Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

Reformulations by Discretization for Piecewise Linear Integer Multicommodity Network Flow Problems

Bernard Gendron
Luis Gouveia

January 2014

CIRRELT-2014-01

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Reformulations by Discretization for Piecewise Linear Integer Multicommodity Network Flow Problems†

Bernard Gendron[1,*], Luis Gouveia[2]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Université de Montréal, Pavillon André-Aisenstadt, Department of Computer Science and Operations Research, P.O. Box 6128, Station Centre-Ville, Montréal, Canada H3C 3J7

[2] Departamento de Estatística e Investigação Operacional, Faculdade de Ciências, Universidade de Lisboa, Campo Grande, Edifício C6 – 4º piso, (sala 6.4.11), 1749-016 Lisboa, Portugal

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 cutset 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 cutset inequalities with or without discretization), when used within the Lagrangian relaxation approach.

Keywords. Multicommodity network flow problem, piecewise linear cost, reformulation, discretization, disaggregation, valid inequalities.

Acknowledgements. We are grateful to Serge Bisaillon for his help with implementing and testing the algorithms. We also gratefully acknowledge financial support for this project. Bernard Gendron research is partially funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant 184122-09. Luis Gouveia has been supported by the Fundação para a Ciência e Tecnologia under project PEst-OE/MAT/UI0152.

†This paper is dedicated to our colleague and friend Tom Magnanti.

---

* Corresponding author: Bernard.Gendron@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2014

1 Introduction

We consider the piecewise linear multicommodity network flow problem (PMF) studied in [12]. 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 $S_a = \{1, 2, ..., |S_a|\}$. 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, $l_a^s = b_a^{s-1}$ and $v_a^s = b_a^s$ (the breakpoints, 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 semi-continuous (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 non-decreasing (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 [3, 9, 17, 29, 30] 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. Further, we introduce new formulations for piecewise linear multicommodity network flow problems that exploit this integrality constraint.

Following [12], we present a mixed-integer programming (MIP) formulation of the PMFI, which we call the *basic 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. 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 $F_i = \{a \in A | t(a) = i\}$ and $B_i = \{a \in A | h(a) = i\}$, respectively, the sets of forward and backward arcs incident to node $i \in N$, the basic model, denoted BM, can be expressed as follows:

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

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

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

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

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

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

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

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 *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 breakpoints of that segment, i.e., $b_a^{s-1} = l_a^s \leq x_a^s \leq v_a^s = b_a^s$. 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.

It is well-known that the basic 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 *cutset inequalities*, which have been used to strengthen the LP relaxation bounds of a large number of problems related to the PMFI [1, 4, 5, 6, 7, 16, 23, 28, 31, 32]. Another approach, called *flow disaggregation* [12, 14, 15], consists in 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 [20, 21, 22]. Discretization is to be combined with the two other approaches, addition of cutset 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 BM, that use the segment set in the definition of the variables. Following the developments in [8, 22], we derive valid inequalities from cutset 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 [12]. Our main results state that: 1) discretization provides stronger cutset inequalities than those obtained from segment-based models; 2) discretization, when combined with flow disaggregation, does not improve upon 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: cutset 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 cutset 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 to the *Lagrangian dual optimization* procedure that computes lower bounds on the optimal value of the PMFI, while 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 $\overline{M}$ 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 first strengthening the basic model, as explained in Section 2.1, and then, by defining the point-based model (Section 2.2). In Section 2.3, we derive cutset-based inequalities for both the segment-based and the point-based models, yielding stronger reformulations of the PMFI. In Section 2.4, we investigate the combination of flow disaggregation and discretization. Finally, Section 2.5 summarizes our main results and presents models that combine the strength of point-based cutset inequalities with segment-based flow disaggregation.

2.1 Strengthening the Basic Model

The integrality constraint on the flows implies that x_a is either 0 or can take any integer value $q \in Q_a = \{1, 2, \ldots, u_a\}$. Furthermore, since $g_a(x_a)$ is lower semi-continuous and non-decreasing, if $x_a = b_a^{s-1} > 0$ then $x_a = x_a^{s-1}$, i.e., x_a lies in segment $s-1$. This implies 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. As a consequence, we can strengthen the lower flow bounds in constraints (4) by using $l_a^s = b_a^{s-1}+1$ instead of $l_a^s = b_a^{s-1}$. We obtain in this way the *basic segment-based model*, which we denote BS. We show next, however, that this improvement on the lower flow bounds does not improve the LP relaxation bound of the original basic model.

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

Proof: For both models, $\overline{BM}$ and $\overline{BS}$, we note that there always exists an optimal solution such that $x_a^s = v_a^s y_a^s$, for each arc a and segment s; 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 $\frac{x_a^s}{v_a^s}$ and maintain feasibility, as well as optimality, since $f_a^s \geq 0$. Therefore, any optimal solution to $\overline{BM}$ is also optimal for $\overline{BS}$ and vice-versa. ■

In spite of this result, we will use BS instead of BM as our basic segment-based formulation. The reason for this is that stronger valid inequalities can be obtained from BS, as we will see in Section 2.3.

The proof of Proposition 1 also implies that $\overline{BS}$ can be simplified by projecting out the x_a^s variables, which yields the following model that will be subsequently used in our developments:

$$v(\overline{BS}) = \min \sum_{a \in A} \sum_{s \in S_a} (v_a^s c_a^s + f_a^s) y_a^s \tag{8}$$

$$\sum_{a \in F_i} x_a^k - \sum_{a \in B_i} x_a^k = d_i^k, \quad i \in N, k \in K, \tag{9}$$

$$\sum_{k \in K} x_a^k = \sum_{s \in S_a} v_a^s y_a^s, \quad a \in A, \tag{10}$$

$$\sum_{s \in S_a} y_a^s \leq 1, \quad a \in A, \tag{11}$$

$$x_a^k \geq 0, \quad a \in A, k \in K, \tag{12}$$

$$y_a^s \geq 0, \quad a \in A, s \in S_a. \tag{13}$$

2.2 Point-Based Model

Using the integrality constraint on the flow variables, 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:

$$\min \sum_{a \in A} \sum_{s \in S_a} \sum_{q \in Q_a^s} (c_a^s x_a^q + f_a^s y_a^q) \tag{14}$$

$$\sum_{a \in F_i} x_a^k - \sum_{a \in B_i} x_a^k = d_i^k, \quad i \in N, k \in K, \tag{15}$$

$$\sum_{k \in K} x_a^k = \sum_{q \in Q_a} x_a^q = x_a \text{ integer}, \quad a \in A, \tag{16}$$

$$x_a^q = q y_a^q, \quad a \in A, q \in Q_a, \tag{17}$$

$$\sum_{q \in Q_a} y_a^q \leq 1, \quad a \in A, \tag{18}$$

$$x_a^k \geq 0, \quad a \in A, k \in K, \tag{19}$$

$$y_a^q \in \{0, 1\}, \quad a \in A, q \in Q_a. \tag{20}$$

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 (17) with two inequalities as follows:

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

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 (17) 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:

$$v(BP) = \min \sum_{a \in A} \sum_{s \in S_a} \sum_{q \in Q_a^s} (qc_a^s + f_a^s) y_a^q \tag{21}$$

$$\sum_{a \in F_i} x_a^k - \sum_{a \in B_i} x_a^k = d_i^k, \quad i \in N, k \in K, \tag{22}$$

$$\sum_{k \in K} x_a^k = \sum_{q \in Q_a} qy_a^q, \quad a \in A, \tag{23}$$

$$\sum_{q \in Q_a} y_a^q \le 1, \quad a \in A, \tag{24}$$

$$x_a^k \ge 0, \quad a \in A, k \in K, \tag{25}$$

$$y_a^q \in \{0, 1\}, \quad a \in A, q \in Q_a. \tag{26}$$

Note that $\overline{BP}$, the LP relaxation of (21)-(26), and $\overline{BS}$, the model defined by (8)-(13), 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 [13]):

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

Proof: 1) First, we show that $v(\overline{BP}) \ge v(\overline{BS})$. 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) Second, we show that $v(\overline{BP}) \le v(\overline{BS})$. 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 the flow variables in BP can be projected out using equations (23). Model BP then reduces to:

$$\min \sum_{a \in A} \sum_{s \in S_a} \sum_{q \in Q_a^s} (qc_a^s + f_a^s) y_a^q \tag{27}$$

$$\sum_{a\in F_i}\sum_{q\in Q_a} qy_a^q - \sum_{a\in B_i}\sum_{q\in Q_a} qy_a^q = d_i, \quad i\in N, \tag{28}$$

$$\sum_{q\in Q_a} y_a^q \le 1, \quad a\in A, \tag{29}$$

$$y_a^q \in \{0,1\}, \quad a\in A, q\in Q_a. \tag{30}$$

This model, containing only the binary variables y_a^q, is similar to the reformulations by discretization described in the literature [8, 20, 21, 22].

2.3 Cutset Inequalities

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

$$\sum_{a\in F_U} x_a - \sum_{a\in B_U} x_a = D_U, \quad U\in\mathcal{U}, \tag{31}$$

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 cutset equations with constraints (3) and (4), we obtain the following *segment-based cutset inequalities* for models BM and BS:

$$\sum_{a\in F_U}\sum_{s\in S_a} v_a^s y_a^s - \sum_{a\in B_U}\sum_{s\in S_a} l_a^s y_a^s \ge D_U, \quad U\in\mathcal{U}, \tag{32}$$

$$\sum_{a\in F_U}\sum_{s\in S_a} l_a^s y_a^s - \sum_{a\in B_U}\sum_{s\in S_a} v_a^s y_a^s \le D_U, \quad U\in\mathcal{U}. \tag{33}$$

These inequalities are redundant for the LP relaxations, $\overline{BM}$ and $\overline{BS}$, 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 $conv(CUT)$ and $conv(CUT_S)$ can be used to strengthen $\overline{BM}$ and $\overline{BS}$, respectively, where CUT and CUT_S are the sets of 0-1 solutions to the multiple choice constraints (5) and the cutset inequalities (32)-(33) with $l_a^s = b_a^{s-1}$ for CUT and $l_a^s = b_a^{s-1}+1$ for CUT_S. By adding all the facet-defining inequalities for $conv(CUT)$ and $conv(CUT_S)$ to $\overline{BM}$ and $\overline{BS}$, respectively, we obtain stronger LP relaxations, which we denote $\overline{BM+}$ and $\overline{BS+}$.

It is obvious that $conv(CUT_S)\subseteq conv(CUT)$, which implies the following result:

Proposition 3 $v(\overline{BS+}) \ge v(\overline{BM+})$.

This observation justifies our selection of BS as a better segment-based model than BM, in spite of the equality $v(\overline{BM}) = v(\overline{BS})$ (Proposition 1).

For model BP, a similar derivation yields the following *point-based cutset equations*, which can also be obtained directly from (32)-(33) for the case where $l_a^s = b_a^{s-1} + 1 = b_a^s = v_a^s$:

$$\sum_{a \in F_U} \sum_{q \in Q_a} q y_a^q - \sum_{a \in B_U} \sum_{q \in Q_a} q y_a^q = D_U, \quad U \in \mathcal{U}. \tag{34}$$

Let us define CUT_P as the set of 0-1 solutions that satisfy these cutset equations, along with the multiple choice constraints (18), and $\overline{BP+}$ as the LP relaxation of BP obtained by adding all the facet-defining inequalities for $conv(CUT_P)$ to formulation $\overline{BP}$. We then have the following result:

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

Proof: Let $y(i)$, $i \in I$, be the extreme points of $conv(CUT_P)$; any $y(i)$, $i \in I$, can be mapped to a solution of $conv(CUT_S)$ by the same construction used in the proof of Proposition 2: 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 $conv(CUT_P)$: $y_a^q = \sum_{i \in I} \lambda(i) y_a^q(i)$, $a \in A, q \in Q_a$, 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 2: 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 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 $conv(CUT_S)$ obtained by mapping the extreme points of $conv(CUT_P)$: $y_a^s = (q/v_a^s) y_a^q = (q/v_a^s) \sum_{i \in I} \lambda(i) y_a^q(i) = \sum_{i \in I} \lambda(i) y_a^s(i)$, i.e., this projected solution belongs to $conv(CUT_S)$, which implies that $v(\overline{BP+}) \geq v(\overline{BS+})$. ■

Note that the aggregation of two point-based cutset equations of the form (34) associated to $U \in \mathcal{U}$ and $W \in \mathcal{U}$, $U \cap W = \emptyset$, is equivalent to the cutset equation associated to $U \cup W$. Indeed, using the notation $\{U, W\} = \{a \in A | t(a) \in U, h(a) \in W\} \cup \{a \in A | t(a) \in W, h(a) \in U\}$, we obtain after summing the cutset equations (34) 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 cutset equation associated to $U \cup W$. This observation allows us to considerably reduce the number of equations needed to characterize CUT_P:

Proposition 5 *CUT_P is equal to the set of 0-1 solutions that satisfy the multiple choice constraints (18) and the point-based single-node cutset equations:*

$$\sum_{a \in F_i} \sum_{q \in Q_a} q y_a^q - \sum_{a \in B_i} \sum_{q \in Q_a} q y_a^q = D_i, \quad i \in N. \tag{35}$$

Proof: Using the same argument as above, all cutset equations of the form (34) can be derived by aggregation of single-node cutset equations of the form (35). ■

This proposition illustrates another important difference between the segment-based cutset inequalities (32)-(33) and the point-based cutset equations (34): the *single-node cutset* *equations* (35) are enough to characterize all point-based cutset equations, while the same is not true for the segment-based cutset 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 cutset inequalities of the form (32) or (33) associated to $U \in \mathcal{U}$ and $W \in \mathcal{U}$, $U \cap W = \emptyset$, is not equivalent to the cutset inequality of the same form associated to $U \cup W$. For example, by summing the cutset inequalities (32) 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 cutset inequality of the form (32) associated to $U \cup W$:

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

Obviously, generating all facet-defining inequalities for $conv(CUT_S)$ and $conv(CUT_P)$ is a hard task. In particular, even for a set defined by a *single* cutset inequality associated to a given cut U, generating all facet-defining inequalities is not trivial. In the context of reformulations by discretization, Chvàtal-Gomory rank 1 inequalities have been derived from a single inequality and proven to be rather exceptionally effective [8, 21, 22]. 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 cutset equations (34), yielding the following valid inequalities, where $P = \max_{a \in A}\{u_a\}$:

$$\sum_{a \in F_U} \sum_{q \in Q_a} \left\lceil \frac{q}{p} \right\rceil y_a^q - \sum_{a \in B_U} \sum_{q \in Q_a} \left\lfloor \frac{q}{p} \right\rfloor y_a^q \geq \left\lceil \frac{D_U}{p} \right\rceil, \quad U \in \mathcal{U}, p = 1, 2, \ldots, P, \tag{36}$$

$$\sum_{a \in F_U} \sum_{q \in Q_a} \left\lfloor \frac{q}{p} \right\rfloor y_a^q - \sum_{a \in B_U} \sum_{q \in Q_a} \left\lceil \frac{q}{p} \right\rceil y_a^q \leq \left\lfloor \frac{D_U}{p} \right\rfloor, \quad U \in \mathcal{U}, p = 1, 2, \ldots, P. \tag{37}$$

When $p = 1$, these inequalities reduce to the point-based cutset equations (34). 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 cutset equations (34), (36)-(37) cannot be reduced to inequalities associated only to single-node cuts.

The technique can be generalized to the segment-based cutset inequalities (32)-(33), giving rise to the following valid inequalities:

$$\sum_{a \in F_U} \sum_{s \in S_a} \left\lceil \frac{v_a^s}{p} \right\rceil y_a^s - \sum_{a \in B_U} \sum_{s \in S_a} \left\lfloor \frac{l_a^s}{p} \right\rfloor y_a^s \geq \left\lceil \frac{D_U}{p} \right\rceil, \quad U \in \mathcal{U}, p = 1, 2, \ldots, P, \tag{38}$$

$$\sum_{a \in F_U} \sum_{s \in S_a} \left\lfloor \frac{l_a^s}{p} \right\rfloor y_a^s - \sum_{a \in B_U} \sum_{s \in S_a} \left\lceil \frac{v_a^s}{p} \right\rceil y_a^s \leq \left\lfloor \frac{D_U}{p} \right\rfloor, \quad U \in \mathcal{U}, p = 1, 2, \ldots, P. \tag{39}$$

Let us define $\widehat{BS+}$ and $\widehat{BP+}$ the models obtained by adding, respectively, inequalities (38)-(39) to $\overline{BS}$ and inequalities (36)-(37) to $\overline{BP}$. Since the Chvàtal-Gomory rank 1 inequalities approximate the convex hull associated to each single cutset 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 6 $v(\widehat{BP+}) \geq v(\widehat{BS+})$.

Proof: We apply the same construction as in part 1) of the proof of Proposition 2. The result follows from the inequalities

$$b_a^{s-1} + 1 = l_a^s \leq q \leq v_a^s = b_a^s, \quad a \in A, s \in S_a, q \in Q_a^s.$$

■

2.4 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* [12]:

$$x_a^{ks} \leq M_a^k y_a^s, \quad a \in A, k \in K, s \in S_a, \tag{40}$$

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 (40) to BS as the *extended segment-based model*, which we denote ES. Obviously, we have $v(\overline{ES}) \geq v(\overline{BS})$.

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:

$$x_a^{kq} \leq M_a^k y_a^q, \quad a \in A, k \in K, q \in Q_a. \tag{41}$$

We refer to the model obtained by adding the nonnegative variables x_a^{kq}, the definitional equations $qy_a^q = \sum_{k \in K} x_a^{kq}$ and $x_a^k = \sum_{q \in Q_a} x_a^{kq}$, and the valid inequalities (41) to BP as the *extended point-based model*, which we denote EP.

We now show that, similarly 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 2 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:

$$v(ES(\pi)) = \min \sum_{a\in A}\sum_{s\in S_a}\sum_{k\in K}(c_a^s - \pi_{t(a)}^k + \pi_{h(a)}^k)x_a^{ks} + \sum_{a\in A}\sum_{s\in S_a} f_a^s y_a^s \tag{42}$$

$$l_a^s y_a^s \leq \sum_{k\in K} x_a^{ks} \leq v_a^s y_a^s, \quad a \in A, s \in S_a, \tag{43}$$

$$0 \leq x_a^{ks} \leq M_a^k y_a^s, \quad a \in A, k \in K, s \in S_a, \tag{44}$$

$$\sum_{s\in S_a} y_a^s \leq 1, \quad a \in A, \tag{45}$$

$$y_a^s \in \{0,1\}, \quad a \in A, s \in S_a. \tag{46}$$

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 (46). 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 to arc $a \in A$, i.e., the constraints defining LAG_a have the following form:

$$l_a^s y_a^s \leq \sum_{k\in K} x_a^{ks} \leq v_a^s y_a^s, \quad s \in S_a, \tag{47}$$

$$0 \leq x_a^{ks} \leq M_a^k y_a^s, \quad k \in K, s \in S_a, \tag{48}$$

$$\sum_{s\in S_a} y_a^s \leq 1, \tag{49}$$

$$y_a^s \in \{0,1\}, \quad s \in S_a. \tag{50}$$

We then have the following polyhedral result, which is extracted from the proof of Theorem 5 in [12] (we reproduce the argument, since it is also used in the proof of Proposition 10 below):

Proposition 7 *$conv(LAG_a) = \overline{LAG_a}$.*

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 $(\widehat{x}, \widehat{y})$ be an extreme point of $\overline{LAG_a}$ with at least one fractional component. Assume that $0 < \widehat{y}_a^r < 1$, 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 $x_a^{kr}(r) = \widehat{x}_a^{kr}/\widehat{y}_a^r$, $x_a^{ks}(r) = 0$, $s \neq r$, $y_a^r(r) = 1$ and $y_a^s(r) = 0$, $s \neq r$. Then, $(\widehat{x}, \widehat{y}) = (1 - \sum_{r\in R}\widehat{y}_a^r)(x(0), y(0)) + \sum_{r\in R}\widehat{y}_a^r(x(r), y(r))$ is a representation of $(\widehat{x}, \widehat{y})$ as a convex combination of $|R| + 1 \geq 2$ distinct points in $\overline{LAG_a}$, contradicting the hypothesis that it is an extreme point of $\overline{LAG_a}$. ■

By Lagrangian duality theory [19], this result implies that $v(\overline{ES}) = \max_\pi\{\sum_{i\in N}\sum_{k\in K}\pi_i^k d_i^k + v(ES(\pi))\}$.

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:

$$v(EP(\pi)) = \min \sum_{a\in A}\sum_{s\in S_a}\sum_{q\in Q_a^s}\sum_{k\in K}(c_a^s - \pi_{t(a)}^k + \pi_{h(a)}^k)x_a^{kq} + \sum_{a\in A}\sum_{s\in S_a}\sum_{q\in Q_a^s} f_a^s y_a^q \tag{51}$$

$$\sum_{k\in K} x_a^{kq} = q y_a^q, \quad a\in A, q\in Q_a, \tag{52}$$

$$0 \le x_a^{kq} \le M_a^k y_a^q, \quad a\in A, k\in K, q\in Q_a, \tag{53}$$

$$\sum_{q\in Q_a} y_a^q \le 1, \quad a\in A, \tag{54}$$

$$y_a^q \in \{0,1\}, \quad a\in A, a\in Q_a. \tag{55}$$

Similarly as for $ES(\pi)$, we can show that this Lagrangian subproblem has the integrality property. By Lagrangian duality theory [19], 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 8 $v(\overline{EP}) = v(\overline{ES})$.

Proof: We have just seen that $v(\overline{ES}) = \max_\pi\{\sum_{i\in N}\sum_{k\in K}\pi_i^k d_i^k + v(ES(\pi))\}$ and $v(\overline{EP}) = \max_\pi\{\sum_{i\in N}\sum_{k\in K}\pi_i^k d_i^k + v(EP(\pi))\}$. Therefore, the result follows if we can prove that the two Lagrangian subproblems, $ES(\pi)$ and $EP(\pi)$, are equivalent.
1) First, 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) Second, 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 the variables assume value 0). ■

2.5 Combining Cutset Inequalities and Flow Disaggregation

Our results from the last two sections highlight that the best LP relaxation bounds can be obtained by combining cutset 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 upon flow disaggregation with segment-based variables; 2) cutset 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 cutset inequalities can improve upon segment-based cutset 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:

$$\sum_{k \in K} x_a^{ks} = \sum_{q \in Q_a^s} q y_a^q, \quad a \in A, s \in S_a, \tag{56}$$

$$y_a^s = \sum_{q \in Q_a^s} y_a^q, \quad a \in A, s \in S_a. \tag{57}$$

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

- $\overline{EP+}$, the model derived from $\overline{ES}$ by adding the linking equations (56)-(57), plus all the facet-defining inequalities for $conv(CUT_P)$;

- $\widehat{EP+}$, the LP relaxation obtained from $\overline{ES}$ by adding the point-based Chvàtal-Gomory rank 1 valid inequalities (36)-(37), plus the linking equations (56)-(57).

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

- $\overline{ES+}$, the formulation obtained by adding to $\overline{ES}$ all the facet-defining inequalities for $conv(CUT_S)$;

- $\widehat{ES+}$, the LP relaxation obtained by adding the segment-based Chvàtal-Gomory rank 1 valid inequalities (38)-(39) to $\overline{ES}$.

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

Proposition 9 *The following bound relationships hold:*
a) $v(\overline{EP+}) \geq v(\overline{ES+}) \geq v(\widehat{ES+})$.
b) $v(\overline{EP+}) \geq v(\widehat{EP+}) \geq v(\widehat{ES+})$.

Thus, from a theoretical perspective, the strongest lower bound is obtained from model $\overline{EP+}$, while relaxation $\widehat{ES+}$ shows the worst lower bound and the other two models correspond to "intermediate" bounds. To compute tight approximations to these bounds, we propose an efficient Lagrangian relaxation method, which we present in the next section.

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 making use of the weaker bounds $v(\overline{BS})$, $v(\overline{ES})$, $v(\widehat{ES+})$, $v(\overline{ES+})$ and $v(\widehat{EP+})$, computing 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 2.5, we exploit the following reformulation of the PMFI, which uses the segment-based and point-based variables in a single model:

$$\min \sum_{a\in A}\sum_{s\in S_a}\sum_{k\in K} c_a^s x_a^{ks} + \sum_{a\in A}\sum_{s\in S_a} f_a^s y_a^s \tag{58}$$

$$\sum_{a\in F_i}\sum_{s\in S_a} x_a^{ks} - \sum_{a\in B_i}\sum_{s\in S_a} x_a^{ks} = d_i^k, \quad i\in N, k\in K, \tag{59}$$

$$l_a^s y_a^s \le \sum_{k\in K} x_a^{ks} \le v_a^s y_a^s, \quad a\in A, s\in S_a, \tag{60}$$

$$0 \le x_a^{ks} \le M_a^k y_a^s, \quad a\in A, k\in K, s\in S_a, \tag{61}$$

$$\sum_{s\in S_a} y_a^s \le 1, \quad a\in A, \tag{62}$$

$$\sum_{k\in K} x_a^{ks} = \sum_{q\in Q_a^s} q y_a^q, \quad a\in A, s\in S_a, \tag{63}$$

$$y_a^s = \sum_{q\in Q_a^s} y_a^q, \quad a\in A, s\in S_a, \tag{64}$$

$$\sum_{a\in F_i}\sum_{q\in Q_a} q y_a^q - \sum_{a\in B_i}\sum_{q\in Q_a} q y_a^q = D_i, \quad i\in N, \tag{65}$$

$$y_a^q \in \{0,1\}, \quad a\in A, q\in Q_a, \tag{66}$$

$$y_a^s \in \{0,1\}, \quad a\in A, s\in S_a. \tag{67}$$

The objective (58), along with constraints (59)-(62) and (67), 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 (63) and (66). Constraints (63)-(64) provide the link between the segment-based variables and the point-based variables y_a^q. The point-based single-node cutset equations (65) 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.

To compute $v(\overline{EP+})$, we consider the Lagrangian relaxation of the flow conservation equations (59) and of the linking equations (63), 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 $LAG_P(\pi,\beta)$:

$$v(LAG_P(\pi,\beta)) = \min \sum_{a\in A}\sum_{s\in S_a}\sum_{k\in K}(c_a^s - \beta_a^s - \pi_{t(a)}^k + \pi_{h(a)}^k) x_a^{ks} + \sum_{a\in A}\sum_{s\in S_a}(f_a^s y_a^s + \sum_{q\in Q_a^s} q\beta_a^s y_a^q) \tag{68}$$

subject to constraints (60)-(62) and (64)-(67).

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

$$v(P_a^s(\pi,\beta)) = \min \sum_{k\in K} (c_a^s - \beta_a^s - \pi_{t(a)}^k + \pi_{h(a)}^k) x_a^{ks} \tag{69}$$

$$l_a^s \leq \sum_{k\in K} x_a^{ks} \leq v_a^s, \tag{70}$$

$$0 \leq x_a^{ks} \leq M_a^k, \quad k \in K. \tag{71}$$

Then, we reformulate the Lagrangian subproblem as follows:

$$v(LAG_P(\pi,\beta)) = \min \sum_{a\in A} \sum_{s\in S_a} \left((v(P_a^s(\pi,\beta)) + f_a^s) y_a^s + \sum_{q\in Q_a^s} q\beta_a^s y_a^q \right) \tag{72}$$

subject to constraints (62) and (64)-(67). The resulting Lagrangian subproblem is a pure 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 10 $v(\overline{EP+}) = max_{\pi,\beta}\{\sum_{i\in N} \sum_{k\in K} \pi_i^k d_i^k + v(LAG_P(\pi,\beta))\}$.

Proof: By Lagrangian duality theory [19], the Lagrangian dual is equivalent to optimizing the objective function (58) over the feasible domain described by the intersection of the set defined by (59) and (63) with the convex hull of the set defined by (60)-(62) and (64)-(67), which we denote by $\{(59),(63)\} \cap conv(\{(60)\text{-}(62),(64)\text{-}(67)\})$. If we can show that this feasible domain is equal to $\{(59)\text{-}(64)\} \cap conv(\{(65)\text{-}(66)\}) \equiv \{(59)\text{-}(64)\} \cap conv(CUT_P)$, i.e., the set defined by (59)-(64) to which we add all the facet-defining inequalities for $conv(CUT_P)$, the result would immediately follow by definition of $\overline{EP+}$. To show that $\{(59),(63)\} \cap conv(\{(60)\text{-}(62),(64)\text{-}(67)\}) = \{(59)\text{-}(64)\} \cap conv(CUT_P)$, we first remark that the inclusion $\subseteq$ is trivial. To show the inclusion $\supseteq$, it suffices to show that every extreme point of $\{(60)\text{-}(62),(64)\} \cap conv(CUT_P)$ is integral. The same argument as in the proof of Proposition 7 can be used to show this result. ■

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 (63) to (66) 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 $s \in S_a$, $\sum_{k\in K} x_a^{ks} > 0$ if and only if $y_a^s = 1$. As a result, we can reformulate the Lagrangian subproblem as follows:

$$\min \sum_{a\in A} \sum_{s\in S_a} (v(P_a^s(\pi)) + f_a^s) y_a^s \tag{73}$$

subject to (62) and (67), where $v(P_a^s(\pi))$ is the optimal value of the following continuous knapsack problem:

$$v(P_a^s(\pi)) = \min \sum_{k\in K} (c_a^s - \pi_{t(a)}^k + \pi_{h(a)}^k) x_a^{ks} \tag{74}$$

$$l_a^s \leq \sum_{k\in K} x_a^{ks} \leq v_a^s, \tag{75}$$

$$0 \leq x_a^{ks} \leq M_a^k, \quad k \in K. \tag{76}$$

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 $\min_{s\in S_a}\{v(P_a^s(\pi)) + f_a^s\} \leq 0$ 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 [2, 10, 25].

- $v(\overline{ES+})$: We simply replace constraints (63) to (66) by the segment-based cutset inequalities (32)-(33) 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 cutset inequalities.

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 cutset 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 [1, 7], which show that single-node cutset inequalities are responsible for most of the lower bound improvement obtained by adding cutset inequalities in the context of multicommodity network flow problems.

The reformulation of the Lagrangian subproblem as a pure IP model, i.e., (72) subject to constraints (62) and (64)-(67), 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 a *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 Chvàtal-Gomory rank 1 valid inequalities, i.e., (38)-(39) and (36)-(37), respectively, restricted to *single-node cuts*. The segment-based Chvàtal-Gomory rank 1 valid inequalities are redundant in the resulting MIP model, since the segment-based variables y_a^s are binary. They are also redundant in the LP relaxation of this MIP model, since they are implied by the point-based Chvàtal-Gomory rank 1 valid inequalities. We have observed, however, that their addition helps in solving the model more efficiently. In contrast, the point-based Chvàtal-Gomory rank 1 valid inequalities are no more redundant, since the y_a^q variables are now continuous; in particular, their addition allows 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 (8)-(13), 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 to the linking equations (63). 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 $v(\widehat{ES+})$. 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 all the bounds defined in Section 2.5, 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 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 Chvàtal-Gomory rank 1 valid inequalities (38)-(39) 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 (58)-(67) by relaxing the flow conservation equations (59) (with Lagrange multipliers $\pi = \pi^3$) and by replacing the point-based single-node cutset equations (65) with the point-based Chvàtal-Gomory rank 1 valid inequalities (36)-(37) 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 1d and 2c.

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 1a) and $v(\overline{ES})$ (Step 1b) are extremely fast. Our experiments also confirm that the subgradient method used in Steps 1b and 1c 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 1d and 2c 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, while 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 the 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 1d and 2c, instead of using the subgradient method.

- In Step 2a, 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 $s \in S_a$, $\sum_{k\in K} x_a^{ks} > 0$ if and only if $y_a^s = 1$ is not true anymore; instead, we have that, for each arc $a \in A$ and segment $s \in S_a$, $\sum_{k\in K} x_a^{ks} > 0$ if and only if $y_a^s > 0$. This apparently minor modification makes a huge difference when solving the Lagrangian subproblem, since it is no more 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 non-decomposable 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 2a is computationally preferable to the Lagrangian relaxation approach for approximating the same bound.

- Instead of performing Steps 2a and 2b, we could have solved the approximation to $v(\widehat{EP+})$ defined by model $\widehat{EP+}$ restricted to single-node point-based Chvàtal-Gomory rank 1 valid inequalities (36)-(37). As shown in our computational experiments reported in Section 5, the computing times for solving this LP relaxation are prohibitive. In 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 2b, we also add the segment-based Chvàtal-Gomory rank 1 valid inequalities (38)-(39) 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 9, 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 1d and 2c, respectively. The inequality $v(\widehat{EP+}) \geq v(\widehat{ES+})$ follows from Step 2b. 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 [24]. 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 $\alpha(t) = \lambda(t)(v^* - v(\pi(t-1)))/||\gamma(t)||^2$, where $v(\pi(t-1))$ is the Lagrangian lower bound associated to 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 [26, 27] and multicommodity [11] 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 non-linear 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 non-linear 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:

$$v(MF) = \min \sum_{a \in A} \sum_{k \in K} \bar{c}_a x_a^k \tag{77}$$

$$\sum_{a \in F_i} x_a^k - \sum_{a \in B_i} x_a^k = d_i^k, \quad i \in N, k \in K, \tag{78}$$

$$\sum_{k \in K} x_a^k \leq u_a, \quad a \in A, \tag{79}$$

$$x_a^k \geq 0, \quad a \in A, k \in K. \tag{80}$$

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 in order 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:

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

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 $\bar{x}_a$ derived from solving MF is feasible for the PMFI if $\bar{x}_a$ 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 non-integer 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 non-integer 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 non-integer 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:

$$v(IMF(\bar{x})) = \sum_{a \in A} f_a^{\bar{s}_a} + \min \sum_{a \in A} c_a^{\bar{s}_a} x_a \tag{81}$$

$$\sum_{a \in F_i} x_a^k - \sum_{a \in B_i} x_a^k = d_i^k, \quad i \in N, k \in K, \tag{82}$$

$$\sum_{k \in K} x_a^k = x_a \text{ integer}, \quad a \in A, \tag{83}$$

$$l_a^{\bar{s}_a} \leq x_a \leq v_a^{\bar{s}_a}, \quad a \in A, \tag{84}$$

$$x_a^k \geq 0, \quad a \in A, k \in K. \tag{85}$$

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 upon 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 1a 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 1b of the procedure. Subsequently, we call the slope scaling procedure in two modes: 1) in conjunction with the subgradient method used in Steps 1b and 1c; 2) as part of solving of the MIP Lagrangian subproblem in Steps 1d and 2c. 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 *post-optimization 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 1b and 1c:

- 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 1b and 1c. The pool of feasible solutions thus obtained is used in the post-optimization procedure outlined below.

When solving the MIP Lagrangian subproblem in Steps 1d and 2c, 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 1d and 2c, we give as input to the post-optimization 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 1b, 1c, 1d and 2c of the Lagrangian dual optimization procedure, the following post-optimization 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) (where $l_a^s = b_a^{s-1} + 1$, $a \in A$, $s \in S_a$) with the addition of the constraints:

$$\sum_{s \in S_a} y_a^s = 0, \quad a \in \bar{A}. \tag{86}$$

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. In 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 post-optimization procedure is the best feasible solution found this way, which is used to improve upon 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 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 cutset 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.

5.1 Set of Instances

We obtained the problem instances from a network generator similar to the one described in [10] 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 $|D| \leq |N| - |O|$, 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 [12]: 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 $b_a^s = \frac{s^2 D}{S^2}$, for each arc a, so that the segment length increases as s increases, as is typical of transportation costs [2]. We then adjusted the number of segments on each arc a so that $b_a^{|S_a|} = \min\{T, u_a\}$. Given variable costs, breakpoints, 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 $b_a^s = \left\lceil \frac{T}{S} \right\rceil$, 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 1000.

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 [12]: 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), with the strengthened segment lower bounds $l_a^s = b_a^{s-1} + 1$, $a \in A$, $s \in S_a$.
- ES: The extended segment-based model defined by (58)-(62) and (67).
- $ES+$: This is model ES with the addition of the segment-based Chvàtal-Gomory rank 1 valid inequalities (38)-(39) restricted to single-node cuts.
- $EP+$: This is model ES with the addition the constraints defining the point-based variables, (63), (64) and (66), along with the point-based Chvàtal-Gomory rank 1 valid inequalities (36)-(37) 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 1b), $v(\overline{ES+})$ (Step 1c), $v(\overline{EP+})$ (Steps 1d, 2c 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 Chvàtal-Gomory rank 1 valid inequalities (36)-(39) 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 (8)-(13); $v(\overline{ES})$, 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 Chvàtal-Gomory rank 1 valid inequalities (36)-(39) for $p > 1$ are declared as *lazy* constraints.

- *BB*, the B&B method of CPLEX (with default options) performed for a limit of 1 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 1 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 1b of method *LD*), *ES+* (in Step 1c of method *LD*) and *EP+* (in Steps 1d, 2c 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 1a to 1d of the Lagrangian dual optimization procedure presented in Section 3.2.

- *LPS*, the LP-based strategy performed in Steps 2a to 2c 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 1d 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 1 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 the strategies generate similar lower bounds, with a slight edge for the combined approach *LAG+LPS*. We observed that, for some instances, *LAG*

$(\lvert N\rvert, \lvert A\rvert)$	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, 13829
(20,100)	0.94, 1.02, 22	0.93, 1.04, 40	0.94, 1.02, 62	0.94, 1.02, 24391
(25,100)	0.92, 1.01, 17	0.93, 1.04, 26	0.93, 1.01, 44	0.92, 1.01, 31380
(25,150)	0.92, 1.05, 36	0.92, 1.07, 33	0.92, 1.04, 69	0.92, 1.05, 75612
Small	0.93, 1.03, 23	0.93, 1.05, 28	0.93, 1.02, 51	0.93, 1.03, 36303
(20,75)	0.93, 1.01, 33	0.94, 1.03, 37	0.94, 1.01, 70	0.93, 1.01, 17769
(20,100)	0.94, 1.01, 13	0.94, 1.03, 32	0.94, 1.01, 45	0.94, 1.01, 31774
(25,100)	0.93, 1.02, 34	0.93, 1.04, 28	0.93, 1.02, 62	0.93, 1.02, 34363
(25,150)	0.91, 1.02, 27	0.91, 1.05, 65	0.91, 1.02, 92	0.91, 1.02, 80222
Medium	0.93, 1.02, 27	0.93, 1.04, 41	0.93, 1.02, 68	0.93, 1.01, 41302
(20,75)	0.95, 1.01, 19	0.96, 1.01, 50	0.96, 1.00, 69	0.95, 1.01, 5374
(20,100)	0.95, 1.00, 19	0.95, 1.01, 66	0.95, 1.00, 85	0.95, 1.00, 14613
(25,100)	0.94, 1.01, 40	0.95, 1.01, 127	0.95, 1.00, 166	0.94, 1.01, 25644
(25,150)	0.94, 1.01, 188	0.95, 1.03, 165	0.95, 1.00, 353	0.94, 1.01, 43103
Large	0.95, 1.01, 67	0.95, 1.02, 102	0.95, 1.00, 168	0.95, 1.01, 22184

Table 1: Strategies to approximate $v(\overline{EP+})$: lower bound GAP, upper bound GAP, CPU

produces better lower bounds than *LPS*, while 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 improving 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 1d of the Lagrangian dual optimization procedure.

Table 2 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 1, for each combination of model and lower bounding method described in Section 5.2 (with the exception of *BB* whose results are shown below in Table 4).

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 (40).

- The Lagrangian dual optimization procedure provides effective lower bound approx-

$(\|N\|, \|A\|)$	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, 1294
	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, 1169
	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, 1095
	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, 1103
	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

Table 2: Lower bounds: lower bound GAP, CPU

imations, 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. Small improvements (on the order of 1-2%) are obtained by adding segment-based cutset inequalities through formulation $ES+$. When adding to this last model the point-based cutset 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 1 hour. Table 3 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 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 non-optimal 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. In contrast, the strongest model $EP+$ identifies a feasible solution within 1 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 cutset 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 3 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, .

$(\|N\|, \|A\|)$	Algo	BS	ES	$ES+$	$EP+$
(20,75)	BB	24, 20, 1066	24, 20, 1260	24, 21, 935	24, 8, 3175
	LH	24, 0, 0	24, 0, 1	24, 0, 8	24, 0, 30
(20,100)	BB	24, 18, 1311	24, 18, 1489	24, 19, 1307	24, 8, 3222
	LH	24, 0, 0	24, 0, 0	24, 0, 9	24, 0, 62
(25,100)	BB	24, 16, 1924	24, 13, 2361	24, 15, 2151	22, 0, 3613
	LH	24, 0, 0	24, 0, 3	24, 0, 7	24, 0, 44
(25,150)	BB	24, 5, 3103	24, 9, 2774	24, 8, 2828	12, 0, 3632
	LH	24, 0, 0	24, 0, 1	24, 0, 10	24, 0, 69
Small	BB	96, 59, 1851	96, 60, 1971	96, 63, 1805	82, 16, 3411
	LH	96, 0, 0	96, 0, 1	96, 0, 9	96, 0, 51
(20,75)	BB	24, 12, 2099	24, 10, 2545	24, 12, 2295	17, 2, 3584
	LH	24, 0, 0	24, 0, 2	24, 0, 10	24, 0, 70
(20,100)	BB	24, 9, 2492	24, 10, 2554	24, 9, 2509	17, 2, 3578
	LH	24, 0, 0	24, 0, 1	24, 0, 5	24, 0, 45
(25,100)	BB	24, 8, 2914	23, 6, 2937	24, 8, 3008	1, 0, 3644
	LH	24, 0, 1	24, 0, 3	24, 0, 11	24, 0, 62
(25,150)	BB	24, 3, 3274	18, 2, 3419	20, 3, 3364	0, 0, 3730
	LH	24, 0, 1	24, 0, 3	24, 0, 5	24, 0, 92
Medium	BB	96, 32, 2695	89, 28, 2864	92, 32, 2794	35, 4, 3634
	LH	96, 0, 1	96, 0, 2	96, 0, 8	96, 0, 67
(20,75)	BB	24, 20, 968	21, 12, 2685	24, 15, 2193	0, 0, 3642
	LH	24, 0, 1	24, 0, 3	24, 0, 5	24, 0, 69
(20,100)	BB	24, 11, 2337	21,5, 3382	24, 6, 3152	0, 0, 3680
	LH	24, 0, 1	24, 0, 7	24, 0, 8	24, 0, 85
(25,100)	BB	24, 8, 2742	14, 0, 3642	22, 1, 3597	0, 0, 3721
	LH	24, 0, 1	24, 0, 18	24, 0, 19	24, 0, 166
(25,150)	BB	24, 3, 3385	11, 0, 3648	19, 1, 3662	0, 0, 3818
	LH	24, 0, 1	24, 0, 155	24, 0, 161	24, 0, 353
Large	BB	96, 42, 2358	67, 17, 3339	89, 22, 3151	0, 0, 3715
	LH	96, 0, 1	96, 0, 46	96, 0, 48	96, 0, 168

Table 3: Upper bounds: Nfeas, Nopti, CPU

As a further comparison between *BB* and *LH*, Table 4 shows the results obtained with the best method-model combination for each of the two methods. For *BB*, as just seen in Table 3, the best model is BS, while 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 1d, 2c 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 3. Table 4 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 times 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 1 minute on all instances. On Large instances with size $(25, 150)$, the computing time is around 5 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 gets 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 (PMFI). 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 [12]. We have strengthened the discretized models either by adding valid inequalities derived from cutset inequalities or by using flow disaggregation techniques, obtaining a model similar to the so-called *extended (segment-based) formulation* introduced in [12].

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

- Discretization provides stronger cutset inequalities than those obtained from segment-based models.
- Discretization, when combined with flow disaggregation, does not improve upon 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 of both techniques: cutset inequalities based on discretization and flow disag-

$(\|N\|, \|A\|)$	Algo-Model	
(20,75)	BB-BS	1.00, 1.00, 1066
	LH-$EP+$	0.94, 1.02, 30
(20,100)	BB-BS	0.99, 1.00, 1311
	LH-$EP+$	0.94, 1.02, 62
(25,100)	BB-BS	1.00, 1.00, 1924
	LH-$EP+$	0.93, 1.01, 44
(25,150)	BB-BS	0.97, 1.00, 3103
	LH-$EP+$	0.92, 1.04, 69
Small	BB-BS	0.99, 1.00, 1851
	LH-$EP+$	0.93, 1.02, 51
(20,75)	BB-BS	0.99, 1.00, 2099
	LH-$EP+$	0.94, 1.01, 70
(20,100)	BB-BS	0.98, 1.00, 2492
	LH-$EP+$	0.94, 1.01, 45
(25,100)	BB-BS	0.97, 1.00, 2914
	LH-$EP+$	0.93, 1.02, 62
(25,150)	BB-BS	0.92, 1.02, 3274
	LH-$EP+$	0.91, 1.02, 92
Medium	BB-BS	0.97, 1.01, 2695
	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, 2337
	LH-$EP+$	0.95, 1.00, 85
(25,100)	BB-BS	0.97, 1.01, 2742
	LH-$EP+$	0.95, 1.00, 166
(25,150)	BB-BS	0.95, 1.03, 3385
	LH-$EP+$	0.95, 1.00, 353
Large	BB-BS	0.98, 1.01, 2358
	LH-$EP+$	0.95, 1.00, 168

Table 4: Upper bounds: lower bound GAP, upper bound GAP, CPU

gregation with segment-based variables. In order 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 cutset inequalities with or without discretization), when used within the Lagrangian relaxation approach. The results derived from these experiments show 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, while 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 the best of 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 [14, 15, 18]).

Acknowledgments

We are grateful to Serge Bisaillon for his help with implementing and testing the algorithms. We also gratefully acknowledge financial support for this project. The work of the first author has been supported by NSERC (Canada) under grant 184122-09. The work of the second author has been supported by the Fundação para a Ciência e Tecnologia under project PEst-OE/MAT/UI0152.

References

[1] A. Atamtürk. On Capacitated Network Design Cut-Set Polyhedra. *Mathematical Programming*, 92:425–437, 2002.

[2] A. Balakrishnan and S. Graves. A Composite Algorithm for a Concave-Cost Network Flow Problem. *Networks*, 19:175–202, 1989.

[3] A. Balakrishnan, T.L. Magnanti, and P. Mirchandani. Network Design. In M. Dell'Amico, F. Maffioli, and S. Martello, editors, *Annotated Bibliographies in Combinatorial Optimization*, pages 311–334. John Wiley and Sons, 1997.

[4] F. Barahona. Network Design Using Cut Inequalities. *SIAM Journal on Optimization*, 6:823–837, 1996.

[5] D. Bienstock, S. Chopra, O. Günlük, and C. Tsai. Minimum Cost Capacity Installation for Multicommodity Network Flows. *Mathematical Programming*, 81:177–199, 1998.

[6] D. Bienstock and O. Günlük. Capacitated Network Design–Polyhedral Structure and Computation. *INFORMS Journal of Computing*, 8:243–259, 1996.

[7] M. Chouman, T.G. Crainic, and B. Gendron. Commodity Representations and Cutset-Based Inequalities for Multicommodity Capacitated Fixed-Charge Network Design. Publication cirrelt-2011-56, CIRRELT, Université de Montréal, 2011.

[8] I. Correia, L. Gouveia, and F. Saldanha da Gama. Discretized Formulations for Capacitated Location Problems with Modular Distribution Costs. *European Journal of Operational Research*, 204:237–244, 2010.

[9] T.G. Crainic. Service Network Design in Freight Transportation. *European Journal of Operational Research*, 122:272–288, 2000.

[10] T.G. Crainic, A. Frangioni, and B. Gendron. Bundle-Based Relaxation Methods for Multicommodity Capacitated Fixed Charge Network Design Problems. *Discrete Applied Mathematics*, 112:73–99, 2001.

[11] T.G. Crainic, B. Gendron, and G. Hernu. A Slope Scaling/Lagrangean Perturbation Heuristic with Long-Term Memory for Multicommodity Capacitated Fixed-Charge Network Design. *Journal of Heuristics*, 10:525–545, 2004.

[12] K.L. Croxton, B. Gendron, and T.L. Magnanti. Variable Disaggregation in Network Flow Problems with Piecewise Linear Costs. *Operations Research*, 55:146–157, 2007.

[13] C. Duhamel, L. Gouveia, P. Moura, and M. Souza. Minimum Cost Degree Constrained Spanning Trees with Node-Degree Costs. *Networks*, 50:1–18, 2012.

[14] A. Frangioni and B. Gendron. 0-1 Reformulations of the Multicommodity Capacitated Network Design Problem. *Discrete Applied Mathematics*, 157:1229–1241, 2009.

[15] A. Frangioni and B. Gendron. A Stabilized Structured Dantzig-Wolfe Decomposition Method. *Mathematical Programming*, 140:45–76, 2013.

[16] V. Gabrel, K. Knippel, and M. Minoux. A Comparison of Heuristics for the Discrete Cost Multicommodity Network Optimization Problem. *Journal of Heuristics*, 9:429–445, 2003.

[17] B. Gendron, T.G. Crainic, and A. Frangioni. Multicommodity Capacitated Network Design. In Soriano, P. and Sansò, B., editors, *Telecommunications Network Planning*, pages 1–19. Kluwer Academics Publisher, 1999.

[18] B. Gendron and M. Larose. Branch-and-Price-and-Cut for Large-Scale Multicommodity Capacitated Fixed-Charge Network Design. Publication cirrelt-2012-74, CIRRELT, Université de Montréal, 2012.

[19] A. M. Geoffrion. Lagrangian Relaxation for Integer Programming. *Mathematical Programming Studies*, 2:82–114, 1974.

[20] L. Gouveia. A $2n$ Constraint Formulation for the Capacitated Minimal Spanning Tree Problem. *Operations Research*, 43:130–141, 1995.

[21] L. Gouveia and P. Moura. Enhancing Discretized Formulations: the Knapsack Reformulation and the Star Reformulation. *TOP*, 20:52–74, 2012.

[22] L. Gouveia and F. Saldanha da Gama. On the Capacitated Concentrator Location Problem: a Reformulation by Discretization. *Computers and Operations Research*, 33:1242–1258, 2006.

[23] O. Günlük. A Branch-and-Cut Algorithm for Capacitated Network Design Problems. *Mathematical Programming*, 86:17–39, 1999.

[24] M. Held, P. Wolfe, and H.P. Crowder. Validation of Subgradient Optimization. *Mathematical Programming*, 6:62–88, 1974.

[25] K. Holmberg and D. Yuan. A Lagrangean Heuristic Based Branch-and-Bound Approach for the Capacitated Network Design Problem. *Operations Research*, 48:461–481, 2000.

[26] D. Kim and P.M. Pardalos. A Solution Approach to the Fixed Charge Network Flow Problem Using a Dynamic Slope Scaling Procedure. *Operations Research Letters*, 24:195–203, 1999.

[27] D. Kim and P.M. Pardalos. Dynamic Slope Scaling and Trust Interval Techniques for Solving Concave Piecewise Linear Network Flow Problems. *Networks*, 35:216–222, 2000.

[28] T.L. Magnanti, P. Mirchandani, and R. Vachani. The Convex Hull of Two Core Capacitated Network Design Problems. *Mathematical Programming*, 60:233–250, 1993.

[29] T.L. Magnanti and R.T. Wong. Network Design and Transportation Planning: Models and Algorithms. *Transportation Science*, 18:1–55, 1984.

[30] M. Minoux. Network Synthesis and Optimum Network Design Problems: Models, Solution Methods and Applications. *Networks*, 19:313–360, 1989.

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

[32] C. Raack, A.M.C.A. Koster, S. Orlowski, and R. Wessäly. On cut-based inequalities for capacitated network design polyhedra. *Networks*, 57:141–156, 2011.