Computers & Operations Research 66 (2016) 105–115

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Unsplittable non-additive capacitated network design using set functions polyhedra

Amal Benhamiche [a,*], A. Ridha Mahjoub [a], Nancy Perrot [c], Eduardo Uchoa [b]

[a] Université Paris-Dauphine, LAMSADE, France
[b] Universidade Federal Fluminense, Departamento de Engenharia de Produção, Brazil
[c] Orange Labs R&D, Issy-Les-Moulineaux, France

ARTICLE INFO

Available online 1 September 2015

Keywords:
Capacitated Network Design
Set functions
Polyhedra
Branch-and-Cut

ABSTRACT

In this paper, we address the *Unsplittable Non-Additive Capacitated Network Design* problem, a variant of the Capacitated Network Design problem where the flow of each commodity cannot be split, even between two facilities installed on the same link. We propose a compact formulation and an aggregated formulation for the problem. The latter requires additional inequalities from considering each individual arc-set. Instead of studying those particular polyhedra, we consider a much more general object, the unitary step monotonically increasing set function polyhedra, and identify some families of facets. The inequalities that are obtained by specializing those facets to the Bin Packing function are separated in a Branch-and-Cut for the problem. Several series of experiments are conducted on random and realistic instances to give an insight on the efficiency of the introduced valid inequalities.

1. Introduction

The design of optimal networks has become one of the major economic issues for nowadays telecommunications industry. Many variants of this problem have been considered in the literature, addressing the topology aspects as well as the installation of capacities and the traffic routing. One of the network design problems that has received a big attention is the so-called capacitated network design (CND) problem. Given a network with a set of commodities and a set of potential capacitated link facilities together with their costs, the problem consists of determining the facilities to install on the network so that the commodities can be routed and the total cost is minimum.

In this paper, we consider a variant of the CND problem. This concerns the case where the commodities cannot be split. More precisely, for its routing, each commodity must go from its origin to its destination through only one path and must use at most one facility on each link of the network. The latter constraint makes impossible to aggregate the capacities installed over a link, and will be referred to as the *non-additivity* of the facilities. This problem arises in the design of telecommunication networks. In particular, we are interested in optical networks holding a set of multiplexer devices interconnected by optical fibres and using the so-called OFDM (*Orthogonal Frequency Division Multiplexing*) technology. Indeed, this technology consists in setting up several facilities referred to as *subbands* on the links of a network. Every subband has a certain capacity and a non-negative cost. In this context, given an optical network, a set of commodities and a set of available subbands, the aim is to identify the minimum cost subbands to install on the links of the network so that the traffic may be routed. In particular, we focus on the problem which concerns the installation of the subbands, which will permit an optimal routing. In fact, an efficient algorithm for solving this restricted version of the problem, which is already NP-hard, as it will be shown later, may be useful for solving the problem of the more general multilayer version. This is our motivation for considering the problem which will be called the *Unsplittable Non-Additive Capacitated Network Design* (UNACND) problem.

The purpose of this paper is to devise a Branch-and-Cut algorithm for the UNACND problem. The algorithm is based on an investigation of the polyhedral structure of the problem when it is restricted to a single link. Previous works have already shown the effectiveness of such approach for solving network design problems (see [1–3] and the references therein). Some results in this paper are presented in a very preliminary stage in [4].

To the best of our knowledge, the UNACND problem has not been considered before. However, other versions of the problem have been widely discussed in the literature. In fact, the restriction of CND to one arc has been investigated first by Magnanti et al. [5], for two facilities and splittable flow assumption. Pochet and Wolsey [6] study the polyhedron of a single-arc network design problem with an arbitrary number of facilities and splittable flow assumption. Brockmüller et al. [1] and van Hoesel et al. [2] investigate the

* Corresponding author.

http://dx.doi.org/10.1016/j.cor.2015.08.009
0305-0548/

CND restricted to one edge (the edge capacity problem). They study the integer knapsack problem arising from this relaxation and introduce the so-called c-strong inequalities and give necessary and sufficient conditions for these inequalities to define facets. In [2], the authors give conditions under which the facets of edge capacity polytope also define facets for the CND polytope. In [3], Atamtürk and Rajan study both splittable and unsplittable CND arc-set polyhedra by considering the existing capacity of the arc. They give a linear-time separation procedure for the residual capacity inequalities and show its effectiveness for the splittable CND. They also use the c-strong inequalities and derive a second class of valid inequalities for the unsplittable CND problem. Similar approaches have also been used to study cut-set polyhedra associated with the CND in [7] and CND with survivability constraints in [8].

Besides, the earlier results on the CND problem and the associated polyhedron can be found in [5], where the authors study a multiple commodities-two facilities network design problem restricted to a single arc. They propose several classes of facet defining valid inequalities that completely describe the convex hull of the arc-set CND solutions. In [9], Magnanti et al. propose a more detailed discussion on the CND problem. They propose two approaches to solve the problem: a Lagrangian approach and a cutting planes approach. In particular, they show that the results given in [5] strengthen the CND formulation. Some of the results given in [5] are generalized by Bienstock and Günlük in [10]. They are also extended for the capacity expansion problem, where the overall capacity of the network can be increased by installing several units of capacitated facilities or “batches” on the links. The authors develop a cutting plane approach based on several facet defining inequalities, to solve the problem. Further polyhedral results are presented in [11–16] for different versions of the CND problem under splittable traffic assumption. In particular in [15,12], the authors study two formulations based on the so-called metric inequalities for the minimum cost CND problem. In [12], Bienstock et al. describe two classes of valid inequalities that define facets and are used to obtain a complete characterization of the considered polyhedron for complete three nodes graphs. Moreover, Mattia et al. [15] introduce the so-called tight metric inequalities and show that all the facets of the polyhedron associated with the solutions of the CND are tight metric inequalities. Note that handling the problem by this approach is similar to the Benders decomposition approach (see [17] for more details on this approach).

More recently, some authors have turned their attention to the multi-layer version of the CND problem (see for instance [18–20] and the references therein). Most of the approaches proposed to solve the multi-layer network design problems are based on the results introduced for their single-layer versions.

Our contribution. The objective of this paper is to solve efficiently the UNACND problem by using a Branch-and-Cut algorithm that embeds new classes of valid inequalities. These are obtained by investigating the polyhedra associated with the single arc UNACND problem. Actually, we realized that different possible variants of the single arc CND are in fact associated with the same polyhedron. We refer to these variants as *functions*. We then introduce the polyhedra associated with a general class of functions called *unitary step monotonically increasing* functions, and study their basic properties. We provide two classes of inequalities called *Min Set I* and *Min Set II* that are valid for all considered functions. We give necessary and sufficient conditions for these inequalities to define facets. Our polyhedral results as well as the separation routines remain available for every considered function, by integrating the specificities of each function. We give an application to the *Bin Packing function*, that is in fact equivalent to the arc-set UNACND. In particular, our results for Min Set I inequalities generalize those provided in [1–3] for c-strong inequalities. Both classes of inequalities Min Set I and Min Set II are used within a Branch-and-Cut algorithm to efficiently solve UNACND problem and to strengthen the linear relaxation of the multi-layer version of this problem.

The rest of the paper is organized as follows. In Section 2 we briefly describe the UNACND problem and its restriction to a single arc. In Section 3, we introduce the set functions polyhedra and study their basic properties. We then present the Min Set I and Min Set II inequalities, and investigate their facial structure. In Section 4, we give an application of our polyhedral results to the Bin Packing function, and show the interest of such application for the UNACND problem. Both Min Set I and Min Set II inequalities are embedded within a Branch-and-Cut algorithm described in Section 5. In this section, we also present the separation procedures used to generate the identified valid inequalities. We then show a set of experiments conducted on random and realistic SNDlib based instances in Section 6. Finally, some concluding remarks are given in Section 7.

2. The unsplittable non-additive capacitated network design problem

The UNACND problem can be presented as follows. Consider a bi-directed graph $G=(V,A)$ that represents an optical network. Each node $v \in V$ corresponds to an optical device (multiplexer) and every arc $a=(i,j) \in A$ corresponds to an optical fibre. If an arc (i,j) exists in A, then (j,i) also belongs to A. Let K be a set of commodities. Each commodity $k \in K$ has an origin node $o_k \in V$, a destination node $d_k \in V$ and a traffic $D^k > 0$ that has to be routed between o_k and d_k. We suppose that a set of equivalent modules, each of capacity C, is available. This set will be denoted by W. Assume that $D^k \le C$, for all $k \in K$. A module $w \in W$ installed on an arc (i,j) is a copy of that arc, and yields a cost c_{ij}. Every module w can carry one or many commodities, but a commodity cannot be split on several paths or even on several modules of the same arc. This specificity makes impossible to aggregate the commodities having the same source and destination nodes, to reduce the size of the problem. Thus, there might be several different commodities with the same origin and destination nodes. The UNACND problem is to determine a minimum cost set of modules that have to be installed on the arcs of G so that a routing path is associated with each commodity from its origin to its destination.

Now consider a set $K=\{1,\dots,n\}$ of items (demands) with weights $D^1, D^2, \dots, D^n$ and bins with the same capacity C. The *bin packing problem* (BPP) consists in assigning each item to one bin so that the total weight of the items in each bin does not exceed C and the number of bins used is minimum [21]. We assume, without loss of generality, that the weights D^k and the capacity C are positive integers and $D^k \le C$, for all $k \in K$. The bin packing problem is NP-hard in general [22] and various approaches have been proposed during the last three decades to solve it. In what follows, we use the relationship between UNACND problem and bin packing problem to show that the former is NP-hard.

Proposition 1. *The UNACND problem is NP-hard even if A has a single arc.*

Proof. We will show that the UNACND problem is NP-hard even when the underlying graph consists of only one arc. The reduction is from the bin packing problem. Consider an instance of the bin packing problem, given by a set of items denoted K, each one having a weight $D^k > 0$, $k \in K$. Let W denote a set of available bins, where every bin has a capacity C. We look for the smallest number of bins needed to pack the items of K. Let us construct the graph $G=(V,A)$, where $V=\{u,v\}$ and $A=\{(u,v)\}$. In other words, G consists of two nodes interconnected by a single arc. For each $k \in K$, we must send D^k units of flow from node u to node v. The set W defines the set of available modules with capacity C, the installation costs are unitary. Let B denote the optimal solution of this UNACND problem, that is the number of modules installed over (u, v). Then B is also the optimal solution of the corresponding bin packing problem.□

Next, we give a compact integer linear programming formulation for the problem.

2.1. Compact formulation

Let $y \in \mathbb{R}^{|A||W|}$ be a decision variable vector such that, for each arc $(i,j) \in A$ and for each module $w \in W$, y_{ij}^w takes the value 1 if w is installed on the arc (i,j), and 0 otherwise. We denote by x_{ij}^{kw}, for $k \in K$, $w \in W$ and $(i,j) \in A$ the decision variable that takes the value 1, if k uses the module w, installed on the arc (i,j), and 0 otherwise. The UNACND problem is then equivalent to the following formulation:

$$\min \sum_{(i,j)\in A} \sum_{w \in W} c_{ij} y_{ij}^w$$

$$\sum_{w\in W}\sum_{j\in V} x_{ji}^{kw} - \sum_{w\in W}\sum_{j\in V} x_{ij}^{kw} = \begin{cases} 1 & \text{if } i = d_k, \\ -1 & \text{if } i = o_k, \\ 0 & \text{otherwise}, \end{cases} \quad \forall\, k \in K, \forall\, i \in V, \tag{1}$$

$$\sum_{k\in K} D^k x_{ij}^{kw} \le C y_{ij}^w, \quad \forall\, w \in W, \forall\, (i,j) \in A, \tag{2}$$

$$x_{ij}^{kw} \in \{0,1\}, \quad \forall\, k \in K, \forall\, w \in W, \forall\, (i,j) \in A, \tag{3}$$

$$y_{ij}^{w} \in \{0,1\}, \quad \forall\, w \in W, \forall\, (i,j) \in A. \tag{4}$$

Equalities (1) are the flow conservation constraints. Together with (3), they enforce that a single path between o_k and d_k is used by each commodity k. Inequalities (2) are the capacity constraints for each installed module. They also ensure that the capacity installed on arc (i,j) is large enough to carry the commodities using this arc. Constraints (3) and (4) are the trivial and integrity constraints.

Consider now a single arc $(i,j) \in A$. The polyhedron

$$P_{ij} := conv\{(x,y) \in \{0,1\}^{|K|\times|W|} \times \{0,1\}^{|W|}: \sum_{k\in K} D^k x_{ij}^{kw} \le C y_{ij}^w \ \ \forall\, w \in W, \sum_{w\in W} x_{ij}^{kw} \le 1 \ \ \forall\, k \in K\}$$

is the convex hull of UNACND problem restricted to the variables related to (i,j). Note that polyhedron P_{ij} has many symmetric solutions and does not present a suitable structure to investigate. In fact, there are few chances that such an investigation can bring any relevant information to help in solving UNACND problem.

To overcome this difficulty, we will introduce a new *aggregated* model that does not specify which copy of the arc (i,j) is used for the routing of a commodity k. Indeed, the idea is just to determine the number of modules that has to be installed on (i,j), so that each commodity can be assigned to one of these modules.

2.2. Aggregated formulation

We will define the following additional decision variables. Let $y \in \mathbb{Z}^+$ be such that for each arc $(i,j) \in A$, $y_{ij} = \sum_{w\in W} y_{ij}^w$ is the number of modules installed on (i,j). Let $x \in \mathbb{R}^{|K||A|}$ be such that for each commodity $k \in K$, and for each arc $(i,j) \in A$, $x_{ij}^k = \sum_{w\in W} x_{ij}^{kw}$, and x_{ij}^k takes the value 1, if k uses some module of the arc (i,j) for its routing, and 0 otherwise.

Consider the following ILP:

$$\min \sum_{(i,j)\in A} c_{ij} y_{ij}$$

$$\sum_{j\in V} x_{ji}^k - \sum_{j\in V} x_{ij}^k = \begin{cases} 1 & \text{if } i = d_k, \\ -1 & \text{if } i = o_k, \\ 0 & \text{otherwise}, \end{cases} \quad \begin{matrix} \forall\, k \in K, \\ \forall\, i \in V, \end{matrix} \tag{5}$$

$$\sum_{k\in K} D^k x_{ij}^k \le C y_{ij}, \quad \forall\, (i,j) \in A, \tag{6}$$

$$x_{ij}^k \in \{0,1\}, y_{ij} \in \{0,\dots,|W|\}, \quad \forall\, k \in K, \forall\, (i,j) \in A. \tag{7}$$

As in formulation (1)–(4), equalities (5) are the flow conservation constraints for each commodity of K. Inequalities (6) will be called *aggregated capacity constraints*. They ensure that the overall capacity of the modules installed over (i,j) is not exceeded by the commodities flowing along (i,j), $(i,j) \in A$. This is not enough to model the non-additivity of the capacities. Therefore, in order to obtain a formulation for the UNACND, we should add the following constraints:

$$(x_{ij}, y_{ij}) \in Q_{ij}, \quad \text{with } x_{ij} = (x_{ij}^k, k \in K), \quad \forall\, (i,j) \in A, \tag{8}$$

where

$$Q_{ij} := conv\{(x,y) \in \{0,1\}^{|K|} \times \mathbb{Z}^+: x_{ij}^k = \sum_{w\in W} x_{ij}^{kw}, y_{ij} \ge \sum_{w\in W} y_{ij}^w, \sum_{k\in K} D^k x_{ij}^{kw} \le C y_{ij}^w \ \forall\, w \in W, x_{ij}^{kw} \in \{0,1\}, y_{ij}^w \in \{0,1\}, \forall\, k \in K, \forall\, w \in W\} \tag{9}$$

Q_{ij} is the projection on (x_{ij}^k, y_{ij}) of polyhedron P_{ij}. Observe that the symmetric solutions of P_{ij} will project on a single point, and Q_{ij} would then be more suitable to investigate. Remark that all polyhedra Q_{ij}, $(i,j) \in A$, are identical. Therefore, we may drop the indices (i,j) in its definition and write:

$$Q := conv\{(x_{ij}, y_{ij}) \in \{0,1\}^{|K|} \times \mathbb{Z}^+ : x_{ij}^k = \sum_{w\in W} x_{ij}^{kw}, y_{ij} \ge \sum_{w\in W} y_{ij}^w, \sum_{k\in K} D^k x_{ij}^{kw} \le C y_{ij}^w \quad \forall\, w \in W, x_{ij}^{kw} \in \{0,1\}, y_{ij}^w \in \{0,1\}, \forall\, k \in K, \forall\, w \in W\} \tag{10}$$

Polyhedron Q is related to a general class of polyhedra, associated with possible variants of the CND problem. In what follows, we introduce a family of functions that captures the general structure of those variants.

3. Set function polyhedra

Let $E = \{1, \dots, n\}$ be a ground set with n elements and let $c = (c_i, i \in E)$ be a weight system associated with E. Let $f: 2^E \longrightarrow \mathbb{Z}^+$ be a set function over E. Let S be a subset of E. We denote by $x^S \in \{0,1\}^E$ the vector given by

$$x_i^S = \begin{cases} 1 & \text{if } i \in S, \\ 0 & \text{otherwise}. \end{cases}$$

We may write $f(x^S)$ for $f(S)$ and let $x(S)$ be equal to $\sum_{i\in S} x_i$.

Definition 1. A function f defined on a subset of elements $S \subseteq E$ is called *monotonically increasing function* if

$$f(S \cup \{s\}) - f(S) \ge 0, \quad \forall\, S \subseteq E, \forall\, s \in E \setminus S.$$

A combinatorial interpretation of such a function is that adding any element of $E\setminus S$ to the subset S yields a nonnegative increase of the function value.

Definition 2. The function f is said to be *unitary step monotonically increasing* if

(i) $f(\emptyset)=0$,
(ii) $f(S\cup\{s\})-f(S)\in\{0,1\}, \quad \forall S\in E, \forall s\in E\backslash S$.

In other words, given a subset $S\subseteq E$, a function f is unitary step monotonically increasing if adding any element s to the initial subset S yields an increase of at most one in the value of f. It is clear by the previous definition that the function f is such that $f(S)\le |S|$, $\forall S\subseteq E$, $f(S\backslash T)\ge f(S)-f(T)$, $\forall S\subseteq E$ and $T\subseteq S$.

Given a unitary step monotonically increasing set function f: $\{0,1\}^n \longrightarrow \mathbb{Z}^+$, we define its polyhedron as

$$P_f{:=}conv\{(x^S,y^S)\in\{0,1\}^n\times\mathbb{Z}^+ : S\subseteq E \quad \text{and} \quad y^S\ge f(x^S)\}. \tag{11}$$

In what follows we will discuss the polyhedron P_f. We will introduce two classes of valid inequalities, namely *Min Set I* and *Min Set II*, and give necessary and sufficient conditions for these inequalities to define facets for any polyhedron P_f, when f is unitary step monotonically increasing.

3.1. Properties of P_f

3.1.1. Dimension

Theorem 1. *The polyhedron P_f is full dimensional.*

Proof. We shall exhibit $n+2$ solutions p_i, $i=1,\ldots,n+2$, whose incidence vectors (x^{S_i},y^{S_i}) are affinely independent. First, consider the solutions $(x^{S_i}, f(S_i))$ induced by the subsets $S_i=\{i\}$, for $i\in E$. Moreover, consider the solutions $(x^{\emptyset},1)$ and $(x^{\emptyset},0)$. It can be easily seen that these $n+2$ solutions are affinely independent.□

In what follows, we will be interested in the facial structure of P_f. In particular we study the trivial inequalities $x_i\ge 0$, and $x_i\le 1$, for all $i\in E$.

3.1.2. Trivial inequalities

Theorem 2. *For $i\in E$, $x_i\ge 0$ defines a facet of P_f.*

Proof. Denote by $\mathcal{F}_i$ the face induced by inequality $x_i\ge 0$, that is

$\mathcal{F}_i=\{(x,y)\in P_f : x_i=0\}$,

Similarly, to prove Theorem 2, we have to identify $n+1$ affinely independent solutions whose incidence vectors belong to $\mathcal{F}_i$. First consider the solutions $(x^{S_j},1)$, where $S_j=\{j\}$, for $j\in E\backslash\{i\}$. Also consider the solutions $(x^{\emptyset},1)$ and $(x^{\emptyset},0)$. Clearly, all these solutions are in $\mathcal{F}_i$. Moreover, they are affinely independent. □

Theorem 3. *For $i\in E$, $x_i\le 1$ defines a facet of P_f.*

Proof. Let us denote by $\mathcal{F}_i$ the face induced by inequality $x_i\le 1$, that is

$\mathcal{F}_i=\{(x,y)\in P_f : x_i=1\}$,

Consider the subsets S_j of E such that $S_j=\{j,i\}$, for $j\in E\backslash\{i\}$. Clearly, the solution $(x^{S_j},2)$ for $j\in E\backslash\{i\}$ belongs to $\mathcal{F}_i$. Moreover, the solutions $(x^E, \mathrm{n}+1)$ and $(x^E, \mathrm{n}+2)$ also belong to $\mathcal{F}_i$. We can see that these $n+1$ solutions are affinely independent.□

In what follows, we will show that all the non-trivial facets of the polyhedron P_f have non negative coefficients.

Theorem 4. *All the non-trivial facet defining inequalities of P_f are of the form $\sum_{i\in E}\pi_i x_i\le\pi_0 y+p$, where p is a non negative integer parameter, and $0\le\pi_i\le\pi_0$, for all $i\in E$.*

Proof. Let $F_{\pi,p}=(\sum_{i\in E}\pi_i x_i=\pi_0 y+p)\cap P_f$ be the facet of P_f defined by $\sum_{i\in E}\pi_i x_i\le\pi_0 y+p$. We will first show that $\pi_i\ge 0$, for all $i\in E$. Consider an element j of E. Since $F_{\pi,p}$ is not contained in the face defined by $x_j\ge 0$, there must exist a set $S\subseteq E$ containing j and $y\in\mathbb{Z}^+$ such that the vector (x^S,y) belongs to $F_{\pi,p}$. So, (i) $\sum_{i\in S}\pi_i x_i=\pi_0 y+p$. Consider the subset $S'=S\backslash\{j\}$. Since $(x^{S'},y)$ belongs to P_f, (ii) $-\sum_{i\in S'}\pi_i x_i\ge -(\pi_0 y+p)$. Adding (i) and (ii), we obtain $\pi_j\ge 0$.

Now we shall show that $\pi_i\le\pi_0$, for all $i\in E$. Consider an element j of E. Since $F_{\pi,p}$ is not contained in the face defined by $x_j=1$, there must exist a set $S\subseteq E$ not containing j and $y\in\mathbb{Z}^+$ such that the vector (x^S,y) belongs to $F_{\pi,p}$. So, (i) $\sum_{i\in S}\pi_i x_i=\pi_0 y+p$. Consider the subset $S''=S\cup\{j\}$. Since $(x^{S''}, y+1)$ belongs to P_f, (ii) $-\sum_{i\in S''}\pi_i x_i\ge -(\pi_0(y+1)+p)$. Adding (i) and (ii), we obtain $-\pi_j\ge -\pi_0$.

We also have $p\ge 0$, since (0,0) is a solution of P_f.□

In what follows we introduce two families of valid inequalities and we describe some conditions under which these inequalities may define facets for polyhedron P_f.

3.2. Min Set I inequalities

Proposition 2. *Let S be a subset of E and p a non negative integer such that $p\ge|S|-f(S)$. Then, the following Min Set I inequality*

$$\sum_{i\in S}x_i\le y+p, \tag{12}$$

is valid for P_f.

Proof. Let S' be any subset of E, and let $T=S'\cap S$. As the function f is unitary step monotonically increasing, $|S|-|T|\ge f(S)-f(T)$ or $|T|\le f(T)+|S|-f(S)$. Hence, $|T|\le f(S')+|S|-f(S)$. For any $y\ge f(S')$, we then have $|T|\le y+|S|-f(S)\le y+p$. Thus, (12) is satisfied for $(x^{S'},y)$.□

Theorem 5. *Inequality (12) defines a facet of P_f if and only if the following holds:*

(i) $p=|S|-f(S)$,
(ii) $f(S\cup\{s\})=f(S)=|S|-p$, *for all* $s\in E\backslash S$,
(iii) $f(S\backslash\{s\})=f(S)-1=|S|-p-1$, *for all* $s\in S$.

Proof. *Necessity*:

(i) Trivial.
(ii) Suppose that there exists an element s of $E\backslash S$ such that $f(S\cup\{s\})=|S|-p+1$. Then the inequality (12) with respect to $S\cup\{s\}$ can be written as

$$\sum_{i\in S\cup\{s\}}x_i\le y+(|S|+1)-f(S\cup\{s\})=y+p. \tag{13}$$

However, (13) dominates (12), and therefore the latter cannot define a facet.
(iii) Suppose there exists $s\in E\backslash S$, such that $f(S\backslash\{s\})=|S|-p$. Inequality (12), with respect to $S\backslash\{s\}$ can be written as

$$\sum_{i\in S\backslash\{s\}}x_i\le y+(|S|-1)-f(S\backslash\{s\})=y+p-1.$$

Inequality (12) can be obtained as a linear combination of the inequality above and $x_s\le 1$. Therefore, it cannot define a facet.

Sufficiency:

Assume now that conditions (i), (ii) and (iii) of Theorem 5 are fulfilled. We will denote by $\mathcal{F}$ the face induced by inequality (12). That is

$$\mathcal{F}=\left\{(x,y)\in P_f:\ \sum_{i\in S}x_i=y+p\right\}.$$

First consider the solution $p_0=(x^S, f(S))$. By (i), $p_0\in\mathcal{F}$. Now let us consider the solutions $p_s=(x^{S\cup\{s\}}, f(S))$, for $s\in E\backslash S$. As by (ii), $f(S\cup\{s\})=f(S)$, we have that p_s is a solution of P_f and also of $\mathcal{F}$. Finally, consider the solutions $p_s=(x^{S\backslash\{s\}}, f(S)-1)$ for all $s\in S$. By (iii), it follows that p_s is a solution of P_f, for $s\in S$. Moreover, p_s satisfies (12) with equality, and then it is also a solution of $\mathcal{F}$. Now, one can easily see that p_0, p_s for $s\in E\backslash S$, p_s for $s\in S$ are affinely independent.□

3.3. Min Set II inequalities

Proposition 3. *Let S be a subset of E, p and q two non negative integers, with $q\geq 2$. Then, the Min Set II inequality*

$$\sum_{i\in S}x_i\leq qy+p, \tag{14}$$

is valid for P_f if $p\geq |T|-qf(T)$, for all $T\subseteq S$.

Proof. Let S' be a subset of S. By summing trivial inequalities $x_i\leq 1$ over S', we get $\sum_{i\in S'}x_i\leq |S'|$ which is valid. On the other hand, by definition of the polyhedron P_f, we have that $y\geq f(S')$, for all $(x^{S'}, y)\in P_f$. As $q\geq 0$, it then follows that, $q(y-f(S'))\geq 0$. Thus

$$\sum_{i\in S}x_i^{S'}=\sum_{i\in S'}x_i^{S'}\leq |S'|+q(y-f(S'))=qy+|S'|-qf(S')\leq qy+p,$$

yielding the validity of (14).□

Theorem 6. *Given a subset of elements $S\subseteq E$, two non negative integers $q\geq 2$ and p. The inequality*

$$\sum_{i\in S}x_i\leq qy+p \tag{15}$$

defines a facet of P_f, if the following holds:

(i) *There exists an integer $r\in\mathbb{Z}^+$, $p\leq r\leq |S|-1$, such that for all $S'\subseteq S$ with $|S'|=r$, $f(S')=\frac{|S'|-p}{q}$,*
(ii) *for all $s\in E\backslash S$, there exists $S'\subseteq S$ such that $f(S')=\frac{|S'|-p}{q}=f(S'\cup\{s\})$,*

Proof. We will denote by $\mathcal{F}$ the face induced by inequality (15), i.e.,

$$\mathcal{F}=\{(x,y)\in P_f:\ \sum_{i\in S}x_i=qy+p\}$$

Suppose that conditions (i) and (ii) hold. We will exhibit $n+1$ solutions of $\mathcal{F}$ that are affinely independent. Consider a subset S' of S such that $|S'|=r$. As by (i), $p\leq r\leq |S|-1$, $S'\neq\varnothing$, $S\neq S'$. Let e' and $\bar{e}'$ be elements of S' and $S\backslash S'$, respectively.

Consider sets $S_e=(S'\backslash\{e'\})\cup\{e\}$ for all $e\in S\backslash S'$ and $S_e=(S'\backslash\{e\})\cup\{\bar{e}'\}$ for all $e\in S'$. Clearly, by (i), solutions $(x^S, f(S))$, $(x^{S_e}, f(S_e))$, $e\in S$ all belong to $\mathcal{F}$.

Next, for each $e\in E\backslash S$, by (ii) there exists $S'_e\subseteq S$ such that $f(S'_e)=\frac{|S'_e|-p}{q}=f(S'_e\cup\{e\})$. Hence, the solutions $(x^{S'_e\cup\{e\}}, f(S'_e\cup\{e\}))$ for all $e\in E\backslash S$ all belong to $\mathcal{F}$. Finally, consider the solution $(x^S, f(S)=\frac{|S|-p}{q})$ which is also in $\mathcal{F}$. Now, it is not hard to see that these solutions constitute a set of $n+1$ affinely independent points.□

In the next section, we will study an application that illustrates well how our results for general set functions are still valid for a specific function. We further provide a counter-example showing that conditions (i) and (ii) of Theorem 6 are not necessary.

4. Application to the Bin Packing function

Definition 3. Let BP: $2^K\longrightarrow\mathbb{Z}^+$ be the function over a set of items K (with associated demands D^k, for all $k\in K$) where every subset $S\subseteq K$ is mapped to $BP(S)$, the minimum number of bins with capacity C necessary to pack the items in S. This function will be referred as the *Bin Packing Function* (BP).

The following statement is easily seen to be true:

Proposition 4. *The Bin Packing Function is unitary step monotonically increasing.*

As by Proposition 4, all the results presented in Section 3 remain valid for the Bin Packing Function. The polyhedron corresponding to the Bin Packing Function is then a particular case of (11) over a set of items K and a capacity C, and is defined as

$$P_{BP}(K,C):=conv\{(x^S,y)\in\{0,1\}^n\times\mathbb{Z}^+:\ S\subseteq K\quad\text{and}\quad y\geq BP(S)\}.$$

When the context is clear, we may drop the arguments K and C and refer to such polyhedron as P_{BP}. Polyhedron P_{BP} is essentially equivalent to polyhedron Q used in the definition of the aggregated formulation for the UNACND. The difference between Q and P_{BP} lies in the fact that pair (x,y) may not belong to Q when W is too small. If $|W|\geq BP(K)$, then both polyhedra coincide. In fact, it is clear that $Q\subseteq P_{BP}$. Moreover, by definition, all extreme points of P_{BP} have the format (x^S,y) for some $S\subseteq K$. If $|W|\geq BP(K)\geq BP(S)$, it would be possible to assign values to the x^{kw} variables in (10) corresponding to a packing of items into at most W bins. Therefore, (x^S,y) also belongs to Q. Thus, we give the following proposition.

Proposition 5. *If $|W|\geq BP(K)$ then $Q=P_{BP}$.*

In practice, due to the huge physical capacity of an optical fibre, there are few a priori limitations on the maximum number of modules that can be installed over an arc. Therefore, in practical instances $Q=P_{BP}$. Accordingly, all the results from the previous section apply directly for polyhedron P_{BP}.

In addition, we remark that the c-strong inequalities for the Unsplittable Capacitated Network Design (UCND) introduced in [1] and studied in [2,3] correspond to the Min Set I inequalities over the polyhedron P_{AP} defined by the following unitary step monotonically increasing function:

Definition 4. Let AP: $2^K\longrightarrow\mathbb{Z}^+$ be the function over a set of items K (with associated demands D^k, for all $k\in K$) where every subset $S\subseteq K$ is mapped to $\lceil\sum_{i\in S}D^i/C\rceil$, the minimum number of multiples of an additive capacity C necessary to pack the items in S. This function will be referred as the *Additive Packing Function (AP)*.

P_{BP} and P_{AP} are not the same polyhedra. Actually, for a given set of demands K and a capacity C, $P_{BP}(K,C)\subseteq P_{AP}(K,C)$. However, P_{BP} and P_{AP} have a similar deep structure. In both cases, Min Set I inequalities satisfying the conditions in Theorem 5 are facet-defining.

4.1. Examples

As example, consider a set K of six demands with sizes 12, 9, 8, 7, 3 and 2, respectively. The bins have capacity 15. Then, the following Min Set I inequality:

$$x^1+x^2+x^4\leq y, \tag{16}$$

defines a facet of P_{BP}. However, this inequality is not even valid for P_{AP} since the point $(1,1,0,1,0,0,2)\in P_{AP}$ is cut off by (16). Further Min Set I inequalities defining facets of P_{BP} include

$x^3 + x^4 + x^5 \le y + 1,$
$x^1 + x^3 + x^4 + x^5 + x^6 \le y + 2.$

Now consider a set K of six items with sizes 12, 9, 8, 7, 3 and 2. Assume that each available bin has a capacity of 15. Then, the following inequalities are examples of Min Set II inequality that define facets of P_{BP}

$$x^1 + x^5 + x^6 \le 2y, \tag{17}$$

$$x^1 + x^2 + x^3 + x^4 + x^6 \le 2y, \tag{18}$$

Note that inequality (18) is an example of facet-inducing inequality that does not satisfy conditions of Theorem 6. This implies that conditions of Theorem 6 are not necessary.

In the next section, we devise a Branch-and-Cut algorithm for the UNACND problem. This algorithm uses theoretical results presented in the previous sections. First, we give an outline of the algorithm. Then, we describe the separation procedures used for some valid inequalities.

5. A Branch-and-Cut algorithm for the UNACND problem

In this section, we devise a Branch-and-Cut algorithm for the UNACND problem. The algorithm is based on the aggregated formulation of the problem. Our aim is to address the algorithmic applications of the polyhedral results given in the previous section. In particular, we would like to examine the effectiveness of the Min Set I and Min Set II inequalities.

Suppose that we are given a bi-directed graph $G = (V, A)$ and a weight vector $c \in \mathbb{R}_+^A$ associated with the arcs of G. Let K be a set of commodities to be routed on G and W a set of available facilities per arc. To start the optimization, we consider the following linear program, LP$_{initial}$, given by the flow conservation constraints and the aggregated capacity constraints associated with the arcs of G, together with the trivial inequalities, that is

$$\min \sum_{(i,j)\in A} c_{ij} y_{ij}$$

$$\sum_{j\in V} x_{ji}^k - \sum_{j\in V} x_{ij}^k = \begin{cases} 1 & \text{if } i = d_k, \\ -1 & \text{if } i = o_k, \\ 0 & \text{otherwise}, \end{cases} \quad \forall\, k \in K, \forall\, i \in V, \tag{19}$$

$$\sum_{k\in K} D^k x_{ij}^k \le C y_{ij}, \quad \forall\, (i,j) \in A, \tag{20}$$

$$0 \le x_{ij}^k \le 1, \quad \forall\, k \in K, \forall\, (i,j) \in A, \tag{21}$$

$$0 \le y_{ij} \le |W|, \quad \forall\, (i,j) \in A. \tag{22}$$

Proposition 6. *An integral solution* $(\bar{x}, \bar{y})$ *of* (19)–(22) *is a valid UNACND solution if and only if it satisfies Min Set I inequalities.*

Proof. *Necessity*: Let $(\bar{x}, \bar{y})$ be an integer solution of (19)–(22). Assume that there exists a subset of commodities, say $\bar{S}$, and an arc $(i,j) \in A$, such that $\bar{x}_{ij}^k = 1$, for all $k \in \bar{S}$, and so that

$$\sum_{k\in\bar{S}} \bar{x}_{ij}^k > \bar{y}_{ij} + p, \tag{23}$$

By Proposition 2, $p \ge |\bar{S}| - \mathrm{BP}(\bar{S})$, which leads to $\bar{y}_{ij} < \sum_{k\in\bar{S}} \bar{x}_{ij}^k + BP(\bar{S}) - |\bar{S}| = |\bar{S}| + BP(\bar{S}) - |\bar{S}|$. Thus, we get $\bar{y}_{ij} < BP(\bar{S})$, that is to say $(\bar{x}, \bar{y})$ is not feasible for UNACND problem.

Sufficiency: Let $(\bar{x}, \bar{y})$ be an integer solution of (19)–(22) that satisfies all the Min Set I inequalities. Then this solution guarantees that enough capacity is installed over the arcs of G to ensure the feasibility for UNACND problem. In fact, Min Set I inequalities dominate the capacity constraints (2) and ensure that a solution of the aggregated formulation is feasible for the problem.□

The optimal solution of this program, say $(\bar{x}, \bar{y})$ is feasible for the problem if it is integer and satisfies all the Min Set I inequalities. Usually, this is not the case. Then, at each iteration of the Branch-and-Cut algorithm, one has to generate further inequalities that are valid for the problem but violated by the current solution. For this, we solve the separation problem related to Min Set I and Min Set II inequalities. The separation of these inequalities is performed in the following order:

1 . Min Set I inequalities
2 . Min Set II inequalities

Moreover, at each iteration, we may add more than one violated inequality. Also we move to the separation of a new class of inequalities only if no additional inequalities can be identified in the current class. Note that the cutting plane is a global procedure, applied to all the nodes of the Branch-and-Cut tree. This allows us to enhance the quality of the lower bound for the problem.

In what follows, we discuss separation algorithms for Min Set I and Min Set II given by inequalities (12) and (14), respectively.

5.1. Separation

5.1.1. Min Set I inequalities

For each arc $a \in A$, we look for a Min Set I inequality (12) that is violated by the current fractional solution $(\bar{x}, \bar{y})$. Let a be an arc of A, and S a subset of commodities in K. In order to separate inequalities (12), we consider the inequalities

$$\sum_{k\in S} x_a^k \le y_a + p_r, \text{ for all } a \in A, S \subseteq K, \tag{24}$$

where $p_r = |S| - \sum_{k\in S} \frac{D^k}{C}$. Notice that p_r is obtained by replacing BP (S) by the trivial lower bound $\sum_{k\in S} \frac{D^k}{C}$ in p. In what follows, we describe a procedure for the separation of this relaxed version of inequalities (12). The idea of our heuristic is to separate inequalities (24). This separation allows us to easily exhibit a subset S which might induce a violated Min Set I inequality. Let us introduce the variable α^k, for all $k \in K$, that takes 1 if the item k is in S and 0 if not. For a solution $(\bar{x}, \bar{y})$, and an arc $a \in A$, the separation problem associated with inequality (24) is equivalent to the following integer linear program:

$$\max Z = \sum_{k\in K} (\bar{x}_a^k + \frac{D^k}{C} - 1)\alpha^k - \bar{y}_a \tag{25}$$

$$0 \le \alpha^k \le 1, \alpha^k \in \{0, 1\}, \quad \forall\, k \in K. \tag{26}$$

The ILP (25) and (26) is maximized by setting to 1 the α^k variables corresponding to the set S_a given by

$$S_a = \{k \in K | \bar{x}_a^k + \frac{D^k}{C} - 1 \ge 0\}. \tag{27}$$

If the corresponding optimal solution $Z^* > 0$, then (24) is violated. But, even if it is not, we verify if the stronger Min Set I inequality over S_a is violated.

Actually, instead of computing the exact value of $BP(S_a)$, we use a strong lower bound introduced by Fekete and Shepers [23], which relies on the so-called *dual feasible function*. These functions have been introduced by Lueker [24] and used first by Johnson

[25] and then Lueker [24] to derive lower bounds for bin packing problems (see [26] for detailed description of dual feasible functions). The Fekete and Shepers's bound we shall use corresponds to the bound $L_*^{(2)}(S_a)$ in [23], where S_a is a subset of items. This function uses and strengthen lower bounds previously introduced by Martello and Toth [21,27]. An interesting property of this bound function states that if the items of S_a are larger than $\frac{1}{3}C$, where C is the capacity of a bin, then $L_*^2(S_a)$ equals $BP(S_a)$.

Therefore, our separation routine for the Min Set I inequalities, given in Algorithm 2 can be presented as follows. For each arc $a \in A$, we determine a subset of commodities $S_a \subseteq K$, using Algorithm 1. Then, we compute $L_*^2(S)$ and $p = |S| - L_*^2(S)$, and check whether S_a and p produce a violated Min Set I inequality.

Algorithm 1. Separation heuristic for inequalities (12).

Data: a solution $(\overline{x}, \overline{y})$
Output: a set $\mathcal{M}I$ of Min Set I inequalities violated by $(\overline{x}, \overline{y})$
$\mathcal{M}I \leftarrow \emptyset$;
Forall $a \in A$ **do**
  $S_a = \{k \in K | \overline{x}_a^k + \frac{D^k}{C} - 1 \geq 0\}$
  Compute the parameter $p = |S| - L_*^2(S_a)$
  If $\sum_{k \in S_a} \overline{x}_a^k - \overline{y}_a > p$ **then**
    Denote I_a this inequality;
    $\mathcal{M}I \leftarrow \mathcal{M}I \cup \mathcal{I}_a$;
return the identified violated Min Set I inequalities $\mathcal{M}I$.

Fekete and Shepers's function $L_*^2(S)$ can be computed in $O(|K| \log(|K|))$. In fact, the computational effort consists in sorting the commodities by traffic amount. As the operation is iterated for each arc of A, our separation procedure runs in time $O(m|K| \log(|K|))$, where $m = |A|$. However, if the commodities are already sorted by traffic amount, then we have a complexity of $O(m|K|)$.

The use of Fekete and Shepers's lower bound accelerates a lot the separation. However, a few Bin Packing problems must still be solved to optimality. If $(\overline{x}, \overline{y})$ is integral, then Algorithm 1 uses a MIP model for calculating $BP(S_a)$ (instead of $L_*^2(S_a)$) to check the feasibility of this solution. By Proposition 6, this makes sure that either $(\overline{x}, \overline{y})$ is indeed a feasible UNACND solution or it violates a Min Set I inequality.

5.1.2. Min set II inequality

Now we turn to the separation of Min Set II inequalities. This relies on the separation procedure developed before for Min Set I inequalities. First note that for Min Set II inequalities, the validity condition requires, for a set $S \subseteq K$ and two integers p and q, that $p \geq |S'| - qBP(S')$, for all $S' \subseteq S$. Hence, the separation for inequalities (14) must incorporate the computation of $|S'| - qBP(S')$ or an approximation value of this for all $S' \subseteq S$. As the number of possibilities may be very large ($2^{|S|}$ possibilities), this would not be possible for large S. For this, our separation routine for these inequalities will only consider sets with small size, not exceeding 4 elements. Moreover, we shall also consider the separation for inequalities with $p=0$ and $q=2$. In fact, by our experiments, we have noted that most of the violated Min Set II inequalities are of this type. The procedure works as follows. For every arc $a \in A$, we compute a subset S_a of commodities using (27). If $|S_a| \leq 4$, then for every subset S' of S_a, we compute $L_*^2(S')$. If the inequality induced by S_a is valid, then we verify if it is violated by the current solution $(\overline{x}, \overline{y})$.

6. Computational experiments

In this section, we will present our experimental results. The Branch-and-Cut algorithm given in the previous section has been implemented in C++ using CPLEX 12.5 as a linear solver and to handle the Branch-and-Cut framework. Our algorithm was tested on a Bi-Xeon quad-core E5507 2.27 GHz with 8Go of RAM, running under Linux. Finally, we have fixed a CPU time limit of five hours.

The results given here have been obtained by considering instances from a library dedicated to the optimization of telecommunication networks, namely SNDlib [28]. We have considered random and realistic instances. Each instance is characterized by the node set V, the arc set A, the set of available facilities W, and the set of commodities K. We included instances with randomly generated commodities and realistic commodities. Both classes of instances are characterized by the number of nodes $|V|$, the number of arcs $|A|$, the number of available facilities denoted $|W|$, and the number of commodities $|K|$.

The random demand instances use topologies from SNDlib instances *polska*, *nobel_us*, *newyork*, *geant*, *ta1* and *pioro40*. The sets of nodes and arcs in G correspond to those instances, each edge of the undirected SNDlib instances induces two inversely directed arcs in our instances. We associate with each arc a length that is the rounded euclidean distance between the end nodes of the arc. Moreover, a facility settled on an arc has a cost equivalent to the length of this arc. The available facilities are supposed to have the same capacity and their maximum number is fixed to ten on all arcs ($|W| = 10$). Concerning the traffic matrices, we randomly generate the origin and destination nodes. The traffic demands are picked from a uniform distribution in the interval $[0.2C, C]$. For each SNDlib original instance, we choose different sizes of K and generate five instances for each $|K|$. The reported results in Table 2 are the averages over those sets of five similar instances.

The realistic demand instances also use SNDlib network topologies, from *abilene*, *atlanta*, *nobel_germany*, *france*, *nobel_eu*, *india35*, *cost266* and *zib54* instances. We assume that we can install at most five facilities in each arc ($|W| = 5$). For each SNDlib original instance, we choose different sizes of K, and pick the $|K|$ largest demands from the original instances.

We have performed a simple preprocessing operation on our instances in order to speed up the solution process. Each commodity that is not compatible with any other commodity has its demand increased to the capacity value. In other words, if for a commodity k, $D^k + D^{k'} \geq C$, for all $k' \in K \setminus \{k\}$, then D^k is changed to C. This improves the lower bound provided by the linear relaxation without changing the optimal integral solution.

Next, we present the experiment results obtained by our Branch-and-Cut algorithm. These are reported in the tables given below. The entries of the various tables are:

$\|V\|$:	number of nodes in G,
$\|A\|$:	number of arcs,
$\|K\|$:	number of commodities,
NmsI:	number of generated Min Set I inequalities,
NmsII:	number of generated Min Set II inequalities,
nodes:	number of nodes in the Branch-and-Cut tree,
o/p:	number of instances solved to optimality over number of tested instances (only for tests with randomly generated traffic),
Gap:	the relative error between the best upper bound (optimal solution if the problem has been solved to optimality) and the lower bound at the root node (before branching),
FinalGap:	the relative error between the best upper bound (optimal solution if the problem has been solved to optimality) and the best lower bound over the Branch-and-Cut tree

TT: total CPU time in h:m:s
TTsep: total CPU time spent in performing the constraints separation in seconds.

The runs are stopped if CPU time reaches 5 h. For those instances not solved to optimality, the gap at the root node, as well as the final gap are indicated in italic.

Our first series of experiments was performed on the random demand instances. We compare the results of the compact formulation (1)–(4), solved by CPLEX, with those of the Branch-and-Cut algorithm over the aggregated formulation (AF) (5)–(7) and using Min Set I first, then both Min Set I and Min Set II inequalities. The goal of these experiments is to show the efficiency of the valid inequalities introduced in the previous sections. The results are summarized in Table 1, where it is possible to compare the performances of the three approaches in terms of gap, number of nodes of the Branch-and-Bound (respectively Branch-and-Cut) tree, and CPU time.

The aggregated formulation with valid inequalities (Min Set I + Min Set II) performs better than the compact formulation for all the instances. In fact, we can notice that the Branch-and-Cut approach allows us to solve to optimality all the tested instances in a short time (less than 45 min for all the instances except for *newyork*_20_1, *ta1*_16_1, *ta1*_18_1, and *ta1*_20_1), whereas many instances could not be solved by Branch-and-Bound after 5 h. For example *polska*_14_1 is solved to optimality in less then 10 min by Branch-and-Cut, while it could not be solved by Cplex Branch-and-Bound. The root gaps obtained by the Branch-and-Cut are usually smaller, but not always. This may happen because CPLEX internal cuts are allowed in the experiments with the compact formulation and disabled in the Branch-and-Cut experiments. Note that an optimal solution could be obtained for instance *ta*1_20_1 in less than 25 h, while CPLEX was not able to prove the optimality of this solution for the compact formulation after 10 days of computation.

Overall, the results presented in Table 1 clearly show the gain provided by using the valid inequalities introduced in the previous sections, within a Branch-and-Cut framework. They also indicate that the aggregated formulation allowed us to get over the symmetries of the problem and solve it efficiently. However, as expected, separating Min Set II inequalities, in addition to Min Set I

Table 2
Branch-and-Cut results for SNDlib instances with random demands.

Instance	|V|	|A|	|K|	NmsI	NmsII	Gap	Opt	nodes	TT
polska_10	12	36	10	247.6	3.2	14.61	5/5	66.4	0:00:03
polska_15	12	36	15	533.2	3.6	16.63	5/5	205.8	0:00:25
polska_20	12	36	20	970.8	39.8	17.78	5/5	835.2	0:03:37
polska_30	12	36	30	3294.2	149.2	15.06	4/5	4818	1:14:15
polska_40	12	36	40	7596.8	388.4	16.43	1/5	17,778.6	4:02:08
nobel_us_10	14	42	10	123.6	0.4	21.69	5/5	54	0:00:01
nobel_us_15	14	42	15	795.2	26.4	26.98	5/5	3056.2	0:06:03
nobel_us_20	14	42	20	1609.2	37	26.15	5/5	6399.2	0:29:31
nobel_us_30	14	42	30	3566.8	62.4	22.35	4/5	8436.6	1:17:57
nobel_us_40	14	42	40	8723.6	129.8	*25.37*	0/5	14616	5:00:00
newyork_10	16	98	10	271.2	2.4	10.61	5/5	110.6	0:00:10
newyork_15	16	98	15	598	11.4	12.66	5/5	527.4	0:01:33
newyork_20	16	98	20	1993.6	28.2	14.09	5/5	3778.4	0:34:21
newyork_30	16	98	30	4683.4	101.2	*15.78*	0/5	17,894.6	5:00:00
newyork_40	16	98	40	8994.8	99.4	*60.53*	0/5	10,812.2	5:00:00
geant_10	22	72	10	155.6	0.6	16.65	5/5	35.6	0:00:03
geant_15	22	72	15	312.8	3	14.27	5/5	98.2	0:00:17
geant_20	22	72	20	353	1	13.11	5/5	98.2	0:00:23
geant_30	22	72	30	1496.6	21.6	12.34	5/5	686.8	0:07:15
geant_40	22	72	40	3111.2	37.2	12.60	5/5	2096.4	0:54:45
ta1_10	24	110	10	415.6	7.6	21.03	5/5	778	0:02:04
ta1_15	24	110	15	1120.4	78.4	27.32	4/5	8788.4	1:08:46
ta1_20	24	110	20	1920	49.4	25.25	3/5	10,870	2:45:12
ta1_30	24	110	30	4570	69.6	*25.88*	0/5	9886.8	5:00:00
ta1_40	24	110	40	9187.2	117.8	*52.85*	0/5	13,739.8	5:00:00
pioro_4	40	178	4	174.4	0	66.01	5/5	418.4	0:00:07
pioro_6	40	178	6	217.4	0.4	56.49	5/5	365.8	0:00:13
pioro_8	40	178	8	786.4	11.2	59.67	5/5	8678.8	0:11:42
pioro_10	40	178	10	884	9	51.72	5/5	5719.6	0:10:36
pioro_15	40	178	15	2471.4	82.8	56.60	2/5	46,315.2	3:51:36
pioro_20	40	178	20	3426.4	87.4	*55.55*	0/5	41,249.8	5:00:00

Table 1
Aggregated formulation versus Compact formulation.

				Compact formulation			AF (B&C) with msI				AF (B&C) with msI and msII				
Instance	|V|	|A|	|K|	Gap	Nodes	TT	Gap	Nodes	TT	NmsI	Gap	Nodes	TT	NmsI	NmsII
polska_10_1	12	36	10	14.04	919	0:02:56	12.28	71	0:03:28	384	12.28	71	0:03:28	384	0
polska_12_1	12	36	12	14.15	977	0:02:19	11.33	123	0:05:29	586	11.33	100	0:04:27	499	3
polska_14_1	12	36	14	*18.83*	83,500	5:00:00	18.69	298	0:04:58	606	18.69	269	0:09:46	593	2
polska_16_1	12	36	16	13.48	1241	0:03:56	12.16	244	0:11:52	853	12.16	239	0:11:30	839	1
polska_18_1	12	36	18	12.32	55,299	1:35:48	14.07	483	0:24:40	1402	14.07	428	0:22:31	1347	7
polska_20_1	12	36	20	*27.33*	16,777	5:00:00	23.10	563	0:21:50	687	23.10	645	0:24:57	773	18
nobel_us_10_1	14	42	10	*29.08*	82,768	5:00:00	29.02	110	0:02:31	155	29.02	110	0:02:31	155	0
nobel_us_12_1	14	42	12	*11.75*	71,534	5:00:00	28.90	324	0:09:37	312	28.90	302	0:09:43	344	10
nobel_us_14_1	14	42	14	*27.47*	55,221	5:00:00	22.49	267	0:10:54	538	22.49	302	0:12:44	556	14
nobel_us_16_1	14	42	16	*6.57*	74,429	5:00:00	20.27	305	0:11:49	522	20.27	521	0:19:49	676	9
nobel_us_18_1	14	42	18	*7.48*	64,164	5:00:00	22.34	1420	0:49:24	768	22.34	1001	0:40:04	757	27
nobel_us_20_1	14	42	20	*10.79*	13,291	5:00:00	22.91	430	0:17:57	505	22.91	295	0:12:38	434	6
newyork_10_1	16	98	10	12.65	1360	0:17:06	2.02	18	0:00:15	92	1.82	10	0:00:09	88	1
newyork_12_1	16	98	12	13.94	974	0:07:24	1.83	16	0:00:16	98	1.83	13	0:00:16	109	1
newyork_14_1	16	98	14	21.19	489	0:06:15	8.00	49	0:01:30	223	8.00	46	0:01:39	268	3
newyork_16_1	16	98	16	22.17	1859	0:16:00	5.96	37	0:01:04	214	5.96	186	0:05:53	412	5
newyork_18_1	16	98	18	21.65	5663	0:20:04	10.08	351	0:17:49	1030	10.08	336	0:15:04	857	9
newyork_20_1	16	98	20	*28.18*	8032	5:00:00	18.41	5707	4:55:58	2383	*18.94*	6049	5:00:00	2189	33
ta1_10_1	24	102	10	7.47	2084	0:14:10	6.83	55	0:01:58	282	6.83	55	0:01:58	282	0
ta1_12_1	24	102	12	8.88	11,453	0:32:03	22.89	272	0:08:37	431	22.89	320	0:09:35	380	2
ta1_14_1	24	102	14	*21.84*	25,186	5:00:00	29.65	985	00:41:31	796	29.65	898	0:38:29	770	2
ta1_16_1	24	102	16	*8.14*	23,252	5:00:00	26.94	3084	02:24:17	1417	26.94	3257	2:39:17	1399	8
ta1_18_1	24	102	18	*8.52*	20,941	5:00:00	25.48	3574	02:58:34	1740	25.48	4058	3:39:03	1674	21
ta1_20_1	24	102	20	*30.91*	12,088	5:00:00	*27.27*	5775	5:00:00	1952	*28.50*	6174	5:00:00	1530	26

Table 3
The effect of different demand distributions on Branch-and-Cut. Capacity $C = 100$.

Distribution	Instance	$\|V\|$	$\|A\|$	$\|K\|$	NmsI	NmsII	Gap(%)	FinalGap(%)	nodes	TT	TT(sep)
U(0,25)	Atlanta_10_1	15	44	10	173	0	50.66	0.00	582	0:00:06	<1
U(0,50)	Atlanta_10_2	15	44	10	130	5	32.81	0.00	128	0:00:2	<1
U(25,50)	Atlanta_10_3	15	44	10	170	2	25.51	0.00	178	0:00:05	<1
U(25,75)	Atlanta_10_4	15	44	10	139	0	19.19	0.00	74	0:00:01	<1
U(75,100)	Atlanta_10_5	15	44	10	62	0	4.09	0.00	14	0:00:01	<1
U(0,25)	Atlanta_20_1	15	44	20	711	1	43.22	0.00	2282	0:02:06	2
U(0,50)	Atlanta_20_2	15	44	20	1715	1910	*41.47*	*3.49*	76,342	5:00:00	50
U(25,50)	Atlanta_20_3	15	44	20	2287	1741	*35.86*	*10.24*	65,376	5:00:00	65
U(25,75)	Atlanta_20_4	15	44	20	1073	42	18.58	0.00	1059	0:03:09	3
U(75,100)	Atlanta_20_5	15	44	20	1293	0	5.96	0.00	153	0:00:37	1
U(0,25)	Atlanta_30_1	15	44	30	1954	1254	47.05	0.00	38,492	2:26:03	30
U(0,50)	Atlanta_30_2	15	44	30	2956	3242	*41.86*	*26.67*	39,359	5:00:00	53
U(25,50)	Atlanta_30_3	15	44	30	4497	2922	*34.18*	*21.00*	40,443	5:00:00	83
U(25,75)	Atlanta_30_4	15	44	30	4091	386	*16.23*	*1.14*	48,474	5:00:00	166
U(75,100)	Atlanta_30_5	15	44	30	1450	0	5.90	0.00	175	0:00:45	1
U(0,25)	Atlanta_40_1	15	44	40	2468	1592	*45.73*	*22.98*	46,076	5:00:00	42
U(0,50)	Atlanta_40_2	15	44	40	3084	3804	*43.42*	*30.00*	36,817	5:00:00	54
U(25,50)	Atlanta_40_3	15	44	40	5500	5263	*35.88*	*27.44*	29,548	5:00:00	80
U(25,75)	Atlanta_40_4	15	44	40	8361	166	*19.97*	*8.41*	32,859	5:00:00	117
U(75,100)	Atlanta_40_5	15	44	40	6488	0	5.58	0.00	927	0:12:22	11

Table 4
Branch-and-Cut results for SNDlib instances with realistic traffic.

Instance	$\|V\|$	$\|A\|$	$\|K\|$	NmsI	NmsII	Gap(%)	FinalGap(%)	Nodes	TT	TTsep
abilene	12	30	10	112	0	15.38	0.00	22	0:00:00	0
abilene	12	30	20	620	4	25.25	0.00	231	0:00:14	0
abilene	12	30	30	1801	9	20.48	0.00	1362	0:02:21	3
abilene	12	30	40	3994	576	19.13	0.00	10246	0:45:03	73
abilene	12	30	45	5781	451	16.86	0.00	10005	2:00:09	35
atlanta	15	44	10	76	1	4.61	0.00	27	0:00:00	0
atlanta	15	44	20	297	17	10.17	0.00	127	0:00:08	0
atlanta	15	44	30	1443	305	13.72	0.00	4145	0:13:23	13
atlanta	15	44	40	3771	484	*45.61*	*1.89*	15569	5:00:00	22
nobel_germany	17	52	10	37	1	1.52	0.00	8	0:00:00	0
nobel_germany	17	52	20	400	10	13.47	0.00	80	0:00:04	0
nobel_germany	17	52	30	325	18	33.06	0.00	345	0:00:19	0
nobel_germany	17	52	40	1088	130	30.57	0.00	232	0:45:03	6
nobel_germany	17	52	45	703	44	34.00	0.00	721	0:01:01	2
france	25	90	10	296	10	39.43	0.00	229	0:00:11	0
france	25	90	20	1223	137	27.80	0.00	4610	0:14:57	8
france	25	90	30	4585	951	*35.04*	*21.14*	26412	5:00:00	64
france	25	90	40	5763	1154	*33.06*	22.94	18025	5:00:00	69
france	25	90	45	7865	1497	*61.83*	*56.77*	18521	5:00:00	84
india35	35	160	10	234	1	39.05	0.00	93	0:00:09	0
india35	35	160	20	2349	286	*51.05*	*28.87*	12724	5:00:00	78
india35	35	160	30	2779	419	*75.71*	*69.81*	10017	5:00:00	106
india35	35	160	40	3402	665	*72.93*	*68.24*	6747	5:00:00	166
india35	35	160	45	3789	438	*66.2*	*61.37*	5879	5:00:00	171
cost266	37	102	10	168	10	36.40	0.00	106	0:00:04	0
cost266	37	102	20	414	751	37.67	0.00	4808	1:19:30	35
cost266	37	102	30	2523	646	*42.48*	*22.56*	9710	5:00:00	85
cost266	37	102	40	4224	689	*58.00*	*51.31*	6505	5:00:00	164
cost266	37	102	45	3599	794	*68.62*	*63.95*	6439	5:00:00	168
zib54	54	160	10	869	111	46.62	0.00	975	0:03:83	4
zib54	54	160	20	4050	913	*60.52*	*29.48*	17227	5:00:00	49
zib54	54	160	30	4816	565	*64.20*	*50.55*	11809	5:00:00	58
zib54	54	160	40	3264	246	*81.46*	*72.18*	10289	5:00:00	85
zib54	54	160	45	5245	367	*68.89*	*64.33*	8446	5:00:00	93

inequalities does not bring that much to the efficiency of the Branch-and-Cut algorithm. Actually, only a slight improvement can be noticed for the gap at the root node (see the instance *newyork*_10_1 where the gap value changes from 2.02 to 1.82 by allowing Min Set II separation). We can remark that the CPU time for computation, as well as the number of nodes in the Branch-and-Cut tree decreases significantly for some instances while adding Min Set II inequalities. For instance, *nobel_us*_18_1 is solved in 40 min instead of 49 min by using both Min Set I and Min Set II inequalities.

We give hereafter additional results, only for the Branch-and-Cut, on larger instances with both random and realistic demands. Table 2 reports results for random demand instances with graphs having from 12 up to 40 nodes and from 36 up to 178 arcs, while the number of commodities varies from 10 to 40 (4 to 20 for *pioro*). Note that the results in Table 1 were obtained in a subset of those instances. We can see from Table 2 that 19 among 31 families of instances have been solved to optimality within the fixed time limit (i.e., Opt=5/5). Also remark that for only 6 families of instances, the Branch-and-Cut could not provide any optimal solution within 5 h. It can be seen that the families that could be solved are those that have a reasonably small gap, in only 4 out of 19 it was greater than 30%. Unsolved instances have larger gaps, reaching 66% in instance *pioro_4*.

In order to analyse what, apart from instance size, affects the gap, we have made some experiments on a single network (*atlanta*, 15 nodes and 44 arcs), changing the number of commodities and the demand distribution. We considered five distributions. In the first case the demands are picked from U(0,25); in the second from U(0,50), in the third from U(25,50), the fourth from U(25,75), and finally, in the fifth all demands have value 100. The capacity C is always 100 in all these tests. An instance name will be followed by the extension 1, 2, 3, 4 or 5 in the table, according to the interval that contains its demands. The results are show in Table 3. This table also shows the final gaps, the difference between the global Branch-and-Cut lower bounds at the end of the run and the best known solution.

As expected, the instances where all demands are equal to the capacity are much easier. In that case, the UNACND becomes an integer multicommodity flow network problem. Apart from that case, we can observe the following order:

- The instances where all demands are small (from U(0,25)) are easier, in spite of the larger root gaps. We remark that in that case, the capacity that is wasted by the non-additivity is not likely be large, the UNACND solutions are closer to the UCND.
- The instances where most demands are large (from U(25,75)) are harder.
- The hardest instances are those with many medium-sized demands (from U(0,50) or U(25,50)).

Our last series of experiments were performed on the instances with realistic demands, taken from SNDlib. The results are given in Table 4. It can be seen that 19 among the 34 tested instances were solved to optimality within the fixed time limit, 15 of them within 15 min. The unsolved instances are generally those having more than 30 commodities and/or more than 35 nodes. We can remark that the root gap values are slightly better than those obtained for the instances with random demand. However, they seem to be equally challenging in terms of solvability.

In all experiments described in Tables 2–4, the CPU time spent by the separation procedure is not large, even when thousands of inequalities are generated. Indeed, using the good and fast lower bounds from [23] instead of exact methods for the bin packing subproblems (based on solving the ILP formulation) within the separation routines was important to keep them efficient. Note also that the number of generated Min Set I inequalities is significantly higher than the number of generated Min Set II inequalities. Although the separation procedure for Min Set II inequalities can be possibly enhanced, we do not expect them to be as much effective as Min Set I inequalities.

7. Concluding remarks

In this paper, we have considered the UNACND problem. We focused our attention on the arc-set polyhedron associated with this problem. Actually, we studied a more general family of polyhedra defined by unitary step monotonically increasing set functions. We investigated the basic properties of those polyhedra and derived new classes of valid inequalities. We then described necessary and sufficient conditions for these inequalities to define facets. By considering one of those functions, the Bin Packing set function, the resulting inequalities could be applied on the UNACND problem. We remarked that evaluating the Bin Packing Function is an NP-hard problem. Nevertheless, Min Set I and Min Set II inequalities could be efficiently separated in the Branch-and-Cut algorithm, by using fast and effective lower bounding procedures.

The generality of the Set Function Polyhedra opens opportunities for using quite similar algorithms in other CND variants. For example, consider the UNACND problem with the additional restriction that, for reliability reasons, certain commodities owned by the same client cannot pass by the same facility over a link. Those new arc restrictions correspond to a Bin Packing with Conflicts Set Function (see [29] for more details on the bin packing with conflicts). This function is still unitary step monotonically increasing. Therefore, Min Set I and Min Set II inequalities can still be used in a Branch-and-Cut for the new problem. The only difference is that other lower bounding procedures would have to be used in the separation.

Acknowledgments

Authors are very grateful to the anonymous referees for their comments on a previous version of the paper.

References

[1] Brockmüller B, Günlük O, Wolsey LA. Designing private line networks—polyhedral analysis and computation. CORE Discussion Paper 9647, Université Catholique de Louvain; 1996.
[2] van Hoesel SPM, Koster AMCA, van de Leensel RLMJ, Savelsbergh MWP. Polyhedral results for the edge capacity polytope. Math Program 2002;92(2):335–58.
[3] Atamtürk A, Rajan D. On splittable and unsplittable flow capacitated network design arc-set polyhedra. Math Program 2002;92(2):315–33.
[4] Benhamiche A, Mahjoub AR, Perrot N, Uchoa E. Capacitated network design using bin-packing. Electron Notes Discret Math 2013;41:479–86.
[5] Magnanti TL, Mirchandani P, Vachani R. The convex hull of two core capacitated network design problems. Math Program 1993;60:233–50.
[6] Pochet Y, Wolsey LA. Integer knapsack and flow covers with divisible coefficients: polyhedra, optimization and separation. Discret Appl Math 1995;59(1):57–74.
[7] Atamtürk A. On capacitated network design cut-set polyhedra. Math Program 2002;92(3):425–37.
[8] Bienstock D, Muratore G. Strong inequalities for capacitated survivable network design problems. Math Program 2000;89(1):127–47.
[9] Magnanti TL, Mirchandani P, Vachani R. Modeling and solving the two-facility capacitated network loading problem. Oper Res 1995;43:142–57.
[10] Bienstock D, Günlük O. Capacitated network design—polyhedral structure and computation. Inf J Comput 1994;8:243–59.
[11] Barahona F. Network design using cut inequalities. SIAM J Optim 1996;6(3):823–37.
[12] Chopra DBS, Günlük O, Tsai C-Y. Minimum cost capacity installation for multicommodity network flows. Math Program 1995;81:177–99.
[13] Costa A, Cordeau J-F, Gendron B. Benders, metric and cutset inequalities for multicommodity capacitated network design. Comput Optim Appl 2009;42(3):371–92.
[14] Frangioni A, Gendron B. 01 reformulations of the multicommodity capacitated network design problem. Discret Appl Math 2009;157(6):1229–41.
[15] Avella P, Mattia S, Sassano A. Metric inequalities and the network loading problem. Discret Optim 2007;4(1):103–14.
[16] Raack C, Koster AMCA, Orlowski S, Wessäly R. On cut-based inequalities for capacitated network design polyhedra. Networks 2011;57(2):141–56.
[17] Benders J. Partitioning procedures for solving mixed-variables programming problems. Numer Math 1962;4(1):238–52.
[18] Dahl AMG, Stoer M. Routing through virtual paths in layered telecommunication networks. Oper Res 1999;47(5):693–702.
[19] Fortz B, Poss M. An improved benders decomposition applied to a multi-layer network design problem. Oper Res Lett 2009;37(5):359–64.
[20] Mattia S. A polyhedral study of the capacity formulation of the multilayer network design problem. Networks.

[21] Martello S, Toth P. Knapsack problems: algorithms and computer implementations. New York: Wiley; 1990.
[22] Garey MR, Johnson DS. Computers and intractability: a guide to the theory of NP-Completeness. San Francisco: W.H. Freeman; 1979.
[23] Fekete SP, Schepers J. New classes of fast lower bounds for bin packing problems. Math Program 2001;91(1):11–31.
[24] Lueker GS. Bin packing with items uniformly distributed over intervals [a,b]. In: IEEE symposium on foundations of computer science; 1983. p. 289–297.
[25] Johnson DS. Near-optimal bin packing algorithms. Cambridge, MA: PhD Thesis, MIT; 1973.
[26] Clautiaux F, Alves C, de Carvalho JMV. A survey of dual-feasible and superadditive functions. Ann Oper Res 2010;179(1):317–42.
[27] Martello S, Toth P. Lower bounds and reduction procedures for the bin packing problem. Discret Appl Math 1990;28:59–70.
[28] ⟨http://sndlib.zib.de/home.action⟩.
[29] EGC Jr, Csirik J, Leung J. Variants of classical one-dimensional bin packing. In: Gonzalez TF, editor. Handbook of approximation algorithms and metaheuristics. Chapman & Hall/CRC; 2007 [chapter 33].