Models for the Piecewise Linear Unsplittable Multicommodity Flow Problems

Bernard Fortz[a], Luís Gouveia[b], Martim Joyce-Moniz[a,*]

[a]*Université Libre de Bruxelles, CP 210/01, Boulevard du Triomphe, 1050 Bruxelles, Belgium and INOCS, INRIA Lille Nord-Europe*
[b]*Departamento de Estatística e Investigação Operacional, CMAF-CIO, Faculdade de Ciências da Universidade de Lisboa, Bloco C / 6 - Campo Grande, Cidade Universitária, 1950-409 Lisboa, Portugal*

Abstract

In this paper, we consider multicommodity flow problems, with unsplittable flows and piecewise linear routing costs. We first focus on the case where the piecewise linear routing costs are convex. We show that this problem is $\mathcal{NP}$-hard for the general case, but polynomially solvable when there is only one commodity. We then propose a strengthened mixed-integer programming formulation for the problem. We show that the linear relaxation of this formulation always gives the optimal solution of the problem for the single commodity case. We present a wide array of computational experiments, showing this formulation also produces very tight linear programming bounds for the multi-commodity case. Finally, we also adapt our formulation for the non-convex case. Our experimental results imply that the linear programming bounds for this case, are only slightly than the ones of state-of-the-art models for the splittable flow version of the problem.

Keywords: OR in telecommunications, Networks, Unsplittable Flows, Integer programming, Combinatorial optimization

*Corresponding author
Email addresses: `bernard.fortz@ulb.ac.be` (Bernard Fortz), `legouveia@fc.ul.pt` (Luís Gouveia), `martim.moniz@ulb.ac.be` (Martim Joyce-Moniz)

1. Introduction

In this paper, we study unsplittable multicommodity flow problems with piecewise linear cost functions. These fundamental problems often arise as the backbone of other more intricate optimization problems, with applications in telecommunications, logistics and transportation. One such example can be found in the recent work of Papadimitriou and Fortz [1, 2], in the context of a complex multi-period design and routing problem. Lower bounds resulting from the linear programming (LP) relaxation of the problem are very weak, and part of this weakness is due to the piecewise linear objective function combined with single-path routing. However, to the best of our knowledge, these basic - yet not trivial - problems have never been considered by themselves. We hereby provide and compare mixed-integer programming models for these problems. Stronger formulations provided in this paper could be embedded in models, such as in [1, 2], to improve the lower bounds and make the problems more tractable. We focus on the case where these functions are convex. However, later, we also discuss the situation when the cost functions are non-convex.

We begin by defining the convex unsplittable multicommodity flow (PUMF) problem. Let $G = (V, A)$ be a directed graph, with a set of nodes V, and a set of arcs A. Consider as well a set of commodities K, where each commodity $k \in K$ has a given origin o_k, a destination d_k, and a demand ρ_k to be routed from o_k to d_k.

Each arc $a \in A$ has an associated cost function $g_a(l_a)$ of the load flowing through the arc, l_a. This cost function is continuous, convex and piecewise linear, with the segments being indexed by the finite set $S_a = \{1, 2, ..., |S_a|\}$. Each segment $s \in S_a$ has a lower and upper bound on the flow, represented by the breakpoints b_a^{s-1} and b_a^s. If finite, the breakpoint of the last segment of each arc $a \in A$, $b_a^{|S_a|}$, can be interpreted as the capacity of the arc. However, the case where $b_a^{|S_a|} = \infty$ also holds. A segment is also characterized by a slope c_a^s and an intercept f_a^s. In conclusion, the cost function g_a is such that $g_a(l_a) = f_a^s + c_a^s l_a$ if $b_a^{s-1} \leq l_a \leq b_a^s$, for $s \in S_a$. Figure 1 illustrates this notation. Since in the PUMF problem, the cost functions are convex, these values must be such that $c_a^1 \geq 0$, $c_a^s > c_a^{s-1}$ and $f_a^s \leq 0$, $f_a^s < f_a^{s-1}$. Moreover, as we consider the cost function to be continuous, we assume that $b_a^s c_a^s + f_a^s = b_a^s c_a^{s+1} + f_a^{s+1}$. We also assume that for every arc $a \in A$, $g_a(0) = 0$, and consequently, $f_a^1 = 0$. The PUMF problem is to find a single path for each commodity, such that the sum of the costs associated with the load of
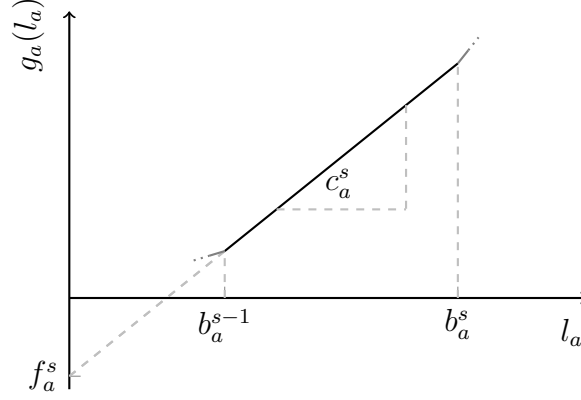
the arcs is minimized.

Figure 1: Notation for each segment of $g_a(l_a)$

In the PUMF problem, the routing costs, given by the piecewise linear functions, play a double role: they confine the load within the arcs' capacities, and at the same time, they help to avoid unnecessary detours in the network - since the function is additive, solutions with short paths will be favored. An example of such a cost function was proposed by Fortz and Thorup [3, 4] and is illustrated in Figure 2.

In this function, the cost of sending flow is cheap for arcs with a small utilization (ratio between the arc's load and its capacity). However, the price quickly rises when the utilization approaches the arc's capacity. Even though it is possible for the utilization to go above 100%, this is so heavily penalized in the cost, that such a solution will likely be avoided.

Objective functions like this have been widely used in problems related with Traffic Engineering (TE) in internet networks. One of the most important and well known of such objectives is the Kleinrock delay function [5]:

$$F = \sum_{a \in A} \frac{l_a}{C_a - l_a},$$

(1)

where C_a denotes the capacity of link $a \in A$. The Kleinrock function helps avoiding congestion by penalizing heavily loaded links. Observe that this objective function is convex, and so it can be optimized using convex programming methods [6]. Nevertheless, due to the discrete nature of the PUMF

problem, it is convenient to approximate the Kleinrock function with a convex piecewise linear function ([3, 4, 7]), leading to a mixed-integer linear optimization problem, that can be solved with the powerful mixed-integer programming solvers available today. Balon et al. [8] and Gourdin [9] discuss various TE objective functions. These authors evaluate how well different objective functions meet TE requirements, and conclude that piecewise linear objectives provide a good trade-off between different measures of quality of service.

Many multicommodity flow problems have been considered in the literature, but the following two can be identified as more closely related to the PUMF problem: the bandwidth packing (BP) problem, and the non-convex piecewise linear multicommodity network flow (NCPMF) problem. We now highlight the main similarities and differences between these two problems and the PUMF problem.

The BP problem was first introduced by Cox *et al.* [10], and has also been referred to as the origin-destination integer multicommodity flow problem in [11], as the unsplittable multicommodity flow problem in [12] and as the minimum cost multiple-source unsplittable flow problem in [13]. Other works on exact methods to this problem include [14], [15] and [16]. This problem shares with the PUMF problem the "unsplitability" of the flows. Nonetheless,
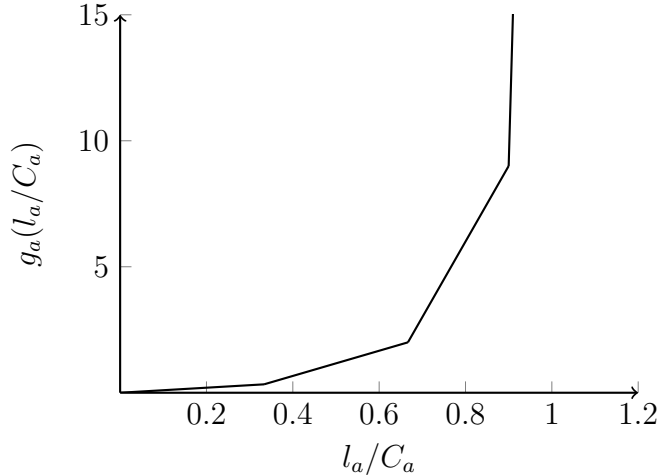

Figure 2: Example of a convex piecewise linear cost function

4

it differs in the two following ways: the flow costs are directly proportional to the load on the arcs; and there are explicit capacity constraints on the amount of flow traversing an arc, whereas in the PUMF problem this is somewhat enforced by the piecewise linear cost functions (as explained above). Another akin class of problems deals with the design of unsplittable multicommodity networks with modular capacities on the arcs [17, 18]. In these problems, the objective is to minimize the costs of the capacity modules; thus, the cost functions are step-wise, rather than piecewise linear. Splittable flow problems dealing with this type of cost functions have also been considered in the literature (*e.g.* [19]).

The NCPMF problem, like the PUMF problem, also imposes costs on the flow via piecewise linear functions. However, unlike the cost functions in the PUMF problem, the piecewise linear functions in the NCPMF problem are non-convex. Later we will consider a version of the PUMF problem where the cost functions are also non-convex. Finally, the major difference between the two problems is that the flows in the NCPMF problems are splittable, *i.e.* this problem considers multipath routing. Croxton *et al.* [20] review three formulations that had been previously used in the literature for generic problems with non-convex piecewise linear costs. These three formulations use integer variables to model the costs, and are shown to be equivalent, with respect to their LP relaxation. In [21], the same authors adapt the formulation with multiple choice constraints [22], to model the NCPMF problem. Recently, Gendron and Gouveia [23] considered a version of the NCPMF problem where the load on the arc must be integer; this is, however, still a problem with splittable flows. Due to the common properties of the two problems, these works provide an important basis for our work.

In Section 2, we discuss the complexity of the PUMF problem. In Section 3, we propose mixed-integer programming formulations that model the problem. In Section 4, we present the results of computational experiments, that help us compare the performance of these formulations in solving instances of the PUMF problem. Later, in Section 5 we study a variant of the PUMF problem, where the cost functions are non-convex. Finally, in Section 6 we draw conclusions on our work, and discuss future developments.

2. Problem complexity

In this section, we analyse the complexity of the PUMF problem.

Lemma 1. *For $|K| = 1$, the PUMF problem is polynomially solvable.*

Proof. Let us consider an instance of PUMF given by a graph $G = (V, A)$ and a single commodity with origin $o \in V$, destination $d \in V$ and demand ρ. To each arc $a \in A$ we associate a length w_a, that represents the potential cost of having the commodity flow through it, i.e. $w_a = f_a^{\bar{s}_a} + c_a^{\bar{s}_a} \rho$, with $\bar{s}_a$ defined such that $\bar{s}_a = \{s : b_a^{s-1} \leq \rho \leq b_a^s\}$; or $w_a = \infty$ if $\rho > b_a^{|S_a|}$. Solving the single-commodity PUMF problem on G is equivalent to identifying the shortest path between o and d in G with respect to lengths w_a.

$\square$

Lemma 1 and its proof will be revisited later on, to show Theorem 5.

Lemma 2. *For $|K| > 1$, the PUMF problem is $\mathcal{NP}$-complete.*

Proof. This can be shown by reducing the bin-packing problem to the PUMF problem. Similar proofs are found in [18, 24] for capacitated unsplittable flow problems; we hereby present a proof for the uncapacitated case.

Consider an instance of the bin-packing decision problem, with $|A|$ bins with capacity $\bar{b}$, and $|K|$ objects with size $\rho_k \in \mathbb{N}$. Consider as well a directed graph $G = (V, A)$ with only two nodes, o and d, and $|A|$ parallel arcs connecting them. The flow cost of every arc $a \in A$ is given by the same piecewise linear function $g(x)$, that has only two segments, separated by the breakpoint $b^1 = \bar{b}$. The slope on the second segment is very steep, in such a way that $g_a(\bar{b}+1) > \sum_{a' \in A} g_{a'}(\bar{b}), a \in A$. Finally, let K be a set of commodities, each $k \in K$ with a given demand ρ_k, origin in o and destination in d. The problem of determining if it is possible to fit the $|K|$ objects in the $|A|$ bins, is equivalent to determining if all the commodities in K are routable in G, with a cost not bigger than $\sum_{a \in A} g_a(\bar{b})$. As the bin-packing problem is $\mathcal{NP}$-complete [25], so is the PUMF problem. $\square$

3. Problem formulation

In this section, we consider three formulations for the PUMF problem that were introduced in [26] and prove some fundamental properties of these formulations. Let λ_i^k be such that $\lambda_{o_k}^k = 1$, $\lambda_{d_k}^k = -1$, and $\lambda_i^k = 0$ for every $i \neq \{o_k, d_k\}$. We define binary variables x_a^k, with $x_a^k = 1$ if arc $a \in A$ is on the unique path chosen to route commodity $k \in K$, and $x_a^k = 0$ otherwise; and y_a^s, with $y_a^s = 1$, if arc $a \in A$ contains a non-zero flow on segment $s \in S_a$, and $y_a^s = 0$ otherwise. For the sake of simplicity, if arc a contains a non-zero flow on segment $s \in S_a$, we say that arc a is on segment $s \in S_a$. We also define

continuous variables l_a^s, that indicate the load going through arc $a \in A$ on segment $s \in S_a$. The PUMF problem can be formulated with the following mixed integer linear program (MIP), which we refer to as the Basic Model 1 (BM1).

The typical multicommodity flow balance constraints (2b) define the path between the origin and destination node of each commodity. Then, constraint sets (2c-2e) identify the segment each arc is on. Naturally, only a single segment per arc may be selected (2c). The choice of segment is implied by the load flowing through the respective arc. This load is given by constraints (2d) and its value is assigned to one of the variables l. To ensure that only the appropriate load variable is non-zero, in (2e) we either bound l_a^s by the breakpoints of the corresponding segment, if $y_a^s = 1$; or we force it to zero, if $y_a^s = 0$.

$$\min \quad \sum_{a \in A} \sum_{s \in S_a} (f_a^s y_a^s + c_a^s l_a^s) \tag{2a}$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(i)} x_a^k - \sum_{a \in \delta^-(i)} x_a^k = \lambda_i^k, \qquad i \in V, k \in K, \tag{2b}$$

$$\sum_{s \in S_a} y_a^s \leq 1, \qquad a \in A, \tag{2c}$$

$$\sum_{s \in S_a} l_a^s = \sum_{k \in K} \rho^k x_a^k, \qquad a \in A, \tag{2d}$$

$$b_a^{s-1} y_a^s \leq l_a^s \leq b_a^s y_a^s, \qquad a \in A, s \in S_a, \tag{2e}$$

$$x_a^k \in \{0,1\}, \qquad a \in A, k \in K, \tag{2f}$$

$$y_a^s \in \{0,1\}, \qquad a \in A, s \in S_a, \tag{2g}$$

$$l_a^s \geq 0, \qquad a \in A, s \in S_a. \tag{2h}$$

Variables x and y are considered as binary (2f-2g), whilst variables l are considered as continuous (2h). Note that if the cost function for every arc is convex, it is not necessary to define explicitly y as binary; instead they can simply be defined as continuous and non-negative. The impact on the effectiveness of the model of defining variables y as binary or continuous is discussed in Section 4.

Theorem 3. *If cost functions g_a are convex for all $a \in A$, then the optimal solution of the LP relaxation of BM1 has binary y-variables.*

7

Proof. Consider a solution of BM1, $\hat{\mathcal{S}}$, with cost $\hat{g}$, and fractional y-variables. Without loss of generality, we assume that this only occurs for one arc - arc a. Then, $\exists s \in S : \hat{y}_a^s \in]0, 1[$. We show that there exists a feasible solution for BM1 $\tilde{\mathcal{S}}$, with $\tilde{x}_a^k = \hat{x}_a^k, a \in A, k \in K$, binary y-variables, and cost $\tilde{g} : \tilde{g} \leq \hat{g}$.

Let us first assume the case where $\sum_{s \in S_a} \hat{y}_a^s < 1$. As $f_a^1 = 0$, it is possible to increase the value of $\hat{y}_a^1$ without increasing $\hat{g}$. Since $b_a^0 = 0$, this new solution would satisfy (2e). As such, let us re-define $\hat{y}_a^1 := 1 - \sum_{s \in S_a \setminus \{1\}} \hat{y}_a^s$ instead. Then, $\sum_{s \in S_a} \hat{y}_a^s = 1$.

Now, let us consider the case where $\sum_{s \in S_a} \hat{y}_a^s = 1$. Then, we show that solution $\tilde{\mathcal{S}}$, with $\tilde{y}_a^{\bar{s}} = 1$ and $\tilde{l}_a^{\bar{s}} = \sum_{s \in S_a} \hat{l}_a^s$ where $\bar{s} = \{s : b_a^{s-1} \leq \sum_{s \in S_a} \hat{l}_a^s \leq b_a^s\}$, is not more expensive than $\hat{\mathcal{S}}$.

First, let us define $\breve{l}_a^s = \frac{\hat{l}_a^s}{\hat{y}_a^s}$ and $\breve{y}_a^s = 1$, if $\hat{y}_a^s > 0$ and $\breve{l}_a^s = 0$ and $\breve{y}_a^s = 0$ otherwise. Values $\breve{l}$ and $\breve{y}_a^s$ satisfy (2e), as for every $s \in S_a$ such that $\hat{y}_a^s > 0$:

$b_a^{s-1} \hat{y}_a^s \leq \hat{l}_a^s \leq b_a^s \hat{y}_a^s \Leftrightarrow b_a^{s-1} \leq \frac{\hat{l}_a^s}{\hat{y}_a^s} \leq b_a^s \Leftrightarrow b_a^{s-1} \breve{y}_a^s \leq \breve{l}_a^s \leq b_a^s \breve{y}_a^s$.

Then, we have that:

$\tilde{g} = f_a^{\bar{s}} \tilde{y}_a^{\bar{s}} + c_a^{\bar{s}} \tilde{l}_a^{\bar{s}} = f_a^{\bar{s}} \sum_{s \in S_a} \hat{y}_a^s + c_a^{\bar{s}} \sum_{s \in S_a} \hat{l}_a^s = f_a^{\bar{s}} \sum_{s \in S_a} \hat{y}_a^s \breve{y}_a^s + c_a^{\bar{s}} \sum_{s \in S_a} \hat{y}_a^s \breve{l}_a^s$.

Note that this implies that solution $\tilde{\mathcal{S}}$ is a convex combination of feasible solutions with load $\breve{l}_a^s$. As the cost functions are convex, we know that:

$\tilde{g} = f_a^{\bar{s}} \sum_{s \in S_a} \hat{y}_a^s \breve{y}_a^s + c_a^{\bar{s}} \sum_{s \in S_a} \hat{y}_a^s \breve{l}_a^s \leq \sum_{s \in S_a} \hat{y}_a^s (f_a^s \breve{y}_a^s + c_a^s \breve{l}_a^s) = \sum_{s \in S_a} (f_a^s \hat{y}_a^s \breve{y}_a^s + c_a^s \hat{y}_a^s \breve{l}_a^s) = \sum_{s \in S_a} (f_a^s \hat{y}_a^s + c_a^s \hat{l}_a^s) = \hat{g}$. $\square$

The BM1 is a multiple choice model, following the terminology in [22]; a single segment is chosen per arc, which allows for the direct pricing of the flow, based on the selected segment's intercept and slope. An alternative way of modelling costs, imposed by a piecewise linear cost function is suggested in [3]. The authors define variables that stand for the routing cost of each arc, to which they impose lower bounds representing the segments of the cost function. Let g_a be the routing cost for arc $a \in A$. We denote the following

MIP as the Basic Model 2 (BM2):

$$\min \quad \sum_{a \in A} g_a \tag{3a}$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(i)} x_a^k - \sum_{a \in \delta^-(i)} x_a^k = \lambda_i^k, \qquad i \in V, k \in K, \tag{3b}$$

$$l_a = \sum_{k \in K} \rho^k x_a^k, \qquad a \in A, \tag{3c}$$

$$g_a \geq f_a^s + c_a^s l_a, \qquad a \in A, s \in S_a, \tag{3d}$$

$$x_a^k \in \{0, 1\}, \qquad a \in A, k \in K, \tag{3e}$$

$$l_a \geq 0, \qquad a \in A, s \in S_a, \tag{3f}$$

$$g_a \geq 0, \qquad a \in A. \tag{3g}$$

Constraints (3c) determine the flow traversing each arc, based on the routing defined in (3b). In (3d), for each segment $s \in S_a$, a lower bound is set for the routing cost g_a of arc $a \in A$. Note that for a given value of l_a, only a single constraint is tight - the one corresponding to the segment arc a is on. Finally, our objective (3a) is to minimize the sum of the routing costs for every arc.

Consider the LP relaxations of BM1 and BM2, where we relax the integrality constraints for variables x and y (when applicable). We denote as $g_{B1}(\mathcal{S})$ ($g_{B2}(\mathcal{S})$), the cost of a feasible solution $\mathcal{S}$ of the LP relaxation of BM1 (BM2), and as $g_{B1}{}^*$ ($g_{B2}{}^*$) the cost of the optimal solution to these relaxations.

Theorem 4. $g_{B1}{}^* = g_{B2}{}^*$.

Proof. We begin by showing that $g_{B2}{}^* \leq g_{B1}{}^*$. Let $\mathcal{P}_{B1}$ and $\mathcal{P}_{B2}$ be the polyhedra defined by the set of feasible solutions of the LP relaxation of BM1 and BM2, respectively. Consider the optimal solution of BM1, $\tilde{\mathcal{S}}^* = \{\tilde{x}_a^k, \tilde{l}_a^s, \tilde{y}_a^s\}$. We construct a solution $\hat{\mathcal{S}} = \{\tilde{x}_a^k, \hat{l}_a, \hat{g}_a\}$ by taking:

- $\hat{l}_a := \sum_{s \in S_a} \tilde{l}_a^s, \quad a \in A$;

- $\hat{g}_a := \max_{s \in S_a}(f_a^s + c_a^s \tilde{l}_a^s), \quad a \in A$.

It is to see that $\hat{\mathcal{S}} \in \mathcal{P}_{B2}$. Let us consider the cost of this solution, $g_{B2}(\hat{\mathcal{S}})$. We have that $g_{B2}(\hat{\mathcal{S}}) = \sum_{a \in A} \hat{g}_a = \sum_{a \in A} \max_{s \in S_a}(f_a^s + c_a^s \tilde{l}_a^s)$. Note that

9

$f_a^s \leq 0$, $a \in A$, and that $\tilde{y}_a^s \geq 0$, $a \in A, s \in S_a$. Then, $\sum_{a \in A} \max_{s \in S_a}(f_a^s + c_a^s \tilde{l}_a^s) \leq \sum_{a \in A} \max_{s \in S_a}(f_a^s \tilde{y}_a^s + c_a^s \tilde{l}_a^s)$. As it has been shown in Theorem 3, variables $\tilde{y}$ are binary, even if their integrality is not explicitly enforced. As such, and following constraints (2c,2e), we know that $f_a^s \tilde{y}_a^s + c_a^s \tilde{l}_a^s$ will only be non-zero for a single $s \in S_a$. Thus, we can conclude that $g_{B2}(\hat{S}) \leq \sum_{a \in A} \max_{s \in S_a}(f_a^s \tilde{y}_a^s + c_a^s \tilde{l}_a^s) \leq \sum_{a \in A} \sum_{s \in S_a}(f_a^s \tilde{y}_a^s + c_a^s \tilde{l}_a^s) = g_{B1}{}^*$, and consequently $g_{B2}{}^* \leq g_{B1}{}^*$.

For the converse, consider the optimal solution of BM2, $\hat{S}^* = \{\hat{x}_a^k, \hat{l}_a, \hat{g}_a\}$. Let us define $\tilde{S} = \{\hat{x}_a^k, \tilde{l}_a^s, \tilde{y}_a^s\}$ by taking:

- $\tilde{l}_a^s := \hat{l}_a$ if $b_a^{s-1} \leq \hat{l}_a \leq b_a^s$; 0 otherwise, $a \in A$.

- $\tilde{y}_a^s := 1$ if $b_a^{s-1} \leq \hat{l}_a \leq b_a^s$; 0 otherwise, $a \in A$.

Naturally, $\tilde{S}$ is a feasible solution in $\mathcal{P}_{B1}$. The cost of solution $\tilde{S}$ is $g_{B1}(\tilde{S}) = \sum_{a \in A, s \in S_a}(f_a^s \tilde{y}_a^s + c_a^s \tilde{l}_a^s)$. As discussed above, for each $a \in A$, $f_a^s \tilde{y}_a^s + c_a^s \tilde{l}_a^s$ is non-zero only for a single $s \in S_a$. As such, $g_{B1}(\tilde{S}) = \sum_{a \in A, s \in S_a} f_a^s \tilde{y}_a^s + c_a^s \tilde{l}_a^s \leq \sum_{a \in A} \max_{s \in S_a} f_a^s + c_a^s \hat{l}_a \leq \sum_{a \in A} \hat{g}_a = g_{B2}{}^*$. Therefore, we also have that $g_{B1}{}^* \leq g_{B2}{}^*$, and thus, $g_{B1}{}^* = g_{B2}{}^*$.   □

The LP relaxation of these basic models can provide very weak lower bounds, even for toy instances. As an example to show how weak these bounds are, consider a graph with only two nodes, o and d, connected by three parallel arcs. Assume a single commodity with origin o, destination d, and demand 3. Finally, assume that the cost function on all three arcs is the same, and is characterized by only two segments, such that the breakpoints are $b^0 = 0$, $b^1 = 1$ and $b^2 = 3$; the slopes are $c^1 = 1$ and $c^2 = 10$; and the intercepts $f^1 = 0$ and $f^2 = -9$. It is easy to see that there are three solutions to the PUMF problem on this graph, all with the same cost of 21: sending the flow through each one of the three available arcs. However, in the LP relaxation of basic models BM1 and BM2, the flow can be equally split among the three arcs, each on the first segment. This results in a LP relaxation optimum value of only 3. By manipulating the structure of the cost functions, this gap can be virtually any value. Therefore, it is important to strengthen this model, in order to solve our problem efficiently.

To this end, we use variable disaggregation, a common technique to strengthen the LP relaxation of MIPs. Thus, consider the binary variables

x_a^{ks}, with $x_a^{ks} = 1$ if arc $a \in A$ is on segment $s \in S_a$ and on the unique path chosen to route commodity $k \in K$, and $x_a^{ks} = 0$ otherwise. We denote as the Disaggregated Model (DM), the following formulation:

$$\min \quad \sum_{a \in A} \sum_{s \in S_a} \left(f_a^s y_a^s + c_a^s \sum_{k \in K} \rho^k x_a^{ks} \right) \tag{4a}$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(i)} \sum_{s \in S_a} x_a^{ks} - \sum_{a \in \delta^-(i)} \sum_{s \in S_a} x_a^{ks} = \lambda_i^k, i \in V, k \in K, \; (\alpha) \tag{4b}$$

$$\sum_{s \in S_a} y_a^s \leq 1, \qquad\qquad\qquad a \in A, \; (\beta) \tag{4c}$$

$$b_a^{s-1} y_a^s \leq \sum_{k \in K} \rho^k x_a^{ks} \leq b_a^s y_a^s, \qquad a \in A, s \in S_a, \; (\gamma) \tag{4d}$$

$$x_a^{ks} \in \{0,1\}, \qquad\qquad a \in A, k \in K, s \in S_a, \tag{4e}$$

$$y_a^s \in \{0,1\}, \qquad\qquad a \in A, s \in S_a. \tag{4f}$$

In this new model the l-variables are no longer necessary as for every $a \in A, s \in S_a$, $l_a^s = \sum_{k \in K} \rho^k x_a^{ks}$, leading to the new objective function (4a) and variable bound constraints (4d). Constraints (4b) define, for each commodity, a path between the origin and destination, now with the disaggregated x-variables. As for BM1, if the cost functions are convex, we can instead define the y-variables as continuous and non-negative. The role of variables α, β and γ will be revealed further on, in the proof of Theorem 5.

It is easy to see that DM, as it has been defined so far, is equivalent to the basic models. However, we can use the disaggregated variables to create new inequalities that considerably strengthen the LP relaxation.

First, note that an arc being traversed by a given commodity cannot be on a segment whose upper breakpoint is smaller than the demand flow. Therefore, we can fix the x-variables as follows:

$$x_a^{ks} = 0, \qquad\qquad a \in A, k \in K, s \in S_a : b_a^s < \rho^k. \tag{5}$$

Furthermore, when combined with inequalities (4d), this variable fixing has a strong impact on the values of the y-variables.

A well-known class of valid inequalities, common with this variable disaggregation, are the following:

$$x_a^{ks} \leq y_a^s, \qquad\qquad k \in K, a \in A, s \in S_a. \qquad (6)$$

These valid inequalities are an obvious choice in cases where the intercepts f_a^s are non-negative (*e.g.* [21]), as they lift the y-variables. This is not the case for PUMF. However, due to (4c), these valid inequalities can still be useful in cutting-off LP solutions; those that for a given arc a have $x_a^{ks} + x_a^{k's'} > 1$, $k' \neq k$, $s' \neq s$.

A related class of valid inequalities, but now making use of the fact that the intercepts f_a^s are negative, is obtained by tightening coefficients in the first inequality in (4d):

$$b_a^{s-1} y_a^s \leq \sum_{k \in K} \min(\rho^k, b_a^{s-1}) x_a^{ks}, \qquad\qquad a \in A, s \in S_a. \ (\zeta) \qquad (7)$$

Empirical results reveal that combining DM with both fixing variables (5) and valid inequalities (7) is highly advantageous, improving both the quality of the lower bounds provided by the LP relaxation, and the computing time of the MIPs. We refer to this strengthening of DM as the Strong Model (SM). We do not include valid inequalities (6) in SM, as empirical results show that they seldom improve the bounds of the LP relaxation (see above), and often cause out-of-memory issues for large instances.

Note that SM is larger than the previous models. Whereas the number of constraints is in the same order of magnitude, $\mathcal{O}(\max\{|V||K|, |A||S|\})$, there are more variables in SM, $\mathcal{O}(|A||S||K|)$, than in the basic models, $\mathcal{O}(\max\{|A||K|, |A||S|\})$. However, as we will see in the next section, the bounds obtained by the LP relaxation of SM are far superior than the ones obtained by the LP relaxations of the basic models.

Finally, we conclude this section by pointing out a relevant result about SM. In Theorem 1, we pointed out that the single-commodity PUMF problem is polynomially solvable. Still, in this case, the LP relaxation of the basic models can be fractional, as it was illustrated with the example following Theorem 4. Theorem 5 shows that the same is not true for SM.

Theorem 5. *When $|K| = 1$, the optimal solution of the linear relaxation of SM is integer and solves PUMF.*

Proof. First, let us recall what was shown in Theorem 1: solving PUMF for $|K| = 1$ in $G = (V, A)$ is equivalent to solving a shortest path problem

between o and d where the length associated with each arc in $a \in A$ is $w_a = f_a^{\bar{s}_a} + c_a^{\bar{s}_a}\rho$, with $\bar{s}_a$ such that $\bar{s}_a = \{s : b_a^{s-1} \leq \rho \leq b_a^s\}$. Let p_i be the length the shortest path from o to i. We can dynamically compute this length for every node by defining $p_o := 0$, and $p_i := \min\{p_j + w_a : a \in \delta^+(i)\}$. Let us consider a primal solution of the single-commodity PUMF with cost $g^* = p_d$.

Next consider the dual of the linear relaxation of SM (SDM). We show that there is a solution of SDM that also has cost g^*.

$$\max \quad \alpha_o - \alpha_d - \sum_{a \in A} \beta_a \tag{8a}$$

$$\alpha_i - \alpha_j + b_a^{\bar{s}_a-1}\gamma_a^{\bar{s}_a} - \rho\zeta_a^{\bar{s}_a} \leq c_a^{\bar{s}_a}\rho, \qquad a = (i,j) \in A \tag{8b}$$

$$\alpha_i - \alpha_j + \rho\gamma_a^s - \rho\zeta_a^s \leq c_a^s\rho, \quad a = (i,j) \in A, s \in S_a : s > \bar{s}_a \tag{8c}$$

$$- \beta_a - b_a^{s-1}\gamma_a^s + b_a^s\zeta_a^s \leq f_a^s, \qquad a \in A, s \in S_a \tag{8d}$$

$$\beta_a \geq 0, \qquad a \in A \tag{8e}$$

$$\gamma_a^s, \zeta_a^s \geq 0, \qquad a \in A, s \in S_a \tag{8f}$$

Dual variables α, β, γ, and ζ correspond, respectively to constraints (4b), (4c), to the left-most constraints in (4d), and to constraints (7). In turn, dual constraints (8b) are linked to the x_a^s-variables when $s = \bar{s}_a$, constraints (8c) to the x_a^s-variables when $s > \bar{s}_a$, and (8d) to the y-variables. Note that we do not include dual constraints associated with x_a^s-variables when $s < \bar{s}_a$, because these variables are fixed to 0 in (5). Let us fix the dual variables as follows:

- $\beta_a = 0$, $a \in A$;

- $\zeta_a^s = 0$, $a \in A, s \in S_a$;

- $\gamma_a^s = -\frac{f_a^s}{b_a^{s-1}}$, if $s \leq \bar{s}_a$; $-\frac{f_a^s}{\rho}$ if $s > \bar{s}_a$, $a \in A$.

Constraints (8d) are satisfied - both when, $s \leq \bar{s}_a$ (obvious); and when $s > \bar{s}_a$, as for those cases $\frac{b_a^{s-1}}{\rho} \geq 1$. The remainder of our dual formulation with fixed variables (SDM-f) is as follows:

$$\max \quad \alpha_o - \alpha_d \tag{9a}$$

$$\alpha_i - \alpha_j \leq f_a^s + c_a^s\rho, \qquad a = (i,j) \in A, s \in S_a : s \geq \bar{s}_a \tag{9b}$$

13

Note that $f_a^{\bar{s}_a} + c_a^{\bar{s}_a}\rho \leq f_a^s + c_a^s\rho$, $a \in A, s \in S_a : s > \bar{s}_a$. SDM-f is thus the dual of the LP formulation that describes the shortest path problem, as detailed above. It is well-known [27] that these dual and primal formulations have dual gap zero. Therefore, there is a solution in SDM with cost g^*. $\quad\square$

4. Computational experiments

In this section, we present the results of computational experiments that were conducted in order to compare the performance of CPLEX using BM1, BM2 and SM for solving instances of the PUMF problem. These experiments were done using two test sets, T_1 and T_2, whose description follows.

Class ID	$\lvert V \rvert$	$\lvert A \rvert$	$\lvert K \rvert$
1	40	936	70
2	40	1092	70
3	40	1092	100
4	60	2478	200
5	60	2832	150
6	60	2832	200
7	80	316	250
8	80	1896	200
9	80	1896	250
10	80	3160	200
11	80	1896	350

Table 1: Description of each class of instances.

In test set T_1, we created 55 instances and grouped them into 11 classes, according to number of nodes, arcs and commodities. Table 1 describes each class of instances. In each of these instances, the distribution of the arcs on the graph is random. Each arc was assigned a capacity of 50, 75 or 100. The traffic demand between the origin and destination node of each commodity was calculated using the following formula, used in [3]:

$$\rho^k = \alpha O_{o_k} D_{d_k} R_{(o_k,d_k)} e^{\frac{-L_2(o_k,d_k)}{2\Delta}} \tag{10}$$

For each node i, two random numbers, O_i and D_i were randomly generated in the interval $[0, 1]$. These values reflect, respectively, the attractiveness

of each node as a sender and as a receiver. Another value, $R_{(i,j)}$, was generated in the same interval, for each pair of nodes. The parameter α was set to 0.6, for all the experiments; we found that this value lead to instances where the majority of the arcs used in the optimal solution were not overloaded. The Euclidian distance (L_2) was substituted by the length of the shortest path between each pair of nodes, with respect to the number of links. Δ is the largest distance in the network. The final values were rounded to the nearest integer.

The routing cost on every arc are given by the function shown in Figure 2, and described in [3]. This function has 6 segments, separated by the following breakpoints: $\{0, \frac{1}{3}C_a, \frac{2}{3}C_a, \frac{9}{10}C_a, C_a, \frac{11}{10}C_a, \infty\}$, where C_a is the capacity of the respective arc. The slopes of each of these 6 segments are respectively 1, 3, 10, 70, 500 and 5000. Note that the intercepts of each segment can be easily calculated in the sense that $f_a^s = b_a^s(c_a^{s-1} - c_a^s) + f_a^{s-1}, a \in A, s \in S_a$.

The results of the computational experiments analyzed in Section 4.1 reveal that solving instances of T_1 with SM is very easy. As such, we propose another test set T_2, where we construct instances that are designedly difficult. In this sense, in test set T_2, we adapted instances of SNDlib [28] to better fit the PUMF problem. These instances are known to be challenging for different network optimization problems. Table 2 details the characteristics of these instances. For each of these SNDlib instances, we created four adapted instances for the PUMF problem, creating a total of 24 instances. For the first instance, we chose as the flow capacity on each arc, either the pre-installed capacity, or, in the cases where the latter was null, the capacity of the first module. So, for example, the capacity assigned to arc (1,6) in instance `atlanta` was 11000, whereas the capacity assigned to arc (1,2) in instance `newyork` was 1000. For the three other adapted instances, we increased or decreased the aforementioned capacities, maintaining the original proportions between the arcs, such that the average arc's utilization was slightly above the breakpoints in the cost function in Figure 2: 38%, 71% and 105%. As the results described in Section 4.1 will show, these instances will be quite more challenging to solve.

The three formulations were implemented using ILOG CLEX 12.6, on a Intel Core i7 CPU 960 @ 3.20GHz with 12GB of memory with 64 bits, and running Ubuntu 14.04.2 LTS (GNU/Linux 3.2.0 − 26−generic x86_64). Tests were done solving both the MIP and LP of each instance. In Subsections 4.1 and 4.2 we present the results of these computational experiments. The results are shown in performance profile graphs. These graphs depict

Instance ID	$|V|$	$|A|$	$|K|$
atlanta–D-B-M-N-C-A-N-N	15	44	210
france–D-B-L-N-C-A-N-N	25	90	300
newyork–D-B-E-N-C-A-N-N	16	98	240
pdh–D-B-E-N-C-A-N-N	11	68	24
sun–D-D-M-N-C-A-N-N	27	102	67
ta1–D-B-E-N-C-A-N-N	24	110	396

Table 2: Description of each class of instances in T_2.

the percentage of instances (spanned over the y-axis) that are solved under different values of each criteria, detailed in the x-axis.

4.1. Results for test set T_1

Let g_{IP}^* and g_{LP}^* be, respectively, the cost of the best integer solution found, and the cost of the solution of the LP relaxation of each model. We denote as LP gap the ratio $\frac{g_{IP}^* - g_{LP}^*}{g_{IP}^*}$. Figure 3 illustrates the performance profile for the LP gaps of the three modes, for instances of T_1. We can observe what had been stated in Theorem 4: the basic models produce the same lower bound. More importantly, we can see how strong the SM really is. The high percentage (93%) of instances that have 0 LP gap with the SM, clearly contrasts with the results for the basic models.

Next, we analyze the performance of the different models, when used by CPLEX to solve the MIPs of T_1 instances. In preliminary experiments we solved the LPs of the instances of T_1, using different optimizer methods offered by CPLEX. We verified that using the Barrier method for solving the LPs of the basic models, allows for a large speed-up, when compared with the optimizer provided by CPLEX's automatic selection. As for SM, CPLEX's default selection seems to provide the best results. We also tested the impact in the solving time of the MIPs, of defining variables y in BM1 and SM, as binary or continuous. The results of these experiments are depicted in Figure 4. We can observe that defining variables y as continuous is highly advantageous, for both BM1 and SM, allowing CPLEX to solve more instances in the time limit of 1 hour. Furthermore, defining y as binary, in the case of SM, led to memory issues. To have a better understanding of this behavior, further computational experiments involving test set T_1 were carried out on CPLEX using the configurations detailed in Table 3.
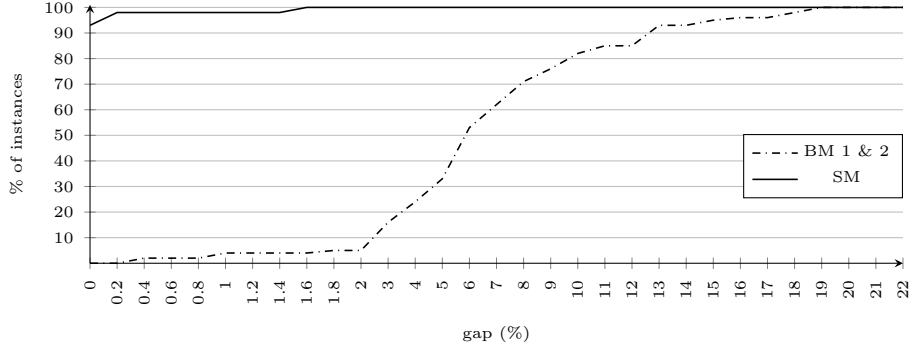
16
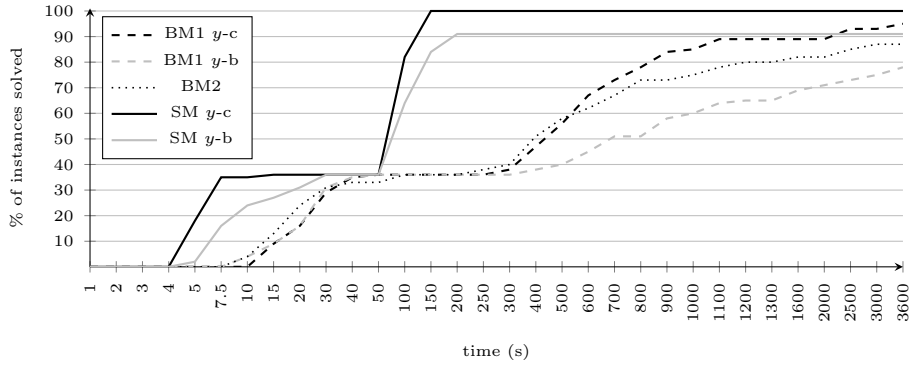
Figure 3: LP gaps for T_1 instances.

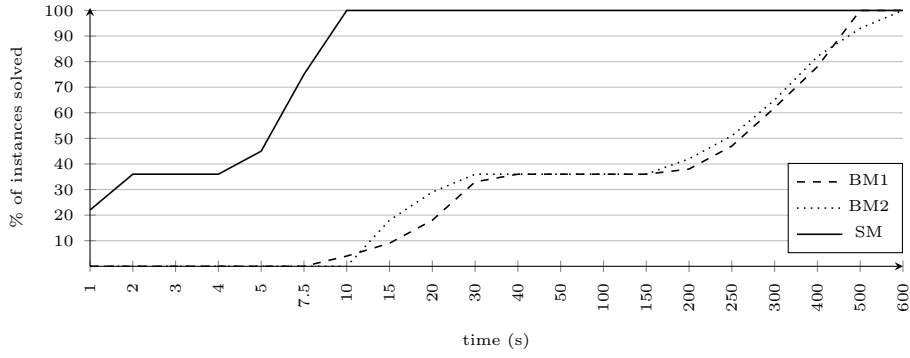Figure 4: Times for solving MIPs for T_1 instances.
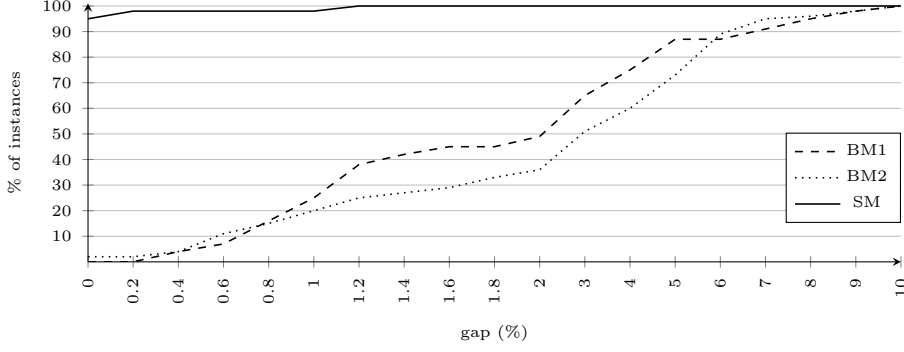

Figure 5: LP solving times for T_1 instances.

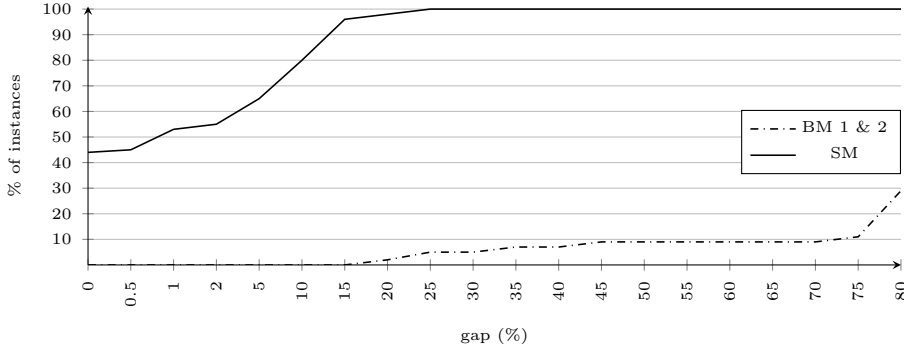17

Figure 6: Gaps at the root node for T_1 instances.

Figure 7: LP gaps for T_1 instances with 10 segments.

Model	LP optimizer	y-variables
SM	Default	Continuous
BM1	Barrier	Continuous
BM2	Barrier	

Table 3: Configuration implemented for instances of T_1

We are now in a better condition for comparing the MIP solving time, for the three different formulations. Figure 4 illustrates these results. They reveal that SM clearly out-performs BM1 and BM2 when solved with CPLEX. Even though, by using BM1, we are able to solve 95% of the instances in the time limit, by using SM we solved every instance, in only under 150 seconds. Finally, we can also observe that BM2 is slower than the other models, only solving 87% of the instances within the time limit.

18

It is easy to understand that the efficiency of SM is closely linked to its stronger lower bounds. The latter also account for the fact that all instances are solved in the root node of the Branch and Bound (B&B) tree. Another important argument for the efficiency of SM is the fast LP solving times, observable in Figure 5. Since in theory the size of SM's LP basis is comparable to the ones of the basic models (see end of previous section), we believe that these short LP solving times are explained by CPLEX's pre-processing procedure, that is particularly effective for SM.

For the basic models, it is interesting to note that despite having the same LP bounds (see Theorem 4 and Figure 3), comparable LP solving times (see Figure 5) and very similarly sized B&B trees, the solving time of the MIPs are distinct. This might be explained by better lower bounds at the end of the root node (g_0), as a result of CPLEX's automatically-generated cuts. The latter can be observed in the performance profile of the gap at the root node, $\frac{g_{IP}^* - g_0}{g_{IP}^*}$, depicted in Figure 6.

We were also interested in investigating how increasing the number of segments in the cost function, effected the performance of the proposed models. As such, we conducted experiments on adapted instances of T_1, where the piecewise linear cost functions described in the beginning of this section, were replaced by functions with 10 segments, and breakpoints $\{0, \frac{1}{20}C_a, \frac{1}{10}C_a, \frac{1}{6}C_a, \frac{1}{3}C_a, \frac{1}{2}C_a, \frac{2}{3}C_a, \frac{9}{10}C_a, C_a, \frac{11}{10}C_a, \infty\}$. The slopes of those segments are respectively 0.1, 0.3, 1, 2, 5, 10, 20, 70, 500 and 5000.

Figure 7 depicts the performance profile of the LP gaps for this case. Once again, the LP gaps of SM are far superior than the LP gaps of the basic models. Nevertheless, the LP gaps of SM seem to be considerably higher for this case, than for the original instances - only 44% instances have null gap, and the maximum value observed is of 23%. It should be noted, however, that these results are possibly "inflated", as for most instances we were unable to find the optimal solution; the LP gaps are calculated using the best found solution, which might not be particularly good. In fact, only 25 instances were solved by SM; for the others, we had out-of-memory issues, due to the large size of the corresponding MIP. These 25 instances were all solved under 250 seconds and their LP gaps were all under 1%. The basic models did not obtain memory problems. Regardless, BM1 was unable to solve any instance in the time limit, whereas BM2 only solved one.
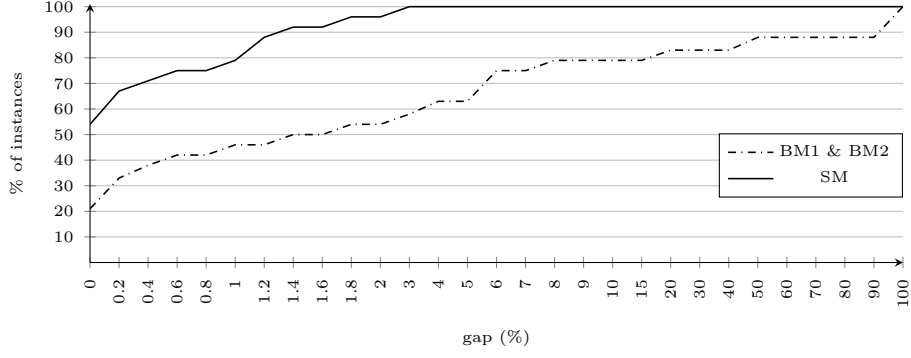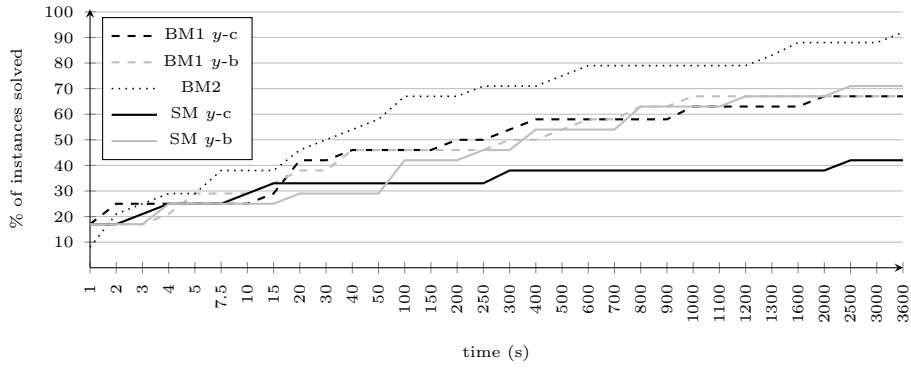
Figure 8: LP gaps for T_2 instances.

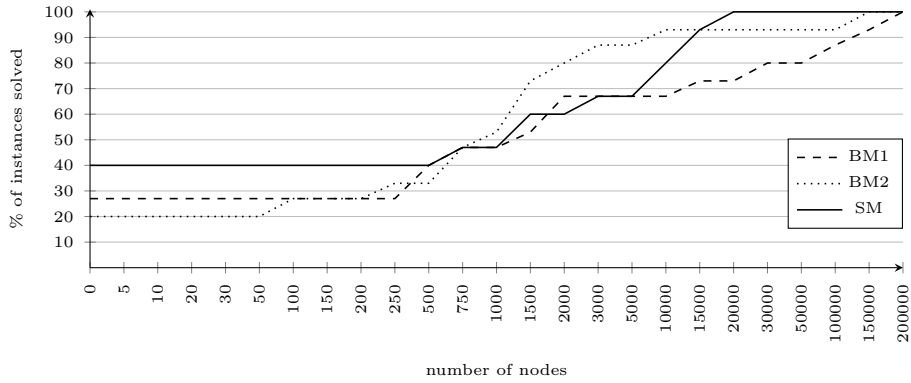Figure 9: Times for solving MIPs for T_2 instances.
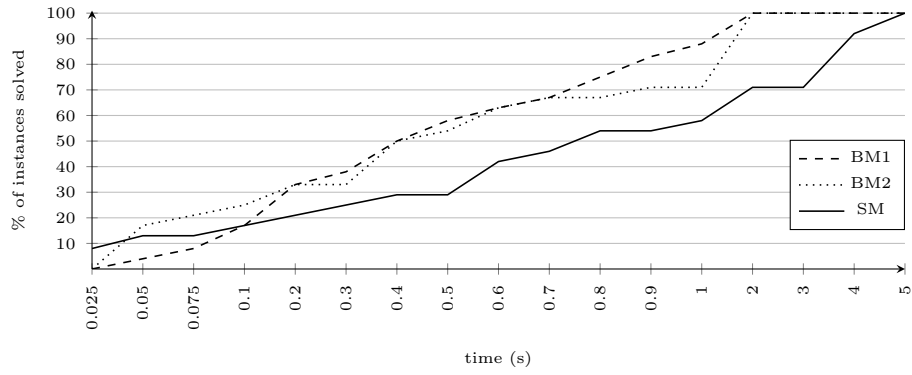

Figure 10: Size of the B&B tree for T_2 instances.
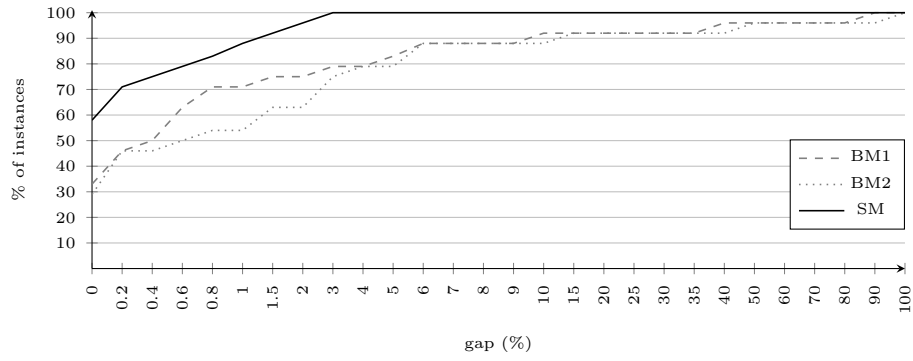
Figure 11: LP solving times for T_2 instances.

Figure 12: Gaps at the root node for T_2 instances.

21

4.2. Results for test set T_2

In this subsection, we analyze the results of the computational experiments done on instances of test set T_2. Figure 8 shows the performance profile for the LP gap (see definition in Subsection 4.1). Once again, SM is able to produce much better lower bounds than both basic models. Nevertheless, the gaps are slightly higher than they were for instances of T_1. Only 54% of the instances have LP gap 0, and the values go as high as 3%. These results imply that the instances of T_2 are harder than the ones of T_1; this is unsurprising, as they were explicitly constructed to be difficult (see beginning of section).

Next, we analyze the performance of CPLEX in solving T_2 instances' MIPs, under our three models. Once again, we analyzed which LP optimizer was better suited for each model. Contrary to what was the case for instances of T_1, for solving the LPs of T_2 instances, CPLEX's default LP optimizer is the faster for all models. We also analyzed the impact of defining y-variables as continuous or binary. The results can be observed in Figure 9. They are quite different than for instances of T_1: for instances of T_2, it seems to be more convenient to define variables y as binary. This is especially observable for SM. We believe that this disparity is a result of the different structures of the networks, in test sets T_1 and T_2, and how they impact the performance of CPLEX's pre-processing. For most instances of T_1, CPLEX is able to reduce the MIP further, when the y-variables are defined as continuous; whereas for instances of T_2, the tendency seems to be the opposite. On further computational experiments, we implement the models on CPLEX following the configurations in Table 4.

Model	LP optimizer	y-variables
SM	Default	Binary
BM1	Default	Binary
BM2	Default	

Table 4: Configuration implemented for instances of T_2

We can now compare the solving time of the MIPs of the three formulations. The performance profiles illustrated in Figure 9 show that for solving instances of T_2, BM2 is the most efficient model. With it, CPLEX can solve 92% of the instances in the 1 hour time limit. Moreover, it reveals that SM does not perform as well as it did for instances of T_1. When using SM,

22

CPLEX only solves 71% of the instances before reaching the time limit and, in general, the solving times are comparable to those obtained with BM1.

The below-par performance of SM with respect to the solving time of the MIPs, especially considering the strength revealed in Figure 8, can be explained by the following two factors: *i)* the LPs of SM are very slow to be solved (see Figure 11), *ii)* the difference between the lower bounds for the basic and strong models at the end of the root node is not as pronounced as in the case of the lower bounds provided by the LP relaxations (see Figure 12). The latter implies that CPLEX is able to produce good cuts for the basic models. The faster MIP solving times for BM2, when compared with BM1, are probably explained by the smaller B&B trees (see Figure 10), as BM1 out-performs BM2 in both the LPs solving time (see Figure 11) and gap at the end of the root node (see Figure 12).

5. Non-convex case

In the previous sections, we have considered the case where the piecewise linear function was convex. However, we can also define the unsplittable multicommodity flow problem with non-convex piecewise linear cost functions (NPUMF). This problem is a direct extension to the NCPMF problem (discussed in [20], [21] and [23]), with the added constraint of having single path routing for each commodity. Note that the complexity results described in Section 2 for the PUMF problem, can be extended to this case.

In order to characterize the non-convex piecewise linear cost functions, we resort to the notation described in Section 1 and illustrated in Figure 1. However, some new assumptions about these functions are made: the slopes c_a^s are non-negative as well, but no monotonicity is imposed and the intercepts f_a^s are not necessarily non-positive. The functions are not necessarily continuous, but we assume them to be lower semi-continuous (*i.e.*, $g_a(l_a) \leq \liminf_{l'_a \to l_a} g_a(l'_a)$, for any sequence l'_a that approaches l_a). Figure 13 illustrates two examples of non-convex piecewise linear functions: one concave (black) and another non-concave (gray).

In this section, we analyze the performance of our proposed models for instances of the NPUMF problem. First, observe that while BM2 cannot be used to model the NPUMF problem, both BM1 and DM are valid formulations for the problem. The SM is also a valid formulation for the NPUMF problem; however, as discussed in Section 3, for cases where $f_a^s \geq 0$, a greater strengthening can be obtained from valid inequalities (6), rather than from
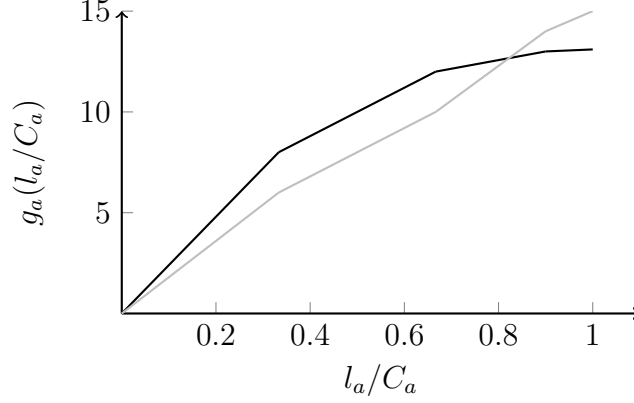
Figure 13: Two examples of non-convex piecewise linear cost functions

inequalities (7). As such, for clarity, we include below SM readapted for this case; we name it as the Strong Model for the Non-Convex Case (SM-n).

$$\min \quad \sum_{a \in A} \sum_{s \in S_a} \left(f_a^s y_a^s + c_a^s \sum_{k \in K} \rho^k x_a^{ks} \right) \tag{4a}$$

$$\text{s.t.} \quad \sum_{a \in A} \sum_{s \in S_a} x_a^{ks} - \sum_{a \in A} \sum_{s \in S_a} x_a^{ks} = \lambda_i^k, \qquad i \in V, k \in K, \tag{4b}$$

$$\sum_{s \in S_a} y_a^s \leq 1, \qquad a \in A, \tag{4c}$$

$$b_a^{s-1} y_a^s \leq \sum_{k \in K} \rho^k x_a^{ks} \leq b_a^s y_a^s, \qquad a \in A, s \in S_a, \tag{4d}$$

$$x_a^{ks} \in \{0, 1\}, \qquad a \in A, k \in K, s \in S_a, \tag{4e}$$

$$y_a^s \in \{0, 1\}, \qquad a \in A, s \in S_a, \tag{4f}$$

$$x_a^{ks} = 0, \qquad a \in A, k \in K, s \in S_a : b_a^s < \rho^k, \tag{5}$$

$$x_a^{ks} \leq y_a^s, \qquad k \in K, a \in A, s \in S_a. \tag{6}$$

Note that since for the NPUMF problem the cost functions are non-convex, the y-variables in BM1 and SM-n need to be considered as binary.

In our analysis, we also explore the impact of imposing unsplittable flows (NPUMF problem), versus allowing each commodity to be sent through multiple paths (NCPMF problem). In this sense, we need to adapt our models

to solve the NCPMF problem. To this purpose, we define the Relaxed Basic Model 1 (BM1-r), where the integrality of variables x (2f) is relaxed. We can also consider the same relaxation in constraint (4e) of DM . As before, this relaxed model can be strengthened by adding valid inequalities (6); we name this formulation as the Relaxed Strong Model (SM-r). Observe that if the flows can be split, we cannot apply valid inequalities (5), and as such, we exclude them from SM-r. Both BM1-r and SM-r have been proposed in [21] to model the NCPMF problem.

In Table 6, we summarize the composition of the strong models for all three problems, PUMF, NPUMF and NCPMF.

Problem	Model	Common constraints	Valid ineqs.	$y_a^s \in$ $[0,1]$	$\{0,1\}$	$x_a^{ks} \in$ $[0,1]$	$\{0,1\}$
PUMF	SM		(5),(7)	✓			✓
NPUMF	SM-n	(4a)-(4d)	(5),(6)		✓		✓
NCPMF	SM-r		(6)		✓	✓	

Table 5: Strong models for PUMF, NPUMF and NCPMF problems.

Network ID	$|V|$	$|A|$	$|K|$
1	20	100	50
2	20	100	50
3	25	100	100
4	25	150	100
5	25	150	100
6	40	300	100
7	40	300	400
8	40	400	400
9	50	400	400
10	50	600	400

Table 6: Description of each class of instances for the NPUMF.

In our computational experiments we tested these models, using instances that had been previously developed for the NCPMF problem [23]. For test set T_n, we selected 40 instances, that describe 10 different networks, whose
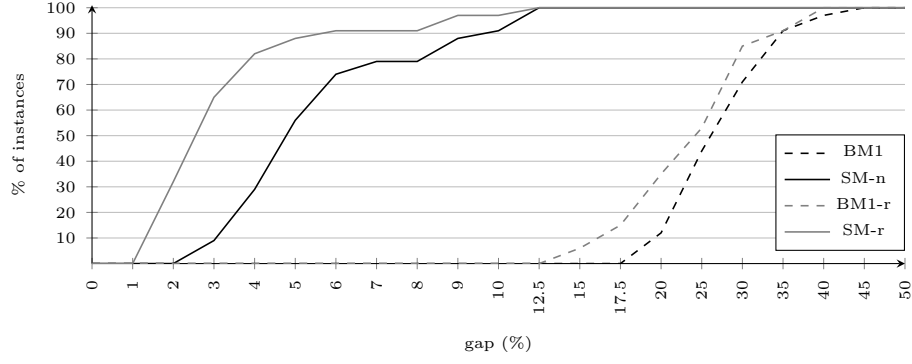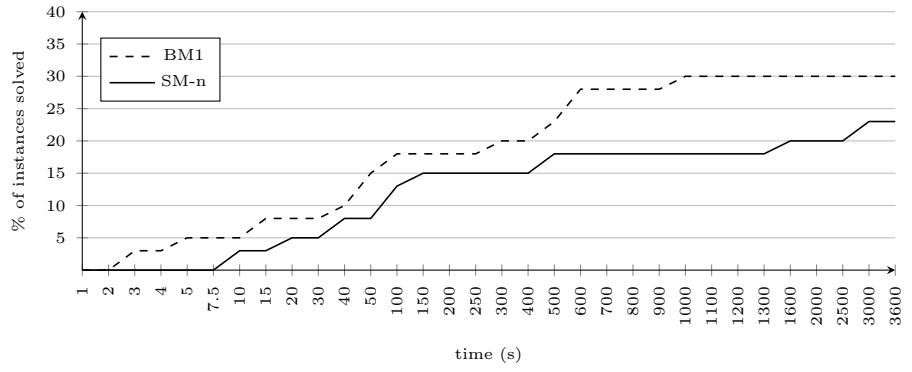
Figure 14: LP gaps for T_n instances.

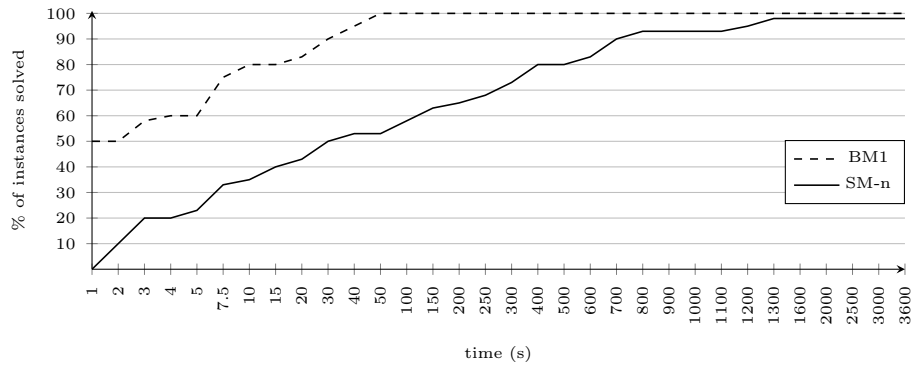Figure 15: Times for solving MIPs for T_n instances.

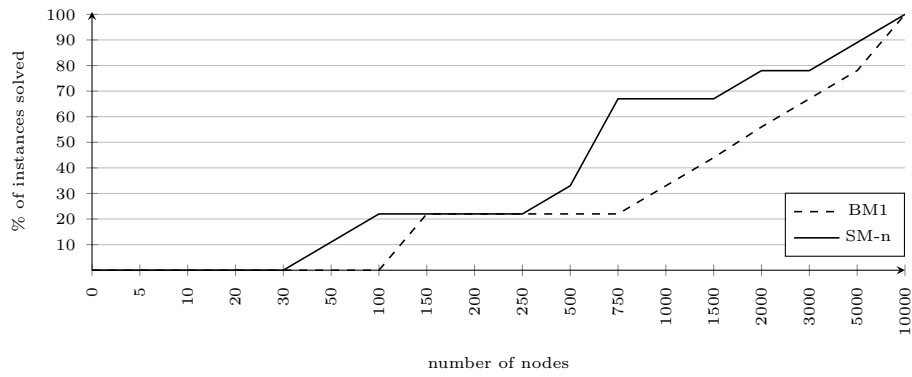Figure 16: LP solving times for T_n instances.

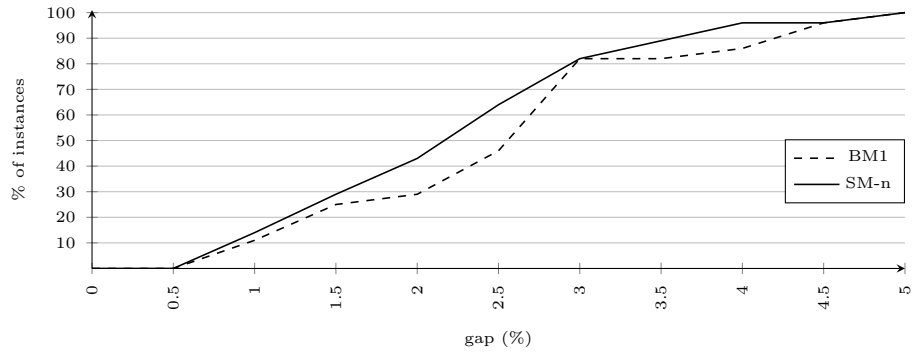Figure 17: Size of the B&B tree for T_n instances.

Figure 18: Gaps at the root node for T_n instances.

characteristics are detailed in Table 6. Despite having the same number of nodes, arcs and commodities, network #1 has tighter arc capacities $b_a^{|S_a|}$, than network #2. The same happens for networks #4 and #5. For each network, there are 4 different instances based on the characteristics of the non-convex piecewise linear cost functions. There are 2 instances with concave functions, and 2 instances with non-concave functions. In some cases, the non-concave functions are also non-continuous. For each type of cost functions, there is an instance with up to 4 segments, and another with up to 8 segments. Every segment has non-negative intercept f, even when the cost functions are non-concave. Finally, note that in the original instances, each commodity had one source and multiple targets. To adapt the instances to the NPUMF problem, we separated each commodity to many, such that each has a single origin and a single destination.

In Figure 14, we present the performance profile of the LP gap, for the NPUMF and the NCPMF problems. For this graph, we only considered the 34 instances, for which we were able to obtain an upper bound for both the NPUMF and NCPMF problems, and lower bounds for all four formulations (BM1, SM-n, BM1-r and SM-r) in the time limit of 1 hour. As before, it reveals that SM-n produces much better lower bounds than BM1. It also shows that, as expected, the gaps for the unsplittable flows case are higher than for the splittable flows one. However, it is noteworthy that the maximum observed gaps for SM-n and SM-r are similar. Moreover, it is easier for CPLEX to find good (or optimal) solutions for the NCPMF problem, than for the NPUMF problem. As such, it is likely that in many cases, the quality of upper bounds used to calculate the LP gaps of SM-n are not very good, and the values for the LP gap are, ultimately, greatly overestimated.

Next, we analyze the behaviour of CPLEX for instances of test set T_n, while using BM1 and SM-n. As it was done for previous experiments, we tested which CPLEX's LP optimizer method was faster to solve the instances of T_n. These preliminary experiments showed that while for BM1, CPLEX's default choice works best, for SM-n the Barrier method generally improves the LP solving time. In Figure 15, we depict the performance profile of the solving time of the MIPs of T_n. Similarly to what happens for instances of T_2 (see Subsection 4.2), BM1 is faster in solving instances of T_n than SM-n. However, in this case, the number of instances solved in the time limit is very low (30%). This is probably explained by the long solving times of the LPs of SM-n (see Figure 16). While all the LPs of BM1 are solved in under 50 seconds, not all the LPs of SM-n were solved in the time limit of

1 hour. Moreover, in Figure 18 we can observe that for BM1, CPLEX is able to automatically generate strong cuts, that highly improve the lower bounds at the root nodes, approaching them to the ones of SM-n. Figure 17 seems to indicate that the B&B trees for SM-n are smaller than for BM1. Notwithstanding, in our performance profiles of the B&B trees nodes (Figures 10 and 17), we only consider instances that were solved in the time limit by all models, in order to keep the results consistent. Seeing that only 9 out of the 40 instances of test set T_n are in this condition, the results depicted in Figure 17 might not be significant.

6. Summary and conclusions

In this work, we present a special case of the multicommodity flow problem, for which the flow of each commodity is unsplittable, and the routing costs on the arcs are given by a piecewise linear function. We focus on the case where the cost functions are convex - the PUMF problem. We show that the PUMF problem is $\mathcal{NP}$-hard for the general case, but polynomial when there is only a single commodity. We begin by proposing two basic MIP formulations for this problem, BM1 and BM2. By disaggregating the flow variables in BM1, we are able to develop valid inequalities that we include in a new MIP formulation: SM. We show that the LP feasible set of SM provides a complete description for the associated polyhedron, in the single-commodity case.

We present results of extensive computational experiments, done over instances of two different test sets: in T_1 we randomly generate our own instances, while in T_2 we adapt instances of the SNDlib. These experiments reveal that the LP relaxation of SM is able to provide very good lower bounds, far better than the ones provided by the LP relaxation of the basic models. In fact, for 93% of the instances in T_1 and for 58% of the instances of T_2, the LP gap of SM is null. In [29], it was shown that for large topologies, splittable multicommodity flow problems with piecewise linear costs (as well as for others types of objective) essentially become unsplittable flow problems, since at optimality each demand tends to use a single path, despite the existence of multiple paths. These results might explain the very good bounds that we were able to obtain for large networks, as it implies that the optimal solutions of the LP relation of the PUMF problem are integer.

We also consider the case where the cost functions are non-convex. This problem, denoted NPUMF, is an extension of the NCPMF problem, where

the flows can be split and sent through multiple paths [20], [21] and [23]. We adapt our SM to better adapt to characteristics of the NPUMF problem, and we name it SM-n. We test the performance of BM1 and SM-n in solving instances proposed in [23]. Our results reveal that, despite the added difficult of the NPUMF problem, in relation to the NCPMF problem, the LP gaps are only slightly worse. This might be explained by valid inequalities (5) that are not valid for the NCPMF problem.

We also tested the efficiency of the formulations in solving the MIPs of the instances, when implemented in CPLEX. Despite being the faster model to solve instances of T_1, SM did not perform as well for instances of T_2 and T_n as BM1 and BM2. This is explained by the large LPs, that can take a long time for CPLEX to solve.

Following these results, in the future, we want to explore the use of decomposition methods, that could eventually reduce the size of the LPs, allowing for a faster solving of the MIPs. Alternatively, we also want to try to project the strong valid inequalities of SM onto BM1, and hopefully gain from having a compact and strong formulation.

Acknowledgements

References

[1] D. Papadimitriou, B. Fortz, Time-dependent combined network design and routing optimization, in: IEEE International Conference on Communications, ICC 2014, Sydney, Australia, June 10-14, 2014, 2014, pp. 1124–1130. `doi:10.1109/ICC.2014.6883472`.
URL `http://dx.doi.org/10.1109/ICC.2014.6883472`

[2] D. Papadimitriou, B. Fortz, Methods for time-dependent combined network design and routing optimization, in: IEEE Global Communications Conference, GLOBECOM 2014, Austin, TX, USA, December 8-12, 2014, 2014, pp. 1303–1309. `doi:10.1109/GLOCOM.2014.7036988`.

[3] B. Fortz, M. Thorup, Internet traffic engineering by optimizing ospf weights, in: INFOCOM 2000. Nineteenth annual joint conference of the IEEE computer and communications societies. Proceedings. IEEE, Vol. 2, IEEE, 2000, pp. 519–528.

[4] B. Fortz, M. Thorup, Increasing internet capacity using local search, Computational Optimization and Applications 29 (1) (2004) 13–48.

[5] M. Gerla, L. Kleinrock, Communication nets: stochastic message flow and delay 25 (1) (1977) 48–60.

[6] C. Lemaréchal, A. Ouorou, G. Petrou, A bundle-type algorithm for routing in telecommunication data networks, Computational Optimization and Applications 44 (3) (2007) 385–409. doi:10.1007/s10589-007-9160-7.
URL http://dx.doi.org/10.1007/s10589-007-9160-7

[7] M. Pióro, D. Medhi, Routing, Flow, and Capacity Design in Communication and Computer Networks, Morgan Kaufman, 2004.

[8] S. Balon, F. Skivée, G. Leduc, How well do traffic engineering objective functions meet TE requirements?, in: Proceedings of IFIP Networking 2006, Coimbra, Vol. 3976, Springer LNCS, 2006.

[9] E. Gourdin, O. Klopfenstein, Comparison of different qos-oriented objectives for multicommodity flow routing optimization, in: Proceedings of the International Conference on Telecommunications (ICT 2006), 2006.

[10] L. A. Cox, L. Davis, Y. Qiu, Dynamic anticipatory routing in circuit-switched telecommunications networks, Handbook of Genetic Algorithms 11 (1991) 229–340.

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

[12] F. Alvelos, J. V. de Carvalho, Comparing branch-and-price algorithms for the unsplittable multicommodity flow problem, in: International Network Optimization Conference, 2003, pp. 7–12.

[13] M. Belaidouni, W. Ben-Ameur, On the minimum cost multiple-source unsplittable flow problem, RAIRO-Operations Research 41 (3) (2007) 253–273.

[14] M. Parker, J. Ryan, A column generation algorithm for bandwidth packing, Telecommunication Systems 2 (1) (1993) 185–195.

[15] K. Park, S. Kang, S. Park, An integer programming approach to the bandwidth packing problem, Management science 42 (9) (1996) 1277–1291.

[16] J. Geffard, A solving method for singly routing traffic demand in telecommunication networks, in: Annales des télécommunications, Vol. 56, Springer, 2001, pp. 140–149.

[17] A. Atamtürk, D. Rajan, On splittable and unsplittable flow capacitated network design arc–set polyhedra, Mathematical Programming 92 (2) (2002) 315–333.

[18] A. Benhamiche, A. R. Mahjoub, N. Perrot, E. Uchoa, Unsplittable non-additive capacitated network design using set functions polyhedra, Computers & Operations Research 66 (2016) 105–115.

[19] F. S. Salman, R. Ravi, J. N. Hooker, Solving the capacitated local access network design problem, INFORMS Journal on Computing 20 (2) (2008) 243–254.

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

[21] K. L. Croxton, B. Gendron, T. L. Magnanti, Variable disaggregation in network flow problems with piecewise linear costs, Operations research 55 (1) (2007) 146–157.

[22] A. Balakrishnan, S. C. Graves, A composite algorithm for a concave-cost network flow problem, Networks 19 (2) (1989) 175–202.

[23] B. Gendron, L. Gouveia, Reformulations by Discretization for Piecewise Linear Integer Multicommodity Network Flow Problems, 2014.

[24] J. M. Kleinberg, Single-source unsplittable flow, in: Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on, IEEE, 1996, pp. 68–77.

[25] R. M. Karp, Reducibility among combinatorial problems, Springer, 1972.

[26] B. Fortz, M. Joyce-Moniz, et al., On the convex piecewise linear unsplittable multicommodity flow problem, in: 2016 12th International Conference on the Design of Reliable Communication Networks (DRCN), IEEE, 2016, pp. 9–13.

[27] L. A. Wolsey, Integer programming, Vol. 42, Wiley New York, 1998.

[28] S. Orlowski, M. Pióro, A. Tomaszewski, R. Wessäly, SNDlib 1.0– Survivable Network Design Library, in: Proceedings of the 3rd International Network Optimization Conference (INOC 2007), Spa, Belgium, 2007, http://sndlib.zib.de, extended version accepted in Networks, 2009.
URL `http://www.zib.de/orlowski/Paper/`
`OrlowskiPioroTomaszewskiWessaely2007-SNDlib-INOC.pdf.gz`

[29] X. Liu, S. Mohanraj, M. Pióro, D. Medhi, Multipath routing from a traffic engineering perspective: How beneficial is it?, in: Network Protocols (ICNP), 2014 IEEE 22nd International Conference on, IEEE, 2014, pp. 143–154.