RAIRO-Oper. Res. 53 (2019) 1279–1295
https://doi.org/10.1051/ro/2019017

RAIRO Operations Research
www.rairo-ro.org

AN EXACT APPROACH FOR THE MULTICOMMODITY NETWORK OPTIMIZATION PROBLEM WITH A STEP COST FUNCTION

Imen Mejri[1], Mohamed Haouari[2], Safa Bhar Layeb[1,*] and Farah Zeghal Mansour[1]

Abstract. We investigate the Multicommodity Network Optimization Problem with a Step Cost Function (MNOP-SCF) where the available facilities to be installed on the edges have discrete step-increasing cost and capacity functions. This strategic long-term planning problem requires installing at most one facility capacity on each edge so that all the demands are routed and the total installation cost is minimized. We describe a path-based formulation that we solve exactly using an enhanced constraint generation based procedure combined with columns and new cuts generation algorithms. The main contribution of this work is the development of a new exact separation model that identifies the most violated bipartition inequalities coupled with a knapsack-based problem that derives additional cuts. To assess the performance of the proposed approach, we conducted computational experiments on a large set of randomly generated instances. The results show that it delivers optimal solutions for large instances with up to 100 nodes, 600 edges, and 4950 commodities while in the literature, the best developed approaches are limited to instances with 50 nodes, 100 edges, and 1225 commodities.

Mathematics Subject Classification. 05C21, 90C11, 90C27.

Received February 20, 2018. Accepted January 22, 2019.

1. Introduction

During the last two decades, and prompted by the rapid development of the telecommunications industry, multicommodity network design problems have been extensively investigated and are still catching the interest of both practitioners and researchers. Indeed, these problems have a wide range of applications mainly in telecommunications, transportation, and logistic systems [4, 22], and have a crucial impact on the business profitability of network operators. Moreover, most multicommodity network design problems are $\mathcal{NP}$-hard [16] and their exact solution poses redoubtable challenges.

In this paper, we investigate a very general network optimization model that is referred to as the Multicommodity Network Optimization Problem with a Step Cost Function (MNOP-SCF). This strategic long-term planning problem is defined as follows. We are given a connected undirected graph $G = (V, E)$ where V is a set of n nodes, and E is a set of m edges. In telecommunication settings, each node may represent a customer that

Keywords. Networks, constraint generation algorithm, cutset inequalities, column generation.

[1] UR-OASIS, National Engineering School of Tunis University of Tunis El Manar, Tunis, Tunisia.

[2] Department of Mechanical and Industrial Engineering, College of Engineering, Qatar University, Doha, Qatar.

*Corresponding author: `Safa.Layeb@enit.utm.tn`

requires to exchange data flows with other customers, and each edge corresponds to a potential physical link on which a transmission facility may be installed for routing the different traffic flows. Hence, we assume that for each pair of nodes $\{i, j\} \subset V^2$ are defined two demand flows d_{ij} and d_{ji} that should be routed from i to j and *vice-versa*, respectively. Each commodity may be routed along several different paths (*i.e.* flows are splittable). The core problem requires designing a network that enables a simultaneous routing of all the point-to-point demands while minimizing the network design cost. In this regard, a peculiar feature of the investigated model is that the structure of the facility cost function is discrete, nonconvex, and step-increasing. More precisely, we assume that L_e different facilities may potentially be installed on each edge $e \in E$. In telecommunications, facilities may correspond to different transmission technologies that transport data at different speed rates and incur different costs. For each edge e and each facility l, we are given a bidirectional capacity u_l^e (expressed in a given unit *e.g.* Mb/s) and a fixed installation cost f_l^e that are discrete step-increasing functions (*i.e.*, $u_1^e < u_2^e < \ldots < u_{L_e}^e$, and $f_1^e < f_2^e < \ldots < f_{L_e}^e$). This very general cost structure offers two significant advantages. First, it allows to model economies of scale that are obtained through installing large capacities when the cost per unit of a facility is a decreasing function of its capacity. Second, it enables advantageous flexibility by explicitly considering a discrete set of multiple link capacities. The MNOP-SCF requires finding which facility capacities should be installed on the edges so that all point-to-point flows are simultaneously routed and the total installation cost is minimized. At this stage, it is worth emphasizing the fact that objective function can readily accommodate flow-dependent, discrete, nonconvex, step-increasing node costs (see [23]). Also, it is noteworthy that although our objective function includes fixed costs only with no variable traffic costs, which models the common case where the total fixed link investment cost overwhelmingly dominates the variable transmission costs, one would readily extend the model and solution approach that are proposed in this paper to accommodate variable costs as well.

To the best of our knowledge, the first reference to the MNOP-SCF goes back to the later 1980s when Minoux [22] introduced a simpler variant that is referred to as the *Optimum Rented Lines Network Problem*. Later, several exact and approximate algorithms were investigated by many authors. In particular, Stoer and Dahl [27] investigated a multicommodity network design problem with a general step-increasing cost function. They derived valid inequalities and facet-defining inequalities and integrated them into a branch-and-cut approach to obtain lower bounds. A rounding procedure is then used to generate approximate integer solutions. They reported the solution of a single network structure with 27 nodes and 51 links with a very sparse demand matrix. Gabrel *et al.* [10] presented a first exact approach for the MNOP-SCF. They proposed an integer-programming formulation with exponentially many constraints (the so-called *metric inequalities* that shall be described in Section 2), and solved it using a pure cutting-plane approach. This algorithm is also thoroughly analyzed in the comprehensive study of the MNOP-SCF that is presented in Minoux [23]. In this paper, the author acknowledges the flexibility and versatility of the MNOP-SCF by highlighting the fact that this model encompasses as special cases many well-studied network design problems including the so-called single- and two-facility capacitated network loading problem, where capacity expansion can be achieved by installing an integer number of a single type or two types of facilities, respectively. Also, he presented different valid integer programming formulations for the basic case as well a more general variant with survivability constraints where both link and node failures have to be taken into consideration. Following the general solution framework of Gabrel *et al.* [10], Mrad and Haouari [24] implemented a similar formulation and solution approach, but including several additional enhanced features, yielding a more effective exact algorithm. They show that the conjunctive use of newly derived valid inequalities, as well as the exact separation of metric inequalities makes it feasible to optimally solve instances with up to 50 nodes and 100 edges.

In addition, efforts have been made for designing effective heuristic algorithms. A local search heuristic is developed by Agarwal [2]. It allows solving instances with up to 20 nodes and 3 facilities within about 5% of lower bound on average. Larger MNOP-SCF instances with up to 50 nodes and 90 links are approximately solved by Gabrel *et al.* [11] using several greedy heuristics based on link-rerouting and flow-rerouting heuristics as well as a heuristic implementation of their previous exact algorithm presented in Gabrel *et al.* [10]. Aloise and Ribeiro [1] propose several effective multi-start heuristics that are based on shortest path and maximum flow algorithms combined with adaptive memory and vocabulary building intensification mechanism. Their approach improve the best solutions of the instances tested by Gabrel *et al.* [11] by as much as 7.4%.

Recently, a robust optimization of the MNOP-SCF under uncertain demand was investigated by Lee *et al.* [19]. They use a Benders decomposition procedure combined with a cut generation scheme. They conduct experiments on real-life telecommunication problems, with up to 27 nodes and 51 edges, and empirically demonstrate that robust solutions with very small penalties in the objective values can be obtained.

A closely related network design problem is the so-called *Network Loading Problem* (NLP). By contrast to the MNOP-SCF, where at most one facility can be installed on each edge, the NLP requires selecting an integer number of facilities so that a preset of traffic demands can be routed simultaneously with a minimum total linear cost [5]. Two problem variants are often considered: the nonbifurcated version, where each traffic demand should be routed along a single path, and the splittable one. The NLP has been subject to a growing interest in the literature and several exact and heuristic approaches have been used so far. Local search and tabu search approaches were proposed by Gendron *et al.* [13] and Gendron *et al.* [14], respectively. In Avella *et al.* [3], a complete description of the convex hull of integer feasible solutions using the so-called tight metric inequalities, is provided. The exact separation of these inequalities is investigated in [21]. Ljubić *et al.* [20] investigate a single-source variant. They introduce several mixed-integer models that they solve to optimality using a Benders-based approach combined with a specific disaggregation technique.

In this paper, we propose a new enhanced cut generation-based approach that outperforms the best algorithms presented in the literature and allows solving larger instances. A distinctive feature of our approach is that it embeds a novel exact separation model that generates the most violated cutest inequalities, as well as a knapsack-based problem that enables the generation of additional cuts. The obtained inequalities allow strengthening the relaxed master problem and reducing the number of violated metric inequalities.

The remainder of this paper is organized as follows. Section 2 investigates several formulations for the MNOP-SCF. Section 3 introduces the new bipartition inequality generators and Section 4 summarizes the overall proposed procedure. Section 5 reports the results of extensive computational experiments. Finally, Section 6 draws conclusions and provides avenues for future research.

2. Valid formulations for the MNOP-SCF

In this section, we successively describe three valid textbook formulations: two mixed-integer programming models and a binary model.

2.1. An arc-flow formulation

A natural mixed-integer programming formulation of the MNOP-SCF can be derived using a bidirected graph $B = (V, A)$ that is obtained from G by replacing each edge $e = \{i, j\} \in E$ by a pair of arcs having opposite directions. Furthermore, we define a continuous flow variables x_{ij}^k for each arc $(i, j) \in A$ and each commodity k ($k = 1, \dots, K$). This yields the following arc-flow model.

$$(\mathbf{AF})\text{: Minimize } \sum_{e \in E} \sum_{l=1}^{L_e} f_l^e y_l^e \tag{2.1}$$

subject to:

$$\sum_{l=1}^{L_e} y_l^e \le 1, \qquad \forall e \in E, \tag{2.2}$$

$$\sum_{j:(i,j)\in A} x_{ij}^k - \sum_{j:(j,i)\in A} x_{ji}^k = \begin{cases} d_k & \text{if } i = s_k \\ 0 & \text{if } i \in V \backslash \{s_k, t_k\} \\ -d_k & \text{if } i = t_k \end{cases}, \qquad \forall k = 1, \dots, K, \tag{2.3}$$

$$\sum_{k=1}^{K} x_{ij}^k + \sum_{k=1}^{K} x_{ji}^k \leq \sum_{l=1}^{L_e} u_l^e y_l^e, \qquad \forall e = \{i,j\} \in E, \tag{2.4}$$

$$x_{ij}^k \geq 0, \qquad \forall (i,j) \in A, \forall k = 1,\dots,K, \tag{2.5}$$

$$y_l^e \in \{0,1\}, \qquad \forall e \in E,\ l = 1,\dots,L_e. \tag{2.6}$$

The objective function (2.1) requires minimizing the total fixed installation costs. Constraints (2.2) enforce that at most one facility is installed on each edge. Constraints (2.3) are the standard flow conservation equality. The bundle constraints (2.4) require that the bidirectional flow that circulates on each edge does not exceed the installed capacity. Constraints (2.5) define the non-negativity of the flow variables and constraints (2.6) impose the integrality of the design variables.

Clearly, Model (**AF**) exhibits a bloc-diagonal structure and includes $O(n^3)$ constraints. Furthermore, if the underlying graph is complete then it includes $O(n^4)$ flow variables and $O(\max_{e\in E}\{L_e\}n^2)$ binary variables.

It is noteworthy that Gabrel *et al.* [10] and Minoux [23] used an alternative arc-flow formulation using different design binary variables. Indeed, they defined $\varkappa_t^e$ as a binary variable that takes value 1 if the facility loaded on edge e is $l \geq t$, and 0 otherwise. Actually, using the following identities

$$y_l^e = \varkappa_l^e - \varkappa_{l+1}^e, \quad \forall l = 1,\dots,L-1,\ \forall e \in E, \tag{2.7}$$

$$y_L^e = \varkappa_L^e,\ \forall e \in E, \tag{2.8}$$

one would readily check that this latter formulation is equivalent to Model (**AF**).

Remark 2.1. Model (**AF**)can easily be extended to the case where multiple facilities of the same type might be installed on the same edge. Indeed, in this case constraint (2.2) should be dropped, and constraint (2.6) should be replaced by

$$y_l^e \in \mathbb{N}, \qquad \forall e \in E,\ l = 1,\dots,L_e. \tag{6'}$$

2.2. A path-based formulation

An alternative formulation of the MNOP-SCF is the so-called path-based formulation. To describe this formulation, we begin first by introducing some additional notation. We denote by P_k the set of all feasible paths between nodes s_k and t_k $(k = 1,\dots,K)$. Let a_{erk} be a binary constant that equals 1 if path $r \in P_k$ includes edge e, and 0 otherwise. We define a continuous nonnegative variable z_r^k that represents the flow of commodity k to be routed along path $r \in P_k$. Given these definitions, a path-based formulation (**PF**) for the MNOP-SCF can be stated as follows.

$$(\mathbf{PF})\text{: Minimize } \sum_{e\in E}\sum_{l=1}^{L_e} f_l^e y_l^e \tag{2.9}$$

subject to: (2.2),(2.6),

$$\sum_{r=1}^{|P_k|} z_r^k = d_k, \qquad \forall k = 1,\dots,K, \tag{2.10}$$

$$\sum_{k=1}^{K}\sum_{r=1}^{|P_k|} a_{erk} z_r^k \leq \sum_{l=1}^{L_e} u_l^e y_l^e, \qquad \forall e \in E, \tag{2.11}$$

$$z_r^k \geq 0, \qquad \forall k = 1,\dots,K,\ r \in P_k. \tag{2.12}$$

Constraint (2.10) requires routing all the flow demands. Constraint (2.11) represents capacity constraints for each edge: the bidirectional flow that circulates on each edge should not exceed the installed capacity. Constraint (2.12) defines the nonnegativity of variables z.

Obviously, Model (**PF**) includes the same number of binary variables as Model (**AF**), an exponential number of (continuous) variables, but much fewer constraints ($O(n^2)$ vs $O(n^3)$).

Fact 1: Models (**AF**) and (**PF**) yield the same linear programming relaxations.

It is noteworthy that, from a computational perspective, large arc flow formulations, that arise embedded within real-world MNOP-SCF instances, are unlikely to be solved directly using commercial LP solvers. By contrast, the path-based formulation exhibits a structure that makes it amenable to be solved by column generation.

2.3. A pure 0-1 programming model

Assume that each edge $e \in E$ is assigned a nonnegative capacity $\bar{u}_e \geq 0$. A necessary and sufficient condition that there exists a multiflow x satisfying (2.3)–(2.5) (or equivalently, a vector z satisfying (2.10)–(2.12)), is given by Farkas Lemma for linear programming duality as follows

Theorem 2.2. *A capacity vector $\bar{u}$ is feasible for a multiflow if and only if*

$$\sum_{e\in E}\lambda_e \bar{u}_e \geq \sum_{k=1}^{K} d_k \mu_k^*(\lambda), \quad \forall \lambda \in \mathcal{R}_+^m,$$

where $\mu_k^*(\lambda)$ $(k = 1, \ldots, K)$ is the value of the shortest path between s_k and t_k in G with respect to the distance matrix $(\lambda_e)_{e\in E}$.

Proof. See Onaga and Kakusho [17] and Croxton *et al.* [8]. □

Hence, a valid formulation for the MNOP-SCF using only binary variables is the following

$$\text{MI: Minimize } \sum_{e\in E}\sum_{l=1}^{L_e} f_l^e y_l^e \tag{2.13}$$

subject to: (2.2),(2.6),

$$\sum_{e\in E}\lambda_e \sum_{l=1}^{L_e} u_l^e y_l^e \geq \sum_{k=1}^{K} d_k \mu_k^*(\lambda), \quad \forall \lambda \in \mathcal{R}_+^m. \tag{2.14}$$

In the literature, constraint (2.14) is referred to as *metric inequalities*. A particularly interesting subset of metric inequalities includes the so-called *bipartition inequalities* (or cutset inequalities) that are defined as follows. Assume that vector λ is the incidence vector of a cutset $\delta(W)$ of G that connects a node subset $W \subset V$ to its complementary subset $\bar{W} = V \setminus W$. In this case, the LHS of (2.14) is equal to the cumulative capacity of the cutset $\delta(W)$. That is, we have

$$\sum_{e\in E}\lambda_e \sum_{l=1}^{L_e} u_l^e y_l^e = \sum_{e\in\delta(W)}\sum_{l=1}^{L_e} u_l^e y_l^e.$$

On the other hand, $\mu_k^*(\lambda) = 1$ if $|\{s_k, t_k\} \cap W| = 1$, and $\mu_k^*(\lambda) = 0$, otherwise. Hence, the RHS of (2.14) is equal to the cumulative demand that traverses cutset $\delta(W)$. Defining $d(W) = \sum_{k:|\{s_k,t_k\}\cap W|=1} d_k$ for all $W \subset V$, we get

$$\sum_{k=1}^{K} d_k \mu_k^*(\lambda) = \sum_{k:|\{s_k,t_k\}\cap W|=1} d_k.$$

Therefore, the bipartition inequalities can be written as follows:

$$\sum_{e\in\delta(W)}\sum_{l=1}^{L_e} u_l^e y_l^e \geq d(W), \qquad \forall W\subset V. \tag{2.15}$$

The significance of bipartition inequalities stems from the following empirical observation: this subclass of inequalities defines a fairly good representation of the convex hull of the set of feasible solutions [10].

It is worth emphasizing that all exact approaches that were proposed so far for the MNOP-SCF are based on solving Model (**MI**) using a constraint-generation procedure. Within this framework, at each stage, a violated metric inequality is generated by solving the following separation problem:

Given a binary vector $\bar{y}$ that is feasible to (2.2) find $\lambda\in\mathcal{R}_+^m$, such that

$$\sum_{e\in E}\lambda_e\bar{z}_e < \sum_{k=1}^{K} d_k\mu_k^*(\lambda) \tag{2.16}$$

or prove that no such vector exists.

Mrad and Haouari [24] show that the exact separation of a metric inequality can be achieved through finding a feasible multiflow using the arc-flow formulation. Since this separation is cumbersome and thereby viable for relatively small-sized (sparse) graphs only, Gabrel *et al.* [10] and Mrad and Haouari [24] resorted to a much simpler, though *inexact*, separation approach that is based on subgradient optimization.

On the other hand, since the exact separation of bipartition inequalities is $\mathcal{NP}$-hard [10], this problem has been approximately solved using tailored heuristics. In this regard, Gabrel *et al.* [10] restated the separation of bipartition inequalities as a *maximum ratio cut* problem that is defined as follows

$$\underset{W\subset V}{\text{Maximize}}\,\rho(W)\equiv\frac{d(W)}{\sum\limits_{e\in\delta(W)} z_e}\cdot$$

and derived approximate solutions using a variable-depth local search heuristic.

A different heuristic strategy for generating bipartition inequalities was implemented by Mrad and Haouari [24]. This strategy requires solving K maximum flow problems.

2.3.0.1. Reformulation of Model (**MI**) Model (**MI**) can be slightly restated as follows. First, we observe that if the metric inequality

$$\sum_{e\in E}\lambda_e\sum_{l=1}^{L_e}u_l^e y_l^e \geq \sum_{k=1}^{K} d_k\mu_k^*(\lambda),$$

holds for some vector $\lambda\in\mathcal{R}_+^m$, then it is also valid for vector $\lambda' = \frac{1}{\max\limits_{e\in E}\{\lambda_e\}}\lambda$ and *vice-versa*. Consequently, we shall restrict the set of multipliers to the hypercube $[0,1]^m$.

Second, we set $\eta_k = \mu_k^*(\lambda)$ for each $\lambda\in[0,1]^m$. Hence, $\eta_k = \min\limits_{r=1,\ldots,|P_k|}\lambda^t a_r^k$ (recall that a_r^k denotes the incidence vector of a path from s_k to $t_k, k=1,\ldots,K, r=1,\ldots,|P_k|$). Thus, we have

$$\eta_k \leq \sum_{e\in E} a_{erk}\lambda_e, \quad \forall r=1,\ldots,|P_k|\,.$$

Therefore, the metric inequality (2.14) shall be restated as follows

$$\sum_{e\in E}\lambda_e\sum_{l=1}^{L_e}u_l^e y_l^e \geq \sum_{k=1}^{K} d_k\eta_k, \tag{2.17}$$

where (λ, η_k) are points of the polyhedron Λ_k defined by

$$\eta_k - \sum_{e \in E} a_{erk} \lambda_e \leq 0, \quad \forall r = 1, \dots, |P_k|,$$
$$0 \leq \lambda_e \leq 1, \ \forall e \in E.$$

Define Π_k as the set of extreme points of Λ_k. We observe that if (2.17) holds for any extreme point of Λ_k then it holds for any point of Λ_k. Therefore, constraint (2.14) can be replaced by the following inequality

$$\sum_{e \in E} \lambda_e \sum_{l=1}^{L_e} u_l^e y_l^e \geq \sum_{k=1}^{K} d_k \eta_k, \quad \forall (\lambda, \eta_k) \in \Pi_k, \ k = 1, \dots, K. \tag{2.18}$$

Fact 2: Model (**MI**) is equivalent to a Benders reformulation of Model (**PF**).

Proof. See Costa [9] and Mrad and Haouari [24]. □

Consequently, Benders cuts are equivalent to metric inequalities.

2.4. Generation of violated metric cuts

To solve Model (**MI**), we use a constraint generation approach, where violated metric cuts are added on the fly until no violated cut is found. More precisely, starting from an initial relaxed master program (RMP) (this initialization process shall be described in Sect. 4.1), we solve it using a general-purpose solver and derive a solution $\tilde{y}$. Next, we check whether the installed capacities allow to simultaneously route all the required commodity flow demands. Actually, a necessary and sufficient condition that $\tilde{y}$ is feasible is that the following problem

$$\text{SPF}(\tilde{y}): h(\tilde{y}) = \text{Minimize} \sum_{e \in E} \varepsilon_e \tag{2.19}$$

subject to: (2.12),

$$\sum_{r=1}^{|P_k|} z_r^k = d_k, \qquad \forall k = 1, \dots, K, \tag{2.20}$$

$$\sum_{k=1}^{K} \sum_{r=1}^{|P_k|} a_{erk} z_r^k - \varepsilon_e \leq \sum_{l=1}^{L_e} u_l^e \tilde{y}_l^e, \qquad \forall e \in E, \tag{2.21}$$

$$\varepsilon_e \geq 0, \qquad \forall e \in E, \tag{2.22}$$

exhibits a zero objective, where ε_e, $e \in E$, is a continuous nonnegative variable that expresses the unrouted demand on edge e, $e \in E$. Thus, we solve SPF$(\tilde{y})$ which allows to generate a violated metric cut, if any, or to prove the optimality of solution $\tilde{y}$. This process is repeated until an optimal solution is found.

Since SPF$(\tilde{y})$ includes an exponentially large number of variables, a column generation algorithm is invoked for its exact solution. For each commodity k, $k = 1, \dots, K$, the reduced cost δ_{rk} of a path r, $r \in P_k$ can be written as follows:

$$\delta_{rk} = \sum_{e \in E} a_{erk} \lambda_e - \eta_k, \qquad \forall k = 1, \dots, K, \forall r \in P_k. \tag{2.23}$$

The reduced cost of a path in (2.23) can be derived by associating to each arc $(i,j) \in A$ a cost $c_{ij} = \lambda_{ij} = \lambda_{ji} = \lambda_e$, $e = \{i,j\} \in E$. Hence, the pricing subproblem reduces to a sequence of shortest path problems, one for each commodity $k, k = 1, \ldots, K$, over digraph B. These subproblems can be efficiently solved using Dijkstra algorithm since all the arc costs are nonnegative. The solution to these pricing subproblems provides a set of columns that are added to the subproblem if they have negative reduced costs. The column generation algorithm iterates until no negative reduced cost column exists. Optimal primal and dual solutions (ε^*, z^*) and (η^*, λ^*) to subproblem SPF$(\tilde{y})$ are thus obtained and a violated metric cut is identified if the objective function has a non-zero value (*i.e.*, $h(\tilde{y}) > 0$).

By duality, we have

$$h(\tilde{y}) = \sum_{k=1}^{K} d_k \eta_k^* - \sum_{e\in E} \lambda_e^* \sum_{l=1}^{L_e} u_l^e \tilde{y}_l^e. \tag{2.24}$$

Thus, the design variable $\tilde{y}$ is feasible for (**PF**) if and only if the inequality

$$\sum_{e\in E} \lambda_e^* \sum_{l=1}^{L_e} u_l^e y_l^e \geq \sum_{k=1}^{K} d_k \eta_k^* \tag{2.25}$$

holds. This metric cut is then appended to the relaxed master program when violated.

3. Bipartition inequalities

Polyhedral properties of Network Design Problem formulations have been received extensive attention in the literature (*e.g.*, [5, 15, 26]). More precisely, Chouman *et al.* [7]) have proposed five classes of valid inequalities for the special case where a single facility and a per unit cost are considered on each edge.

To accelerate the convergence of the proposed constraint-generation approach, we generate multiple bipartition inequalities based on an original exact separation procedure. By contrast to previous authors [10, 24], we propose to solve an *exact* separation problem of the bipartition inequalities.

3.1. Exact separation of bipartition inequalities

Given a solution to the relaxed master program $\tilde{y}$, we derive the most violated bipartition inequality (2.15) by solving an appropriate 0-1 linear program. This program aims at finding a subset $W \subset V$ such that $d(W) - \sum_{e\in\delta(W)} \sum_{l=1}^{L_e} u_l^e \tilde{y}_l^e$ is maximal. Toward this end, we define the following decision variables:

- $\alpha_i = 1$ if node i is selected in the subset W, and 0 otherwise, $i \in V$,
- $\beta_e = 1$ if edge e is selected in the cutset $\delta(W)$, and 0 otherwise, $e \in E$,
- $\varphi_k = 1$ if the flow demand between s_k and t_k crosses the cutset $\delta(W)$, and 0 otherwise, $k = 1, \ldots K$.

The separation problem reads as follows:

$$\mathbf{SB}(\tilde{y})\mathbf{:}\ g(\tilde{y}) = \text{ Maximize } \sum_{k=1}^{K} d_k \varphi_k - \sum_{e\in E} \bar{u}_e \beta_e \tag{3.1}$$

subject to:

$$\varphi_k - \alpha_{s_k} - \alpha_{t_k} \leq 0, \qquad \forall k = 1, \ldots, K, \tag{3.2}$$

$$\varphi_k + \alpha_{s_k} + \alpha_{t_k} \leq 2, \qquad \forall k = 1, \ldots, K, \tag{3.3}$$

$$\beta_e - \alpha_i - \alpha_j \leq 0, \qquad \forall e = \{i, j\} \in E, \tag{3.4}$$

$$\beta_e + \alpha_i + \alpha_j \leq 2, \qquad \forall e = \{i, j\} \in E, \tag{3.5}$$

$$\beta_e - |\alpha_i - \alpha_j| \geq 0, \qquad \forall e = \{i, j\} \in E, \tag{3.6}$$

$$\alpha_1 = 1, \tag{3.7}$$

$$\sum_{i\in V}\alpha_i \leq n-1, \tag{3.8}$$

$$\alpha_i \in \{0,1\}, \qquad \forall i \in V, \tag{3.9}$$

$$\beta_e \in \{0,1\}, \qquad \forall e \in E, \tag{3.10}$$

$$\varphi_k \in \{0,1\}, \qquad \forall k = 1,\dots,K, \tag{3.11}$$

where $\bar{u}_e = \sum_{l=1}^{L_e} u_l^e \tilde{y}_l^e$ is the installed capacity on edge $e \in E$.

The objective function (3.1) maximizes the difference between the total flow that crosses the cutset and the installed cutset capacity. Constraints (3.2) and (3.3) require, for each commodity k, that if flow demand between s_k and t_k crosses the cutset (that is, $\varphi_k = 1$) then either the source node s_k or the sink node t_k (but not both of them) belongs to W (that is, $\alpha_{s_k} + \alpha_{t_k} = 1$). Constraints (3.4)–(3.6) express similar relations between an edge $e \in E$ and its adjacent nodes. Indeed, for each edge $e = \{i,j\} \in E$, if nodes i and j are both in subset $\bar{W}$ (W) then $\alpha_i + \alpha_j = 0$ $(\alpha_i + \alpha_j = 2)$, and constraints (3.4) (respectively (3.5)) prevent edge e to be in the cutset $(\beta_e = 0)$. Otherwise, nodes i and j are in different subsets (*i.e.*, $(i \in W$ and $j \in \bar{W})$ or $(i \in \bar{W}$ and $j \in W))$ then, $\alpha_i + \alpha_j = 1$, and constraints (3.4) and (3.5) together with constraints (3.6) enforce edge e to be in the cutset $(\beta_e = 1)$. Constraint (3.7) breaks symmetry by setting node 1 in subset W (and thereby reducing the set of feasible solutions). Relation (3.8) enforces W to be a proper subset of V. Finally, (3.8)−(3.10) are the integrality constraints.

Clearly, constraints (3.6) can be linearized as follows:

$$\alpha_i - \alpha_j + \beta_e \geq 0, \qquad \forall e = \{i,j\} \in E, \tag{3.12}$$

$$\alpha_i - \alpha_j - \beta_e \leq 0, \qquad \forall e = \{i,j\} \in E. \tag{3.13}$$

The obtained separation problem can be solved using a general-purpose MIP solver.

For a specific solution $\tilde{y}$, when the optimal solution has a nonnegative value (*i.e.*, $g(\tilde{y}) > 0$), a new constraint that corresponds to the most violated bipartition inequality (2.16) is derived and appended to the relaxed master program.

In our computational experiments, we found that this separation problem can be very quickly solved using a general purpose solver.

3.2. Multiple generation of bipartition inequalities

In addition to the generation of the most violated bipartition inequality, we included a procedure that allows the multiple generation of violated inequalities. Indeed, given a solution $\tilde{y}$ of the relaxed master program, and after identifying the most violated bipartition inequality, additional violated bipartition inequalities can be derived by iteratively updating $\tilde{y}$ and solving again the exact separation problem, until no violated constraint exists. More precisely, assume that the most violated bipartition inequality is induced by the node set W and that $d(W) < \sum_{e\in\delta(W)}\sum_{l=1}^{L_e} u_l^e \tilde{y}_l^e$. Then, a new solution $\tilde{y}^{\text{new}}$ can by derived from $\tilde{y}$ by setting $\tilde{y}_e^{\text{new}} = \tilde{y}_e$ for all $e \in E \setminus \delta(W)$, and computing the values of $\tilde{y}_e^{\text{new}}$ for $e \in \delta(W)$ by solving the following 0-1 knapsack-type problem.

$$\textbf{KP}(\textbf{W})\textbf{:}\ \text{Minimize} \sum_{e\in\delta(W)}\sum_{l=1}^{L_e} f_l^e y_l^e \tag{3.14}$$

subject to:

$$\sum_{l=1}^{L_e} y_l^e \leq 1, \qquad \forall e \in \delta(W), \tag{3.15}$$

$$\sum_{e\in\delta(W)} \sum_{l=1}^{L_e} u_l^e y_l^e \geq d(W), \tag{3.16}$$

$$y_l^e \in \{0,1\}, \qquad \forall e \in \delta(W), \forall l = 1,\dots,L_e. \tag{3.17}$$

This problem requires selecting a minimum-cost subset of edges in $\delta(W)$, such that at most one facility is selected for each edge and the cutset capacity is larger than or equal to the total demand. In other words, $\tilde{y}_e^{\text{new}}$ is derived from $\tilde{y}$ by making feasible the capacity of the cutset $\delta(W)$ at a minimum cost. Next, the exact separation problem can be solved for the updated solution $\tilde{y}_e^{\text{new}}$ and a new violated bipartition inequality, that is induced by a node set W^{new}, can be derived and appended to the relaxed master program. Again, a new knapsack-type problem $\mathbf{KP}(\mathbf{W}^{\text{new}})$ is solved and the process is reiterated until no violated bipartition inequality is found. In so doing, starting from any solution of the relaxed master program, multiple bipartition inequalities are derived and appended to the relaxed master program.

4. The overall approach

In this section, we begin by detailing how we initialized the relaxed master program, then, we summarize the overall proposed approach.

4.1. Initialization of the relaxed master program

It is well-documented in the network design literature that adding initial valid inequalities to the relaxed master program is an effective strategy for accelerating the constraint generation process (see, *e.g.*, [9, 20]). Following Mrad and Haouari [24], we implemented two simple (yet effective) valid inequalities that enforce installing a non-empty set of facilities and thereby prevent the initial zero optimal solution of the master problem.

The first constraint (VI1) enforces the connectivity of the graph that is induced by any feasible solution as follows.

$$\sum_{e\in E} \sum_{l=1}^{L_e} y_l^e \geq n-1. \tag{4.1}$$

The second valid inequalities (VI2) are bipartition inequalities that are generated as follows. For each commodity k, $k = 1,\dots,K$, at the first iteration, set W contains only the source node s_k. Then, at each iteration, set W is expanded by adding its adjacent nodes, until reaching the sink node t_k. For each set W, a cut is identified and a corresponding bipartition inequality (2.15) is derived.

4.2. Synthesis of the constraint-generation approach

A synthesis of the overall proposed approach is given below:

Step 0: **Initialization**
1.1 Input a MNOP-SCF instance defined on an undirected graph G.
1.2 Initialize the relaxed master program (RMP_1) with the valid inequalities VI1 and VI2 that are described in Section 4.1. Set $t = 1$.

Step 1: **Solution of the relaxed master program**
Solve Model RMP_t. Let $\tilde{y}_t$ be an optimal solution.

Step 2: **Exact separation of a bipartition inequality**
2.1 Solve Model $SB(\tilde{y}_t)$ defined by (3.1)–(3.11).
Let $\delta(W_t)$ be the optimal cutset.
2.2 If $(g(\tilde{y}_t) \leq 0)$ then Go to Step 4.
2.3 Append to RMP_t the bipartition inequality that is induced by $\delta(W_t)$.
Step 3: **Generation of additional bipartition inequalities**
3.1 Solve Model $KP(W_t)$ defined by (3.14)–(3.17).
Let y^{new} be an optimal solution.
3.2 Solve Model $SB(y^{\text{new}})$ defined by (3.1)–(3.11).
Let $\delta(W^{\text{new}})$ be the optimal cutset.
3.3 If $(g(y^{\text{new}}) \leq 0)$ then Set $t \leftarrow t+1$, Go to Step 1.
3.4 Append to RMP_t the bipartition inequality that is induced by $\delta(W^{\text{new}})$.
3.5 Set $W_t \leftarrow W^{\text{new}}$, Go to Step 3.1.
Step 4: **Exact separation of a metric inequality**
4.1 Solve Model $\text{SPF}(\tilde{y}_t)$ defined by (2.19)–(2.22)
using column-generation. Let (λ^t, η^t) denote the optimal dual solution.
4.2 If $(h(\tilde{y}_t) \leq 0)$ then Output $\tilde{y}_t$ as optimal solution. Stop.
4.3 Append to RMP_t the metric inequality (2.25) that is induced by (λ^t, η^t).
4.4 Set $t \leftarrow t+1$, Go to Step 1.

5. Computational results

To assess the empirical performance of the proposed constraint generation based approach, we conducted three sets of experiments. The first of these investigates the effectiveness of the proposed approach in terms of required computational effort to solve different sets of instances. For the second set of experiments, we analyze the impact of the new generated bipartition inequalities. Finally, in a third set of experiments, we examine the performance of the proposed approach on the Network Loading Problem.

Toward this end, we implemented all the algorithms using $C\#$ language in concert with the commercial MIP solver CPLEX (version 12.5). All the computational experiments were carried out on an $i7$ dual core 2.4 GHz Personal Computer with 12.0 GB RAM. It is worthy mentioning that the datasets used in the experimentation are freely available at [18].

5.1. Performance of the proposed approach

The objective of the first set of experiments is to test the performance of the proposed approach on large instances and to compare it to previous works. For that aim, we conducted computational experiments on three test-beds of instances. The first one consists of 40 instances that were randomly generated using Mrad and Haouari's MNOP-SCF instances generator described in Mrad and Haouari [24]. These instances have different sizes ranging from 10 to 100 nodes, and from 15 to 1000 edges. For all instances, commodities between all pair of nodes should be routed (*i.e.*, $K = \frac{n(n-1)}{2}$) and the number of available facility types is $L_e = 3$, for each edge $e \in E$. The results are summarized in Table 1. For each instance, we provided the value of the optimal solution (Sol), the required total CPU time in seconds (Time (s)), the percentage of CPU time spent in Step i of the overall approach described in Section 4.2 (P_{TimeSi}), $i = 1, \ldots, 4$, the number of Metric Inequalities (MI), and the number of Bipartition Inequalities (BI).

We see from Table 1 that instances having up to 100 nodes and 600 edges are exactly solved, and that the average CPU time is 32.52 min and the maximum CPU time is 2 h 40 min. It is noteworthy that the column generation procedure is consuming an average of 56.4% of the total CPU time. In addition, we observe that the overall approach provides optimal solutions for all instances with up to 85 nodes and 430 edges within less than 1 h of CPU time. These results show that our approach is competitive with previous state-of-the-art procedures. Indeed, to the best of our knowledge, the solution of such large instances has never been reported in the literature. Compared to the work of Mrad and Haouari [24], their exact constraint generation approach

TABLE 1. Performance of the proposed approach on Mrad and Haouari's MNOP-SCF instances.

Inst.	n	m	K	Sol	Time (s)	P_{TimeS1}	P_{TimeS2}	P_{TimeS3}	P_{TimeS4}	MI	BI
MH01	10	15	45	1215	2.76	34.42%	10.87%	6.52%	48.19%	0	16
MH02	15	20	105	2444	3.50	15.14%	6.00%	2.57%	76.29%	0	10
MH03	15	25	105	3187	2.44	32.38%	19.26%	9.84%	38.52%	0	6
MH04	15	30	105	3481	1.87	29.95%	17.65%	10.70%	41.71%	0	5
MH05	30	60	435	16 445	13.29	29.27%	18.43%	8.50%	43.79%	0	13
MH06	35	70	595	28 142	17.20	39.30%	5.29%	2.15%	53.26%	0	7
MH07	45	80	990	43 032	82.56	24.99%	5.92%	2.79%	66.30%	0	36
MH08	50	90	1225	55 820	87.58	35.61%	0.53%	0.15%	63.71%	0	11
MH09	50	100	1225	59 671	105.57	36.84%	15.64%	8.68%	38.85%	0	28
MH10	50	400	1225	59 466	246.08	31.94%	7.34%	3.86%	56.86%	0	6
MH11	55	230	1485	72 572	719.49	34.71%	2.66%	0.77%	61.86%	0	30
MH12	55	240	1485	66 930	224.79	40.91%	3.66%	2.38%	53.05%	0	8
MH13	55	250	1485	81 741	231.79	21.81%	6.82%	4.09%	67.29%	0	5
MH14	55	450	1485	89 516	382.11	25.56%	3.59%	2.24%	68.61%	0	5
MH15	60	260	1770	157 019	328.93	38.11%	5.36%	2.61%	54.05%	0	10
MH16	60	280	1770	141 989	791.77	41.50%	3.00%	0.24%	55.26%	0	31
MH17	60	600	1770	112 325	700.42	31.30%	2.23%	1.66%	64.81%	0	15
MH18	65	290	2080	214 355	695.08	33.95%	1.62%	1.26%	63.17%	0	22
MH19	65	300	2080	229 260	593.76	30.36%	2.42%	1.82%	65.41%	0	18
MH20	65	310	2080	186 004	657.66	33.65%	2.49%	1.72%	62.14%	0	9
MH21	65	700	2080	133 474	1200.66	44.60%	1.83%	1.18%	52.39%	0	6
MH22	70	330	2415	258 516	905.43	36.33%	2.19%	1.95%	59.53%	0	7
MH23	70	750	2415	213 193	2000.54	49.37%	1.66%	1.34%	47.63%	0	20
MH24	75	360	2775	359 943	1620.09	49.15%	1.63%	1.26%	47.96%	0	25
MH25	75	370	2775	388 775	1300.90	44.96%	2.29%	1.96%	50.79%	0	5
MH26	75	800	2775	238 880	2800.05	35.01%	1.58%	1.32%	62.09%	0	5
MH27	80	380	3160	397 327	2284.39	33.69%	1.54%	1.03%	63.74%	0	13
MH28	80	390	3160	438 138	2003.37	49.93%	1.29%	0.67%	48.11%	0	8
MH29	80	400	3160	429 075	1949.44	49.54%	2.14%	1.37%	46.94%	0	19
MH30	80	950	3160	321 676	3400.20	39.29%	0.88%	0.55%	59.28%	0	9
MH31	85	410	3570	488 589	2841.34	36.00%	2.66%	1.36%	59.97%	0	31
MH32	85	420	3570	470 112	3232.38	46.09%	2.93%	1.14%	49.85%	0	14
MH33	85	430	3570	478 810	3047.78	52.28%	2.52%	0.39%	44.81%	0	27
MH34	85	1000	3570	381 612	4811.62	39.05%	1.23%	1.01%	58.70%	0	19
MH35	90	440	4005	551 247	3867.81	46.37%	2.83%	1.68%	49.12%	0	29
MH36	90	450	4005	748 934	3600.79	31.11%	1.92%	0.92%	66.06%	0	14
MH37	90	460	4005	754 492	3843.91	32.75%	2.03%	0.91%	64.31%	0	35
MH38	100	300	4950	1 063 958	9059.9	32.76%	2.04%	2.86%	62.34%	0	147
MH39	100	500	4950	959 200	8813.58	34.21%	1.01%	0.72%	64.07%	0	23
MH40	100	600	4950	826 168	9581.02	40.36%	2.69%	1.05%	55.90%	0	27

is limited to solve instances having up to 50 nodes and 100 edges with 2 instances remained unsolved after one hour of CPU time.

The second test-bed comprises 7 instances of Gabrel and Minoux [12] that were solved optimally by Gabrel *et al.* [10] and already tested in previous works [1, 11]. For these instances, the numbers of nodes and edges range from 15 to 20 and from 26 to 37 respectively. As in the first set of instances, the number of commodities is $K = \frac{n(n-1)}{2}$. However, more facilities are available, reaching 8 facility types for each edge. Table 2 displays the computational results of this test-bed. Column Time G (s) denotes the CPU time in seconds reported by Gabrel *et al.* [11] using a Sun UltraSparc 10. Column Time G (s) denotes the CPU time in seconds required by our approach's code run on a virtual machine with the same characteristics. Columns Time (s), MI and BI present the CPU time in seconds, the number of Metric Inequalities and the number of Bipartition Inequalities obtained

TABLE 2. Performance of the proposed approach on Gabrel *et al.*'s MNOP-SCF instances.

Inst.	n	m	K	Sol	Time G (s)	Time V (s)	Time (s)	MI	BI
rn15$_1$	15	26	105	401	207	126	40	0	28
rn15$_6$	15	26	105	646	467	292	20	0	92
rn15$_7$	15	27	105	678	1041	457	30	0	45
rn20$_9$	20	35	190	884	10 799	931	47	0	45
rn20$_1$0	20	34	190	977	1235	617	229	0	181
rn20$_2$2	20	37	190	799	49 478	11 893	1401	0	252
rn20$_6$1	20	35	190	785	5994	252	20	0	29

TABLE 3. Performance of the proposed approach on SNLib MNOP-SCF instances.

Inst.	N	m	K	Sol	Time (s)	MI	BI
Pdh_sndlib	11	34	24	11 233 089	103.66	0	231
Yuan	11	42	22	656 600	198.67	0	169
nobel_us_sndlib	14	21	91	260	234.08	0	248
nobel_germany_sndlib	17	26	121	320	60.46	0	65

by our approach, respectively. Taking into account the difference between computers speeds, we observe from Table 2 that our approach outperforms Gabrel *et al.*'s algorithm in terms of CPU time effort.

The third test-bed is composed of 4 instances derived from the Survivable Network Design Library (SNDlib [25]). These instances have fewer commodities to route compared with the two previous test-beds, ranging from 22 to 121. The corresponding networks have between 11 and 17 nodes, and 21 and 42 edges. Table 3 illustrates the results of our proposed approach. All the instances are solved to optimality within an average CPU time of 149.22 seconds.

From Tables 1–3, we notice that our approach generates a reasonable number of Bipartition Inequalities that allow deriving optimal solutions without requiring metric inequalities (MI = 0). This point will be further discussed in Section 5.2.

In order to highlight the performance of our approach using path-based model (**PF**), we implemented a basic Benders decomposition procedure with the widely used arc flow model (**AF**), where the restricted master problem and the subproblem are solved by CPLEX (version 12.5). We compared the two approaches on 10 instances of the first test-bed and displayed their results in Table 4. Columns Time (s) and Time ratio present the arc flow basic approach CPU time in seconds and its ratio over our proposed approach CPU time (indicated in Table 1), respectively. Not surprisingly, the arc flow based approach lags far behind our procedure and fails to solve instances beyond 30 nodes. It is worthy to mention that solving the commonly used arc flow model (**AF**) using straightly the general MIP solver CPLEX (version 12.5) as well as the aforementioned basic Benders decomposition procedure fail to solve all instances of Table 3 within 1 h of CPU Time.

5.2. Impact of bipartition inequalities

The objective of this second set of experiments is to assess the impact of the new bipartition inequalities on the performance of the proposed procedure. For that aim, we considered Mrad and Haouari's MNOP-SCF instances. We first dropped all the proposed bipartition inequalities and tested the obtained basic variant. The results obtained within 1 h of CPU time are displayed in Table 5. For each instance, it indicates the solution obtained by the basic variant (Sol), and its gap to the optimal solution (Gap (%)). The last three columns provide the number of Metric Inequalities (MI) generated by the basic variant, its CPU time in seconds ($Time(s)$) and the ratio of this time over the complete proposed approach CPU time ($Time\ ratio$).

TABLE 4. Performance of the arc flow formulation on Mrad and Haouari's MNOP-SCF instances.

Inst.	Sol	Time (s)	Time ratio
MH01	1215	1.95	0.70
MH02	2444	8.58	2.45
MH03	3187	35.07	14.37
MH04	3481	109.73	58.67
MH05*	–	–	–
MH06*	–	–	–
MH07*	–	–	–
MH08*	–	–	–
MH09*	–	–	–
MH10*	–	–	–

Notes. (*)No feasible solution has been obtained.

TABLE 5. Performance of the variant without bipartition inequalities (maximum CPU time = 1 h).

Inst.	Sol	Gap (%)	MI	Time(s)	Time ratio
MH01	1215	0.00	9	10.9	3.94
MH02	2444	0.00	39	183.31	52.37
MH03	3187	0.00	31	217.76	89.24
MH04	3481	0.00	45	1051.67	562.39
MH05	16 106	2.06	26	>3600	–
MH06	27 705	1.55	23	>3600	–
MH07	41 112	4.46	13	>3600	–
MH08	54 899	1.65	8	>3600	–
MH09	57 976	2.84	8	>3600	–
MH10	59 466	0.00	4	>3600	–
MH11	72 064	0.70	5	>3600	–
MH12	66 930	0.00	4	>3600	–
MH13	81 126	0.75	3	>3600	–
MH14	89 516	0.00	3	>3600	–
MH15	156 137	0.56	1	>3600	–
MH16	141 029	0.68	3	>3600	–
MH17	112 325	0.00	2	>3600	–
MH18	211 946	1.12	3	>3600	–
MH19	224 302	2.16	2	>3600	–
MH20	186 004	0.00	3	>3600	–

Table 5 illustrates the effectiveness of the new cutset inequalities. Indeed, it shows that, within the time limit, the basic variant provides the optimality proof for only 4 over the 20 instances and approximate solutions that are within 1.7% of the optimal values in average are derived for the other instances. These latter instances remain unsolved after several hours of computing. This yields to decline the insight that the performance of the proposed approach is due to the path based formulation suggested from Table 4 results.

Interestingly, Table 5 shows that only few metric inequalities were appended to the relaxed master program for instances with more than 50 nodes within the time limit of 1 h. It confirms that their generation process is time consuming within the overall procedure.

Moreover, we tested the proposed constraint generation approach with the exact separation of the most violated bipartition inequalities, but without appending additional cuts generated by the knapsack heuristic (discussed in Section 3.2). The results obtained by this augmented variant within 1 h of CPU time are displayed

TABLE 6. Performance of the variant without multiple bipartition inequalities generation (maximum CPU time = 1 h).

Inst.	Sol	Gap (%)	MI	BI	Time(s)	Time ratio
MH01	1215	0.00	0	4	4.97	1.80
MH02	2444	0.00	0	8	7.73	2.21
MH03	3187	0.00	0	6	3.25	1.33
MH04	3481	0.00	0	5	4.66	2.49
MH05	16 445	0.00	0	13	32.33	2.43
MH06	28 142	0.00	0	7	76.94	4.47
MH07	43 032	0.00	0	54	954.57	11.56
MH08	55 820	0.00	0	16	847.84	9.68
MH09	59 671	0.00	0	49	2316.95	21.95
MH10	59 466	0.00	0	6	813.1	3.30
MH11	72 572*	0.00	0	161	>3600	–
MH12	66 930	0.00	0	5	2036.99	9.06
MH13	81 741	0.00	0	5	706.23	3.05
MH14	89 516	0.00	0	5	618.4	1.62
MH15	157 019	0.00	0	11	1679.71	5.11
MH16	141 989	0.00	0	31	3539.34	4.47
MH17	112 325*	0.00	0	18	>3600	–
MH18	214 355*	0.00	0	31	>3600	–
MH19	229 260*	0.00	0	17	>3600	–
MH20	186 004	0.00	0	8	2745.31	4.17

Notes. (*)Optimality was not proved after 1 h CPU time.

in Table 6. For each instance, it indicates the obtained solution (Sol), the gap to the optimal solution ($Gap(\%)$), the number of Metric Inequalities (MI) the number of Bipartition Inequalities (BI), the CPU time in seconds ($Time(s)$) and the ratio of this time over the complete proposed approach CPU time ($Time\ ratio$).

Table 6 indicates that augmenting the relaxed master program with only the most violated cuts allows to derive optimal solutions for 16 out of the 20 instances, without generating any metric inequality. For the four remained instances, this augmented variant yields to optimal solution but without optimality proof. Furthermore, we observe that bipartition inequalities derived from knapsack heuristics positively impacts the overall efficacy of the proposed approach. Indeed, these additional inequalities enhance the performance of the overall procedure and make it 5.51 times faster in average.

5.3. Performance on the network loading problem

The last set of experiments examines the performance of the proposed approach on Network Loading Problem instances. As the NLP requires finding minimum cost integer facilities that allows simultaneous routing all the point-to-point demands without exceeding any of the installed capacities, then we adopt the proposed approach by mainly considering y integer variables in the path-based model (**PF**) and replacing constraints (2.11) by

$$\sum_{k=1}^{K}\sum_{r=1}^{|P_k|} a_{erk} z_r^k \leq y_e, \qquad \forall e \in E. \tag{5.1}$$

The numerical experiments, summarized in Table 7, were carried on a set of network topologies extracted from the Survivable Network Design Library (SNDlib [25]). These instances are similar to those already tested by Mattia [21]. We observe from this table, that the proposed approach still exhibit a good performance, being

TABLE 7. Performance of the proposed approach on NLP instances.

Inst.	n	m	K	Sol	MI	BI	Time (s)
New York	16	49	54	4 516 400	0	5	3.57
France	25	45	300	22 200*	1	73	>3600
Norway	27	51	210	960 430	0	8	6.33
Nobel-eu	28	41	378	1 426 800	0	414	130.55
Cost266	37	57	1244	15 359 400*	1	811	>3600

Notes. (*)Optimality was not proved after 1 h CPU time.

able to solve exactly medium-size instances within a reasonable CPU effort. Indeed, 3 out of 5 instances are solved to optimality in an average time of 46.8 seconds and without generating any metric inequality.

6. Conclusion

In this paper, we addressed the Multicommodity Network Optimization Problem with a Step Cost Function. This challenging problem has a wealth of pertinence to many areas including telecommunications, transportation, and logistics. We presented an exact separation model that generates the most violated cutest inequalities as well as a knapsack-based problem that derives additional cuts. The obtained inequalities where embedded in a cut generation-based approach that enabled to solve instances with up to 100 nodes, 600 edges, and 4950 commodities within less than 2 h 40 min of CPU time, while in the literature, the best developed approaches are limited to the optimal solution of instances with 50 nodes, 100 edges, and 1225 commodities.

As a direction for future research, we recommend investigating the nonbifurcated variant of the MNOP-SCF where commodities are routed along single paths. A valid formulation for this variant can readily be derived from Model (**PF**) and reads as follows:

$$\textbf{(NB):} \text{ Minimize } \sum_{e\in E}\sum_{l=1}^{L_e} f_l^e y_l^e \tag{6.1}$$

subject to: (2.2),(2.6),

$$\sum_{r=1}^{|P_k|} \boldsymbol{z}_r^k = 1, \qquad \forall k = 1,\ldots,K, \tag{6.2}$$

$$\sum_{k=1}^{K}\sum_{r=1}^{|P_k|} a_{erk} d_k \boldsymbol{z}_r^k \leq \sum_{l=1}^{L_e} u_l^e y_l^e, \qquad \forall e \in E, \tag{6.3}$$

$$\boldsymbol{z}_r^k \in \{0,1\}, \qquad \forall k = 1,\ldots,K,\ r \in P_k, \tag{6.4}$$

where $\boldsymbol{z}_r^k$ is a binary variable that takes value 1 if path $r \in P_k$ is used for routing all the demands of commodity k, and 0 otherwise ($k = 1,\ldots,K$). Clearly, this model can be strengthened by appending the bipartition inequalities (2.15). Hence, the resulting model includes an exponential number of variables and constraints. It would be interesting solving it using a branch-and-cut-and-price approach, where bipartition inequalities are generated using the proposed exact model. Whether this sophisticated implementation would prove effective remains an open question.

Acknowledgements. The authors would like to thank anonymous referees for their valuable suggestions.

References

[1] D. Aloise and C.C. Ribeiro, Adaptive memory in multistart heuristics for multicommodity network design. *J. Heurist.* **17** (2011) 153–179.

[2] Y.K. Agarwal, Design of capacitated multicommodity networks with multiple facilities. *Oper. Res.* **50** (2002) 333–344.

[3] P. Avella, S. Mattia and A. Sassano, Metric inequalities and the network loading problem. *Discret. Opt.* **4** (2007) 103–114.

[4] A. Balakrishnan, T.L. Magnanti and P. Mirchandani, Network design, edited by M. Dell' Amico, F. Maffioli and S. Martello. In: Annotated Bibliographies in Combinatorial Optimization. Wiley, New York, USA (1997) 311–334.

[5] D. Bienstock, S. Chopra, O. Günlük and C.Y. Tsai, Minimum cost capacity installation for multicommodity network flows. *Math. Progr.* **81** (1998) 177–199.

[6] D. Bienstock and O. Günlük, Capacitated network design—polyhedral structure and computation. *Inf. J Comput.* **8** (1996) 243–259.

[7] M. Chouman, T.G. Crainic and B. Gendron, Commodity representations and cut-set-based inequalities for multicommodity capacitated fixed-charge network design. *Transp. Sci.* **51** (2016) 650–667.

[8] K.L. Croxton, B. Gendron and T.L. Magnanti, A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems. *Manage. Sci.* **49** (2003) 1268–1273.

[9] A.M. Costa, A survey on Benders decomposition applied to fixed-charge network design problems. *Comput. Oper. Res.* **32** (2005) 1429–1450.

[10] V. Gabrel, A. Knippel and M. Minoux, Exact solution of multicommodity network optimization problems with general step cost functions. *Oper. Res. Lett.* **25** (1999) 15–23.

[11] V. Gabrel, A. Knippel and M. Minoux, A comparison of heuristic for the discrete cost multicommodity network optimization problem. *J. Heurist.* **9** (2003) 429–445.

[12] V. Gabrel and M. Minoux, LP relaxations better than convexification for multicommodity network optimization problems with step increasing cost functions. *Acta Math. Vietnam.* **22** (1997) 123–145.

[13] B. Gendron, J.Y. Potvin and P. Soriano, Diversification strategies in local search for a nonbifurcated network loading problem. *Eur. J. Oper. Res.* **142** (2002) 231–241.

[14] B. Gendron, J.Y. Potvin and P. Soriano,A tabu search with slope scaling for the multicommodity capacitated location problem with balancing requirements. *Ann. Oper. Res.* **122** (2003) 193–217.

[15] O. Günlük, A branch-and-cut algorithm for capacitated network design problems. *Math. Progr.* **86** (1999) 17–39.

[16] D.S. Johnson, J.K. Lenstra and A.H.G. Rinnooy Kan, The complexity of the network design problem. *Networks* **8** (1978) 279–285.

[17] K. Onaga and O. Kakusho, On feasibility conditions of multicommodity flows in networks. *IEEE Trans. Circ. Theory* **18** (1971) 425–429.

[18] S.B. Layeb, MNOP-SCF Instances. Available at: https://www.researchgate.net/publication/330162116_MNOP-SCF_Instances (2019)

[19] C. Lee, K. Lee and S. Park, Benders decomposition approach for the robust network design problem with flow bifurcations. *Networks* **62** (2013) 1–16.

[20] I. Ljubić, P. Putz and J.J. Salazar-González, Exact approaches to the single-source network loading problem. *Networks* **59** (2012) 89–106.

[21] S. Mattia, Separating tight metric inequalities by bilevel programming. *Oper. Res. Lett.* **40** (2012) 568–572.

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

[23] M. Minoux, Discrete cost multicommodity network optimization problems and exact solution methods. *Ann. Oper. Res.* **106** (2001) 19–46.

[24] M. Mrad and M. Haouari, Optimal solution of the discrete cost multicommodity network design problem. *Appl. Math. Comput.* **204** (2008) 745–753.

[25] S. Orlowski, R. Wessäly, M. Pióro and A. Tomaszewski, SNDlib 1.0—survivable network design library. *Networks* **55** (2010) 276–286.

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

[27] M. Stoer and G. Dahl, A polyhedral approach to multicommodity survivable network design. *Numer. Math.* **68** (1994) 149–167.