Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Reformulations by Discretization for Piecewise Linear Integer Multicommodity Network Flow Problems

Bernard Gendron, Luis Gouveia

To cite this article:
Bernard Gendron, Luis Gouveia (2017) Reformulations by Discretization for Piecewise Linear Integer Multicommodity Network Flow Problems. Transportation Science 51(2):629-649. https://doi.org/10.1287/trsc.2015.0634

informs
http://pubsonline.informs.org/journal/trsc/

TRANSPORTATION SCIENCE
Vol. 51, No. 2, May 2017, pp. 629–649
ISSN 0041-1655 (print), ISSN 1526-5447 (online)

Reformulations by Discretization for Piecewise Linear Integer Multicommodity Network Flow Problems

Bernard Gendron,[a,b] **Luis Gouveia**[c]

[a] Département d'informatique et de recherche opérationnelle, Université de Montréal, Montréal, Québec H3T 1J4, Canada;
[b] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), Montréal, Québec H3C 3J7, Canada;
[c] Department of Statistics and Operations Research–CMAFCIO, Faculdade de Ciencias da Universidade de Lisboa, 1749-016 Lisboa, Portugal

Contact: bernard.gendron@cirrelt.ca (BG); legouveia@fc.ul.pt (LG)

Received: May 7, 2014
Revised: March 29, 2015
Accepted: May 27, 2015
Published Online in Articles in Advance: June 20, 2016

https://doi.org/10.1287/trsc.2015.0634

Abstract. We consider the piecewise linear multicommodity network flow problem with the addition of a constraint specifying that the total flow on each arc must be an integer. This problem has applications in transportation and logistics, where total flows might represent vehicles or containers filled with different products. We introduce formulations that exploit this integrality constraint by adapting to our problem a technique known as discretization that has been used to derive mixed-integer programming models for several combinatorial optimization problems. We enhance the discretized models either by adding valid inequalities derived from cut-set inequalities or by using flow disaggregation techniques. Since the size of the formulations derived from discretization and flow disaggregation rapidly increases with problem dimensions, we develop an efficient and effective Lagrangian relaxation method to compute lower and upper bounds. We perform computational results on a large set of randomly generated instances that allow us to compare the relative efficiency of the different modeling alternatives (flow disaggregation plus addition of cut-set inequalities with or without discretization), when used within the Lagrangian relaxation approach.

Funding: The work of the first author has been supported by the Natural Sciences and Engineering Research Council of Canada [Grant 184122-09]. The work of the second author has been supported by the Fundação para a Ciência e Tecnologia [Project PEst-OE/MAT/UI0152].

Keywords: multicommodity network flow problem • piecewise linear cost • reformulation • discretization • disaggregation • valid inequalities

1. Introduction

We consider the piecewise linear multicommodity network flow problem (PMF) studied in Croxton, Gendron, and Magnanti (2007). Given a directed network $G = (N, A)$, with node set N, arc set A, supplies and demands of multiple commodities at the nodes, and arc capacities, the problem is to find the minimum cost multicommodity flow when the objective is the sum of $|A|$ piecewise linear functions. If we denote x_a the total flow on each arc a, the cost $g_a(x_a)$ is a piecewise linear function such that $g_a(0) = 0$. The pieces, or *segments*, of the cost function for arc a are represented by the finite set $S_a = \{1, 2, \ldots, |S_a|\}$. For each arc a, each segment $s \in S_a$ has a slope $c_a^s \geq 0$ (the linear cost), an intercept $f_a^s \geq 0$ (the fixed cost), and lower and upper flow bounds, b_a^{s-1} and b_a^s (the break points, assumed to be integers), which satisfy $0 = b_a^0 \leq b_a^{s-1} < b_a^s \leq u_a$, where u_a is the integer arc capacity. The function is not necessarily continuous, but we assume it is lower semicontinuous (i.e., $g_a(x_a) \leq \liminf_{x'_a \to x_a} g_a(x'_a)$ for any sequence x'_a that approaches x_a). We also assume it is nondecreasing (i.e., $g_a(x_a) \leq g_a(x'_a)$ whenever $x_a < x'_a$); this mild assumption is typically always satisfied in practice. To complete the problem definition, we let K denote the set of commodities, and d^k the vector of size $|N|$ representing supplies and demands for commodity k: for each node i and each commodity k, $d_i^k > 0$ denotes an origin node with integer supply d_i^k, $d_i^k < 0$ denotes a destination node with integer demand $-d_i^k$, and $d_i^k = 0$ denotes a transshipment node.

Applications of the PMF in transportation, logistics, telecommunications, and production planning (Balakrishnan, Magnanti, and Mirchandani 1997; Crainic 2000; Gendron, Crainic, and Frangioni 1999; Magnanti and Wong 1984; Minoux 1989) often require the flows to take integer values. In the piecewise linear *integer* multicommodity network flow problem (PMFI) that we study, we assume that the total flow on each arc, x_a, must be an integer. In applications in transportation and logistics, total flows might represent vehicles or containers filled with different products, and therefore must assume integer values. Often, this integrality constraint is ignored when modeling and solving the problem, and the final continuous solution is used as an approximation of the optimal integer solution. In this paper, we adopt a different point of view and explicitly state the integrality constraint on the total flows. Furthermore, we introduce new formulations for piece-

wise linear multicommodity network flow problems that exploit this integrality constraint.

Following Croxton, Gendron, and Magnanti (2007), we present a mixed-integer programming (MIP) formulation of the PMFI, which we call the *basic segment-based model*. In this model, the flow x_a on each arc a is decomposed in two ways, by commodity or by segment, with x_a^k and x_a^s representing the flow of commodity k and the flow on segment s, where x_a^s is the total flow on arc a if that flow lies in segment s, and is 0 otherwise. Since $g_a(x_a)$ is lower semicontinuous and nondecreasing, if $x_a = b_a^{s-1} > 0$ then $x_a = x_a^{s-1}$, i.e., x_a lies in segment $s-1$. Because of the integrality of x_a, this observation implies that if x_a lies in segment s, then $l_a^s \equiv b_a^{s-1} + 1 \le x_a^s \le b_a^s \equiv v_a^s$. We also define binary variables y_a^s, with $y_a^s = 1$ if $x_a^s > 0$, and $y_a^s = 0$ otherwise. If we denote by $t(a) = i$ and $h(a) = j$, respectively, the tail and the head of each arc $a = (i, j)$ and by $F_i = \{a \in A \mid t(a) = i\}$ and $B_i = \{a \in A \mid h(a) = i\}$, respectively, the sets of forward and backward arcs incident to node $i \in N$, the basic segment-based model, denoted BS, can be expressed as follows:

$$v(BS) = \min \sum_{a \in A} \sum_{s \in S_a} (c_a^s x_a^s + f_a^s y_a^s), \tag{1}$$

$$\sum_{a \in F_i} x_a^k - \sum_{a \in B_i} x_a^k = d_i^k, \quad i \in N, k \in K, \tag{2}$$

$$\sum_{k \in K} x_a^k = \sum_{s \in S_a} x_a^s = x_a \text{ integer}, \quad a \in A, \tag{3}$$

$$l_a^s y_a^s \le x_a^s \le v_a^s y_a^s, \quad a \in A, s \in S_a, \tag{4}$$

$$\sum_{s \in S_a} y_a^s \le 1, \quad a \in A, \tag{5}$$

$$x_a^k \ge 0, \quad a \in A, k \in K, \tag{6}$$

$$y_a^s \in \{0, 1\}, \quad a \in A, s \in S_a. \tag{7}$$

Constraints (2) are the flow balance constraints typical in a multicommodity network flow formulation. Constraints (3) define the flow by commodity and by segment, and also impose integrality requirements on the total flow on each arc. The *multiple choice constraints*, (5), ensure that we choose at most one segment variable y_a^s to be equal to 1 on each arc a. The *basic forcing constraints*, (4), state that if $y_a^s = 0$, then $x_a^s = 0$, but if $y_a^s = 1$, then x_a^s must lie between the bounds of that segment, i.e., $b_a^{s-1} + 1 = l_a^s \le x_a^s \le v_a^s = b_a^s$. Note that, without the integrality constraint on the total flows, the lower flow bounds $l_a^s = b_a^{s-1} + 1$ must be replaced by $l_a^s = b_a^{s-1}$ to obtain a valid model. When the integrality constraint on the total flows is enforced, replacing $l_a^s = b_a^{s-1} + 1$ by $l_a^s = b_a^{s-1}$ would still yield another valid, but weaker, model (we comment further on this observation in Section 2.2).

It is well known that the basic segment-based model provides a weak linear programming (LP) relaxation bound. To improve this bound, one might add valid inequalities that can be violated by the solutions to the LP relaxation. One approach is to exploit necessary feasibility conditions for the underlying multicommodity network flow structure, giving rise to the so-called *cut-set inequalities*, which have been used to strengthen the LP relaxation bounds of a large number of problems related to the PMFI (Atamtürk 2002; Barahona 1996; Bienstock et al. 1998; Bienstock and Günlük 1996; Chouman, Crainic, and Gendron 2017; Gabrel, Knippel, and Minoux 2003; Günlük 1999; Magnanti, Mirchandani, and Vachani 1993; Ortega and Wolsey 2003; Raack et al. 2011). Another approach, called *flow disaggregation* (Croxton, Gendron, and Magnanti 2007; Frangioni and Gendron 2009, 2013), consists of defining additional flow variables that are linked to the other variables through simple valid inequalities that can improve the LP relaxation bound. A third approach, which exploits the integrality of the flows and is the focus of this paper, is *discretization*, a technique that has been used to derive MIP models for several combinatorial optimization problems (Gouveia 1995; Gouveia and Moura 2012; Gouveia and Saldanha da Gama 2006). Discretization can be combined with the two other approaches, addition of cut-set inequalities and flow disaggregation, with the goal of deriving models that improve the LP relaxation bounds.

In this paper, we show that the formulation obtained by discretization can be viewed as having the same structure as the basic model, except that the segment set on each arc is replaced by a set of integer points, each point corresponding to one of the possible values of the total flow on the arc. For this reason, we denote these models as "point-based" in contrast to "segment-based" models, such as BS, that use the segment set in the definition of the variables. Following the developments in Correia, Gouveia, and Saldanha da Gama (2010), Gouveia and Saldanha da Gama (2006), we derive valid inequalities from cut-set inequalities for both the segment-based and the point-based models. Then, we combine the point-based models with flow disaggregation techniques to derive a model similar to the so-called *extended* (*segment-based*) *formulation* introduced in Croxton, Gendron, and Magnanti (2007). Our main results state that (1) discretization provides stronger cut-set inequalities than those obtained from segment-based models; (2) discretization, when combined with flow disaggregation, does not improve on the LP relaxation of the extended segment-based model. We exploit these results by deriving a reformulation of the problem that combines the strength of both techniques: cut-set inequalities based on discretization and flow disaggregation with segment-based variables. An efficient Lagrangian relaxation method is developed to compute lower and upper bounds for this reformulation, but also for the other models introduced in this paper. Such a method is essential to compute effective bounds in reasonable time, since the size of the formulations

derived from discretization and flow disaggregation rapidly increases with problem dimensions. We perform computational results on a large set of randomly generated instances that allow us to compare the relative efficiency of the different modeling alternatives (flow disaggregation, plus addition of cut-set inequalities with or without discretization), when used within the Lagrangian relaxation approach.

The paper is organized as follows. In Section 2, we present and compare the different formulations of the PMFI, focusing on the relative strength of their LP relaxations. Then, we present the Lagrangian relaxation method. Section 3 describes the *Lagrangian dual optimization* procedure that computes lower bounds on the optimal value of the PMFI, and Section 4 presents the *Lagrangian heuristic* approach used to derive upper bounds. Section 5 analyzes the results of our computational experiments. We present conclusions and directions for further research in Section 6. Throughout the paper, we use the following notation: $v(M)$ denotes the optimal value of any model M and $\bar{M}$ denotes the LP relaxation of any MIP model M; in addition, conv(T) designates the convex hull of any set T.

2. Reformulations by Discretization

We exploit the integrality constraint on the flows by defining the point-based model in Section 2.1. In Section 2.2, we derive cut-set-based inequalities for both the segment-based and the point-based models, yielding stronger reformulations of the PMFI. In Section 2.3, we investigate the combination of flow disaggregation and discretization. Finally, Section 2.4 summarizes our main results and presents models that combine the strength of point-based cut-set inequalities with segment-based flow disaggregation.

2.1. Point-Based Model

The integrality constraint on the flows implies that x_a is either 0 or can take any integer value $q \in Q_a = \{1,2,\ldots,u_a\}$. Note that we can partition Q_a into $|S_a|$ subsets $Q_a^s = \{b_a^{s-1}+1, b_a^{s-1}+2,\ldots,b_a^s\}$, $s \in S_a$, such that $x_a \in Q_a^s$ if x_a lies in segment s. We now present a reformulation of the PMFI which, instead of decomposing the flow x_a on each arc a by segment, separates the flow x_a by each possible positive integer value $q \in Q_a = \{1,2,\ldots,u_a\}$. Namely, we introduce variables x_a^q, which are equal to q if $x_a = q$, along with binary variables y_a^q, which take value 1 if $x_a = q$, and value 0 otherwise. We then obtain the following point-based model for the PMFI:

$$\min \sum_{a\in A}\sum_{s\in S_a}\sum_{q\in Q_a^s}(c_a^s x_a^q + f_a^s y_a^q) \tag{8}$$

subject to (2), (6), and

$$\sum_{k\in K} x_a^k = \sum_{q\in Q_a} x_a^q = x_a \text{ integer}, \quad a \in A, \tag{9}$$

$$x_a^q = q y_a^q, \quad a \in A, q \in Q_a, \tag{10}$$

$$\sum_{q\in Q_a} y_a^q \le 1, \quad a \in A, \tag{11}$$

$$y_a^q \in \{0,1\}, \quad a \in A, q \in Q_a. \tag{12}$$

This model has a structure similar to that of *BS*, except that here each "segment" corresponds to a "point" q, i.e., any possible positive integer value of the flow x_a on each arc a. To obtain the same structure as *BS*, one would simply write down constraints (10) with two inequalities as follows:

$$q y_a^q \le x_a^q \le q y_a^q, \quad a \in A, q \in Q_a.$$

Indeed, when $l_a^s = b_a^{s-1}+1 = b_a^s = v_a^s$ for each $a \in A$ and each $s \in S_a$, *BS* reduces to the point-based model.

Because constraints (10) are expressed as equalities, we can project out the flow variables x_a^q and remove the integrality constraints on the total flows, which are redundant, obtaining the following equivalent formulation, called the *basic point-based model* and denoted *BP*:

$$v(BP) = \min \sum_{a\in A}\sum_{s\in S_a}\sum_{q\in Q_a^s}(q c_a^s + f_a^s) y_a^q \tag{13}$$

subject to (2), (6), (11), (12), and

$$\sum_{k\in K} x_a^k = \sum_{q\in Q_a} q y_a^q, \quad a \in A. \tag{14}$$

We now compare $\overline{BP}$, the *LP* relaxation of *BP*, to $\overline{BS}$, the LP relaxation of the basic segment-based model, (1)–(7). Note that $\overline{BS}$ can be simplified using the following observation: there always exists an optimal solution such that $x_a^s = v_a^s y_a^s$, for each arc a and segment s, because otherwise, if $x_a^s < v_a^s y_a^s$ for some pair (a,s) in an optimal solution, we could always decrease y_a^s down to x_a^s/v_a^s and maintain feasibility, as well as optimality, since $f_a^s \ge 0$. As a result, $\overline{BS}$ can be simplified by projecting out the x_a^s variables, which yields the following model that will be subsequently used in our developments:

$$v(\overline{BS}) = \min \sum_{a\in A}\sum_{s\in S_a}(v_a^s c_a^s + f_a^s) y_a^s \tag{15}$$

subject to (2), (5), (6), and

$$\sum_{k\in K} x_a^k = \sum_{s\in S_a} v_a^s y_a^s, \quad a \in A, \tag{16}$$

$$y_a^s \ge 0, \quad a \in A, s \in S_a. \tag{17}$$

Note that $\overline{BP}$, the LP relaxation of *BP*, and $\overline{BS}$, the model defined by (15)–(17), along with (2), (5)–(6), have similar structures: in $\overline{BP}$, points q are used in place of segments s with their upper flow bounds v_a^s in $\overline{BS}$. Given the similarity of the two LP relaxations, the following proposition is not surprising (a similar result is proven in Duhamel et al. 2012).

Proposition 1. $v(\overline{BP}) = v(\overline{BS})$.

Proof. (1) We show that $v(\overline{BP}) \geq v(\overline{BS})$. Consider an optimal solution to $\overline{BP}$; for any a such that $y_a^q > 0$ for some q in this optimal solution, we let $y_a^s = (q/v_a^s)y_a^q$ whenever $q \in Q_a^s$ for some s (all other variables remain at the same values). This defines a feasible solution to $\overline{BS}$ with objective value $v(\overline{BP})$.

(2) We show that $v(\overline{BP}) \leq v(\overline{BS})$. Consider an optimal solution to $\overline{BS}$; for any a such that $y_a^s > 0$ for some s in this optimal solution, we define $y_a^q = y_a^s$ for $q = v_a^s$ (all other variables remain at the same values). This defines a feasible solution to $\overline{BP}$ with objective value $v(\overline{BS})$. □

When $|K| = 1$, we obtain the single-commodity case and all of the flow variables in BP can be projected out using Equations (14). Model BP then reduces to

$$\min \sum_{a\in A}\sum_{s\in S_a}\sum_{q\in Q_a^s}(qc_a^s + f_a^s)y_a^q \tag{18}$$

subject to (11), (12), and

$$\sum_{a\in F_i}\sum_{q\in Q_a} q y_a^q - \sum_{a\in B_i}\sum_{q\in Q_a} q y_a^q = d_i, \quad i \in N. \tag{19}$$

This model, containing only the binary variables y_a^q, is similar to the reformulations by discretization described in the literature (Correia, Gouveia, and Saldanha da Gama 2010; Gouveia 1995; Gouveia and Moura 2012; Gouveia and Saldanha da Gama 2006).

2.2. Cut-Set Inequalities

We denote by $\mathcal{U}$ the collection of nonempty proper subsets of N. For any cut $U \in \mathcal{U}$, we define its corresponding cut sets $F_U = \{a \in A \mid t(a) \in U, h(a) \notin U\}$ and $B_U = \{a \in A \mid t(a) \notin U, h(a) \in U\}$. By summing the flow conservation Equations (2) for all $i \in U$ and all $k \in K$, we obtain the following *flow cut-set equations*, after canceling equal terms:

$$\sum_{a\in F_U} x_a - \sum_{a\in B_U} x_a = D_U, \quad U \in \mathcal{U}, \tag{20}$$

where $D_U = \sum_{i\in U}\sum_{k\in K} d_i^k$ is the net supply across cut $U \in \mathcal{U}$. When $U = \{i\}$, $i \in N$, we obtain a *single-node* cut and we use the notation $D_i \equiv D_{\{i\}}$.

By combining the flow cut-set equations with constraints (3) and (4), we obtain the following *segment-based cut-set inequalities* for model BS:

$$\sum_{a\in F_U}\sum_{s\in S_a} v_a^s y_a^s - \sum_{a\in B_U}\sum_{s\in S_a} l_a^s y_a^s \geq D_U, \quad U \in \mathcal{U}, \tag{21}$$

$$\sum_{a\in F_U}\sum_{s\in S_a} l_a^s y_a^s - \sum_{a\in B_U}\sum_{s\in S_a} v_a^s y_a^s \leq D_U, \quad U \in \mathcal{U}. \tag{22}$$

These inequalities are redundant for the LP relaxation $\overline{BS}$, since they are obtained by linear combinations of constraints of the original model. However, inequalities derived from them by exploiting the integrality of the y variables might be violated by LP optimal solutions. In particular, every facet-defining inequality for $\text{conv}(CUT_S)$ can be used to strengthen $\overline{BS}$, where CUT_S is the set of 0–1 solutions to the multiple choice constraints (5) and the cut-set inequalities (21)–(22). By adding all of the facet-defining inequalities for $\text{conv}(CUT_S)$ to $\overline{BS}$, we obtain a stronger LP relaxation, which we denote $\overline{BS}+$.

Note that the lower flow bounds l_a^s do not appear in the LP relaxation $\overline{BS}$ defined by (15)–(17), but they play an important role in the cut-set inequalities (21)–(22). Indeed, if we use the weaker $l_a^s = b_a^{s-1}$ instead of $l_a^s = b_a^{s-1} + 1$, the cut-set inequalities would still be valid, but weaker. This observation further justifies our use of $l_a^s = b_a^{s-1} + 1$ as lower flow bounds in constraints (4), in spite of the fact that these stronger lower flow bounds do not improve the LP relaxation $\overline{BS}$.

For model BP, a similar derivation yields the following *point-based cut-set equations*, which can also be obtained directly from (21)–(22) for the case where $l_a^s = b_a^{s-1} + 1 = b_a^s = v_a^s$:

$$\sum_{a\in F_U}\sum_{q\in Q_a} q y_a^q - \sum_{a\in B_U}\sum_{q\in Q_a} q y_a^q = D_U, \quad U \in \mathcal{U}. \tag{23}$$

Let us define CUT_P as the set of 0–1 solutions that satisfy these cut-set equations, along with the multiple choice constraints (11), and $\overline{BP}+$ as the LP relaxation of BP obtained by adding all of the facet-defining inequalities for $\text{conv}(CUT_P)$ to formulation $\overline{BP}$. We then have the following result:

Proposition 2. $v(\overline{BP}+) \geq v(\overline{BS}+)$.

Proof. Let $y(i)$, $i \in I$, be the extreme points of $\text{conv}(CUT_P)$; any $y(i)$, $i \in I$, can be mapped to a solution of $\text{conv}(CUT_S)$ by the same construction used in the proof of Proposition 1: for any a such that $y_a^q(i) > 0$ for some q, we let $y_a^s(i) = (q/v_a^s)y_a^q(i)$ whenever $q \in Q_a^s$ for some s. Consider an optimal solution to $\overline{BP}+$; when projected over the space of y_a^q variables, this solution can be expressed as a convex combination of the extreme points of $\text{conv}(CUT_P)$: $y_a^q = \sum_{i\in I}\lambda(i)y_a^q(i)$, $a \in A, q \in Q_a$, with $\sum_{i\in I}\lambda(i) = 1$ and $\lambda(i) \geq 0$, $i \in I$. Again, we construct a feasible solution to $\overline{BS}+$ as in the proof of Proposition 1: for any a such that $y_a^q > 0$ for some q in this optimal solution, we let $y_a^s = (q/v_a^s)y_a^q$ whenever $q \in Q_a^s$ for some s (all other variables remain at the same values). This solution satisfies all of the constraints of model $\overline{BS}$; in addition, its projection over the space of y_a^s variables can be expressed as a convex combination of the solutions of $\text{conv}(CUT_S)$ obtained by mapping the extreme points of $\text{conv}(CUT_P)$: $y_a^s = (q/v_a^s)y_a^q = (q/v_a^s)\sum_{i\in I}\lambda(i)y_a^q(i) = \sum_{i\in I}\lambda(i)y_a^s(i)$, i.e., this projected solution belongs to $\text{conv}(CUT_S)$, which implies that $v(\overline{BP}+) \geq v(\overline{BS}+)$. □

Note that the aggregation of two point-based cut-set equations of the form (23) associated with $U \in \mathcal{U}$ and

$W \in \mathcal{U}$, $U \cap W = \emptyset$, is equivalent to the cut-set equation associated with $U \cup W$. Indeed, using the notation $\{U, W\} = \{a \in A \mid t(a) \in U, h(a) \in W\} \cup \{a \in A \mid t(a) \in W, h(a) \in U\}$, we obtain after summing the cut-set Equations (23) for U and W

$$\sum_{a \in F_{U \cup W}} \sum_{q \in Q_a} q y_a^q - \sum_{a \in B_{U \cup W}} \sum_{q \in Q_a} q y_a^q + \sum_{a \in \{U,W\}} \sum_{q \in Q_a} (q - q) y_a^q = D_{U \cup W},$$

which is the same as the cut-set equation associated with $U \cup W$. This observation allows us to considerably reduce the number of equations needed to characterize CUT_P.

Proposition 3. *CUT_P is equal to the set of 0–1 solutions that satisfy the multiple choice constraints (11) and the point-based single-node cut-set equations*

$$\sum_{a \in F_i} \sum_{q \in Q_a} q y_a^q - \sum_{a \in B_i} \sum_{q \in Q_a} q y_a^q = D_i, \quad i \in N. \tag{24}$$

Proof. Using the same argument as above, all cut-set equations of the form (23) can be derived by aggregation of single-node cut-set equations of the form (24). □

Proposition 3 illustrates another important difference between the segment-based cut-set inequalities (21)–(22) and the point-based cut-set equations (23): the *single-node cut-set Equations* (24) are enough to characterize all point-based cut-set equations, whereas the same is not true for the segment-based cut-set inequalities, i.e., by restricting these inequalities to single-node cuts, we only obtain a subset of CUT_S. To see why, observe that the aggregation of two cut-set inequalities of the form (21) or (22) associated with $U \in \mathcal{U}$ and $W \in \mathcal{U}$, $U \cap W = \emptyset$, is not equivalent to the cut-set inequality of the same form associated with $U \cup W$. For example, by summing the cut-set inequalities (21) for U and W, one obtains

$$\sum_{a \in F_{U \cup W}} \sum_{s \in S_a} v_a^s y_a^s - \sum_{a \in B_{U \cup W}} \sum_{s \in S_a} l_a^s y_a^s + \sum_{a \in \{U,W\}} \sum_{s \in S_a} (v_a^s - l_a^s) y_a^s \geq D_{U \cup W},$$

which is dominated by the cut-set inequality of the form (21) associated with $U \cup W$

$$\sum_{a \in F_{U \cup W}} \sum_{s \in S_a} v_a^s y_a^s - \sum_{a \in B_{U \cup W}} \sum_{s \in S_a} l_a^s y_a^s \geq D_{U \cup W}.$$

Obviously, generating all facet-defining inequalities for $\text{conv}(CUT_S)$ and $\text{conv}(CUT_P)$ is a hard task. In particular, even for a set defined by a *single* cut-set inequality associated with a given cut U, generating all facet-defining inequalities is not trivial. In the context of reformulations by discretization, Chvátal–Gomory rank 1 inequalities have been derived from a single inequality and proven to be rather exceptionally effective (Correia, Gouveia, and Saldanha da Gama 2010; Gouveia and Moura 2012; Gouveia and Saldanha da Gama 2006). The technique is simple: each possible value in the discrete set is tried as a divisor of every coefficient in the inequality; then, the resulting coefficients are rounded up or down to obtain inequalities that are valid for the MIP model, but not necessarily for its LP relaxation. The technique is easy to illustrate on the point-based cut-set Equations (23), yielding the following valid inequalities, where $P = \max_{a \in A}\{u_a\}$:

$$\sum_{a \in F_U} \sum_{q \in Q_a} \left\lceil \frac{q}{p} \right\rceil y_a^q - \sum_{a \in B_U} \sum_{q \in Q_a} \left\lfloor \frac{q}{p} \right\rfloor y_a^q \geq \left\lceil \frac{D_U}{p} \right\rceil, \quad U \in \mathcal{U}, p = 1, 2, \ldots, P, \tag{25}$$

$$\sum_{a \in F_U} \sum_{q \in Q_a} \left\lfloor \frac{q}{p} \right\rfloor y_a^q - \sum_{a \in B_U} \sum_{q \in Q_a} \left\lceil \frac{q}{p} \right\rceil y_a^q \leq \left\lfloor \frac{D_U}{p} \right\rfloor, \quad U \in \mathcal{U}, p = 1, 2, \ldots, P. \tag{26}$$

When $p = 1$, these inequalities reduce to the point-based cut-set Equations (23). Note that a large number of these inequalities can be removed, since they can be easily shown to be dominated by others. So, even though there is a large number of valid inequalities, P, for each cut U, only a small subset of them will be generated. We also observe that, contrary to the point-based cut-set Equations (23), (25)–(26) cannot be reduced to inequalities associated only to single-node cuts.

The technique can be generalized to the segment-based cut-set inequalities (21)–(22), giving rise to the following valid inequalities:

$$\sum_{a \in F_U} \sum_{s \in S_a} \left\lceil \frac{v_a^s}{p} \right\rceil y_a^s - \sum_{a \in B_U} \sum_{s \in S_a} \left\lfloor \frac{l_a^s}{p} \right\rfloor y_a^s \geq \left\lceil \frac{D_U}{p} \right\rceil, \quad U \in \mathcal{U}, p = 1, 2, \ldots, P, \tag{27}$$

$$\sum_{a \in F_U} \sum_{s \in S_a} \left\lfloor \frac{l_a^s}{p} \right\rfloor y_a^s - \sum_{a \in B_U} \sum_{s \in S_a} \left\lceil \frac{v_a^s}{p} \right\rceil y_a^s \leq \left\lfloor \frac{D_U}{p} \right\rfloor, \quad U \in \mathcal{U}, p = 1, 2, \ldots, P. \tag{28}$$

Let us define $\widehat{BS+}$ and $\widehat{BP+}$ the models obtained by adding, respectively, inequalities (27)–(28) to $\overline{BS}$ and inequalities (25)–(26) to $\overline{BP}$. Since the Chvátal–Gomory rank 1 inequalities approximate the convex hull associated with each single cut-set inequality, we have the obvious bound relationships $v(\overline{BS+}) \geq v(\widehat{BS+})$ and $v(\overline{BP+}) \geq v(\widehat{BP+})$. In addition, we have the following result.

Proposition 4. $v(\widehat{BP+}) \geq v(\widehat{BS+})$.

Proof. We apply the same construction as in part (1) of the proof of Proposition 1. The result follows from the inequalities $b_a^{s-1} + 1 = l_a^s \leq q \leq v_a^s = b_a^s$, $a \in A$, $s \in S_a$, $q \in Q_a^s$. □

2.3. Flow Disaggregation

Another approach to improve the basic segment-based model is to define additional variables x_a^{ks} as the flow of commodity k on arc a if the total flow x_a on the arc lies in segment s, and equal zero otherwise. These variables are related to the previous ones via the *definitional equations*: $x_a^s = \sum_{k\in K} x_a^{ks}$ and $x_a^k = \sum_{s\in S_a} x_a^{ks}$. Using these variables, we can define the following *extended forcing constraints* (Croxton, Gendron, and Magnanti 2007):

$$x_a^{ks} \le M_a^k y_a^s, \quad a \in A, k \in K, s \in S_a, \tag{29}$$

where M_a^k is an integer upper bound on the flow of commodity k circulating through arc a; for instance, one can simply use $M_a^k = \min\{u_a, \frac{1}{2}\sum_{i\in N}|d_i^k|\}$. We refer to the model obtained by adding the nonnegative variables x_a^{ks}, the definitional equations, and the valid inequalities (29) to BS as the *extended segment-based model*, which we denote ES. Obviously, we have $v(\overline{ES}) \ge v(\overline{BS})$.

We now define another reformulation of the PMFI by applying a similar variable disaggregation technique to the basic point-based model, BP. Additional variables x_a^{kq} are defined, representing the flow of commodity k on arc a if the total flow $\sum_{k\in K} x_a^k$ on the arc is q, and equal zero otherwise. Using them, we can define the following valid inequalities:

$$x_a^{kq} \le M_a^k y_a^q, \quad a \in A, k \in K, q \in Q_a. \tag{30}$$

We refer to the model obtained by adding the nonnegative variables x_a^{kq}, the definitional equations $q y_a^q = \sum_{k\in K} x_a^{kq}$ and $x_a^k = \sum_{q\in Q_a} x_a^{kq}$, and the valid inequalities (30) to BP as the *extended point-based model*, which we denote EP.

We now show that, similar to what happens for the basic models, the LP relaxations of the extended point-based and segment-based models are equivalent. Note that the proof of Proposition 1 does not apply to the extended models, hence we have to use more elaborate arguments. Our proof of the equivalence of these two models makes use of the Lagrangian relaxation with respect to the flow conservation Equations (2) for both models, ES and EP. After projecting out all flow variables, except variables x_a^{ks}, we obtain for model ES the following Lagrangian subproblem, denoted by $ES(\pi)$, where $\pi = (\pi_i^k)_{i\in N}^{k\in K}$ are the Lagrange multipliers:

$$v(ES(\pi)) = \min\Bigg\{\sum_{a\in A}\sum_{s\in S_a}\sum_{k\in K}(c_a^s - \pi_{t(a)}^k + \pi_{h(a)}^k)x_a^{ks} + \sum_{a\in A}\sum_{s\in S_a} f_a^s y_a^s\Bigg\}, \tag{31}$$

subject to (5), (7), and

$$l_a^s y_a^s \le \sum_{k\in K} x_a^{ks} \le v_a^s y_a^s, \quad a \in A, s \in S_a, \tag{32}$$

$$0 \le x_a^{ks} \le M_a^k y_a^s, \quad a \in A, k \in K, s \in S_a. \tag{33}$$

Note that we could remove the integrality constraint on the variables $x_a = \sum_{k\in K}\sum_{s\in S_a} x_a^{ks}$, since it is implicitly satisfied in the Lagrangian subproblem. Indeed, for any value of $y_a^s \in \{0,1\}$, $a \in A, s \in S_a$, all x_a^{ks} variables must assume integer values, because M_a^k, l_a^s, and v_a^s are integers.

This Lagrangian subproblem has the integrality property, i.e., we can solve it by relaxing the integrality requirements (7). To see why, first note that we can solve it independently for each arc $a \in A$. Let us denote LAG_a and $\overline{LAG}_a$ the sets of feasible solutions to, respectively, the Lagrangian subproblem and its LP relaxation associated with arc $a \in A$, i.e., the constraints defining LAG_a have the following form:

$$l_a^s y_a^s \le \sum_{k\in K} x_a^{ks} \le v_a^s y_a^s, \quad s \in S_a, \tag{34}$$

$$0 \le x_a^{ks} \le M_a^k y_a^s, \quad k \in K, s \in S_a, \tag{35}$$

$$\sum_{s\in S_a} y_a^s \le 1, \tag{36}$$

$$y_a^s \in \{0,1\}, \quad s \in S_a. \tag{37}$$

We then have the following polyhedral result, which is extracted from the proof of Theorem 5 in Croxton, Gendron, and Magnanti (2007).

Proposition 5. $\text{conv}(LAG_a) = \overline{LAG}_a$.

Proof. The inclusion $\subseteq$ is trivial. To show the inclusion $\supseteq$, it suffices to prove that every extreme point of $\overline{LAG}_a$ is integral. If not, then let $(\hat{x}, \hat{y})$ be an extreme point of $\overline{LAG}_a$ with at least one fractional component. Assume that $0 < \hat{y}_a^r < 1$, for $r \in R \neq \emptyset$. Define the following $|R| + 1$ points in $\overline{LAG}_a$: $(x(0), y(0)) = (0,0)$ and $(x(r), y(r))$, for $r \in R$, with $x_a^{kr}(r) = \hat{x}_a^{kr}/\hat{y}_a^r$, $x_a^{ks}(r) = 0$, $s \neq r$, $y_a^r(r) = 1$ and $y_a^s(r) = 0$, $s \neq r$. Then, $(\hat{x}, \hat{y}) = (1 - \sum_{r\in R}\hat{y}_a^r)(x(0), y(0)) + (\sum_{r\in R}\hat{y}_a^r(x(r), y(r)))$ is a representation of $(\hat{x}, \hat{y})$ as a convex combination of $|R| + 1 \ge 2$ distinct points in $\overline{LAG}_a$, contradicting the hypothesis that it is an extreme point of $\overline{LAG}_a$. □

By Lagrangian duality theory (Geoffrion 1974), this result implies that

$$v(\overline{ES}) = \max_{\pi}\Bigg\{\sum_{i\in N}\sum_{k\in K}\pi_i^k d_i^k + v(ES(\pi))\Bigg\}.$$

In the same way, we define for model EP the Lagrangian subproblem, $EP(\pi)$, resulting from the Lagrangian relaxation with respect to the flow conservation equations; $EP(\pi)$ has the following form, where $\pi = (\pi_i^k)_{i\in N}^{k\in K}$ are the Lagrange multipliers:

$$v(EP(\pi)) = \min\Bigg\{\sum_{a\in A}\sum_{s\in S_a}\sum_{q\in Q_a^s}\sum_{k\in K}(c_a^s - \pi_{t(a)}^k + \pi_{h(a)}^k)x_a^{kq} + \sum_{a\in A}\sum_{s\in S_a}\sum_{q\in Q_a^s} f_a^s y_a^q\Bigg\}, \tag{38}$$

subject to (11), (12), and

$$\sum_{k\in K} x_a^{kq} = q y_a^q, \quad a \in A, q \in Q_a, \tag{39}$$

$$0 \le x_a^{kq} \le M_a^k y_a^q, \quad a \in A, k \in K, q \in Q_a. \tag{40}$$

Similarly, as for $ES(\pi)$, we can show that this Lagrangian subproblem has the integrality property. By Lagrangian duality theory (Geoffrion 1974), it follows that $v(\overline{EP}) = \max_\pi \{\sum_{i\in N}\sum_{k\in K} \pi_i^k d_i^k + v(EP(\pi))\}$.

Proposition 6. $v(\overline{EP}) = v(\overline{ES})$.

Proof. We have just seen that

$$v(\overline{ES}) = \max_\pi \left\{ \sum_{i\in N}\sum_{k\in K} \pi_i^k d_i^k + v(ES(\pi)) \right\} \quad \text{and}$$

$$v(\overline{EP}) = \max_\pi \left\{ \sum_{i\in N}\sum_{k\in K} \pi_i^k d_i^k + v(EP(\pi)) \right\}.$$

Therefore, the result follows if we can prove that the two Lagrangian subproblems, $ES(\pi)$ and $EP(\pi)$, are equivalent.

(1) We show that $v(EP(\pi)) \ge v(ES(\pi))$. Consider an optimal solution to $EP(\pi)$; for any a such that $y_a^q = 1$ for some q in this optimal solution, we let $y_a^s = 1$ and $x_a^{ks} = x_a^{kq}$ whenever $q \in Q_a^s$ for some s. This defines a feasible solution to $ES(\pi)$ with objective value $v(EP(\pi))$.

(2) We show that $v(EP(\pi)) \le v(ES(\pi))$. Consider an optimal solution to $ES(\pi)$; for any a such that $y_a^s = 1$ for some s in this optimal solution, the values of x_a^{ks} can be obtained by solving a continuous knapsack problem, with both lower and upper integer capacities, l_a^s and v_a^s, and integer bounds M_a^k on each variable. We conclude that there always exists an optimal solution to $ES(\pi)$ such that the total segment flow x_a^s (if it is positive) is an integer $q \in Q_a^s$. As a consequence, we can derive a feasible solution to $EP(\pi)$ with value $v(ES(\pi))$ as follows: $x_a^{kq} = x_a^{ks}$, $k \in K$, and $y_a^q = 1$ whenever $x_a^s = q$ (otherwise, all of the variables assume value 0). □

2.4. Combining Cut-Set Inequalities and Flow Disaggregation

Our results from Sections 2.2 and 2.3 highlight that the best LP relaxation bounds can be obtained by combining cut-set inequalities with flow disaggregation. To combine them into a single model, we have to consider the following facts: (1) flow disaggregation with point-based variables does not bring any bound improvement on flow disaggregation with segment-based variables; (2) cut-set inequalities that use either segment-based or point-based variables can improve the lower bound when added to the extended models (the results shown for the basic models easily generalize to the extended models); (3) point-based cut-set inequalities can improve on segment-based cut-set inequalities.

These observations motivate the definition of new models obtained from the extended-segment based model by adding the point-based binary variables y_a^q through the linking equations

$$\sum_{k\in K} x_a^{ks} = \sum_{q\in Q_a^s} q y_a^q, \quad a \in A, s \in S_a, \tag{41}$$

$$y_a^s = \sum_{q\in Q_a^s} y_a^q, \quad a \in A, s \in S_a. \tag{42}$$

The two following relaxations are then obtained by adding point-based cut-set inequalities:

- $\overline{EP+}$, the model derived from $\overline{ES}$ by adding the linking Equations (41)–(42), plus all of the facet-defining inequalities for $\text{conv}(CUT_P)$;
- $\widehat{EP+}$, the LP relaxation obtained from $\overline{ES}$ by adding the point-based Chvátal–Gomory rank 1 valid inequalities (25)–(26), plus the linking Equations (41)–(42).

The next two relaxations are "pure" segment-based models that are included to assess what can be gained by adding the point-based cut-set inequalities:

- $\overline{ES+}$, the formulation obtained by adding to $\overline{ES}$ all of the facet-defining inequalities for $\text{conv}(CUT_S)$;
- $\widehat{ES+}$, the LP relaxation obtained by adding the segment-based Chvátal–Gomory rank 1 valid inequalities (27)–(28) to $\overline{ES}$.

The following proposition relates the optimal values of these four relaxations of the PMFI.

Proposition 7. *The following bound relationships hold*:

(a) $v(\overline{EP+}) \ge v(\overline{ES+}) \ge v(\widehat{ES+})$.
(b) $v(\overline{EP+}) \ge v(\widehat{EP+}) \ge v(\widehat{ES+})$.

These results, as well as all of the relationships between the LP relaxation bounds presented in Section 2, are summarized in Table 1. The symbol "≤" on an empty line relates the LP relaxation bounds of the formulations above and below the line (the symbol "≤≥" means that there is no dominance between the bounds). Thus, from a theoretical perspective, the strongest lower bound is obtained from model $\overline{EP+}$, and relaxation $\overline{BS}$ shows the worst lower bound.

We illustrate these different LP relaxation bounds on an instance of the PMFI with a single segment on each arc and three commodities. In spite of its apparent simplicity, this small example is complex enough to illustrate the impact of discretization, flow disaggregation, and their combination on the improvement of the LP relaxation bounds. Figure 1 shows the network $G = (N, A)$ for this instance, and Tables 2 and 3 specify the data related to the arcs and to the commodities, respectively. For each arc a, let x_a^k denote the flow of commodity $k \in \{1, 2, 3\}$ and y_a the binary variable that equals 1, if $\sum_{k=1}^3 x_a^k > 0$, and 0, otherwise.

An optimal solution consists in sending two units of flow of commodity 1 on path $(r, m, 1)$, two units of flow of commodity 2 on path $(r, m, 2)$, and two units of flow

Table 1. Summary of Formulations and Techniques Used to Improve the Weaker Models, and the Relationships Between the LP Relaxation Bounds

Technique	Segment based		Point based
None	$v(\overline{BS})$	=	$v(\overline{BP})$
	$\leq$		$\leq$
Chvátal–Gomory cut-set inequalities	$v(\widehat{BS+})$	$\leq$	$v(\widehat{BP+})$
	$\leq$		$\leq$
Convex hull of cut-set inequalities	$v(\overline{BS+})$	$\leq$	$v(\overline{BP+})$
	$\leq\geq$		$\leq\geq$
Flow disaggregation	$v(\overline{ES})$	=	$v(\overline{EP})$
	$\leq$		$\leq$
Flow disaggregation + Chvátal–Gomory cut-set inequalities	$v(\widehat{ES+})$	$\leq$	$v(\widehat{EP+})$
	$\leq$		$\leq$
Flow disaggregation + Convex hull of cut-set inequalities	$v(\overline{ES+})$	$\leq$	$v(\overline{EP+})$

of commodity 3 on path $(r,n,3)$. This way, only two arcs with a fixed cost of 10, (r,m) and (r,n), and only one arc with a linear cost of 2, $(m,2)$, are selected, for an optimal value $v(BS)=24$. The optimal values of the y variables are therefore $y_{rm}=y_{rn}=1$ and $y_{ro}=0$.

By contrast, the LP relaxation of the basic segment-based model utilizes the three arcs with positive fixed costs, to save on the linear costs: in its optimal solution, two units of flow of commodity 1 are sent on path $(r,m,1)$, two units of flow of commodity 2 on path $(r,n,3)$, and two units of flow of commodity 3 on path $(r,o,2)$. The optimal values of the y variables are $y_{rm}=1/3$ and $y_{rn}=y_{ro}=2/5$, with an optimal value $v(\overline{BS})=11\frac{1}{3}$.

From the single-node cut $\{r\}$, we derive the following segment-based cut-set inequality (which is the only relevant one, since other cuts do not involve arcs with positive fixed costs):

$$6y_{rm}+5y_{rn}+5y_{ro}\geq 6. \tag{43}$$

Figure 1. PMFI Instance Used to Illustrate the Different LP Relaxation Bounds

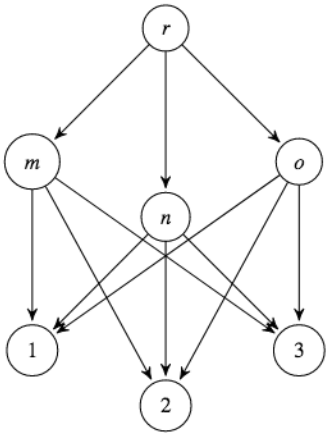

Table 2. Data for Each Arc of the PMFI Instance of Figure 1

Arcs	Capacity	Fixed cost	Linear cost
(r,m)	6	10	0
$(r,n),(r,o)$	5	10	0
$(m,1),(n,3),(o,2)$	2	0	0
$(m,2),(n,1),(o,3)$	2	0	2
$(m,3),(n,2),(o,1)$	2	0	10

Table 3. Data for Each Commodity of the PMFI Instance of Figure 1

Commodity	Demand	Origin	Destination
1	2	r	1
2	2	r	2
3	2	r	3

The only nonredundant Chvátal–Gomory inequality of the form (27)–(28), derived from (43) for $p=5$, is

$$2y_{rm}+y_{rn}+y_{ro}\geq 2. \tag{44}$$

By adding it to $\overline{BS}$, we obtain $\widehat{BS+}$, whose optimal solution has the same flow values as that of $\overline{BS}$, but the optimal values of the y variables are $y_{rm}=3/5$ and $y_{rn}=y_{ro}=2/5$, with an optimal value $v(\widehat{BS+})=14$. Compared to the optimal solution of $\overline{BS}$, we note that the value of y_{rm} is lifted by the cut, the other variables remaining at the same values.

To compute $v(\overline{BS+})$, we consider the convex hull of 0–1 solutions satisfying inequality (43), which is given by the two following facet-defining inequalities, along with the trivial facets $y_{rm}\leq 1$, $y_{rn}\leq 1$, and $y_{ro}\leq 1$:

$$y_{rm}+y_{rn}\geq 1, \tag{45}$$

$$y_{rm}+y_{ro}\geq 1. \tag{46}$$

We note that inequality (44) is obtained by aggregating these two facet-defining inequalities. In spite of being stronger, model $\overline{BS+}$ provides the same optimal solution as that of $\widehat{BS+}$, thus we have $v(\overline{BS+})=14$.

Considering now the point-based models, we have to add 34 additional binary variables, which correspond to the variables y_a^q for each arc $a\in A$ and for each possible flow value q on arc a, along with Equations (41)–(42). The resulting point-based formulations not only have a large number of variables, but the number of point-based cut-set inequalities can also be significant. For instance, CUT_P is described by seven single-node cut-set equations of the form (24) and 12 multiple choice inequalities of the form (11). To identify all of the facet-defining inequalities of $\text{conv}(CUT_P)$, we used PORTA version 1.4.1 (Christof and Loebel 2015). Unfortunately, after several days of computations, PORTA could not provide the facets of $\text{conv}(CUT_P)$ and started experiencing numerical issues. For that reason, for all point-based models, we use the relaxations defined by adding

only the variables and the inequalities associated with the single-node cut $\{r\}$; to simplify the notation, the resulting relaxations will be identified with their corresponding "complete" formulation, i.e., $\widehat{BP+}$, $\overline{BP+}$, $\widehat{EP+}$, and $\overline{EP+}$. As we will see, considering these relaxations will be sufficient to illustrate the bound improvements obtained by moving from segment-based to point-based models.

For the single-node cut $\{r\}$, we add 16 binary variables y^q_a for $a \in \{(r,m),(r,n),(r,o)\}$ and for each flow value $q \in \{1,\ldots,5\}$, with the additional value $q = 6$ on arc (r,m). Using these variables to generate the Chvátal–Gomory inequalities of the form (25)–(26), we obtain five nonredundant inequalities, two of the form (25) and three of the form (26). The optimal solution to $\widehat{BP+}$ has the same flow values as that of $\overline{BS}$, but the optimal values of the y variables are $y_{rm} = y^6_{rm} = \frac{1}{3}$, $y_{ro} = y^5_{ro} = \frac{2}{5}$ and $y_{rn} = \frac{14}{15}$, $y^1_{rn} = \frac{2}{3}$, $y^5_{rn} = \frac{4}{15}$ (all other y variables are at value 0), with an optimal value $v(\widehat{BP+}) = 16\frac{2}{3}$. Compared to the optimal solution of $\overline{BS}$, we see that now, it is the value of y_{rn} that is lifted. There is only one Chvátal–Gomory inequality, of the form (25) for $p = 5$, that is responsible for this lifting

$$\sum_{q=1}^{5} y^q_{rm} + 2y^6_{rm} + \sum_{q=1}^{5} y^q_{rn} + \sum_{q=1}^{5} y^q_{ro} \geq 2. \tag{47}$$

We can easily see that this point-based inequality dominates the segment-based Chvátal–Gomory inequality (44) by writing down the latter inequality in the space of point-based variables, using Equations (42)

$$\sum_{q=1}^{6} 2y^q_{rm} + \sum_{q=1}^{5} y^q_{rn} + \sum_{q=1}^{5} y^q_{ro} \geq 2. \tag{48}$$

The relaxation of $\overline{BP+}$ restricted to the single-node cut $\{r\}$ can be obtained by finding the facet-defining inequalities for the convex hull of 0–1 solutions to the cut-set equation of the form (24) for $i = r$, along with three multiple choice constraints of the form (11). Again, PORTA was used for that purpose, resulting this time in the identification of all of the 149 facet-defining inequalities. It is noteworthy that, even though the Chvátal–Gomory inequality (47) is not facet defining (it is dominated by a facet found by PORTA), adding it to $\overline{BS}$ provides the same LP relaxation bound than adding *all* of the 149 facet-defining inequalities, since we verified using CPLEX that $v(\overline{BP+}) = 16\frac{2}{3}$, the same bound as $v(\widehat{BP+})$.

Now, turning our attention to flow disaggregation, we can see that the LP relaxation of the extended formulation, $\overline{ES}$, splits the flow values to save on the fixed costs, since its optimal solution sends one unit of flow on the following paths: $(r,m,1)$, $(r,n,1)$, $(r,m,2)$, $(r,o,2)$, $(r,n,3)$, and $(r,o,3)$. The optimal values of the y variables are $y_{rm} = y_{rn} = y_{ro} = 1/2$, with an optimal value $v(\overline{ES}) = 21$. Note that the nontrivial facet-defining inequalities (45) and (46) are satisfied by this optimal solution, hence $v(\widehat{ES+}) = v(\overline{ES+}) = 21$.

If we add the point-based binary variables to $\overline{ES}$, we note that the point-based Chvátal–Gomory cut-set inequality (47) is violated by the optimal solution to $\overline{ES}$. The optimal solution to $\widehat{EP+}$ is obtained by rerouting the flows as follows to satisfy this cut-set inequality in the most economical way: $x^1_{rm} = x^2_{rm} = \frac{5}{4}$, $x^1_{rn} = \frac{3}{4}$, $x^3_{rn} = \frac{5}{4}$, and $x^2_{ro} = x^3_{ro} = \frac{3}{4}$. The optimal values of the y variables are $y_{rm} = \frac{5}{8}$, $y^1_{rm} = \frac{1}{4}$, $y^6_{rm} = \frac{3}{8}$, $y_{rn} = \frac{5}{8}$, $y^1_{rn} = \frac{9}{32}$, $y^2_{rn} = \frac{11}{32}$, and $y_{ro} = \frac{3}{8}$, $y^1_{ro} = \frac{3}{32}$, $y^5_{ro} = \frac{9}{32}$ (all other y variables are equal to 0), with optimal value $v(\widehat{EP+}) = 21\frac{3}{4}$.

After adding the point-based binary variables to $\overline{ES}$, along with the 149 facet-defining inequalities identified by PORTA, the same optimal solution as that of $\widehat{EP+}$ is obtained, so we have $v(\overline{EP+}) = 21\frac{3}{4}$. Again, we note that the single Chvátal–Gomory inequality (47) is responsible for the lower bound improvement from segment-based to point-based models.

To summarize, we obtain for this instance the following LP relaxation bounds:

$$\begin{aligned} v(\overline{BS}) &= 11\tfrac{1}{3} < 14 = v(\widehat{BS+}) = v(\overline{BS+}) < v(\widehat{BP+}) \\ &= v(\overline{BP+}) = 16\tfrac{2}{3} < 21 = v(\overline{ES}) = v(\widehat{ES+}) \\ &= v(\overline{ES+}) < v(\widehat{EP+}) = v(\overline{EP+}) = 21\tfrac{3}{4} < 24 \\ &= v(BS). \end{aligned}$$

In particular, this example shows the impact of the point-based Chvátal–Gomory cut-set inequalities, since we have $v(\widehat{BS+}) < v(\widehat{BP+}) = v(\overline{BP+})$ and $v(\widehat{ES+}) < v(\widehat{EP+}) = v(\overline{EP+})$.

3. Lagrangian Dual Optimization

In this section, we outline a Lagrangian relaxation method that provides lower bounds on the optimal value of the PMFI. The algorithm computes a tight approximation to $v(\overline{EP+})$, the strongest lower bound that we derived for the PMFI. This approximate lower bound is obtained by computing in sequence the weaker bounds $v(\overline{BS})$, $v(\overline{ES})$, $v(\widehat{ES+})$, $v(\overline{ES+})$, and $v(\widehat{EP+})$, using tight approximations to them. In Section 3.1, we describe the Lagrangian subproblem for computing the approximation to the lower bound $v(\overline{EP+})$; by slightly modifying this Lagrangian subproblem, we also show how to compute approximations to $v(\overline{ES})$ and $v(\overline{ES+})$. In Section 3.2, we outline our algorithm to obtain approximate lower bounds. In Section 3.3, we present the subgradient algorithm used to find effective Lagrange multipliers.

3.1. Lagrangian Subproblems

Based on our observations in Section 2.4, we exploit the following reformulation of the PMFI, which uses the segment-based and point-based variables in a single model:

$$\min\left\{\sum_{a\in A}\sum_{s\in S_a}\sum_{k\in K} c_a^s x_a^{ks} + \sum_{a\in A}\sum_{s\in S_a} f_a^s y_a^s\right\}, \quad (49)$$

$$\sum_{a\in F_i}\sum_{s\in S_a} x_a^{ks} - \sum_{a\in B_i}\sum_{s\in S_a} x_a^{ks} = d_i^k, \quad i\in N, k\in K, \quad (50)$$

$$l_a^s y_a^s \le \sum_{k\in K} x_a^{ks} \le v_a^s y_a^s, \quad a\in A, s\in S_a, \quad (51)$$

$$0 \le x_a^{ks} \le M_a^k y_a^s, \quad a\in A, k\in K, s\in S_a, \quad (52)$$

$$\sum_{s\in S_a} y_a^s \le 1, \quad a\in A, \quad (53)$$

$$\sum_{k\in K} x_a^{ks} = \sum_{q\in Q_a^s} q y_a^q, \quad a\in A, s\in S_a, \quad (54)$$

$$y_a^s = \sum_{q\in Q_a^s} y_a^q, \quad a\in A, s\in S_a, \quad (55)$$

$$\sum_{a\in F_i}\sum_{q\in Q_a} q y_a^q - \sum_{a\in B_i}\sum_{q\in Q_a} q y_a^q = D_i, \quad i\in N, \quad (56)$$

$$y_a^q \in \{0,1\}, \quad a\in A, q\in Q_a, \quad (57)$$

$$y_a^s \in \{0,1\}, \quad a\in A, s\in S_a. \quad (58)$$

The objective (49), along with constraints (50)–(53) and (58), correspond to the extended segment-based model *ES*, where all flow variables, except the x_a^{ks} variables, are projected out. Note that the integrality constraints on the total flow variables x_a are not included in the model, since they are implied by (54) and (57). Constraints (54)–(55) provide the link between the segment-based variables and the point-based variables y_a^q. The point-based single-node cut-set Equations (56) complete the formulation; these equations are redundant, both in this model and in its LP relaxation, but they will be used to improve the lower bound derived by Lagrangian relaxation.

We now show how to compute $v(\overline{EP+})$ using a Lagrangian relaxation of the reformulation of the PMFI defined by (49)–(58). We consider the Lagrangian relaxation of the flow conservation Equations (50) and of the linking Equations (54), where $\pi = (\pi_i^k)_{i\in N}^{k\in K}$ and $\beta = (\beta_a^s)_{a\in A}^{s\in S_a}$ are the respective Lagrange multipliers. This relaxation gives the following Lagrangian subproblem, noted $LAG_P(\pi,\beta)$:

$$v(LAG_P(\pi,\beta)) = \min\left\{\sum_{a\in A}\sum_{s\in S_a}\sum_{k\in K}(c_a^s - \beta_a^s - \pi_{t(a)}^k + \pi_{h(a)}^k)x_a^{ks} + \sum_{a\in A}\sum_{s\in S_a}(f_a^s y_a^s + \sum_{q\in Q_a^s} q\beta_a^s y_a^q)\right\}, \quad (59)$$

subject to constraints (51)–(53) and (55)–(58).

It is obvious that there exists an optimal solution to the Lagrangian subproblem such that, for each arc $a\in A$ and segment $s\in S_a$, $\sum_{k\in K} x_a^{ks} > 0$ only if $y_a^s = 1$. Hence, we can solve the Lagrangian subproblem as follows: for each arc $a\in A$ and segment $s\in S_a$, we first solve the following continuous knapsack problem:

$$v(P_a^s(\pi,\beta)) = \min\sum_{k\in K}(c_a^s - \beta_a^s - \pi_{t(a)}^k + \pi_{h(a)}^k)x_a^{ks}, \quad (60)$$

$$l_a^s \le \sum_{k\in K} x_a^{ks} \le v_a^s, \quad (61)$$

$$0 \le x_a^{ks} \le M_a^k, \quad k\in K. \quad (62)$$

Then, we reformulate the Lagrangian subproblem as follows:

$$v(LAG_P(\pi,\beta)) = \min\sum_{a\in A}\sum_{s\in S_a}\left((v(P_a^s(\pi,\beta)) + f_a^s)y_a^s + \sum_{q\in Q_a^s} q\beta_a^s y_a^q\right), \quad (63)$$

subject to constraints (53) and (55)–(58). The resulting Lagrangian subproblem is a pure integer programming (IP) model expressed only in terms of the segment-based and the point-based variables y_a^s and y_a^q.

Solving the corresponding Lagrangian dual allows us to compute $v(\overline{EP+})$, as stated next.

Proposition 8. $v(\overline{EP+}) = \max_{\pi,\beta}\{\sum_{i\in N}\sum_{k\in K}\pi_i^k d_i^k + v(LAG_P(\pi,\beta))\}$.

Proof. By Lagrangian duality theory (Geoffrion 1974), the Lagrangian dual is equivalent to optimizing the objective function (49) over the feasible domain described by the intersection of the set defined by (50) and (54) with the convex hull of the set defined by (51)–(53) and (55)–(58), which we denote by {(50), (54)} ∩ conv({(51)–(53), (55)–(58)}). If we can show that this feasible domain is equal to {(50)–(55)} ∩ conv(CUT_P), i.e., the set defined by (50)–(55) to which we add all of the facet-defining inequalities for conv(CUT_P), the result would immediately follow by definition of $\overline{EP+}$. To show that {(50), (54)} ∩ conv({(51)–(53), (55)–(58)}) = {(50)–(55)} ∩ conv(CUT_P), we first remark that the inclusion ⊆ follows by Proposition 3. To show the inclusion ⊇, it suffices to show that every extreme point of {(51)–(53), (55)} ∩ conv(CUT_P) is integral. Yet, this is immediate, since every extreme point of conv(CUT_P) satisfies the integrality constraints on the point-based variables, (57), along with $\sum_{q\in Q_a^s} y_a^q \le \sum_{q\in Q_a} y_a^q \le 1$, $a\in A$, $s\in S_a$. Hence, by (55), the integrality constraints on the segment-based variables, (58), are satisfied. □

Slight variations of this Lagrangian relaxation approach yield the following lower bounds, provided the optimal Lagrange multipliers are computed:

- $v(\overline{ES})$. It suffices to drop constraints (54) to (57) and to apply the same Lagrangian relaxation. In a similar way as above, there exists an optimal solution to the resulting Lagrangian subproblem such that, for each

arc $a \in A$ and segment $s \in S_a$, $\sum_{k\in K} x_a^{ks} > 0$ only if $y_a^s = 1$. As a result, we can reformulate the Lagrangian subproblem as follows:

$$\min \sum_{a\in A} \sum_{s\in S_a} (v(P_a^s(\pi)) + f_a^s) y_a^s, \tag{64}$$

subject to (53) and (58), where $v(P_a^s(\pi))$ is the optimal value of the following continuous knapsack problem:

$$v(P_a^s(\pi)) = \min \sum_{k\in K} (c_a^s - \pi_{t(a)}^k + \pi_{h(a)}^k) x_a^{ks}, \tag{65}$$

subject to (61) and (62). The Lagrangian subproblem is thus solvable by finding the smallest Lagrangian cost $v(P_a^s(\pi)) + f_a^s$ for each arc a, i.e., if $\min_{s\in S_a}\{v(P_a^s(\pi)) + f_a^s\} \le 0$ then for one $s \in S_a$ that achieves this minimum, we set $y_a^s = 1$; otherwise, we set $y_a^s = 0$, $s \in S_a$. A similar approach has been used to solve other problems related to the PMFI (Balakrishnan and Graves 1989; Crainic, Frangioni, and Gendron 2001; Holmberg and Yuan 2000).

- $v(\overline{ES+})$. We simply replace constraints (54)–(57) by the segment-based cut-set inequalities (21)–(22) and apply the same approach as for $v(\overline{ES})$ when evaluating the Lagrangian subproblem. Here, however, we obtain a pure IP model, in a similar way as when computing $v(\overline{EP+})$, but expressed only in terms of the y_a^s variables. Note that this IP model contains an exponential number of cut-set inequalities, in contrast to the IP model used when computing $v(\overline{EP+})$, which has a polynomial number of point-based single-node cut-set equations.

3.2. Computing Approximate Lower Bounds

As we have just seen, each of the Lagrangian subproblems solved when computing $v(\overline{ES+})$ contains an exponential number of cut-set inequalities. Although we could add them iteratively using a cutting-plane approach, these inequalities are difficult to separate for general multicommodity network flow problems. An alternative to a cutting-plane approach is to generate a priori a small subset of these inequalities. In our implementation, we adopted this approach, since our objective is not to obtain the exact lower bounds, but rather to compute efficiently tight approximations of them. Hence, we generate only the inequalities based on single-node cuts, a choice that is justified by computational experiments on similar problems (Atamtürk 2002; Chouman, Crainic, and Gendron 2017), which show that single-node cut-set inequalities are responsible for most of the lower bound improvement obtained by adding cut-set inequalities in the context of multicommodity network flow problems.

The reformulation of the Lagrangian subproblem as a pure IP model, i.e., (63) subject to constraints (53) and (55)–(58), is difficult to solve because of the large number of binary variables involved and also because the model exhibits a lot of symmetry, i.e., many solutions have very close objective values. To circumvent these issues, we solve instead an *MIP relaxation* of this reformulation obtained by dropping the integrality of the y_a^q variables and by adding the segment-based and the point-based Chvátal–Gomory rank 1 valid inequalities, i.e., (27)–(28) and (25)–(26), respectively, restricted to *single-node cuts*. The segment-based Chvátal–Gomory rank 1 valid inequalities are then redundant, but we have observed that their addition helps in solving the model more efficiently. By contrast, the point-based Chvátal–Gomory rank 1 valid inequalities are no more redundant, since the y_a^q variables are now continuous; in particular, their addition allows us to derive a tighter LP relaxation.

To compute tight approximations to $v(\overline{EP+})$, we propose two incremental strategies that are called one after the other and combined to produce the best approximate lower bound. The first strategy, called the *Lagrangian strategy* (or *LAG*), initializes the Lagrange multipliers π to the values obtained when solving $\overline{BS}$, the model defined by (15)–(17), with a state-of-the-art LP solver. The strategy then updates the Lagrange multipliers π by a subgradient method (to be detailed in Section 3.3) that derives tight approximations to $v(\overline{ES})$ and $v(\overline{ES+})$. As a final step, strategy *LAG* solves the Lagrangian subproblem defined in Section 3.1 by using the best values for the Lagrange multipliers π found so far and by setting to zero the Lagrange multipliers β associated with the linking Equations (54). The second strategy, called the *LP-based strategy* (or *LPS*), initializes the Lagrange multipliers π to the values obtained when solving an LP-based approximation to $v(\widehat{ES+})$ restricted to single-node cuts. The strategy then computes values for the Lagrange multipliers β by solving an LP-based approximation to $v(\widehat{EP+})$. The final step of strategy *LPS* solves the Lagrangian subproblem of Section 3.1 by using the best Lagrange multipliers π and β found so far. By combining these two incremental strategies, we obtain a unified procedure that makes use of tight approximations to the bounds defined in Section 2.4, i.e., $v(\overline{ES+})$, $v(\widehat{ES+})$, $v(\widehat{EP+})$, and $v(\overline{EP+})$.

The *Lagrangian dual optimization* procedure is outlined as follows:

1. *Lagrangian strategy (LAG).*

(a) Compute $v(\overline{BS})$; let π^0 be the optimal Lagrange multipliers obtained from the optimal LP dual solution.

(b) Given initial Lagrange multipliers π^0, apply a subgradient method to find an approximation to $v(\overline{ES})$; let π^1 be the best Lagrange multipliers found by the subgradient method.

(c) Given initial Lagrange multipliers π^1, apply a subgradient method to find an approximation to $v(\overline{ES+})$; let π^2 be the best Lagrange multipliers found by the subgradient method.

(d) Given Lagrange multipliers π^2, find an approximation to $v(\overline{EP+})$ by solving the *MIP relaxation* of the Lagrangian subproblem $LAG_P(\pi^2, 0)$, as outlined above.

2. *LP-based strategy (LPS).*

(a) Compute an approximation to $v(\widehat{ES+})$ by solving the LP relaxation obtained by restricting the segment-based Chvátal–Gomory rank 1 valid inequalities (27)–(28) to *single-node cuts*; let π^3 be the optimal Lagrange multipliers obtained from the optimal LP dual solution.

(b) Given Lagrange multipliers π^3, find an approximation to $v(\widehat{EP+})$ by solving the *LP relaxation of the Lagrangian subproblem* obtained from model (49)–(58) by relaxing the flow conservation Equations (50) (with Lagrange multipliers $\pi = \pi^3$) and by replacing the point-based single-node cut-set Equations (56) with the point-based Chvátal–Gomory rank 1 valid inequalities (25)–(26) restricted to *single-node cuts*; let π^4 and β^4 be the optimal Lagrange multipliers obtained from the optimal LP dual solution.

(c) Given Lagrange multipliers π^4 and β^4, find an approximation to $v(\overline{EP+})$ by solving the *MIP relaxation* of the Lagrangian subproblem $LAG_P(\pi^4, \beta^4)$, as outlined above.

3. Return as the approximation to $v(\overline{EP+})$ the best of the two approximations found in steps 1(d) and 2(c).

A few remarks are in order to fully understand the procedure:

- As shown in our computational experiments reported in Section 5, the computations of $v(\overline{BS})$ (step 1(a)) and $v(\overline{ES})$ (step 1(b)) are extremely fast. Our experiments also confirm that the subgradient method used in steps 1(b) and 1(c) generally performs better when it is provided with "good" initial Lagrange multipliers. These observations explain the incremental approach used in the Lagrangian strategy.
- Our experiments, reported in Section 5, show that the MIP relaxation used in steps 1(d) and 2(c) is solved efficiently, but requires a much more significant time than the Lagrangian subproblem used to compute the approximation to $v(\overline{ES+})$. In particular, although the subgradient optimization algorithm is both efficient and effective for computing this lower bound, it is not practical for computing an approximation to $v(\overline{EP+})$. On one hand, the computing times become prohibitive, because of the increased number of Lagrange multipliers and because of the difficulty in solving the Lagrangian subproblems. On the other hand, the lower bound obtained by the combination of the two incremental strategies is already very effective, to the point that the subgradient optimization algorithm provides only minor bound improvement, as shown in Section 5. These observations explain why we solve only one Lagrangian subproblem in steps 1(d) and 2(c), instead of using the subgradient method.
- In step 2(a), we solve the corresponding LP relaxation by using a state-of-the-art LP solver. Another approach would be to solve the same LP relaxation by using the subgradient optimization algorithm in conjunction with the Lagrangian relaxation of the flow conservation equations. At first, this approach appears very similar to the one used to compute the approximation to $v(\overline{ES+})$. There is a major difference, however: the resulting Lagrangian subproblem is defined in terms of continuous variables only. As a consequence, the property that, for each arc $a \in A$ and segment $s \in S_a$, $\sum_{k \in K} x_a^{ks} > 0$ only if $y_a^s = 1$ is not true anymore; instead, we have that, for each arc $a \in A$ and segment $s \in S_a$, $\sum_{k \in K} x_a^{ks} > 0$ only if $y_a^s > 0$. This apparently minor modification makes a huge difference when solving the Lagrangian subproblem, since it is not possible to solve it through decomposition into a collection of continuous knapsack problems followed by the solution of a model expressed only in terms of the y_a^s variables. Instead, the Lagrangian subproblem would be solved as a nondecomposable LP model involving both the flow variables x_a^{ks} and the segment-based variables y_a^s. As a result, the direct solution of the LP relaxation in step 2(a) is computationally preferable to the Lagrangian relaxation approach for approximating the same bound.
- Instead of performing steps 2(a) and 2(b), we could have solved the approximation to $v(\widehat{EP+})$ defined by model $\widehat{EP+}$ restricted to single-node point-based Chvátal–Gomory rank 1 valid inequalities (25)–(26). As shown in our computational experiments reported in Section 5, the computing times for solving this LP relaxation are prohibitive. By contrast, the LP-based incremental strategy computes effective lower bounds, while ensuring low computational requirements. These observations explain the incremental approach used in the LP-based strategy.
- When solving the LP relaxation of the Lagrangian relaxation in step 2(b), we also add the segment-based Chvátal–Gomory rank 1 valid inequalities (27)–(28) restricted to single-node cuts. Although these inequalities are redundant, we observed that their addition generally improves the computing times.
- The combination of the two incremental strategies has the nice characteristic that it preserves most of the bound relationships of Proposition 7, where each theoretical bound is replaced by its approximation given by the procedure. Indeed, the inequalities $v(\overline{EP+}) \geq v(\overline{ES+})$ and $v(\overline{EP+}) \geq v(\widehat{EP+})$ are guaranteed by steps 1(d) and 2(c), respectively. The inequality $v(\widehat{EP+}) \geq v(\widehat{ES+})$ follows from step 2(b). Because approximations of the two bounds $v(\overline{ES+})$ and $v(\widehat{ES+})$ are computed independently, only the inequality $v(\overline{ES+}) \geq v(\widehat{ES+})$ might be violated, although, in practice, it is generally satisfied.

3.3. Subgradient Method

The subgradient method is a simple implementation of the classical Held–Wolfe–Crowder approach (Held, Wolfe, and Crowder 1974). At every iteration $t > 0$, the new Lagrange multipliers $\pi(t)$ are computed by taking a step $\alpha(t)$ in the direction of a subgradient $\gamma(t)$: $\pi(t) = \pi(t-1) + \alpha(t)\gamma(t)$. The subgradient $\gamma(t)$ is equal to the difference between the right- and left-hand sides of the flow conservation equations evaluated at the optimal solution of the current Lagrangian subproblem. The step is computed as $\alpha(t) = \lambda(t)(v^* - v \cdot (\pi(t-1)))/\|\gamma(t)\|^2$, where $v(\pi(t-1))$ is the Lagrangian lower bound associated with Lagrange multipliers $\pi(t-1)$, v^* is an upper bound on the optimal value of the Lagrangian dual (we use the best upper bound obtained by the Lagrangian heuristic method described in Section 4), $\lambda(t)$ is a parameter that takes its initial value $\lambda(0)$ in the interval $(0,2]$ and is typically decreased (divided by $\omega_1 > 1$) every time $v(\pi(t))$ has not improved for some number ω_2 of consecutive iterations. The algorithm stops when the lower bound has not improved for some number ω_3 of consecutive iterations or when a maximum number ω_4 of iterations has been attained. In our experiments, we use the following values for these parameters: $\lambda(0) = 1$, $\omega_1 = 2$, $\omega_2 = 15$, $\omega_3 = 30$, and $\omega_4 = 400$.

4. Lagrangian Heuristic

In this section, we present the Lagrangian heuristic method used to compute feasible solutions to the PMFI, yielding upper bounds on the optimal value of the problem. As in any Lagrangian heuristic method, we make use of the values $\bar{y}_a^s$ obtained from solving any Lagrangian subproblem to derive feasible solutions to the PMFI. To derive effective feasible solutions from the Lagrangian subproblem solutions, we use a *slope scaling* procedure, which has been used successfully in the context of single-commodity (Kim and Pardalos 1999; Kim and Pardalos 2000) and multicommodity (Crainic, Gendron, and Hernu 2004) network flow problems. The novelty here is to embed it within a traditional Lagrangian heuristic method that uses the Lagrangian subproblem primal solutions to guide the search for feasible solutions. Section 4.1 gives the details of the slope scaling procedure, which solves a sequence of linear *continuous* multicommodity network flow problems. In Section 4.2, we explain how to use the solutions obtained by the slope scaling procedure to drive the search for effective *integer* multicommodity flow solutions. Section 4.3 presents how the slope scaling procedure is combined with the Lagrangian dual optimization approach described in Section 3.2 to produce a complete Lagrangian relaxation method that generates lower and upper bounds on the optimal value of the PMFI.

4.1. Slope Scaling Procedure

The guiding principle of a slope scaling approach is extremely simple: given a feasible solution to a nonlinear network flow problem, the objective function is linearized in such a way that, if the resulting linear network flow problem provides as optimal solution the same feasible solution, the optimal value of the linear problem corresponds to the nonlinear objective function value.

At every step of the slope scaling approach, we consider the following linear multicommodity network flow problem, denoted *MF*, where the linear arc costs $\bar{c}_a$ are to be adjusted using the slope scaling guiding principle:

$$v(MF) = \min \sum_{a \in A} \sum_{k \in K} \bar{c}_a x_a^k, \tag{66}$$

subject to (2), (6), and

$$\sum_{k \in K} x_a^k \le u_a, \quad a \in A. \tag{67}$$

This problem can be solved with any existing efficient method for linear multicommodity network flow problems; in our implementation, we use a state-of-the-art LP solver. Note that the costs and the capacities do not depend on the commodities. As a consequence, when some commodities share the same origin (or the same destination), they can be aggregated into a single commodity, thus reducing the size of the problem.

Suppose we solve *MF* and obtain a feasible solution with flows $\bar{x}_a^k$; for each arc a, we then let $\bar{x}_a = \sum_{k \in K} \bar{x}_a^k$ and $\bar{s}_a$ the segment of the piecewise linear objective function of PMFI such that $x_a^{\bar{s}_a} = \bar{x}_a$. The linear cost $\bar{c}_a$ at the next slope scaling iteration is then adjusted using the formula

$$\bar{c}_a = \begin{cases} c_a^{\bar{s}_a} + (f_a^{\bar{s}_a}/\bar{x}_a), & \text{if } \bar{x}_a > 0, \\ \bar{c}_a, & \text{if } \bar{x}_a = 0. \end{cases}$$

Thus, when there is flow on arc a and the same flow appears again in the solution obtained after computing *MF*, the associated linear cost reflects the piecewise linear cost: $(c_a^{\bar{s}_a} + (f_a^{\bar{s}_a}/\bar{x}_a))\bar{x}_a = c_a^{\bar{s}_a} x_a^{\bar{s}_a} + f_a^{\bar{s}_a}$. When there is no flow on arc a, the slope scaling update must intuitively assign a sufficiently large linear cost to arc a, but not too large to avoid "freezing" the solution too early. The cost used at the previous iteration was precisely large enough for the flow not to transit through arc a and is thus used for that purpose.

The slope scaling approach iterates between the solution of *MF* and the linear cost update until the same solution is repeated or a maximum number of iterations is achieved (we use 50 in our implementation). To start this iterative process, we need initial linear costs; this is where we use the Lagrangian optimal solutions $\bar{y}_a^s$ in the spirit of a classical Lagrangian heuristic method.

More precisely, we initialize the linear cost $\bar{c}_a$ on each arc with the formula

$$\bar{c}_a = (c_a + f_a/u_a)\left(1 + M\left(1 - \sum_{s \in S_a} \bar{y}_a^s\right)\right),$$

where $c_a = c_a^{|S_a|}$, $f_a = f_a^{|S_a|}$, and M is a sufficiently large number (we use 10 in our implementation). When arc a is used in the Lagrangian solution, i.e., $\sum_{s \in S_a} \bar{y}_a^s = 1$, the rationale behind this formula is then to use the linear lower approximation $c_a + f_a/u_a$ that corresponds to the line connecting the origin to the objective function value at full usage of the arc, i.e., its capacity $u_a = b_a^{|S_a|}$. When arc a is not used in the Lagrangian solution, i.e., $\sum_{s \in S_a} \bar{y}_a^s = 0$, the linear cost should be sufficiently large to reflect the fact that arc a is not "interesting" according to the Lagrangian solution.

4.2. Deriving Integer Solutions

Any solution $\bar{x}$ derived from solving *MF* is feasible for the PMFI if $\bar{x}$ is integer. The upper bound corresponding to this feasible solution is $v(\bar{x}) = \sum_{a \in A}(c_a^{\bar{s}_a}\bar{x}_a + f_a^{\bar{s}_a})$. During any call to the slope scaling procedure, we thus keep track of the best integer solution $\bar{x}$ with its corresponding value $v(\bar{x})$. To prevent against the possibility that no integer solution $\bar{x}$ is found during an entire call to the slope scaling procedure, we also keep track of the best noninteger solution, the one with the best piecewise linear objective function value $v(\bar{x}) = \sum_{a \in A}(c_a^{\bar{s}_a}\bar{x}_a + f_a^{\bar{s}_a})$. If the best integer and noninteger solutions are "close" enough (in our implementation, if they differ by less than 1%), the slope scaling procedure is stopped; otherwise (if no integer solution is found or only a "poor" integer solution is found), we then solve again the *MF* that gave the best noninteger solution, but this time with the addition of the integrality constraint on the total flows. If the resulting integer solution is better than the currently best integer solution, it replaces it. By proceeding in this way, we also ensure that we obtain an integer solution $\bar{x}$ of value $v(\bar{x})$ at the end of the slope scaling procedure.

Note that $v(\bar{x})$ is the piecewise linear objective function value of the integer solution $\bar{x}$ derived from solving *MF*. Thus, $\bar{x}$ is optimal when using the linear costs adjusted with the slope scaling formula, but it is not necessarily the best solution for the restriction of the PMFI that uses the same arcs as $\bar{x}$ at the same lower and upper limits. To determine this solution, we solve the following integer multicommodity flow problem, $IMF(\bar{x})$, using the best integer solution $\bar{x}$ found by the slope scaling procedure

$$v(IMF(\bar{x})) = \sum_{a \in A} f_a^{\bar{s}_a} + \min \sum_{a \in A} c_a^{\bar{s}_a} x_a, \tag{68}$$

subject to (2), (6), and

$$\sum_{k \in K} x_a^k = x_a \text{ integer}, \quad a \in A, \tag{69}$$

$$l_a^{\bar{s}_a} \le x_a \le v_a^{\bar{s}_a}, \quad a \in A. \tag{70}$$

Note that this problem always has a feasible solution, namely, $\bar{x}$. By optimizing over the "true" piecewise linear objective function, we can thus only improve on the value $v(\bar{x})$. Thus, as an *intensification step* after every call to the slope scaling procedure, we solve $IMF(\bar{x})$ and use its optimal value $v(IMF(\bar{x}))$ to possibly update the best upper bound v^*.

4.3. Combining Slope Scaling and Lagrangian Dual Optimization

The slope scaling procedure is called just after computing $v(\overline{BS})$ in the Lagrangian strategy (i.e., step 1(a) of the Lagrangian dual optimization procedure presented in Section 3.2), this time using the optimal solution $\bar{y}_a^s$ to $\overline{BS}$; this provides an initial upper bound v^* given to the subgradient method used in step 1(b) of the procedure. Subsequently, we call the slope scaling procedure in two modes: (1) in conjunction with the subgradient method used in steps 1(b) and 1(c); (2) as part of solving the MIP Lagrangian subproblem in steps 1(d) and 2(c). Within each of these two modes, the slope scaling procedure is called several times, thus producing a pool of "good" feasible solutions, out of which we apply a *postoptimization procedure* that produces an improved feasible solution.

We use the following rules to decide when to call the slope scaling procedure in conjunction with the subgradient method in steps 1(b) and 1(c):

- Call the slope scaling procedure using the solution $\bar{y}_a^s$ that corresponds to the best Lagrangian subproblem obtained at the end of the step.
- Call the slope scaling procedure using solution $\bar{y}_a^s$ if the lower bound has improved "significantly" since the last time the upper bound was computed; the "significant" improvement is measured by the test $(v(last) - v(current))/v(last) > \delta$, where δ is a parameter (set to 1%) and $v(last)$ and $v(current)$ are, respectively, the lower bound computed at the current iteration and the lower bound obtained the last time the slope scaling procedure was called.
- Call the slope scaling procedure every nth ($n = 10$) iteration of the subgradient method (to avoid too early "freezing" of upper bound computations in case δ is too large).

The slope scaling procedure is thus called several times in conjunction with the subgradient method, both in steps 1(b) and 1(c). The pool of feasible solutions thus obtained is used in the postoptimization procedure outlined below.

When solving the MIP Lagrangian subproblem in steps 1(d) and 2(c), we use a state-of-the-art MIP solver that implements a branch-and-bound (B&B) algorithm. For each integer solution found during the exploration of the B&B tree, which provides binary values $\bar{y}_a^s$ for the segment-based variables, we invoke the slope scaling procedure. Thus, at the end of each of steps 1(d)

and 2(c), we give as input to the postoptimization procedure the pool of feasible solutions obtained from calling the slope scaling procedure heuristic multiple times, one for each integer solution.

At the end of steps 1(b), 1(c), 1(d), and 2(c) of the Lagrangian dual optimization procedure, the following postoptimization procedure is applied. We assume we have kept in memory a pool of the feasible solutions found during the corresponding step by the slope scaling procedure. Out of the solutions in this pool, we extract only the best solutions $\bar{x}$, i.e., those with a value $v(IMF(\bar{x}))$ sufficiently close to the best upper bound v^* (in our tests, we consider $\bar{x}$ if the relative gap between $v(IMF(\bar{x}))$ and v^* is less than 1%). We denote by $\mathcal{P}$ the pool consisting of these best solutions. We then define $\bar{A} = \{a \in A \,|\, \bar{x}_a = 0, \forall \bar{x} \in \mathcal{P}\}$, the subset of the arcs for which every solution in $\mathcal{P}$ displays no flow circulating on these arcs. We solve the MIP formulation $BS(\bar{A})$, which is the basic segment-based model of the PMFI restricted to the arcs in $A \setminus \bar{A}$. This model is defined by (1)–(7) with the addition of the constraints

$$\sum_{s \in S_a} y_a^s = 0, \quad a \in \bar{A}. \tag{71}$$

It is obvious that each solution $\bar{x} \in \mathcal{P}$ defines a feasible solution to $BS(\bar{A})$. As mentioned above, $IMF(\bar{x})$ is optimizing over the true piecewise linear objective function, but it does so by fixing the segment of the cost function for each arc. By contrast, $BS(\bar{A})$ fixes only the arcs that are not used in every solution $\bar{x} \in \mathcal{P}$, while optimizing over all segments of the cost function for the other arcs. Hence, the best value $v(IMF(\bar{x}))$ for any $\bar{x} \in \mathcal{P}$, which is given as the best incumbent value when starting to solve $BS(\bar{A})$, can only be improved as a result of solving $BS(\bar{A})$. The output of this postoptimization procedure is the best feasible solution found this way, which is used to improve on the value v^* obtained at the end of the Lagrangian heuristic method.

5. Computational Experiments

We present computational results on a large set of randomly generated instances with different cost structures. Our objective is twofold:

- To assess the performance of the Lagrangian relaxation method. To this purpose, we compare its results to those obtained by a state-of-the-art LP/MIP solver. This way, we are able to compare the Lagrangian-based lower bounds with their corresponding equivalent LP relaxation bounds for these models. We also compare the upper bounds from the Lagrangian heuristic method with those from the MIP solver with a limited CPU (central processing unit) time.
- To assess the quality of the different formulations with respect to various network configurations and cost structures. In particular, we are interested in evaluating the improvements in the bounds obtained by discretization combined with the addition of cut-set inequalities and flow disaggregation.

The Lagrangian relaxation method was implemented in C++, using CPLEX version 12.5.1.0 as the MIP/LP solver. The code was compiled with g++ 4.4.7 and run on an Intel Xeon X5675, operating at 3.07 GHz, in single-threaded mode. Before analyzing the results in Section 5.3, we first describe the set of instances used in our experiments in Section 5.1 and then present the design of the experiments in Section 5.2.

5.1. Set of Instances

We obtained the problem instances from a network generator similar to the one described in Crainic, Frangioni, and Gendron (2001) for multicommodity capacitated fixed-charge problems. When provided with target values for $|N|$ and $|A|$, this generator creates arcs by connecting two randomly selected nodes (no parallel arcs are allowed). The commodities are generated as follows: given target values $|O| < |N|$ and $|D| \le |N| - |O|$, the number of origins and destinations, respectively, it selects the origins at random. Then, for each origin, it selects $|D|$ destinations at random among the nodes in $N \setminus O$, where O is the set of origins. The number of commodities is therefore equal to $|K| = |O| \times |D|$. The generator also creates the variable costs, capacities, and demands as uniformly distributed over user-provided intervals. The capacities can then be scaled by adjusting the capacity ratio, $C = |A|T/\sum_{a \in A} u_a$, to user-provided values (in this formula, $T = \frac{1}{2}\sum_{k \in K}\sum_{i \in N} |d_i^k|$, the total demand flowing through the network). When C equals 1, the average arc capacity $\sum_{a \in A} u_a/|A|$ equals the total demand, and the network is lightly capacitated. It becomes more tightly capacitated as C increases.

For each network, we generated two cost structures, as in Croxton, Gendron, and Magnanti (2007): concave and nonconcave. For both types of instances, we provided the maximum number of segments, S, of the cost function as a parameter. For concave instances, we randomly generated a set of decreasing variable costs within the specified interval for each arc. We also set $b_a^s = s^2 D/S^2$, for each arc a, so that the segment length increases as s increases, as is typical of transportation costs (Balakrishnan and Graves 1989). We then adjusted the number of segments on each arc a so that $b_a^{|S_a|} = \min\{T, u_a\}$. Given variable costs, break points, and f_a^1, the initial fixed cost, we can then compute the appropriate fixed costs for the remaining segments so that the resulting function is concave. We obtained nonconcave instances by imposing $b_a^s = \lceil T/S \rceil$, for each arc a, so that each segment is of equal size, except the last one. We then adjusted the number of segments to account for the capacities on the arcs by eliminating segments beyond any arc's capacity. The network generator provided the variable costs,

which are not necessarily decreasing, as in the concave case. Given an initial fixed cost f_a^1 for each arc, we compute the remaining fixed costs as $f_a^s = s f_a^1$. Thus, when $f_a^1 > 0$, we obtain a staircase cost function (with variable costs). In our experiments, we consider four different cost structures: concave and nonconcave, with $f_a^1 = 100$ and 1,000.

We classify the instances according to the number of commodities, which is one of the main characteristics in assessing the difficulty of solving the models. We consider three classes of instances:

- *Small instances* ($|K| = 25$). The following network dimensions are used for instances in this class: $(|N|, |A|) = (20, 75)$, $(20, 100)$, $(25, 100)$, $(25, 150)$. For each of these four combinations, we select five origins and, for each of them, five destinations, i.e., $|O| = 5$ and $|D| = 5$, so the number of commodities is $|K| = 25$.
- *Medium instances* ($|K| = 50$). We use the same four network dimensions as for Small instances. We then select $|O| = 5$ origins at random and, for each of them, we select $|D| = 10$ destinations at random among the $|N| - |O|$ remaining nodes. The number of commodities is therefore $|K| = 50$.
- *Large instances* ($|K| = 100$). We use the same four network dimensions as for Small and Medium instances. These instances have $|K| = 100$ commodities obtained by selecting $|O| = 10$ origins and, for each of them, $|D| = 10$ destinations among the $|N| - |O|$ remaining nodes.

Thus, in each category of instances, there are four combinations of network dimensions $(|N|, |A|)$. For each of these combinations, we generate 24 instances by varying the different parameters in a similar way as in Croxton, Gendron, and Magnanti (2007): in addition to the four different cost structures, we vary the number of segments $(4, 6, 8)$ and the capacity ratio C $(2, 4)$. Our generation procedure thus results into 96 instances in each category, for a total of 288 instances.

5.2. Design of the Experiments

Our experiments consider four MIP formulations of the PMFI:

- *BS*: The basic segment-based model defined by (1)–(7).
- *ES*: The extended segment-based model defined by (49)–(53) and (58).
- *ES+*: This is model *ES* with the addition of the segment-based Chvátal–Gomory rank 1 valid inequalities (27)–(28) restricted to single-node cuts.
- *EP+*: This is model *ES* with the addition of the constraints defining the point-based variables, (54), (55), and (57), along with the point-based Chvátal–Gomory rank 1 valid inequalities (25)–(26) restricted to single-node cuts.

The following methods are used to compute lower and upper bounds based on these four formulations:

- *LD*, the Lagrangian dual optimization procedure presented in Section 3.2. The following bounds, approximated by this method, are reported: $v(\overline{ES})$ (step 1(b)), $v(\overline{ES+})$ (step 1(c)), $v(\overline{EP+})$ (steps 1(d), 2(c), and 3). When solving the Lagrangian subproblems used to compute $v(\overline{ES+})$ and $v(\overline{EP+})$, the B&B method of CPLEX (with default options) is used. In addition, the Chvátal–Gomory rank 1 valid inequalities (25)–(28) for $p > 1$ are declared as *lazy* constraints, which ensures that CPLEX is adding only a small number of them, in a cutting-plane fashion.
- *LP*, the LP solver of CPLEX (with default options). The following bounds are computed by this method: $v(\overline{BS})$, using the model defined by (15)–(17); $v(\overline{ES})$, using the LP relaxation of MIP model *ES*; $v(\widehat{ES+})$, using the LP relaxation of MIP model *ES+*; and $v(\widehat{EP+})$, using the LP relaxation of MIP model *EP+*.
- BB_0, the root node computations of the B&B method of CPLEX (with default options). Lower bounds for the four MIP formulations are computed with this method. Because of CPLEX preprocessing and cutting-plane procedures, these lower bounds dominate those computed by method *LP*. Again, the Chvátal–Gomory rank 1 valid inequalities (25)–(28) for $p > 1$ are declared as *lazy* constraints.
- *BB*, the B&B method of CPLEX (with default options) performed for a limit of one hour. This method generates both lower and upper bounds based on the four MIP formulations, except for the instances for which CPLEX cannot find any feasible solution within the limit of one hour, in which case only a lower bound is obtained.
- *LH*, the Lagrangian heuristic method described in Section 4. Upper bounds are computed based on formulations *ES* (in step 1(c) of method *LD*), *ES+* (in step 1(c) of method *LD*), and *EP+* (in steps 1(d), 2(c), and 3 of method *LD*).

For each instance I, these five methods are performed for the four models, producing several lower and upper bounds on the optimal value of the PMFI for instance I. The best of these upper bounds, denoted $v^*(I)$, is used as a reference for computing lower and upper bound gaps. More precisely, for any bound (lower or upper) $v(I)$, we compute the ratio with respect to the best-known upper bound $v^*(I)$ for each instance, i.e., $GAP(I) = v(I)/v^*(I)$, which implies that values closer to 1 are better.

5.3. Analysis of the Computational Results

We first analyze the results obtained with different strategies to approximate $v(\overline{EP+})$. We compare the following approaches:

- *LAG*, the Lagrangian strategy performed in steps 1(a) to 1(d) of the Lagrangian dual optimization procedure presented in Section 3.2.

Table 4. Strategies to Approximate $v(\overline{EP+})$: Lower Bound GAP, Upper Bound GAP, CPU

$(\|N\|, \|A\|)$	*LAG*	*LPS*	*LAG + LPS*	*LAG + SUB*
(20, 75)	0.93, 1.03, 16	0.93, 1.03, 14	0.94, 1.02, 30	0.93, 1.03, 13,829
(20, 100)	0.94, 1.02, 22	0.93, 1.04, 40	0.94, 1.02, 62	0.94, 1.02, 24,391
(25, 100)	0.92, 1.01, 17	0.93, 1.04, 26	0.93, 1.01, 44	0.92, 1.01, 31,380
(25, 150)	0.92, 1.05, 36	0.92, 1.07, 33	0.92, 1.04, 69	0.92, 1.05, 75,612
Small	0.93, 1.03, 23	0.93, 1.05, 28	0.93, 1.02, 51	0.93, 1.03, 36,303
(20, 75)	0.93, 1.01, 33	0.94, 1.03, 37	0.94, 1.01, 70	0.93, 1.01, 17,769
(20, 100)	0.94, 1.01, 13	0.94, 1.03, 32	0.94, 1.01, 45	0.94, 1.01, 31,774
(25, 100)	0.93, 1.02, 34	0.93, 1.04, 28	0.93, 1.02, 62	0.93, 1.02, 34,363
(25, 150)	0.91, 1.02, 27	0.91, 1.05, 65	0.91, 1.02, 92	0.91, 1.02, 80,222
Medium	0.93, 1.02, 27	0.93, 1.04, 41	0.93, 1.02, 68	0.93, 1.01, 41,302
(20, 75)	0.95, 1.01, 19	0.96, 1.01, 50	0.96, 1.00, 69	0.95, 1.01, 5,374
(20, 100)	0.95, 1.00, 19	0.95, 1.01, 66	0.95, 1.00, 85	0.95, 1.00, 14,613
(25, 100)	0.94, 1.01, 40	0.95, 1.01, 127	0.95, 1.00, 166	0.94, 1.01, 25,644
(25, 150)	0.94, 1.01, 188	0.95, 1.03, 165	0.95, 1.00, 353	0.94, 1.01, 43,103
Large	0.95, 1.01, 67	0.95, 1.02, 102	0.95, 1.00, 168	0.95, 1.01, 22,184

- *LPS*, the LP-based strategy performed in steps 2(a) to 2(c) of the Lagrangian dual optimization procedure.
- *LAG + LPS*, the whole Lagrangian dual optimization procedure, combining the two previous strategies.
- *LAG + SUB*, the same as strategy *LAG*, except that the solution to the single Lagrangian subproblem in step 1(d) is replaced by a call to an adaptation of the subgradient method described in Section 3.3, where both π and β multipliers are adjusted.

Table 4 summarizes the computational results obtained for all instances. Lower and upper bound GAPs and CPU times in seconds are reported on average for the four network dimensions $(|N|, |A|) = (20, 75), (20, 100), (25, 100), (25, 150)$ (each class contains 24 instances), as well as for the 96 instances in each class, Small ($|K| = 25$), Medium ($|K| = 50$), and Large ($|K| = 100$).

These results show that all of the strategies generate similar lower bounds, with a slight edge for the combined approach *LAG + LPS*. We observed that, for some instances, *LAG* produces better lower bounds than *LPS*, whereas for other instances, the opposite is true. Thus, by combining the two strategies, we obtain better overall lower bounds. Although strategy *LAG* generates better upper bounds than *LPS* on average, the same observation holds for the upper bounds: no dominance exists across all instances, which implies that the combination of the two strategies produces better overall upper bounds. This can be seen on the average values for some of the instance classes, for example, Small instances with size (20, 75) for both the lower bound and the upper bound GAPs and Small instances with size (25, 150) for the upper bound GAP. As shown in column *LAG + SUB*, the subgradient method does not help improve the bounds and its computing times are prohibitive. These results justify solving a single Lagrangian subproblem instead of a call to the subgradient method in step 1(d) of the Lagrangian dual optimization procedure.

Table 5 displays the lower bound GAPs and the CPU times in seconds to compute the different lower bounds (excluding the times for upper bound computations), averaged for each problem class as in Table 4, for each combination of model and lower bounding method described in Section 5.2 (with the exception of *BB* whose lower bound results are shown in Table 7).

From these results, we draw the following conclusions:

- As expected, formulation *BS* is weak, producing lower bound gaps around 25% on average. The B&B method of CPLEX at the root node reduces these gaps by 5%–10%, thanks to its preprocessing and cutting-plane procedures. The extended models, on the other hand, improve these gaps by 15%–20%, which shows the strength of the extended forcing constraints (29).
- The Lagrangian dual optimization procedure provides effective lower bound approximations, independently of the formulation. As can be seen from column *ES*, the subgradient method provides a tight approximation (within 1% on average) of the theoretical bound $v(\overline{ES})$ computed by the LP solver of CPLEX. It is also noteworthy that CPLEX provides only slight improvements (on the order of 1%–2%) by adding its sophisticated preprocessing and cutting-plane features at the root node (method BB_0).
- For the same model, the Lagrangian dual optimization procedure is in general significantly faster than the LP solver of CPLEX and the difference in computing times increases with the number of commodities. In particular, method *LD* solves the extended models for Medium and Large instances much faster than *LP*. For the same instances, the computing times for *LD* are also generally better than those for BB_0, except for model *EP+* where the CPU times are similar.
- Formulation *ES* produces gaps around 5%–10% on average, with better results on Large instances.

Table 5. Lower Bounds: Lower Bound GAP, CPU

(|N|, |A|)	Algo	*BS*	*ES*	*ES+*	*EP+*
(20,75)	*LD*	0.75, 0	0.92, 0	0.93, 7	0.94, 28
	LP	0.75, 0	0.93, 1	0.93, 1	0.93, 65
	BB_0	0.83, 0	0.94, 2	0.95, 2	0.95, 13
(20,100)	*LD*	0.74, 0	0.91, 0	0.94, 9	0.94, 49
	LP	0.74, 0	0.92, 1	0.92, 1	0.92, 56
	BB_0	0.84, 0	0.94, 2	0.95, 3	0.95, 18
(25,100)	*LD*	0.74, 0	0.92, 0	0.92, 4	0.93, 40
	LP	0.74, 0	0.92, 1	0.92, 2	0.92, 273
	BB_0	0.84, 1	0.94, 3	0.95, 3	0.95, 25
(25,150)	*LD*	0.70, 0	0.91, 1	0.92, 10	0.92, 68
	LP	0.70, 0	0.91, 3	0.92, 4	0.92, 204
	BB_0	0.79, 1	0.92, 5	0.94, 8	0.94, 43
Small	*LD*	0.73, 0	0.92, 0	0.93, 8	0.93, 46
	LP	0.73, 0	0.92, 2	0.92, 2	0.92, 150
	BB_0	0.83, 1	0.94, 3	0.95, 4	0.95, 25
(20,75)	*LD*	0.76, 0	0.92, 1	0.93, 9	0.94, 68
	LP	0.76, 0	0.93, 3	0.93, 4	0.93, 124
	BB_0	0.83, 1	0.94, 5	0.95, 7	0.95, 28
(20,100)	*LD*	0.73, 0	0.93, 0	0.94, 4	0.94, 42
	LP	0.73, 0	0.94, 6	0.94, 7	0.94, 228
	BB_0	0.83, 1	0.94, 9	0.95, 11	0.96, 39
(25,100)	*LD*	0.70, 1	0.92, 1	0.93, 9	0.93, 60
	LP	0.70, 1	0.93, 7	0.93, 8	0.93, 433
	BB_0	0.79, 2	0.93, 12	0.95, 16	0.95, 54
(25,150)	*LD*	0.66, 1	0.90, 2	0.91, 4	0.91, 90
	LP	0.66, 1	0.91, 25	0.92, 35	0.92, 1,294
	BB_0	0.76, 3	0.91, 28	0.92, 49	0.93, 140
Medium	*LD*	0.71, 1	0.92, 1	0.93, 7	0.93, 65
	LP	0.71, 1	0.93, 10	0.93, 14	0.93, 520
	BB_0	0.80, 2	0.93, 14	0.94, 21	0.95, 65
(20,75)	*LD*	0.80, 1	0.95, 2	0.95, 3	0.96, 64
	LP	0.80, 1	0.96, 12	0.96, 18	0.96, 310
	BB_0	0.87, 1	0.96, 18	0.96, 17	0.96, 54
(20,100)	*LD*	0.78, 1	0.95, 1	0.95, 2	0.95, 72
	LP	0.78, 1	0.95, 24	0.95, 35	0.95, 1,169
	BB_0	0.82, 3	0.96, 34	0.96, 40	0.96, 89
(25,100)	*LD*	0.74, 1	0.94, 1	0.94, 2	0.95, 123
	LP	0.74, 1	0.95, 39	0.95, 63	0.95, 1,095
	BB_0	0.80, 3	0.95, 54	0.95, 59	0.95, 130
(25,150)	*LD*	0.76, 1	0.94, 3	0.94, 3	0.95, 189
	LP	0.76, 1	0.95, 67	0.95, 106	0.95, 1,103
	BB_0	0.81, 7	0.95, 81	0.95, 107	0.95, 227
Large	*LD*	0.77, 1	0.95, 2	0.95, 3	0.95, 112
	LP	0.77, 1	0.95, 36	0.95, 56	0.95, 919
	BB_0	0.83, 4	0.96, 47	0.96, 56	0.96, 125

Small improvements (on the order of 1%–2%) are obtained by adding segment-based cut-set inequalities through formulation *ES+*. When adding to this last model the point-based cut-set inequalities, a similar behavior is observed.

We now compare the upper bounds obtained by the Lagrangian heuristic method of Section 4, *LH*, with those computed by the B&B method of CPLEX, *BB*, with a limit of one hour. Table 6 displays three measures for each class of instances: Nfeas, the number of instances in the corresponding class for which each method-model combination found a feasible solution; Nopti, the number of instances in the corresponding class for which each method-model combination provided a certificate of optimality; CPU, the computing time in seconds for each method-model combination. We note that, independently of the model used, method *LH* always generates a feasible solution. Moreover, the best solution it generates has never been shown to be optimal for any of the instances. Hence, the values of Nfeas and Nopti are easy to interpret for method *LH*: for 100% of the instances, *LH* found feasible, but nonoptimal solutions.

These results show that, in spite of being the weakest model in terms of the quality of its LP relaxation bound, *BS* gives the best performance for computing upper bounds with the B&B method of CPLEX. In particular, it is the only model for which *BB* finds feasible solutions to all instances. By contrast, the strongest model *EP+* identifies a feasible solution within one hour for only 117 of the 288 instances. Formulations *ES* and *ES+* show intermediate results, with 252 and 277 instances, respectively, for which they could find feasible solutions. The performance in terms of CPU times and number of optimal solutions found are similar: *BS* is generally faster than the other models and is able to prove optimality for a larger number of instances. The only exception is for Small instances for which model *ES+* is slightly better than both *BS* and *ES*, which indicates that the addition of cut-set inequalities can help in solving the problem, at least for instances with few commodities. Overall, these results show that, although the extended models generate much stronger lower bounds than the basic model, their size is an issue for a stand-alone MIP solver like CPLEX and that decomposition methods must be used to exploit their strength. Table 6 also shows that the Lagrangian heuristic is fast, its CPU times being one to two orders of magnitude smaller than those of the B&B method of CPLEX.

As a further comparison between *BB* and *LH*, Table 7 shows the results obtained with the best method-model combination for each of the two methods. For *BB*, as just seen in Table 6, the best model is *BS*, whereas for *LH*, we selected *EP+* as the best model. Indeed, the incremental strategy used when the Lagrangian heuristic is combined with the Lagrangian dual optimization procedure (see Section 4.3) guarantees that the upper bound obtained after solving *EP+* (steps 1(d), 2(c), and 3) dominates any other upper bound found during the course of the Lagrangian heuristic. In practice, we observed that the upper bound found when solving *ES* is already very good, as it is about 1% away from the best feasible solution found by the Lagrangian heuristic when solving *EP+*. Nonetheless, *EP+* is to be preferred, as it produces the best lower and upper bounds, with a modest additional computational effort, as shown in Table 6. Table 7

Table 6. Upper Bounds: Nfeas, Nopti, CPU

(|N|, |A|)	Algo	*BS*	*ES*	*ES+*	*EP+*
(20, 75)	*BB*	24, 20, 1,066	24, 20, 1,260	24, 21, 935	24, 8, 3,175
	LH	24, 0, 0	24, 0, 1	24, 0, 8	24, 0, 30
(20, 100)	*BB*	24, 18, 1,311	24, 18, 1,489	24, 19, 1,307	24, 8, 3,222
	LH	24, 0, 0	24, 0, 0	24, 0, 9	24, 0, 62
(25, 100)	*BB*	24, 16, 1,924	24, 13, 2,361	24, 15, 2,151	22, 0, 3,613
	LH	24, 0, 0	24, 0, 3	24, 0, 7	24, 0, 44
(25, 150)	*BB*	24, 5, 3,103	24, 9, 2,774	24, 8, 2,828	12, 0, 3,632
	LH	24, 0, 0	24, 0, 1	24, 0, 10	24, 0, 69
Small	*BB*	96, 59, 1,851	96, 60, 1,971	96, 63, 1,805	82, 16, 3,411
	LH	96, 0, 0	96, 0, 1	96, 0, 9	96, 0, 51
(20, 75)	*BB*	24, 12, 2,099	24, 10, 2,545	24, 12, 2,295	17, 2, 3,584
	LH	24, 0, 0	24, 0, 2	24, 0, 10	24, 0, 70
(20, 100)	*BB*	24, 9, 2,492	24, 10, 2,554	24, 9, 2,509	17, 2, 3,578
	LH	24, 0, 0	24, 0, 1	24, 0, 5	24, 0, 45
(25, 100)	*BB*	24, 8, 2,914	23, 6, 2,937	24, 8, 3,008	1, 0, 3,644
	LH	24, 0, 1	24, 0, 3	24, 0, 11	24, 0, 62
(25, 150)	*BB*	24, 3, 3,274	18, 2, 3,419	20, 3, 3,364	0, 0, 3,730
	LH	24, 0, 1	24, 0, 3	24, 0, 5	24, 0, 92
Medium	*BB*	96, 32, 2,695	89, 28, 2,864	92, 32, 2,794	35, 4, 3,634
	LH	96, 0, 1	96, 0, 2	96, 0, 8	96, 0, 67
(20, 75)	*BB*	24, 20, 968	21, 12, 2,685	24, 15, 2,193	0, 0, 3,642
	LH	24, 0, 1	24, 0, 3	24, 0, 5	24, 0, 69
(20, 100)	*BB*	24, 11, 2,337	21, 5, 3,382	24, 6, 3,152	0, 0, 3,680
	LH	24, 0, 1	24, 0, 7	24, 0, 8	24, 0, 85
(25, 100)	*BB*	24, 8, 2,742	14, 0, 3,642	22, 1, 3,597	0, 0, 3,721
	LH	24, 0, 1	24, 0, 18	24, 0, 19	24, 0, 166
(25, 150)	*BB*	24, 3, 3,385	11, 0, 3,648	19, 1, 3,662	0, 0, 3,818
	LH	24, 0, 1	24, 0, 155	24, 0, 161	24, 0, 353
Large	*BB*	96, 42, 2,358	67, 17, 3,339	89, 22, 3,151	0, 0, 3,715
	LH	96, 0, 1	96, 0, 46	96, 0, 48	96, 0, 168

summarizes the results obtained with the two method-model combinations, *BB-BS* and *LH-EP+*. Three performance measures are provided: the lower bound GAP, the upper bound GAP, and the CPU time in seconds.

These results show that, on Small and Medium instances, the upper bounds obtained by *LH* are on average within 2% of the best-known solutions. On Large instances, the Lagrangian heuristic generally computes the best-known upper bounds, with *BB* being 1% away from them on average, and even 3% away on average on the largest instances with (25, 150). The computational effort to obtain such effective upper bounds is reasonable, as the CPU time is typically on the order of one minute on all instances. On Large instances with size (25, 150), the computing time is around five minutes on average. The lower bounds computed by the B&B of CPLEX are, as expected, better than the Lagrangian lower bounds, but only slightly so. In particular, for Small, Medium, and Large instances with size (25, 150), the two lower bounds are close, and get closer as the number of commodities increases. Indeed, the final gaps produced by the Lagrangian heuristic are on average better for Large instances with size (25, 150).

6. Conclusions

We have considered the piecewise linear integer multicommodity network flow problem. We have introduced formulations that exploit the integrality of the flows by using discretization. We have shown that the basic model obtained by discretization can be viewed as a particular case of the basic segment-based formulation introduced in Croxton, Gendron, and Magnanti (2007). We have strengthened the discretized models either by adding valid inequalities derived from cut-set inequalities or by using flow disaggregation techniques, obtaining a model similar to the so-called *extended (segment-based) formulation* introduced in Croxton, Gendron, and Magnanti (2007).

When comparing the relative strength of the different formulations, our main results state the following:

- Discretization provides stronger cut-set inequalities than those obtained from segment-based models.
- Discretization, when combined with flow disaggregation, does not improve on the LP relaxation of the extended segment-based model.

We have exploited these results by deriving a reformulation of the problem that combines the strength

Table 7. Upper Bounds: Lower Bound GAP, Upper Bound GAP, CPU

(|N|, |A|)	Algo-model	
(20,75)	*BB-BS*	1.00, 1.00, 1,066
	LH-EP+	0.94, 1.02, 30
(20,100)	*BB-BS*	0.99, 1.00, 1,311
	LH-EP+	0.94, 1.02, 62
(25,100)	*BB-BS*	1.00, 1.00, 1,924
	LH-EP+	0.93, 1.01, 44
(25,150)	*BB-BS*	0.97, 1.00, 3,103
	LH-EP+	0.92, 1.04, 69
Small	*BB-BS*	0.99, 1.00, 1,851
	LH-EP+	0.93, 1.02, 51
(20,75)	*BB-BS*	0.99, 1.00, 2,099
	LH-EP+	0.94, 1.01, 70
(20,100)	*BB-BS*	0.98, 1.00, 2,492
	LH-EP+	0.94, 1.01, 45
(25,100)	*BB-BS*	0.97, 1.00, 2,914
	LH-EP+	0.93, 1.02, 62
(25,150)	*BB-BS*	0.92, 1.02, 3,274
	LH-EP+	0.91, 1.02, 92
Medium	*BB-BS*	0.97, 1.01, 2,695
	LH-EP+	0.93, 1.02, 68
(20,75)	*BB-BS*	1.00, 1.00, 968
	LH-EP+	0.96, 1.00, 69
(20,100)	*BB-BS*	0.98, 1.00, 2,337
	LH-EP+	0.95, 1.00, 85
(25,100)	*BB-BS*	0.97, 1.01, 2,742
	LH-EP+	0.95, 1.00, 166
(25,150)	*BB-BS*	0.95, 1.03, 3,385
	LH-EP+	0.95, 1.00, 353
Large	*BB-BS*	0.98, 1.01, 2,358
	LH-EP+	0.95, 1.00, 168

of both techniques: cut-set inequalities based on discretization and flow disaggregation with segment-based variables. To overcome the large size of the resulting model, we developed an efficient and effective Lagrangian relaxation method to compute lower and upper bounds.

Computational experiments on a large set of randomly generated instances allowed us to compare the relative efficiency of the different modeling alternatives (flow disaggregation plus addition of cut-set inequalities with or without discretization), when used within the Lagrangian relaxation approach. The results derived from these experiments show that the Lagrangian relaxation method is both efficient and effective. For all instances, it produces lower and upper bounds in relatively small computing times and with gaps on the order of 5%–10%. On all instances, Lagrangian lower bounds are computed in less time than that required by CPLEX to solve the LP relaxation, with similar gaps; moreover, high-quality upper bounds are obtained in reasonable time, whereas the B&B method of CPLEX on any of the tested models often does not converge to optimality within the one-hour time limit.

This work opens the way for many research avenues. The models that we study are generic and include as special cases a large number of problems with applications in transportation and logistics, but also in other areas such as telecommunications and production planning. To our knowledge, apart from the references already cited, no other work on reformulations by discretization has been performed on such problems. It would be interesting to investigate the impact of discretization on such problems, as well as on other problems with a similar structure. The formulations we have introduced involve a large number of variables and constraints. We handled the large size of the models by developing Lagrangian relaxation methods. It would be interesting to investigate other approaches, such as column-and-cut generation (recent examples of such methods on problems similar to the PMFI include Frangioni and Gendron 2009, 2013; Gendron and Larose 2014).

Acknowledgments

The authors thank two anonymous referees whose comments have helped improve the paper. The authors are grateful to Serge Bisaillon for his help with implementing and testing the algorithms. The authors also gratefully acknowledge financial support for this project. This paper is dedicated to our colleague and friend Tom Magnanti.

References

Atamtürk A (2002) On capacitated network design cut-set polyhedra. *Math. Programming* 92:425–437.

Balakrishnan A, Graves S (1989) A composite algorithm for a concave-cost network flow problem. *Networks* 19:175–202.

Balakrishnan A, Magnanti TL, Mirchandani P (1997) Network design. Dell'Amico M, Maffioli F, Martello S, eds. *Annotated Bibliographies in Combinatorial Optimization* (John Wiley and Sons, Hoboken, NJ), 311–334.

Barahona F (1996) Network design using cut inequalities. *SIAM J. Optim.* 6:823–837.

Bienstock D, Günlük O (1996) Capacitated network design–polyhedral structure and computation. *INFORMS J. Comput.* 8:243–259.

Bienstock D, Chopra S, Günlük O, Tsai C (1998) Minimum cost capacity installation for multicommodity network flows. *Math. Programming* 81:177–199.

Chouman M, Crainic TG, Gendron B (2017) Commodity representations and cutset-based inequalities for multicommodity capacitated fixed-charge network design. *Transportation Sci.* 51(2): 650–667.

Christof T, Loebel A (2015) Porta (version 1.4.1). http://www.iwr.uni-heidelberg.de/groups/comopt/software/PORTA.

Correia I, Gouveia L, Saldanha da Gama F (2010) Discretized formulations for capacitated location problems with modular distribution costs. *Eur. J. Oper. Res.* 204:237–244.

Crainic TG (2000) Service network design in freight transportation. *Eur. J. Oper. Res.* 122:272–288.

Crainic TG, Frangioni A, Gendron B (2001) Bundle-based relaxation methods for multicommodity capacitated fixed charge network design problems. *Discrete Appl. Math.* 112:73–99.

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

Croxton KL, Gendron B, Magnanti TL (2007) Variable disaggregation in network flow problems with piecewise linear costs. *Oper. Res.* 55:146–157.

Duhamel C, Gouveia L, Moura P, Souza M (2012) Minimum cost degree constrained spanning trees with node-degree costs. *Networks* 50:1–18.

Frangioni A, Gendron B (2009) 0–1 reformulations of the multicommodity capacitated network design problem. *Discrete Appl. Math.* 157:1229–1241.

Frangioni A, Gendron B (2013) A stabilized structured Dantzig–Wolfe decomposition method. *Math. Programming* 140:45–76.

Gabrel V, Knippel K, Minoux M (2003) A comparison of heuristics for the discrete cost multicommodity network optimization problem. *J. Heuristics* 9:429–445.

Gendron B, Larose M (2014) Branch-and-price-and-cut for large-scale multicommodity capacitated fixed-charge network design. *Eur. J. Computat. Optim.* 2:55–75.

Gendron B, Crainic TG, Frangioni A (1999) Multicommodity capacitated network design. Soriano P, Sansò B, eds. *Telecommunications Network Planning* (Kluwer Academic Publishers, Dordrecht, Netherlands), 1–19.

Geoffrion AM (1974) Lagrangian relaxation for integer programming. *Math. Programming Stud.* 2:82–114.

Gouveia L (1995) A $2n$ constraint formulation for the capacitated minimal spanning tree problem. *Oper. Res.* 43:130–141.

Gouveia L, Moura P (2012) Enhancing discretized formulations: The knapsack reformulation and the star reformulation. *TOP* 20: 52–74.

Gouveia L, Saldanha da Gama F (2006) On the capacitated concentrator location problem: A reformulation by discretization. *Comput. Oper. Res.* 33:1242–1258.

Günlük O (1999) A branch-and-cut algorithm for capacitated network design problems. *Math. Programming* 86:17–39.

Held M, Wolfe P, Crowder HP (1974) Validation of subgradient optimization. *Math. Programming* 6:62–88.

Holmberg K, Yuan D (2000) A Lagrangean heuristic based branch-and-bound approach for the capacitated network design problem. *Oper. Res.* 48:461–481.

Kim D, Pardalos PM (1999) A solution approach to the fixed charge network flow problem using a dynamic slope scaling procedure. *Oper. Res. Lett.* 24:195–203.

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

Magnanti TL, Wong RT (1984) Network design and transportation planning: Models and algorithms. *Transportation Sci.* 18: 1–55.

Magnanti TL, Mirchandani P, Vachani R (1993) The convex hull of two core capacitated network design problems. *Math. Programming* 60:233–250.

Minoux M (1989) Network synthesis and optimum network design problems: Models, solution methods and applications. *Networks* 19:313–360.

Ortega F, Wolsey LA (2003) A branch-and-cut algorithm for the single commodity uncapacitated fixed charge network flow problem. *Networks* 41:143–158.

Raack C, Koster AMCA, Orlowski S, Wessäly R (2011) On cut-based inequalities for capacitated network design polyhedra. *Networks* 57:141–156.