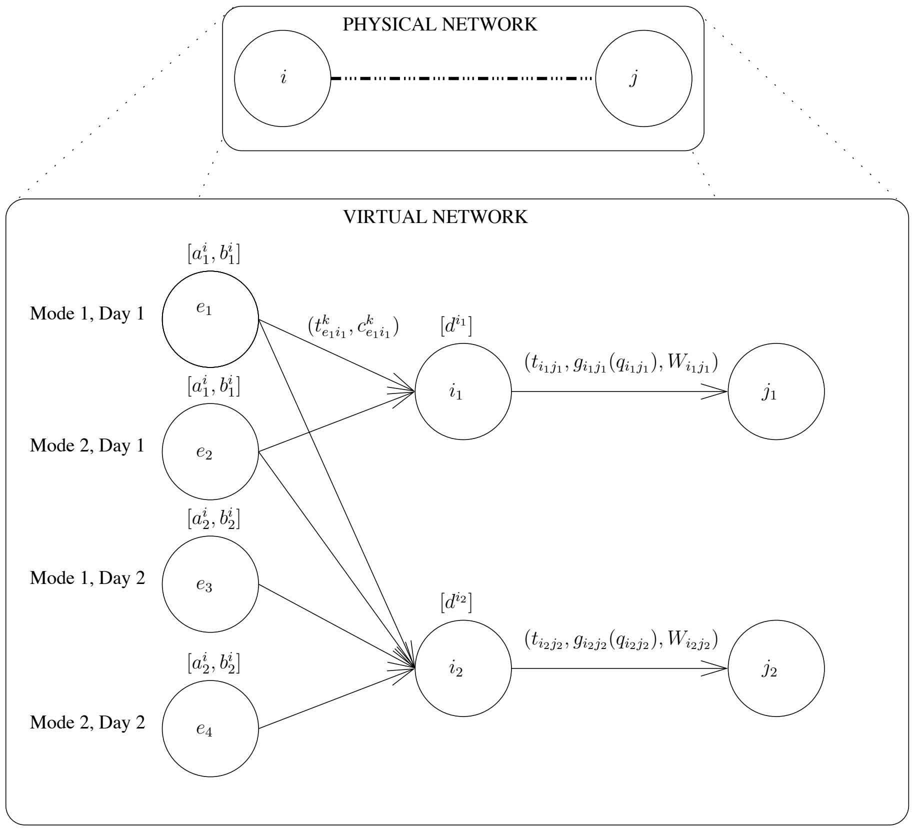

Figure 2: Example of exploding the physical network to the virtual one

Figure 2 depicts the following example: two terminals linked by two TCSs, e.g. shuttle trains, in a two-day planning horizon, and with two possible entering modes, e.g. truck and rail. We designate the departure of the two services by two nodes, i_1 and i_2, with collapsed time windows. A collapsed time window for a generic node i is such that $a^i = b^i = d^i$, where d^i indicates the departure time. The entrance to terminal i requires, in our example, four nodes to describe the possible combinations between modes and opening hours in the two-day planning horizon. We assume one opening time window per day: $[a_1^i, b_1^i]$ for day one, and $[a_2^i, b_2^i]$ for day two. We link the entrance nodes with the compatible departure nodes. We can have an arc between an entrance node e and a departure node i only if $a^e + \min_{k \in K} t^k_{ei} \leq d^i$, where t^k_{ei} is the required time for the shipment k to move between the nodes e and i. In the example of Figure 2 we assume that the service represented by the node i_1 departs on day one and is therefore compatible only with entrance nodes e_1 and e_2 which represent day one arrivals. Conversely, the service i_2 departs on day two and is compatible with all the four entrance nodes e_1 to e_4.

In this representation, costs and times on the arcs between entrance and departure nodes are related to three attributes: ingoing mode, outgoing mode, and shipment. The arc leaving a departure node connects with the corresponding arrival node and presents a PL cost function which depends upon the cumulated quantity, e.g. for a generic arc (i, j) we have $g_{ij}(q_{ij})$, where q_{ij} is the arc load. The arc load is the sum of the quantities q_{ij}^k, i.e. the arc specific quantity of the shipment k, for all the transport orders that use (i, j). As we will show in Section 4.2, the MILP model of the PL cost function will ensure that the capacity constraint related to the q_{ij}^k unit of measure will also be satisfied. For notational simplicity let us assume that we have to take into account only a second type of capacity constraint on the arc (i, j). For instance, the q_{ij}^k values express volume while there is an additional capacity constraint for the weight. We denote by W_{ij} the arc capacity and with w_{ij}^k the shipment requests. Therefore, consolidation service arcs will be modeled as capacitated.

3.3 Flexible consolidation services

Flexible consolidation services (FCSs) are not constrained by fixed departure dates. However, we still need to take into account the synchronization with the arrival of the assigned shipments. First consider that FCSs usually have a time window for the departure. Therefore, we model an FCS with a sufficiently large set of nodes with fixed departure times equally spaced along the departure time window. The optimal routing will look for consolidation and, consequently, a minimal number of departure nodes will be used. We implicitly assume that the number of available vehicles of the FCS is large enough to cover the assigned transportation demand. Since FCSs relate to trucks, this hypothesis is not restrictive.

3.4 Dedicated services

A dedicated service arises whenever the freight forwarder cannot, for technical or managerial reasons, share transportation means between different shipments. In a more general way we define a dedicated service whenever the cost function is separable by shipments. Usually dedicated services are performed by truck, and therefore they have flexible departure times. Transportation by truck between the shipment origins to terminals or from terminals to destinations, i.e. drayage operations, are examples of this type of service. These *flexible dedicated* *services* (FDSs) are the easiest to represent on a digraph: an arc between two geographically related nodes with traveling time and cost that are shipment dependent. However, in a multimodal network we can have transportation services whose cost function is that of an FDS, but having timetables or guaranteed arrival times. Examples are short sea shipping line services where there are scheduled departures and arrivals at ports. Capacities of these transportation services are relatively large with respect to the requests of a freight forwarder. Thus, the negotiated contract for this type of services has a per container cost. These *timetabled dedicated services* (TDSs) can be represented with nodes featuring collapsed time windows, as in the TCS case. However, the arcs are uncapacitated and have separable cost functions, i.e. c^k_{ij} values. The last type of time synchronization that we consider, the guaranteed arrival time, requires a departure node with a time window for the consignment and an arrival node with a collapsed time window for the arrival time.

3.5 Direct origin-destination arcs

The digraph on which the problem is defined contains an arc between the origin and the destination nodes for each shipment. A direct origin-destination arc yields the minimum cost route that can be obtained by dedicated services only. This cost could be computed solving an optimization problem in a separated graph for each shipment. It is easy to see that we can determine the minimum cost route by applying a shortest path algorithm with time windows on the digraph of the dedicated services. Because of the availability of effective SPPTW algorithms and the relatively small size of the reduced digraph, the problem is not difficult. In practice we have to compare a few alternatives. We have to choose between direct origin-destination transportation by truck, or a combination, if available, of drayage operations and dedicated services between terminals. The existence of these “virtual” direct origin-destination arcs, (o^k, d^k), helps our solution approach, as explained in Section 5.5. Additionally, we have a cost estimate when shipment consolidation is not considered.

3.6 Digraph node and arc sets

To summarize, the M++RP can be represented on a digraph with the following nodes:

- origin and destination nodes, sets O and D;
- pickup and delivery nodes with time windows, sets $\Omega = \cup_{k\in K}\Omega(k)$ and $\Gamma = \cup_{k\in K}\Gamma(k)$;

- terminal entrance nodes with time windows, set N^e;
- scheduled departure nodes or time guaranteed arrival nodes, i.e. with collapsed time window, set N^d;
- consignment nodes with time windows, set N^c.

Let N be the union of the above listed node sets. For notational convenience we denote by N^{tw} the set of the time windowed nodes, i.e. $N^{tw} = \Omega \cup N^e \cup N^c \cup \Gamma$. The arc set $A \subset N \times N$ is defined according to operational constraints as outlined in the above examples. We distinguish two disjoint subsets of A:

- the set A^v, where the arc cost function is separable by shipment, i.e. we have costs $c^k_{ij} > 0, \forall (i,j) \in A^v$; these arcs represent transportation costs of dedicated services, transfer costs inside terminals, penalty costs for late arrivals, etc.
- the set A^{pl}, where the arc cost function $g_{ij}()$ is PL and depends upon the total arc load q_{ij}.

4 Mathematical Models

In this section we model the problem as an origin-destination integer multicommodity flow problem with PL costs, time windows, and side constraints over the digraph G defined in the previous section. We first introduce, in Section 4.1, the variables and constraints needed to model the PL cost functions. Here we discuss also properties relevant to our approach. We then present a node-arc formulation, $\mathcal{F}_1$, which is described in Section 4.2. However, since $\mathcal{F}_1$ results in very large integer programs, we devise a decomposition scheme using a path based formulation, $\mathcal{F}_2$, which is presented in Section 4.3. An approximate compact formulation, $\mathcal{F}_2\mathcal{L}$, is introduced in Section 4.4 and its relevance is discussed in Section 4.5. Finally, valid inequalities from the literature are adapted to both formulations $\mathcal{F}_1$ and $\mathcal{F}_2$ in Section 4.6.

4.1 Modeling the PL cost functions

We have in our problem $|A^{pl}|$ PL cost functions, possibly distinct. We indicate with $S_{ij} = \{1, ..., |S_{ij}|\}$ the set of linear segments of the cost function $g_{ij}, (i,j) \in A^{pl}$. Each segment $s \in S_{ij}$ has a variable cost or slope, c^s_{ij}, a non-negative fixed cost, f^s_{ij}, and the breakpoints, r^{s-1}_{ij} and r^s_{ij}. We denote by α^s_{ij} the slope of the line joining the origin with the point of segment maximum flow, i.e. $\alpha_{ij}^s = g_{ij}(r_{ij}^s)/r_{ij}^s$. We do not assume continuity, but we require that the function be lower semicontinuous: $g_{ij}(q) \leq \liminf_{q' \to q} g_{ij}(q')$. To a zero arc load corresponds a zero cost: $g_{ij}(0) = 0$. An illustration is provided in Figure 3, where, for notational simplicity, we drop the subscript ij. We use the so called *multiple choice model* (MCM) to represent these PL functions

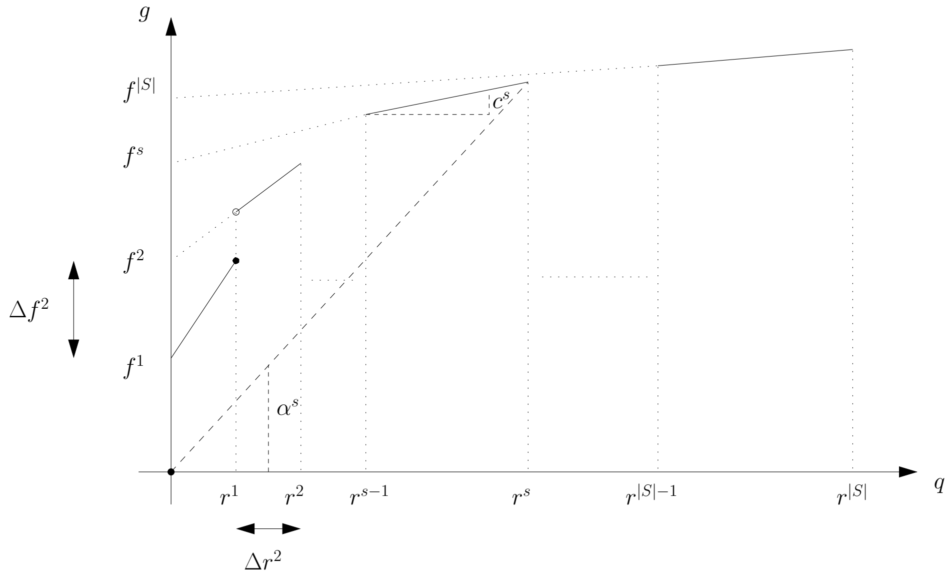

Figure 3: Notation of the PL cost function

(see, e.g., Croxton et al. (2003a)). The MCM employs the following variables:

- $l_{ij}^s \in \Re^+, \forall (i,j) \in A^{pl}, \forall s \in S_{ij}$, expresses the arc load on segment s; if $l_{ij}^s > 0$ implies $l_{ij}^u = 0, \forall u \in S_{ij} \setminus \{s\}$ and $r_{ij}^{s-1} \leq l_{ij}^s \leq r_{ij}^s$;
- $y_{ij}^s \in \{0, 1\}, \forall (i,j) \in A^{pl}, \forall s \in S_{ij}$, where $y_{ij}^s = 1$ if the arc load belongs to the segment s of the cost function g_{ij}; otherwise $y_{ij}^s = 0$.

Then, dropping the subscript ij, for a given arc load q, we obtain the following MCM mixed-integer linear formulation for $g(q)$:

$$\text{minimize} \sum_{s \in S} (c^s l^s + f^s y^s) \tag{1}$$

subject to

$$\sum_{s \in S} l^s \geq q \tag{2}$$

$$r^{s-1} y^s \leq l^s \leq r^s y^s \qquad \forall s \in S, \tag{3}$$

$$\sum_{s \in S} y^s \leq 1 \tag{4}$$

$$l^s \in \Re^+ \qquad \forall s \in S, \tag{5}$$

$$y^s \in \{0,\, 1\} \qquad \forall s \in S. \tag{6}$$

Because of binary requirement upon y^s variables and constraints (4) we must select only one segment. Constraints (2) and (3) link the arc load, q, and the choice of the segment s such that $r^{s-1} \leq q \leq r^s$. As a result, the objective function (1) expresses $g(q)$ by the fixed cost and slope of the chosen segment.

Croxton et al. (2003a) proved that a piecewise linear cost function will be approximated by its lower convex envelope when relaxing integrality constraints in the multiple choice model. The lower convex envelope of a function g is the greatest convex function majorized by g (Rockafellar (1970, p. 36)). Croxton et al. derived this result for a second formulation (the convex combination model), and then they extended it to the MCM and to a third MILP formulation by proving the equivalence of their LP relaxations. Here we discuss a particular case of this result. Croxton et al. observed that under concavity the lower convex envelope approximation assumes a linear form: it is the line joining the origin with the point of maximum feasible flow. We prove that this property holds under a milder assumption: i.e whenever the minimum α^s value is associated to the last segment.

Proposition 1 *Let $\hat{s}$ be such that $\alpha^{\hat{s}} = min_{s \in S} \alpha^s$. Then $\alpha^{\hat{s}} q$ is a lower bound for the LP relaxation of the MCM model. If $q \leq r^{\hat{s}}$ then $\alpha^{\hat{s}} q$ represents also the optimal solution.*

Proof — Since the binary requirement is relaxed and because of constraints (3), we will have $l^s / r^s \leq y^s \leq l^s / r^{s-1}$ in any feasible solution. Furthermore, the non-negative coefficients of y^s in the objective function (1) will lead, in an optimal solution, to the equality $y^s = l^s / r^s$. We can then project out the y^s variables. The objective function becomes $\sum_{s \in S} (c^s l^s + f^s l^s / r^s) = \sum_{s \in S} \alpha^s l^s$. Constraints (3) can be eliminated, constraint (2) and (5) hold, while constraint (4) can be expressed as

$$\sum_{s \in S} \frac{l^s}{r^s} \leq 1. \tag{7}$$

The resulting model is a continuous knapsack problem with α^s assignment costs in addition to constraint (7). It is immediate that $\alpha^{\hat{s}}q$ is a valid lower bound and, if the hypothesis $q \leq r^{\hat{s}}$ holds, then the assignment of the total arc flow to the segment $\hat{s}$ is feasible (constraint (7) is satisfied), and $\alpha^{\hat{s}}q$ is the optimum, thus proving the proposition. □

Assumption 1 *Assume that we have a PL cost function such that* $\alpha^{|S|} = min_{s \in S} \alpha^s$.

Observation 1 *Under Assumption 1 and a feasible flow* q*, the optimal cost of the linear relaxation of the MCM is equal to* $\alpha^{|S|}q$ *and an optimal solution can be characterized as follows: the selected segment is the last,* $l^{|S|} = q$*, and* $y^{|S|} = q/r^{|S|} > 0$*, while* $l^s = y^s = 0, \forall s \in \{1, ..., |S| - 1\}$.

Observation 2 *Under Assumption 1 and a feasible flow* q*, the optimal dual multiplier associated to the constraint (2) is equal to* $\alpha^{|S|}$.

Proof — Let (π, μ) be the pair of dual multipliers associated to constraint (2) and (7), respectively. Note that the pair $(\alpha^{|S|}, 0)$ is dual feasible and has a dual cost equal to $\alpha^{|S|}q$. Therefore, by duality theory it is also dual optimal. □

Cost functions related to consolidation generally satisfy Assumption 1. To clarify this statement we introduce a characterization of PL cost functions induced by consolidation. It is certainly legitimate to assume that the c^s values, segment slopes, are non-increasing in s. Furthermore, define the jumps of the fixed costs, Δf^s, and the length of the segments, Δr^s, as:

$$\Delta f^s = \begin{cases} f^s - f^{s-1} & \text{if } \ s \in \{2, ..., |S|\} \\ f^1 & \text{if } s = 1, \end{cases}$$

$$\Delta r^s = \begin{cases} r^s - r^{s-1} & \text{if } \ s \in \{2, ..., |S|\} \\ r^1 & \text{if } s = 1. \end{cases}$$

The length of the segments where price discounts apply generally do not decrease, while the fixed cost jumps do not increase, see Figure 3. This is summarized in the following Assumption:

Assumption 2 *We assume a PL cost function such that the c^s and Δf^s values are non-increasing in the index s, while the Δr^s series is non-decreasing.*

We now prove that the Assumption 2, motivated by consolidation legitimate features, implies Assumption 1, and is thus stronger.

Proposition 2 *Under Assumption 2, the α^s series is non-increasing.*

Proof — Because α^s is the sum of c^s and f^s/r^s terms and we use the non-increasing assumption on the c^s values, it remains to prove that the f^s/r^s series is non-increasing as well, i.e. $f^{s+1}/r^{s+1} \leq f^s/r^s, \forall s \in \{1, ..., |S|-1\}$. Observe that the non-increasing assumption on the Δf^s series and the non-decreasing Δr^s values, allow to write the following inequality:

$$\frac{f^{s+1}}{r^{s+1}} = \frac{f^s + \Delta f^{s+1}}{r^s + \Delta r^{s+1}} \leq \frac{f^s + \Delta f^s}{r^s + \Delta r^s}. \tag{8}$$

For $s = 1$ the inequality (8) becomes $f^2/r^2 \leq f^1/r^1$, as required. This leads to a proof by induction. Let $f^s/r^s \leq f^{s-1}/r^{s-1}$ be the induction hypothesis. It is easy to verify that the proposition is verified by using the induction hypothesis:

$$\frac{f^s}{r^s} \leq \frac{f^{s-1}}{r^{s-1}} \Rightarrow \frac{f^s + \Delta f^s}{r^s + \Delta r^s} = \frac{2f^s - f^{s-1}}{2r^s - r^{s-1}} \leq \frac{f^s}{r^s}. \tag{9}$$

The above implication can be proved by contradiction negating the last inequality in (9):

$$\frac{2f^s - f^{s-1}}{2r^s - r^{s-1}} > \frac{f^s}{r^s} \Rightarrow \frac{f^s}{r^s} > \frac{f^{s-1}}{r^{s-1}}. \tag{10}$$

□

In view of Proposition 2, we do not consider Assumption 1 to be limiting. The above results allow us to handle more general non-convex cost functions that are not necessarily concave, e.g. staircase functions, often found in practice.

4.2 Node-arc based formulation $\mathcal{F}_1$

The following additional notation is introduced to state the formulation. Let I be the set of intermediate nodes between origins and destinations, i.e. $I = N \setminus (O \cup D)$. For each node $i \in I$ the set $\delta(i)^+$ represents the nodes $j \in N$ such that $(i, j) \in A$. Similarly, $\delta(i)^-$ represents the nodes $j \in N$ such that $(j, i) \in A$.

The decision variables of the node-arc based M++RP formulation, $\mathcal{F}_1$, are:

- $x_{ij}^k \in \{0, 1\}, \forall (i,j) \in A, \forall k \in K$, where $x_{ij}^k = 1$ if shipment k uses arc (i,j), and $x_{ij}^k = 0$ otherwise;
- $T_i^k \in \Re^+, \forall k \in K, \forall i \in I$, indicates the arrival time of shipment k at node i; $T_i^k > 0$ if the node is visited by the shipment, otherwise $T_i^k = 0$.

Using this notation, the node-arc formulation can now be presented:

$$\text{minimize} \sum_{k \in K} \sum_{(i,j) \in A^v} c_{ij}^k x_{ij}^k + \sum_{(i,j) \in A^{pl}} \sum_{s \in S_{ij}} (c_{ij}^s l_{ij}^s + f_{ij}^s y_{ij}^s) \tag{11}$$

subject to

$$\sum_{j \in \Omega(k) \cup \{d^k\}} x_{o^k j}^k = 1 \qquad \forall k \in K, \tag{12}$$

$$\sum_{i \in \Gamma(k) \cup \{o^k\}} x_{id^k}^k = 1 \qquad \forall k \in K, \tag{13}$$

$$\sum_{j \in \delta^+(i)} x_{ij}^k - \sum_{j \in \delta^-(i)} x_{ji}^k = 0 \qquad \forall k \in K, \forall i \in I, \tag{14}$$

$$T_i^k + t_{ij}^k - T_j^k \leq (1 - x_{ij}^k) M_{ij}^k \qquad \forall k \in K,\ \forall (i,j) \in A \cap I \times I, \tag{15}$$

$$a^i \sum_{j \in \delta^+(i)} x_{ij}^k \leq T_i^k \leq b^i \sum_{j \in \delta^+(i)} x_{ij}^k \qquad \forall k \in K,\ \forall i \in N^{tw}, \tag{16}$$

$$T_i^k = d^i \sum_{j \in \delta^+(i)} x_{ij}^k \qquad \forall k \in K, \forall i \in N^d, \tag{17}$$

$$\sum_{s \in S_{ij}} l_{ij}^s \geq \sum_{k \in K} q_{ij}^k x_{ij}^k \qquad \forall (i,j) \in A^{pl}, \tag{18}$$

$$r_{ij}^{s-1} y_{ij}^s \leq l_{ij}^s \leq r_{ij}^s y_{ij}^s \qquad \forall (i,j) \in A^{pl}, \forall s \in S_{ij}, \tag{19}$$

$$\sum_{s \in S_{ij}} y_{ij}^s \leq 1 \qquad \forall (i,j) \in A^{pl}, \tag{20}$$

$$\sum_{k \in K} w_{ij}^k x_{ij}^k \leq W_{ij} \qquad \forall (i,j) \in A^{pl}, \tag{21}$$

$$T_i^k \in \Re^+ \qquad \forall k \in K, \forall i \in I, \tag{22}$$

$$l_{ij}^s \in \Re^+ \qquad \forall (i,j) \in A^{pl}, \forall s \in S_{ij}, \tag{23}$$

$$y_{ij}^s \in \{0, 1\} \qquad \forall (i,j) \in A^{pl}, \forall s \in S_{ij}, \tag{24}$$

$$x_{ij}^k \in \{0, 1\} \qquad \forall (i,j) \in A, k \in K. \tag{25}$$

Here M_{ij}^k assumes the value $b^i + t_{ij}^k$ if $i \in N^{tw}$, it is set equal to $d^i + t_{ij}^k$ if $i \in N^d$, and it is set equal to a sufficiently large number otherwise. The objective function (11) minimizes the total cost incurred by using arcs with a shipment specific cost, i.e. arcs belonging to A^v, and by using consolidation services, arcs in A^{pl}, were PL cost functions hold. Constraints (12) and (13) define the degree of the origin and destination nodes, respectively. Flow conservation for the remaining nodes is ensured by constraints (14). Propagation of time variables T_i^k is enforced by constraints (15), while time windows and timetables are enforced by constraints (16) and (17), respectively. Constraints (18) to (20) are similar to the MCM constraints (2) to (4). They link the arc load, $\sum_{k\in K} q_{ij}^k x_{ij}^k$, and the choice of the segment s of the arc cost function such that $r_{ij}^{s-1} \leq \sum_{k\in K} q_{ij}^k x_{ij}^k \leq r_{ij}^s$. Note that (19) is also useful to enforce capacity constraints over the quantities q_{ij}^k. In fact, the arc load $\sum_{k\in K} q_{ij}^k x_{ij}^k$ must not be greater than $r_{ij}^{|S_{ij}|}$. The resource consumption upper bounds over w_{ij}^k values are modeled by constraints (21).

4.3 Path based formulation $\mathcal{F}_2$

A path based formulation focuses on paths between the origin and the destination nodes over the digraph G. For each shipment $k \in K$, let $P(k)$ represent the set of all feasible paths on G. Each path must start from the origin node o^k, end at the destination node d^k and respect the time related constraints. Concisely, a path satisfies constraints (12) to (17), and (25). For each feasible path $p \in P(k)$, we define a binary decision variable z_p^k, where $z_p^k = 1$ if and only if shipment k is assigned to path p. Let $P = \bigcup_{k\in K} P(k)$. We also introduce the following notation:

- $\phi_{ij}^p, \forall (i,j) \in A, \forall p \in P$, ϕ_{ij}^p equals to one if the arc (i,j) belongs to the path p, and zero otherwise.
- $c_p^k = \sum_{(i,j)\in A^v} c_{ij}^k \phi_{ij}^p, \forall p \in P, k \in K$, thus c_p^k indicates the cost component of path p that is not affected by consolidation.

Hence, the path based formulation $\mathcal{F}_2$ is the following:

$$\text{minimize} \sum_{k\in K} \sum_{p\in P(k)} c_p^k z_p^k + \sum_{(i,j)\in A^{pl}} \sum_{s\in S_{ij}} (c_{ij}^s l_{ij}^s + f_{ij}^s y_{ij}^s) \tag{26}$$

subject to

$$\sum_{p\in P(k)} z_p^k = 1 \qquad \forall k \in K, \tag{27}$$

$$\sum_{s\in S_{ij}} l_{ij}^s \geq \sum_{k\in K}\sum_{p\in P(k)} q_{ij}^k \phi_{ij}^p z_p^k \qquad \forall (i,j) \in A^{pl}, \tag{28}$$

$$\sum_{k\in K}\sum_{p\in P(k)} w_{ij}^k \phi_{ij}^p z_p^k \leq W_{ij} \qquad \forall (i,j) \in A^{pl}, \tag{29}$$

$$r_{ij}^{s-1} y_{ij}^s \leq l_{ij}^s \leq r_{ij}^s y_{ij}^s \qquad \forall (i,j) \in A^{pl}, \forall s \in S_{ij}, \tag{30}$$

$$\sum_{s\in S_{ij}} y_{ij}^s \leq 1 \qquad \forall (i,j) \in A^{pl}, \tag{31}$$

$$l_{ij}^s \in \Re^+ \qquad \forall (i,j) \in A^{pl}, \forall s \in S_{ij}, \tag{32}$$

$$y_{ij}^s \in \{0,\,1\} \qquad \forall (i,j) \in A^{pl}, \forall s \in S_{ij}, \tag{33}$$

$$z_p^k \in \{0,\,1\} \qquad \forall k \in K, p \in P(k). \tag{34}$$

One must choose exactly one path for each shipment, as stated in constraints (27). Constraints (28) play the same role as constraints (18). Capacities over the w_{ij}^k values are modeled by constraints (29). Constraints (30) and (31) are similar to (19) and (20).

4.4 Approximated path based formulation $\mathcal{F}_2\mathcal{L}$

Here we introduce a simplified formulation where the PL cost functions are replaced by linear costs $\alpha_{ij} = \min_{s\in S_{ij}} \alpha_{ij}^s$. We call this formulation $\mathcal{F}_2\mathcal{L}$:

$$\text{minimize} \sum_{k\in K}\sum_{p\in P(k)} (c_p^k + \sum_{(i,j)\in A^{pl}} \alpha_{ij} q_{ij}^k \phi_{ij}^p) z_p^k \tag{35}$$

subject to

$$\sum_{p\in P(k)} z_p^k = 1 \qquad \forall k \in K, \tag{36}$$

$$\sum_{k\in K}\sum_{p\in P(k)} q_{ij}^k \phi_{ij}^p z_p^k \leq r_{ij}^{|S_{ij}|} \qquad \forall (i,j) \in A^{pl}, \tag{37}$$

$$\sum_{k\in K}\sum_{p\in P(k)} w_{ij}^k \phi_{ij}^p z_p^k \leq W_{ij} \qquad \forall (i,j) \in A^{pl}, \tag{38}$$

$$z_p^k \in \Re^+ \qquad \forall k \in K, p \in P(k). \tag{39}$$

Where constraints (37) define the upper bounds over the arc flows, and constraints (36) and (38) are similar to (27) and (29), respectively. The merits of $\mathcal{F}_2\mathcal{L}$ are discussed in the next section.

4.5 Lower bounds strength

Let LB_1 denote the lower bound obtained by relaxing the integrality constraints (24) and (25) in $\mathcal{F}_1$, while LB_2 is the the lower bound when relaxing (33) and (34) in $\mathcal{F}_2$. We indicate by V_{2L} the optimal solution value of $\mathcal{F}_2\mathcal{L}$. Afterwards, we present the two following propositions.

Proposition 3 $V_{2L} \leq LB_2$

Proof — Note that a feasible solution z for $\mathcal{F}_2\mathcal{L}$ is feasible for the $\mathcal{F}_2$ linear relaxation, and vice versa. The two objective functions (26) and (35) differ for the PL related costs. In (26) the PL costs derive from the MCM, whereas in (35) they are expressed as $\alpha_{ij}q_{ij}$ values. As noted in Proposition 1, $\alpha_{ij}q_{ij}$ is a lower bound of the corresponding relaxed MCM, thus proving the inequality. □

Proposition 4 *If* $\alpha_{ij} = \min_{s \in S_{ij}} \alpha_{ij}^s = \alpha_{ij}^{|S_{ij}|}, \forall (i,j) \in A^{pl}$, *then* $V_{2L} = LB_2 \geq LB_1$

Proof — The equality $V_{2L} = LB_2$ is a consequence of Proposition 3 and Observation 1. The inequality $LB_2 \geq LB_1$ derives from the decomposition choice. A path p satisfies constraints (12) to (17), and (25). This set of constraints does not exhibit the integrality property. In fact, the (12) to (17) polytope has a polynomial number of constraints while describing the feasible region of an $\mathcal{NP}$-hard problem, the SPPTW. Therefore (Geoffrion, 1974), we can expect that solving the LP relaxation of formulation $\mathcal{F}_2$ will yield tighter lower bounds than solving that of formulation $\mathcal{F}_1$. □

Since the formulation $\mathcal{F}_2\mathcal{L}$ is considerably more compact than $\mathcal{F}_2$, the previous result highlights the convenience of using the approximate formulation for bounding purposes. When Assumption 1 holds, the strength of the lower bound is not compromised by solving the compact formulation. Furthermore, $\mathcal{F}_2\mathcal{L}$ could also prove its usefulness when the above mentioned assumption does not apply. This would be the case when very large instances could not be tackled alternatively.

4.6 Valid inequalities

We adapt to formulations $\mathcal{F}_1$ and $\mathcal{F}_2$ the valid inequalities proposed by Croxton et al. (2007). We start with the valid inequalities for formulation $\mathcal{F}_1$. The *strong forcing constraints* state that when on a given arc no segment is chosen, the flow of each shipment is zero on that arc. With our notation:

$$x_{ij}^k \leq \sum_{s \in S_{ij}} y_{ij}^s \qquad \forall k \in K, \forall (i,j) \in A^{pl}. \tag{40}$$

The strong forcing constraints (40) aggregate segments while disaggregating shipments. Introducing additional non-negative variables x_{ij}^{ks} allows us to disaggregate both shipments and segments. These new variables are related to the previous ones by the following equations:

$$x_{ij}^k = \sum_{s \in S_{ij}} x_{ij}^{ks} \qquad \forall k \in K, \forall (i,j) \in A^{pl}, \tag{41}$$

$$l_{ij}^s \geq \sum_{k \in K} q_{ij}^k x_{ij}^{ks} \qquad \forall (i,j) \in A^{pl}, \forall s \in S_{ij}. \tag{42}$$

The *extended forcing constraints* can be stated as:

$$x_{ij}^{ks} \leq y_{ij}^s \qquad \forall (i,j) \in A^{pl}, \forall k \in K, \forall s \in S_{ij}. \tag{43}$$

We call $\mathcal{F}_1\mathcal{S}$ the formulation obtained by adding constraints (40) to $\mathcal{F}_1$, while $\mathcal{F}_1\mathcal{E}$ is the formulation incorporating constraints (41) to (43) and the variables x_{ij}^{ks}.

The equivalent strengthened $\mathcal{F}_2$ formulation, $\mathcal{F}_2\mathcal{S}$, would require constraints (40) to be reformulated in the z variables:

$$\sum_{p \in P(k)} \phi_{ij}^p z_p^k - \sum_{s \in S_{ij}} y_{ij}^s \leq 0 \qquad \forall k \in K, \forall (i,j) \in A^{pl}. \tag{44}$$

Croxton et al. (2007) proved that the extended forcing constraints describe the convex hull of the Lagrangean subproblem when relaxing flow conservation constraints in network flow problems with piecewise linear costs. Frangioni and Gendron (2007) established the equivalence between extended forcing constraints and residual capacity inequalities. However, computational experiments of Croxton et al. (2007) indicate that when there are relatively large initial fixed costs the extended forcing constraints do not significantly improve upon the strong forcing ones.

5 Column Generation Algorithms

The column generation approach solves a large linear program (LP) without explicitly including all columns, i.e. variables, in the constraint matrix. The full problem is called the *master problem* (MP), while the LP with only a subset of the MP columns is called the *restricted master problem* (RMP). In most problems, only a very small subset of all columns will belong to an optimal solution, and all other (non-basic) columns can be ignored. In a minimization problem all columns with positive reduced cost can in fact be ignored. The column generation algorithm finds an optimal solution to the MP by solving a series of several smaller RMPs. When the optimal solution of an RMP is found, the algorithm looks for columns of negative reduced cost not included in the RMP. This is called the pricing problem. If no column can be found by the pricing routine, then the current optimal solution of the RMP is also optimal for the MP. Otherwise, one or more columns with negative reduced costs are added to the RMP and the algorithm iterates.

We present three algorithms to obtain lower bounds for our problem:

- CG solves the linear relaxation of $\mathcal{F}_2$ by column generation;
- CGL solves $\mathcal{F}_2\mathcal{L}$ by column generation;
- CGS solves the linear relaxation of $\mathcal{F}_2\mathcal{S}$ by row and column generation.

Finally, feasibility is reached by a branch-and-cut algorithm that chooses paths between those generated by one of these three algorithms.

5.1 $\mathcal{F}_2$ linear relaxation, algorithm CG

We devise the following pricing routine to find paths, i.e. columns, with negative reduced cost for the formulation $\mathcal{F}_2$. Let $\sigma_k, \forall k \in K$, $\pi_{ij} \geq 0, \forall (i,j) \in A^{pl}$, and $\lambda_{ij} \leq 0, \forall (i,j) \in A^{pl}$ be dual multipliers associated with constraints (27), (28), and (29) respectively. The reduced cost of the variable z_p^k, denoted as $\bar{c}_p^k$, is:

$$\bar{c}_p^k = \sum_{(i,j)\in A^v} c_{ij}^k \phi_{ij}^p - \sum_{(i,j)\in A^{pl}} \phi_{ij}^p (w_{ij}^k \lambda_{ij} - q_{ij}^k \pi_{ij}) - \sigma_k. \tag{45}$$

We introduce modified arc costs, $\tilde{c}^k_{ij}$, over the digraph G, as follows:

$$\tilde{c}^k_{ij} = \begin{cases} q^k_{ij}\pi_{ij} - w^k_{ij}\lambda_{ij} & \text{if } (i,j) \in A^{pl}, \\ c^k_{ij} & \text{otherwise.} \end{cases}$$

Given this modified cost structure, every path $p \in P(k)$ will have a cost $\tilde{c}^k_p$ equal to $\bar{c}^k_p + \sigma_k$. Therefore, in order to find a column with a negative reduced cost we look for a path p such that $\tilde{c}^k_p < \sigma_k$. This can be accomplished by solving for each shipment a SPPTW over the digraph G with the cost structure modified by the current dual multipliers. Observe that the modified arc costs are non-negative. The pricing algorithm is detailed in Section 5.4.

At the first iteration of the algorithm we insert in the $\mathcal{F}_2$ formulation the paths (o^k, d^k), i.e. the direct origin-destination arcs. This brings noteworthy advantages: we avoid using artificial variables with "big M" costs to initialize the process, and the corresponding first LP is immediately solved, thus providing "good" dual values. Therefore, at the first iteration the σ_k dual multipliers assume value $c_{o^k d^k}$, while the other dual multipliers associated to PL related constraints are equal to zero because these (o^k, d^k) paths do not use consolidation arcs. Consequently, the pricing algorithm looks for paths that cost less than the direct origin-destination arcs. However, it uses a modified digraph cost structure where the arcs with PL cost functions are seen as arcs at zero cost. As we will see in the next section, this undesiderable behavior is intrinsically avoided by algorithm CGL.

5.2 Solving $\mathcal{F}_2\mathcal{L}$, algorithm CGL

The CGL algorithm uses a modified pricing problem. Let $\eta_{ij} \leq 0, \forall (i,j) \in A^{pl}$ be dual multipliers associated with constraints (37). These dual variables, together with the λ and σ introduced earlier, allow us to write the reduced cost of the variable z^k_p, $\bar{c}^k_p$, in the formulation $\mathcal{F}_2\mathcal{L}$, as:

$$\bar{c}^k_p = \sum_{(i,j)\in A^v} c^k_{ij}\phi^p_{ij} - \sum_{(i,j)\in A^{pl}} \phi^p_{ij}(q^k_{ij}(\eta_{ij} - \alpha_{ij}) + w^k_{ij}\lambda_{ij}) - \sigma_k. \tag{46}$$

Let the dual modified arc costs, $\tilde{c}^k_{ij}$, be:

$$\tilde{c}^k_{ij} = \begin{cases} -q^k_{ij}(\eta_{ij} - \alpha_{ij}) - w^k_{ij}\lambda_{ij} & \text{if } (i,j) \in A^{pl}, \\ c^k_{ij} & \text{otherwise.} \end{cases}$$

Similarly to the previously defined pricing scheme in CG, we use an SPPTW algorithm to find paths p such that $\tilde{c}_p^k < \sigma_k$, i.e. columns with negative reduced cost. Observe the difference between the dual modified digraph cost structure of CGL respect to the one of CG. In CGL we have $\tilde{c}_{ij}^k \geq q_{ij}^k \alpha_{ij}, \forall (i,j) \in A^{pl}$, and the equality is attained when the η and λ dual multipliers are equal to zero. As discussed in the previous section, this happens at the first iteration of the column generation. Here the pricing phase of CGL uses a more realistic cost structure. In fact, recalling Observation 2, it is clear that when dealing with PL cost functions that satisfy Assumption 1, the π_{ij} dual multipliers in CG converge to α_{ij}. Therefore, CGL, where the α_{ij} values play a role since the beginning, has a faster overall convergence. The advantage of $\mathcal{F}_2\mathcal{L}$ is obvious: it implicitly exploits the knowledge about the optimal dual multipliers π_{ij}. This fact leads, together with the smaller size of the LPs, to the better performance of CGL compared to CG, as we will show in our computation experiments.

5.3 $\mathcal{F}_2\mathcal{S}$ linear relaxation, algorithm CGS

Solving the $\mathcal{F}_2\mathcal{S}$ linear relaxation requires larger LPs than in CG because of constraints (44). These constraints, when dealing with very large instances, cannot all be introduced because of excessive memory requirements. Therefore, we resort to a column and row generation algorithm, where the generated rows are the violated constraints (44). The algorithm CGS consists of two phases. During the first phase, CGS-P1, row generation does not occur. CGS-P1 consists of an improved version of the algorithm CG. The algorithm CGS-P1 differs from CG by exploiting the knowledge about the optimal dual variables π, i.e. the π_{ij} multipliers are set to α_{ij}. In addition, the algorithm CGS-P1 is useful to assess the improvement due to the dual information. At the end of CGS-P1 the second phase, CGS-P2, starts and violated inequalities are searched by complete enumeration and appended, if they exist, to the current model. Afterward, a new column generation procedure, which we now outline, is called. The algorithm iterates until no new violated inequalities are found. The column generation procedure for CGS-P2 differs from the one of CG by considering the dual multipliers associated with generated constraints, i.e. a subset of (44). We indicate these dual multipliers by $\nu_{ij}^k \leq 0, \forall k \in K, \forall (i,j) \in A^{pl}$. Consequently, the reduced cost of the variable z_p^k, $\bar{c}_p^k$, are:

$$\bar{c}_p^k = \sum_{(i,j)\in A^v} c_{ij}^k \phi_{ij}^p - \sum_{(i,j)\in A^{pl}} \phi_{ij}^p (w_{ij}^k \lambda_{ij} - q_{ij}^k \pi_{ij} + \nu_{ij}^k) - \sigma_k. \tag{47}$$

The modified arc costs, $\tilde{c}^k_{ij}$, can be expressed as:

$$\tilde{c}^k_{ij} = \begin{cases} q^k_{ij}\pi_{ij} - w^k_{ij}\lambda_{ij} - \nu^k_{ij} & \text{if } (i,j) \in A^{pl}, \\ c^k_{ij} & \text{otherwise.} \end{cases}$$

5.4 Solving the pricing problem

This section describes how the pricing problems in our column generation algorithms are solved. As described earlier, because of our representation of timetables as collapsed time windows, our pricing problem is an SPPTW. We are given a directed graph where each arc has an associated cost c_{ij} and travel time t_{ij}. Each node i has a time window $[a^i, b^i]$. We are allowed to arrive to a node i before a^i, but we must wait until time a^i before the node can be processed and the node cannot be visited after b^i. The objective of the SPPTW is to construct a minimum cost path (with respect to the arc costs c_{ij}) from a start node o^k to an end node d^k such that time windows are respected. We assume that travel times are non-negative, but in general we do not make assumptions on the travel costs (they could be negative). Negative travel costs can lead to cycles in the optimal shortest path but as long as the travel time on all cycles are positive the optimal shortest path is finite. In the shortest path problem considered in this paper we know that all arc costs are non-negative. This means that the shortest paths returned from an SPPTW algorithm are without loops. Consequently nothing is gained by considering the more difficult elementary shortest path algorithms.

Labeling algorithms for solving the SPPTW are described by Desrochers and Soumis (1988), Feillet et al. (2004), and Irnich and Desaulniers (2005). Such algorithms build partial paths starting from the start-node s. Each partial path $(o^k, i_1, \ldots, i_n)$ is represented by a label $[i_n, c, t, h]$ where i_n is the end node of the partial path, c is the cost of the partial path, t is the arrival time at node i_n, and h is a pointer to the parent label (the label corresponding to the path $(o^k, i_1, \ldots, i_{n-1})$). The algorithm maintains two sets of labels: *processed* and *unprocessed* labels. The algorithm starts with a single label $[o^k, 0, 0, \text{NULL}]$ in the set of unprocessed labels. At each iteration the algorithm considers a label $[i, c, t, h]$ from the set of unprocessed labels, and this label is extended by considering all arcs originating in node i. Extending the label to node j results in the label $[j, c + c_{ij}, \max\{t + t_{ij}, a^j\}, h']$ where h' is a pointer to the original label. If $\max\{t + t_{ij}, a^j\} > b^j$ then the label is discarded as it corresponds to a partial path that violates the time window. The new generated labels are moved to the set of unprocessed labels while the label we extended are moved to the set of processed labels. The algorithm terminates when the set of unprocessed labels is empty. In this case the label corresponding to the shortest path from o^k to d^k is the label associated with node d^k that has the lowest cost. The algorithm described so far enumerates all feasible paths. To speed up the algorithm *dominance rules* are introduced. If we have two labels $[i, c, t, h]$ and $[i', c', t', h']$ such that $i = i'$, $c \leq c'$ and $t \leq t'$ then the second label is dominated and can be discarded. In fact, all possible ways of completing the partial path corresponding to the second label can be used to complete the partial path corresponding to the first label. As a consequence, the partial path corresponding to the first label leads to completions with cost at least as good as the completions of the partial path associated to the second label.

In this paper we use the bidirectional algorithm proposed by Righini and Salani (2006). This algorithm improves the algorithm outlined above by extending paths from both the start node o^k and the end node d^k simultaneously. Informally speaking, the algorithm stops when the paths from the start node meets with paths from the end node. The algorithm is described in detail by Righini and Salani (2006). It provides significant speedups compared with the unidirectional algorithm that only extends paths from the start node (see results in Righini and Salani (2006)).

In each iteration of the column generation process we are interested in generating variables (columns, paths) with negative reduced costs. It is not necessary to generate the variable that has the most negative reduced cost. Consequently we are free to use heuristics for generating the variables. We only need to solve the pricing problem to optimality to prove that no variable with negative reduced cost exists, that is, the exact algorithm needs only be called when the heuristic algorithm no longer finds variables with negative reduced costs. It is well known (e.g., Irnich and Desaulniers (2005)) that the labeling algorithm for the shortest path problem easily can be turned into a heuristic. However, the pricing problems that we encounter are all relatively easy and therefore, for simplicity, we always use the exact algorithm.

5.5 Heuristics to obtain upper bounds, algorithms H-CG, H-CGL, H-CGS

The column generation algorithms outlined above will usually yield a fractional solution, meaning that some shipments use more than one path. We propose a very simple heuristic scheme aimed at obtaining a feasible solution. It consists of performing a time limited branch-and-cut search on the set of generated columns. This amounts to solving the final RMP as an integer program. By doing so, we turn the column generation algorithms into heuristics. We use a state-of-the-art MILP solver for this branch-and-cut algorithm. When reporting computational results we will indicate these heuristics as H-CG, H-CGL, H-CGS. Note that the branch-and-cut for H-CGL is initiated by loading the generated columns into the $\mathcal{F}_2$ formulation, since $\mathcal{F}_2\mathcal{L}$ is not equivalent to $\mathcal{F}_2$ when converted to an integer program.

In general the outlined heuristic approach fails because there is no guarantee of integer feasibility in the RMP. However, in our problem it is always possible to get a feasible integer solution because of the direct origin-destination arc for each shipment. As described previously, these paths are inserted at the beginning of the column generation, and hence they belong to the set of paths on which the branch-and-cut algorithm is applied. Therefore, their usefulness is twofold: they ensure faster convergence when computing a lower bound (by avoiding the "big M" to initialize the process), and they guarantee integer feasibility when looking heuristically for an upper bound.

6 Computational Results

We now present computational experiments. We first describe how test instances were generated, and we then provide results obtained with the various algorithms.

6.1 Generation of test instances

As mentioned in Section 1, this study was motivated by a real-life application. The largest customer requires shipments from 10 factories located in Northern Italy to 10 regional distribution warehouses in Central-South Italy. The planning horizon is two weeks and 28 block trains can be activated. There are five train stations where loading can take place and four arrival stations where these block trains end. Once arrived at one of these four stations the shipment can be sent to its final destination by truck. Two of the four arrival stations offer the opportunity to re-route the shipment by rail to seven other train stations. This additional set of railway links has a cost structure similar to a dedicated service, i.e. there are no quantity discounts and the cost is per railcar. The available mode and consolidation combinations yield six alternatives:

1. dedicated origin-destination service by truck;
2. consolidated service by truck to an intermediate platform; final link by dedicated truck service;

3. consolidated service by truck to a loading train station; consolidation by block train to arrival station; final link by dedicated truck service;

4. as option three, but with an additional step before the last link: a dedicated rail service between two compatible train stations;

5. as option three, but with dedicated truck service between factory and loading train station;

6. dedicated truck service between factory and loading train station; consolidation by block train to intermediate station; dedicated rail service to arrival station; final link by dedicated truck service.

With 122 shipments in two weeks, pickup and delivery time windows, virtual nodes and arcs to represent mode or vehicle transfers, etc. the resulting digraph has a rather large size: $|N| = 1507, |A| = 4900$, and $|A^{pl}| = 2094$. Instances of this size are clearly beyond the capabilities of exact approaches. To test the efficacy of our algorithms we created smaller instances that are subsets of the real-life one with $|K|$ equal to 10, 30, and 60. The instance set contains three instances for each choice of $|K|$, labeled as i-10-01, i-10-02, i-10-03, etc. and the realistic sized instance denoted as i-122-01.

6.2 Implementation details and results

The algorithms were implemented in C++ using CPLEX 10.0. Computational experiments were run on a 2.5 GHz Pentium IV Linux server with 4 GB of memory. When reporting the experiments with the three formulations $\mathcal{F}_1$, $\mathcal{F}_1\mathcal{S}$, and $\mathcal{F}_1\mathcal{E}$ implemented in CPLEX we will denote their results by the name of the formulation, i.e. $\mathcal{F}_1$ will indicate the formulation as well as its CPLEX implementation. We have not tweaked the CPLEX parameters. We have only set time limits: 10 hours for the $\mathcal{F}_1$, $\mathcal{F}_1\mathcal{S}$, and $\mathcal{F}_1\mathcal{E}$ algorithms, 15 minutes for H-CG2 and H-CGL, and 30 minutes for H-CGS.

We first compare in Table 1 results by $\mathcal{F}_1$ and H-CGL. H-CGL clearly outperforms $\mathcal{F}_1$ by taking considerably less computational time while obtaining better upper bounds than the 10 hours of the truncated CPLEX branch-and-cut upon $\mathcal{F}_1$. Note that whenever $\mathcal{F}_1$ requires less than the time limit, it means that an optimal solution has been found. On these instances, solved to optimality, H-CGL obtains the same solution value, but in a shorter time. Results on the real-life i-122-01 instance are worth a remark: H-CGL provides a 10 percentage points better solution by using three orders of magnitude less computational time than $\mathcal{F}_1$.

Table 2 clarifies why we have chosen to benchmark H-CGL with $\mathcal{F}_1$. This formulation is the only one able to handle the large instance with $|K| = 122$. In fact, because of memory limits $\mathcal{F}_1\mathcal{S}$ manages to load instances up to $|K| = 60$ while the heaviest $\mathcal{F}_1\mathcal{E}$ stops at $|K| = 30$. While examining the lower bounds obtained at the root nodes (results that we do not report here), $\mathcal{F}_1\mathcal{S}$ improves over $\mathcal{F}_1$ but with a slightly longer computational time. As Table 2 shows, the better bounding capabilities of the strong forcing constraints do not uniformly guarantee better overall performance, due to slower node examination. In spite of their theoretical strength, the extended forcing constraints in $\mathcal{F}_1\mathcal{E}$ are disappointing: they yield a lower bound improvement similar to that of the strong forcing constraints but with computational time larger by an order of magnitude. Consequently, $\mathcal{F}_1\mathcal{E}$ results are dominated by those of $\mathcal{F}_1$ and $\mathcal{F}_1\mathcal{S}$. Customized branch-and-cut algorithms based on the discussed valid inequalities would have been more competitive than our straightforward implementation. However, results on the small instances, $|K| = 30$, indicate that the lower bound improvement is not enough to compensate the additional computational burden. Furthermore, this improvement is already captured by the more compact strong forcing constraints. In fact, Croxton et al. (2007) report that when there are relatively large initial fixed costs the extended formulation does not significantly improve upon the strong one. This is so in our case and it explains why we did implement a column and row generation algorithm based upon the strong forcing constraints only.

We now discuss the advantages of using the $\mathcal{F}_2\mathcal{L}$ formulation to compute lower bounds. Table 3 reports the computational times of three algorithms: CG, CGS-P1, and CGL. These three algorithms compute the same lower bound. As expected, CG is the slowest. The algorithm CGS-P1 improves in average with respect to CG because it exploits the knowledge about the optimal π dual values. The algorithm CGL considerably improves with respect to CGS-P1 because of the compactness of the corresponding formulation. These two combined effects lead to CGL being 40 times faster in average than CG.

The performance of the three column generation based heuristics are summarized in Table 4. The merits of the more compact $\mathcal{F}_2\mathcal{L}$ formulation can be further appreciated. The algorithm H-CGL is better not only because CGL has a faster column generation convergence. In addition, the heuristic phase, i.e. the branch-and-cut search upon the set of generated columns, is more effective in H-CGL. This happens because H-CG receives a larger set of columns from CG than

H-CGL does from CGL. However, these additional paths are not useful, because, as observed in Section 5.1, they are generated with a zero cost estimate of the PL functions at the beginning of the CG algorithm. The H-CGS algorithm corrects this unfavorable characteristic. However, it does not obtain appreciably better results, even within an extended time limit of 30 minutes.

Table 5 assesses the merit of the pricing algorithm. We indicate by H-CGL-LSA1 a modification of the H-CGL algorithm where the label setting routine returns only one path, if any. The computational results indicate that this modification considerably degrades solution quality although it yields a faster algorithm. The algorithm H-CGL-SPCPLEX replaces the label setting routine by a MILP model solved by CPLEX. Here both solution quality and computational time worsen. Moreover, memory limit is reached by the medium and large instances.

The algorithm CGS proves its usefulness when computing lower bounds. In Table 6 we report lower bounds provided by $\mathcal{F}_1$, $\mathcal{F}_1\mathcal{S}$ and $\mathcal{F}_1\mathcal{E}$ that are obtained at the termination of the branch-and-cut algorithms. The computational times to compute these lower bounds are the ones of Table 2. The last two columns of Table 6 provide the results of the CGS algorithm. In spite of the relatively small computational times, the lower bounds provided by CGS are of high quality, since the algorithm obtains three times the best lower bound. Finally, using these lower bounds, the solution quality of the heuristic can be assessed. In the worst case the solution values produced by using H-CGL lie within a few percentage points from optimality.

7 Conclusions

We have described, formulated and solved a new and feature-rich routing problem. The construction of an appropriate network representation turned out being non-trivial. We have devised a solution approach that exploits specific problem characteristics like the cost function properties and the realistic upper bounds upon feasible paths. Computational experiments showed the efficacy of the proposed heuristic algorithm based on decomposition.

8 Acknowledgements

This work was partly supported by the MUR (Italy) under project PILOT, and by the Natural Sciences and Engineering Council of Canada under grants 227837-00 and 39682-05. These supports are gratefully acknowledged. Thanks are also due to Manlio Gaudioso, M. Flavia Monaco, and Gabriella Messina for fruitful discussions.

References

Aldaihani, M. and Dessouky, M. M. (2003). Hybrid scheduling methods for paratransit operations. *Computers & Industrial Engineering*, 45(1):75–96.

Amiri, A. and Pirkul, H. (1997). New formulation and relaxation to solve a concave-cost network flow problem. *Journal of the Operational Research Society*, 48:278–287.

Barnhart, C., Hane, C. A., and Vance, P. H. (2000). Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research*, 48(2):318–326.

Barnhart, C. and Ratliff, H. D. (1993). Modeling intermodal routing. *Journal of Business Logistics*, 14(1):205 – 223.

Boardman, B. S., Malstrom, E. M., Butler, D. P., and Cole, M. H. (1997). Computer assisted routing of intermodal shipments. *Computers & Industrial Engineering*, 33(1-2):311–314.

Bontekoning, Y. M., Macharis, C., and Trip, J. J. (2004). Is a new applied transportation research field emerging?–A review of intermodal rail-truck freight transport literature. *Transportation Research Part A: Policy and Practice*, 38(1):1–34.

Caramia, M., Dell'Olmo, P., Gentili, M., and Mirchandani, P. B. (2007). Delivery itineraries and distribution capacity of a freight network with time slots. *Computers & Operations Research*, 34(6):1585–1600.

Chang, T.-S. (2007). Best routes selection in international intermodal networks. *Computers & Operations Research*, In Press, Corrected Proof.

Crainic, T. G. (2000). Service network design in freight transportation. *European Journal of Operational Research*, 122(2):272 – 288.

Crainic, T. G., Florian, M., Guélat, J., and Spiess, H. (1990). Strategic planning of freight transportation: STAN, an interactive-graphic system. *Transportation Research Record*, 1283:97–124.

Crainic, T. G., Gendron, B., and Hernu, G. (2004). A slope scaling/Lagrangean perturbation heuristic with long-term memory for multicommodity capacitated fixed-charge network design. *Journal of Heuristics*, 10(5):525–545.

Crainic, T. G. and Kim, K. H. (2006). Intermodal transportation. In Barnhart, C. and Laporte, G., editors, *Transportation*, volume 14 of *Handbooks in Operations Research and Management Science*, chapter 8, pages 467–537. Elsevier.

Crainic, T. G. and Laporte, G. (1997). Planning models for freight transportation. *European Journal of Operational Research*, 97(3):409 – 438.

Croxton, K. L., Gendron, B., and Magnanti, T. L. (2003a). A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems. *Management Science*, 49(9):1268 – 1273.

Croxton, K. L., Gendron, B., and Magnanti, T. L. (2003b). Models and methods for merge-in-transit operations. *Transportation Science*, 37(1):1 – 21.

Croxton, K. L., Gendron, B., and Magnanti, T. L. (2007). Variable disaggregation in network flow problems with piecewise linear costs. *Operations Research*, 55(1):146 – 157.

Desaulniers, G., Desrosiers, J., and Solomon, M. M., editors (2005). *Column Generation*. Springer, Boston.

Desrochers, M. and Soumis, F. (1988). A generalized permanent labeling algorithm for the shortest path problem with time windows. *INFOR*, 26:191–212.

Feillet, D., Dejax, P., Gendreau, M., and Gueguen, C. (2004). An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, 44:216–229.

Frangioni, A. and Gendron, B. (2007). 0-1 reformulations of the multicommodity capacitated network design problem. Technical Report 29, CIRRELT — Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport.

Gendron, B., Crainic, T., and Frangioni, A. (1998). Multicommodity capacitated network design. In Sansò, B. and Soriano, P., editors, *Telecommunications Network Planning*, pages 1–19. Kluwer, Boston.

Geoffrion, A. M. (1974). Lagrangean relaxation for integer programming. *Mathematical Programming Study*, 2:82–113.

Guélat, J., Florian, M., and Crainic, T. G. (1990). A multimode multiproduct network assignment model for strategic planning of freight flows. *Transportation Science*, 24(1):25 – 39.

Ibaraki, T., Imahori, S., Kubo, M., Masuda, T., Uno, T., and Yagiura, M. (2005). Effective local search algorithms for routing and scheduling problems with general time-window constraints. *Transportation Science*, 39(2):206–232.

Irnich, S. and Desaulniers, G. (2005). Shortest path problems with resource constraints. In Desaulniers, G., Desrosiers, J., and Solomon, M. M., editors, *Column Generation*, chapter 2, pages 33–65. Springer, Boston.

Jourquin, B., Beuthe, M., and Demilie, C. L. (1999). Freight bundling network models: Methodology and application. *Transportation Planning and Technology*, 23:157–177.

Keha, A. B., de Farias, I. R., and Nemhauser, G. L. (2004). Models for representing piecewise linear cost functions. *Operations Research Letters*, 32(1):44–48.

Keha, A. B., de Farias, I. R., and Nemhauser, G. L. (2006). A branch-and-cut algorithm without binary variables for nonconvex piecewise linear optimization. *Operations Research*, 54(5):847–858.

Kim, D. and Pardalos, P. M. (1999). A solution approach to the fixed charge network flow problem using a dynamic slope scaling procedure. *Operations Research Letters*, 24(4):195–203.

Kim, D. and Pardalos, P. M. (2000). Dynamic slope scaling and trust interval techniques for solving concave piecewise linear network flow problems. *Networks*, 35(3):216–222.

Macharis, C. and Bontekoning, Y. (2004). Opportunities for OR in intermodal freight transport research: A review. *European Journal of Operational Research*, 153(2):400 – 416.

Min, H. (1991). International intermodal choices via chance-constrained goal programming. *Transportation Research Part A: Policy and Practice*, 25(6):351–362.

Righini, G. and Salani, M. (2006). Symmetry helps: bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. *Discrete Optimization*, 3(3):255–273.

Rockafellar, R. T. (1970). *Convex Analysis*. Princeton University Press, Princeton.

Southworth, F. and Peterson, B. E. (2000). Intermodal and international freight network modeling. *Transportation Research Part C: Emerging Technologies*, 8(1-6):147–166.

Toth, P. and Vigo, D. (2002). *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia.

Xu, H., Chen, Z.-L., Rajagopal, S., and Arunapuram, S. (2003). Solving a practical pickup and delivery problem. *Transportation Science*, 37(3):347–364.

Ziliaskopoulos, A. and Wardell, W. (2000). An intermodal optimum path algorithm for multimodal networks with dynamic arc travel times and switching delays. *European Journal of Operational Research*, 125(3):486 – 502.

	H-CGL		$\mathcal{F}_1$	
	Sol. Value	Time (min.)	Sol. Value	Time (min.)
i-10-01	**100.0**	**0.2**	**100.0**	600.0
i-10-02	**100.0**	**0.1**	**100.0**	2.5
i-10-03	**100.0**	**0.4**	**100.0**	2.6
i-30-01	**100.0**	**0.1**	136.3	600.0
i-30-02	**100.0**	**0.2**	100.9	600.0
i-30-03	**100.0**	**15.0**	**100.0**	246.7
i-60-01	**100.0**	**0.5**	113.6	600.1
i-60-02	**100.0**	**2.5**	113.6	600.1
i-60-03	**100.0**	**15.0**	100.3	600.3
i-122-01	**100.0**	**0.7**	110.0	600.4
Average	**100.0**	**3.5**	*107.5*	445.3

Table 1: Computational results with $\mathcal{F}_1$ and H-CGL. The solution values are scaled to 100, and bold entries correspond to the best entry for each row, for computational time as well as solution quality.

	$\mathcal{F}_1$		$\mathcal{F}_1\mathcal{S}$		$\mathcal{F}_1\mathcal{E}$	
	Sol. Value	Time (min.)	Sol. Value	Time (min.)	Sol. Value	Time (min.)
i-10-01	100.0	600	100.0	18	100.0	600
i-10-02	100.0	3	100.0	1	100.0	43
i-10-03	100.0	3	100.0	4	100.0	30
i-30-01	100.0	600	100.0	600	100.0	600
i-30-02	100.0	600	105.0	600	152.3	600
i-30-03	100.0	247	100.0	600	100.0	600
i-60-01	100.0	600	100.0	600	n.a.	n.a.
i-60-02	100.0	600	105.6	600	n.a.	n.a.
i-60-03	100.5	600	100.0	600	n.a.	n.a.
Average	*100.1*	*428*	*101.2*	*403*		

Table 2: Computational results with $\mathcal{F}_1$, $\mathcal{F}_1\mathcal{S}$ and $\mathcal{F}_1\mathcal{E}$. The solution values are scaled to 100.

	CG (sec.)	CGS-P1 (sec.)	CGL (sec.)	CG/CGS-P1	CGS-P1/CGL	CG/CGL
i-10-01	7.8	4.0	0.1	2.0	28.3	55.5
i-10-02	10.9	7.0	0.2	1.5	36.9	57.2
i-10-03	57.4	48.7	0.7	1.2	70.6	83.1
i-30-01	26.7	23.9	1.1	1.1	22.5	25.2
i-30-02	39.2	34.6	1.6	1.1	21.6	24.5
i-30-03	141.7	223.5	12.2	0.6	18.3	11.6
i-60-01	116.7	85.2	3.2	1.4	26.6	36.4
i-60-02	167.2	128.5	5.2	1.3	24.7	32.1
i-60-03	541.2	843.5	29.8	0.6	28.3	18.1
i-122-01	667.7	548.2	12.0	1.2	45.7	55.6
Average				*1.2*	*32.3*	*39.9*

Table 3: Computational times of three column generation algorithms: CG, CGS-P1, CGL.

	H-CGL		H-CG		H-CGS	
	Sol. Value	Time (min.)	Sol. Value	Time (min.)	Sol. Value	Time (min.)
i-10-01	**100.0**	**0.2**	**100.0**	15.0	**100.0**	28.2
i-10-02	**100.0**	**0.1**	**100.0**	3.3	**100.0**	3.8
i-10-03	**100.0**	**0.4**	**100.0**	15.0	**100.0**	30.0
i-30-01	**100.0**	**0.1**	100.4	15.0	**100.0**	7.9
i-30-02	**100.0**	**0.2**	**100.0**	15.0	**100.0**	4.0
i-30-03	**100.0**	**15.0**	**100.0**	15.0	**100.0**	30.0
i-60-01	**100.0**	**0.5**	101.1	15.0	100.4	30.0
i-60-02	100.1	**2.5**	100.4	15.0	**100.0**	30.0
i-60-03	**100.0**	**15.0**	102.1	15.0	106.2	30.0
i-122-01	**100.0**	**0.7**	104.0	15.0	102.8	30.0
Average	**100.0**	**3.5**	*100.8*	*13.8*	*100.9*	*22.4*

Table 4: Computational results with the three heuristic algorithms. The solution values are scaled to 100, and bold entries correspond to the best entry for each row, for computational time as well as solution quality.

	H-CGL		H-CGL-LSA1		H-CGL-SPCPLEX	
	Sol. Value	Time (min.)	Sol. Value	Time (min.)	Sol. Value	Time (min.)
i-10-01	**100.0**	0.2	104.8	**0.01**	101.0	0.6
i-10-02	**100.0**	0.1	101.9	**0.01**	101.9	0.8
i-10-03	**100.0**	0.4	104.2	**0.02**	101.1	4.5
i-30-01	**100.0**	0.1	116.6	**0.02**	112.0	10.7
i-30-02	**100.0**	0.2	112.7	**0.02**	n.a.	10.0
i-30-03	**100.0**	15.0	116.8	**0.05**	n.a.	10.0
i-60-01	**100.0**	0.5	104.7	**0.07**	n.a.	10.0
i-60-02	**100.0**	2.5	103.3	**0.08**	n.a.	10.0
i-60-03	**100.0**	15.0	103.7	**0.43**	n.a.	10.0
i-122-01	**100.0**	0.7	103.4	**0.29**	n.a.	10.0
Average	**100.0**	*3.5*	*107.2*	**0.10**		

Table 5: Assessment of the pricing algorithm. The solution values are scaled to 100, and bold entries correspond to the best entry for each row, for computational time as well as solution quality.

	H-CGL/best LB	$\mathcal{F}_1$	$\mathcal{F}_1\mathcal{S}$	$\mathcal{F}_1\mathcal{E}$	CGS	CGS Time (min.)
i-10-01	100.1	95.6	**100.0**	96.1	84.7	0.2
i-10-02	100.0	**100.0**	**100.0**	**100.0**	82.3	0.1
i-10-03	100.0	**100.0**	**100.0**	**100.0**	75.5	1.7
i-30-01	109.7	93.3	94.0	92.7	**100.0**	0.6
i-30-02	104.3	**100.0**	98.9	95.7	91.4	1.0
i-30-03	100.1	99.5	**100.0**	**100.0**	83.0	7.5
i-60-01	108.0	93.6	94.8	n.a.	**100.0**	2.2
i-60-02	106.1	97.3	**100.0**	n.a.	97.9	3.4
i-60-03	100.9	**100.0**	**100.0**	n.a.	86.6	26.9
i-122-01	105.9	87.9	n.a.	n.a.	**100.0**	13.7
Average	103.5	96.7	98.6	97.4	90.1	5.7

Table 6: H-CGL solution quality compared with the best known lower bounds. The lower bound values are scaled to 100, and bold entries correspond to the best entry for each row.