Mixed-Integer Nonlinear Programs featuring “on/off” constraints

Hassan Hijazi · Pierre Bonami · Gérard Cornuéjols · Adam Ouorou

July 14, 2011

Abstract In this paper, we study MINLPs featuring “on/off” constraints. An “on/off” constraint is a constraint $f(\mathbf{x}) \leq 0$ that is activated whenever a corresponding 0-1 variable is equal to 1. Our main result is an explicit characterization of the convex hull of the feasible region when the MINLP consists of simple bounds on the variables and one “on/off” constraint defined by an isotone function f. When extended to general convex MINLPs, we show that this result yields tight lower bounds compared to classical formulations. This allows us to introduce new models for the delay-constrained routing problem in telecommunications. Numerical results show gains in computing time of up to one order of magnitude compared to state-of-the-art approaches.

Keywords mixed-integer nonlinear programming, “on/off” constraints, disjunctive constraints, convex programming, delay-constrained routing problems.

1 Introduction

A very active area of research in recent years has been the study of Mixed-Integer Nonlinear Programs (MINLPs). A special case of interest is that of *convex MINLPs* where the objective function to minimize is convex and the feasible region obtained by dropping the integrality requirements on

First author is supported by Digiteo Emergence PASO, Digiteo Chair 2009-14D RMNCCO and Digiteo Emergence 2009-55D ARM. A major part of this work was accomplished while first author was working at Orange Labs R&D

Second author is supported by ANR grant ANR06-BLAN-0375 and by a Google Research Award.

Third author is supported by NSF grant CMMI1024554 and ONR grant N00014-03-1-0133.

H. Hijazi
LIX, École Polytechnique, F-91128 Palaiseau, France.

P. Bonami
LIF, CNRS-Aix Marseille Universite, Parc Scientifique et Technologique de Luminy, Marseille, France.

G. Cornuéjols
Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA 15213, USA.

A. Ouorou
Orange Labs R&D/CORE-TPN, 38-40 rue du Général Leclerc, 92794 Issy-Les-Moulineaux cedex 9, France.

the variables is a convex set. For such problems, several algorithms have been developed [10,20,24, 7] and have been implemented in solvers such as FilMINT [1] or Bonmin[7] which are able to solve problems of medium size. A challenge is to solve larger problems.

In this paper, we are considering convex MINLPs which feature certain specific structures. Given convex functions $g : \mathbb{R}^{n+K} \rightarrow (\mathbb{R} \cup \{\infty\})^m$, $h : \mathbb{R}^{n+K} \rightarrow \mathbb{R} \cup \{\infty\}$, $f^k : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$, $\forall k \in \{1,\ 2, ..., K\}$, and two vectors $\mathbf{l}$ and $\mathbf{u} \in \mathbb{R}^n$, we are interested in optimization problems of the form:

$$\begin{aligned} \min\ & h(\mathbf{x}, \mathbf{z}) \\ \text{s.t. } & g(\mathbf{x}, \mathbf{z}) \leq 0, \\ & f^k(\mathbf{x}) \leq 0 \text{ if } z_k = 1, \quad \forall k \in \{1,\ 2, ..., K\}, \\ & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \\ & \mathbf{x} \in \mathbb{R}^n,\ \mathbf{z} \in \{0,1\}^K. \end{aligned} \tag{1}$$

Our main object is to deal with the constraints $f^k(\mathbf{x}) \leq 0$ if $z_k = 1$. We call these constraints "on/off" because $f^k(\mathbf{x}) \leq 0$ is imposed to hold only when the corresponding indicator variable z_k is equal to 1. These constraints being not formulated using algebraic expressions, (1) can not be passed to algebraic modeling languages such as AMPL or GAMS. A simple way to formulate (1) would consist of transforming the "on/off" constraints into: $z_k f^k(\mathbf{x}) \leq 0,\ \forall k \in \{1,\ 2, ..., K\}$. In this case, even the continuous relaxation of (1) becomes non-convex. Since all variables are bounded, another alternative is to use a big-M formulation that leads to a compact convex continuous relaxation. Unfortunately, these models often provide weak lower bounds.

A third alternative is to use disjunctive programming (see [5,8,22,13]). For $k = 1, \ldots, K$, let us define $\Gamma_0^k = \{(\mathbf{x},\ z_k) \in \mathbb{R}^n \times \mathbb{B} : z_k = 0,\ \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}$ and $\Gamma_1^k = \{(\mathbf{x},\ z_k) \in \mathbb{R}^n \times \mathbb{B} : z_k = 1,\ f^k(\mathbf{x}) \leq 0,\ \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}$ ($\mathbb{B}$ denotes the set $\{0,1\}$). Problem (1) can be reformulated as a disjunctive program:

$$\begin{aligned} \min\ & h(\mathbf{x}, \mathbf{z}) \\ \text{s.t. } & g(\mathbf{x}, \mathbf{z}) \leq 0, \\ & (\mathbf{x}, z_k) \in \Gamma_0^k \cup \Gamma_1^k, \qquad \forall k \in \{1,\ 2, ..., K\}, \\ & \mathbf{x} \in \mathbb{R}^n,\ \mathbf{z} \in \{0,1\}^K. \end{aligned} \tag{2}$$

Taking the convex hull of each union $\Gamma_0^k \cup \Gamma_1^k$ separately, (2) can be rewritten as:

$$\begin{aligned} \min\ & h(\mathbf{x}, z) \\ \text{s.t. } & g(\mathbf{x}, \mathbf{z}) \leq 0, \\ & (\mathbf{x}, z_k) \in conv(\Gamma_0^k \cup \Gamma_1^k), \quad \forall k \in \{1,\ 2, ..., K\}, \\ & \mathbf{x} \in \mathbb{R}^n,\ \mathbf{z} \in \{0,1\}^K. \end{aligned} \tag{3}$$

The advantage of (3) is that the integrity requirement on $\mathbf{z}$ can be dropped to obtain a continuous relaxation that is convex and typically gives a good lower bound on the value of (1). The convex hull $conv\left(\Gamma_0^k \cup \Gamma_1^k\right)$ has an explicit description in an extended space using results of Stubbs and Mehrotra [22], Ceria and Soares [8], and Grossman and Lee [13]. However, since additional variables are introduced for each "on/off" constraint in the problem, the resulting formulation is very large and its continuous relaxation can be very difficult to solve. Convex formulations that do not require the introduction of additional variables are of great interest. In [14], Günlük and Linderoth suggest a formulation of $conv\left(\Gamma_0^k \cup \Gamma_1^k\right)$ defined in the original space of variables in the case where $\Gamma_0^k =$

$\{(\mathbf{x},\ z_k) \in \mathbb{R}^n \times \mathbb{B} : z_k = 0,\ \mathbf{x} = \mathbf{p}\}$, that is the set Γ_0^k is restricted to a single point. Under the same assumptions, Aktürk, Atamtürk and Gürel [2] have given a strong characterization of such convex hulls for a particular function used in machine scheduling problems.

In this work, we study the more general case where Γ_0^k is defined by finite bounds on the $\mathbf{x}$ variables. In Section 2, based on the work of [8], we start by studying extended formulations of $conv\left(\Gamma_0^k \cup \Gamma_1^k\right)$. Our main result is a characterization of $conv\left(\Gamma_0^k \cup \Gamma_1^k\right)$ in the space of original variables when the functions f^k satisfy a certain monotonicity property.

In Section 3, we study the application of this result to a problem in telecommunications introduced by Ben-Ameur and Ouorou [3]: the Delay-Constrained Routing Problem (DCRP). This model allows telecommunication operators to provide a "differentiated quality of service" by guaranteeing a maximum end-to-end transfer delay for each type of commodity. We propose new mathematical programming models for this problem based on the convex hull formulations introduced in Section 2. Finally, in Section 4, we report computational results with the different models we proposed for DCRP.

An extended abstract of this paper appeared in the proceedings of ISCO 2010 [15] and we use the same notation as there: given a set $\Gamma \in \mathbb{R}^n$, we denote by $cl(\Gamma)$ its topological closure and by $proj_{(x_1,\ldots,x_j)}(\Gamma)$ its projection onto the $(x_1, ..., x_j)$ space.

2 Convex Hull Of $\Gamma_0 \cup \Gamma_1$

We start by studying a simple example with one "on/off" constraint (i.e. K = 1) in $\mathbb{R}^3$:

$$\begin{aligned} &\min\ h(\mathbf{x}, z)\\ &\text{s.t. } (\mathbf{x}, z) \in conv\left(\Gamma_0 \cup \Gamma_1\right),\\ &\quad \Gamma_0 = \{\ (\mathbf{x},\ z) \in \mathbb{R}^2 \times \mathbb{B} : z = 0,\ l_1 \le x_1 \le u_1,\ l_2 \le x_2 \le u_2\},\\ &\quad \Gamma_1 = \{\ (\mathbf{x},\ z) \in \mathbb{R}^2 \times \mathbb{B} : z = 1,\ f(\mathbf{x}) \le 0,\ l_1 \le x_1 \le u_1,\ l_2 \le x_2 \le u_2\}, \end{aligned} \tag{4}$$

where $f(\mathbf{x}) = \frac{1}{c_1 - x_1} + \frac{1}{c_2 - x_2} - d$, $u_1 \le c_1$ and $u_2 \le c_2$.

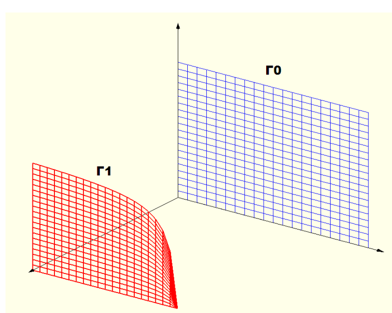

Fig. 1 The sets Γ_0 and Γ_1.

Figure 1 gives a representation of the sets Γ_0 and Γ_1 in $\mathbb{R}^3$. Γ_0 is the rectangle on the right hand side of the figure and Γ_1 is the convex set on the left hand side.

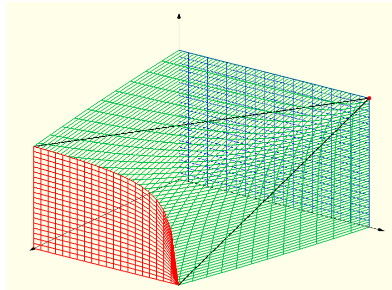

Fig. 2 conv$(\mathit{\Gamma}_0 \cup \mathit{\Gamma}_1)$.

Figure 2 gives a representation of $conv(\mathit{\Gamma}_0 \cup \mathit{\Gamma}_1)$ in $\mathbb{R}^3$. An interesting remark is that this convex hull can be derived using *perspective* functions. The perspective function $\tilde{f} : \mathbb{R}^{n+1} \rightarrow \mathbb{R} \cup \{+\infty\}$ of f is defined by: $\tilde{f}(\mathbf{x}, z) \equiv \begin{cases} zf(\mathbf{x}/z) & \text{if } z > 0, \\ +\infty & \text{if } z \leq 0. \end{cases}$

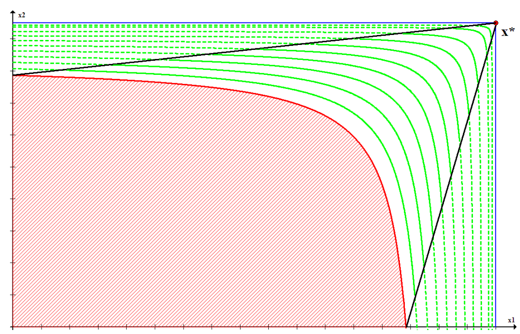

Fig. 3 Level curves of the perspective function $\tilde{f}(\mathbf{x} - (1-z)\mathbf{x}^*, z)$ in $\mathbb{R}^3$.

In our example, the nonlinear constraint needed to describe $conv\,(\mathit{\Gamma}_0 \cup \mathit{\Gamma}_1)$ is obtained by taking the perspective function of f from the point $\mathbf{x}^* = (u_1, u_2)$. Figure 3 plots the level curves of $\tilde{f}(\mathbf{x} - (1-z)\mathbf{x}^*, z)$ in $\mathbb{R}^3$. Note that $conv\,(\mathit{\Gamma}_0 \cup \mathit{\Gamma}_1)$ can be obtained by using the perspective function when $z > 0$ and taking a topological closure.

For this specific example, $conv\,(\mathit{\Gamma}_0 \cup \mathit{\Gamma}_1)$ is written as follows:

$$conv\,(\mathit{\Gamma}_0 \cup \mathit{\Gamma}_1) = cl \left\{ \begin{array}{l} (\mathbf{x},\ z) \in \mathbb{R}^2 \times]0,1] : \\ \tilde{f}(\mathbf{x} - (1-z)\mathbf{x^*},\ z) \leq 0 \\ l_1 \leq x_1 \leq u_1,\ l_2 \leq x_2 \leq u_2. \end{array} \right\}$$

$$= cl \left\{ \begin{array}{l} (\mathbf{x},\ z) \in \mathbb{R}^2 \times]0,1] : \\ \frac{z}{zc_1 - x_1 + (1-z)u_1} + \frac{z}{zc_2 - x_2 + (1-z)u_2} \leq d \\ l_1 \leq x_1 \leq u_1,\ l_2 \leq x_2 \leq u_2. \end{array} \right\}.$$

We now return to the general case dealing with an "on/off" constraint in $\mathbb{R}^n$. First, we recall a result of Ceria and Soares [8] characterizing the convex hull of a union of closed convex sets in an extended space. Consider a closed convex set $P \subseteq \mathbb{R}^n$ defined by $P \equiv cl\ conv(K),\ K \equiv \bigcup_{i=1}^{p} K^i$, where every set K^i is a closed convex set having the representation $K^i \equiv \{\mathbf{x} \in \mathbb{R}^n : G^i(\mathbf{x}) \leq 0\}$, and $G^i : \mathbb{R}^n \to \mathbb{R}^m$ is a vector mapping whose components are closed convex functions. The following theorem gives a characterization of P (note that the statement is slightly different from [8] but both statements are equivalent).

Theorem 1 **([8])** *Let $I \equiv \{i : K^i \neq \emptyset\}$. If the set K is bounded below or above, then*

$$P \equiv cl \left\{ x \in \mathbb{R}^n : \forall i \in I,\ \exists \lambda_i > 0 \ and\ \mathbf{x}^i \in \mathbb{R}^n \ with\ \mathbf{x} = \sum_{i \in I} \mathbf{x}^i,\ \sum_{i \in I} \lambda_i = 1,\ \tilde{G}^i(\mathbf{x}^i, \lambda_i) \leq 0,\ i \in I \right\}.$$

Let us emphasize that the above formulation introduces $(2n+4)$ variables to convexify one "on/off" constraint ($i = 2$ in our case), thus a total of $|K|(2n+4)$ variables must be added to convexify all the disjunctive constraints in (2). Therefore reducing the space dimension can have a very important impact when dealing with large optimization problems including many "on/off" constraints. In the following lemma, for our case of interest, we show that the convex hull formulation corresponding to one "on/off" constraint can be obtained by adding only n new variables.

Lemma 1 **([15])** *Let $f : E \to \mathbb{R},\ E \subseteq \mathbb{R}^n$, be a closed convex function,*

$$\begin{aligned} \mathit{\Gamma}_0 &= \{\ (\mathbf{x},\ z) \in \mathbb{R}^n \times \mathbb{B} : z = 0,\ \mathbf{l}^0 \leq x \leq \mathbf{u}^0\}, \\ \mathit{\Gamma}_1 &= \{\ (\mathbf{x},\ z) \in \mathbb{R}^n \times \mathbb{B} : z = 1,\ f(\mathbf{x}) \leq 0,\ \mathbf{l}^1 \leq \mathbf{x} \leq \mathbf{u}^1\} \neq \emptyset. \end{aligned}$$

Then $\boldsymbol{conv\,(\mathit{\Gamma}_0 \cup \mathit{\Gamma}_1) = pr_{(\mathbf{x},z)} cl(\mathit{\Gamma})}$*, with*

$$\mathit{\Gamma} = \left\{ \begin{array}{l} (\mathbf{x}, \mathbf{y}, z) \in \mathbb{R}^{2n+1} : \\ z f\,(\mathbf{y}/z) \leq 0, \\ \mathbf{x} - (1-z)\mathbf{u}^0 \leq \mathbf{y} \leq \mathbf{x} - (1-z)\mathbf{l}^0, \\ z\mathbf{l}^1 \leq \mathbf{y} \leq z\mathbf{u}^1, \\ 0 < z \leq 1. \end{array} \right\}$$

Proof. *Theorem 1 in [8], allows to write the exact formulation of $conv(\Gamma_0 \cup \Gamma_1)$ as follows: $conv(\Gamma_0 \cup \Gamma_1) = pr_{(\mathbf{x},z)} cl(\Gamma)$, where*

$$\Gamma = \left\{ \begin{array}{l} (\mathbf{x}, z, \lambda_0, \lambda_1, z_0, z_1, \mathbf{x}^0, \mathbf{x}^1) \in \mathbb{R}^{3n+5} : \\ \mathbf{x} = \mathbf{x}^0 + \mathbf{x}^1, \\ z = z_0 + z_1, \\ \lambda_0 + \lambda_1 = 1, \\ \tilde{f}\left(\mathbf{x}^1, \lambda_1\right) \leq 0, \\ \mathbf{l}^0 \lambda_0 \leq \mathbf{x}^0 \leq \mathbf{u}^0 \lambda_0, \\ \mathbf{l}^1 \lambda_1 \leq \mathbf{x}^1 \leq \mathbf{u}^1 \lambda_1, \\ z_0 = 0, \\ z_1 = \lambda_1, \\ 0 < \lambda^1, \ 0 \leq \lambda^0. \end{array} \right\} \equiv \left\{ \begin{array}{l} (\mathbf{x}, z, \lambda_0, \mathbf{x}^0, \mathbf{x}^1) \in \mathbb{R}^{3n+2} : \\ \mathbf{x} = \mathbf{x}^0 + \mathbf{x}^1, \\ \lambda_0 + z = 1, \\ \tilde{f}\left(\mathbf{x}^1, z\right) \leq 0, \\ \mathbf{l}^0 \lambda_0 \leq \mathbf{x}^0 \leq \mathbf{u}^0 \lambda_0, \\ \mathbf{l}^1 z \leq \mathbf{x}^1 \leq \mathbf{u}^1 z, \\ 0 \leq \lambda_0. \\ 0 < z. \end{array} \right\}.$$

Substituting $\mathbf{x}^0 = \mathbf{x} - \mathbf{x}^1$ and $\lambda_0 = 1 - z$, we obtain:

$$\Gamma \equiv \left\{ \begin{array}{l} (\mathbf{x}, z, \mathbf{x}^1) \in \mathbb{R}^{2n+1} : \\ \tilde{f}\left(\mathbf{x}^1, z\right) \leq 0, \\ \mathbf{l}^0 \lambda_0 \leq \mathbf{x} - \mathbf{x}^1 \leq \mathbf{u}^0 \lambda_0, \\ \mathbf{l}^1 z \leq \mathbf{x}^1 \leq \mathbf{u}^1 z, \\ 0 \leq 1 - z. \\ 0 < z. \end{array} \right\} \equiv \left\{ \begin{array}{l} (\mathbf{x}, \mathbf{y}, z) \in \mathbb{R}^{2n+1} : \\ \tilde{f}\left(\mathbf{y}, z\right) \leq 0, \\ \mathbf{x} - (1-z)\mathbf{u}^0 \leq \mathbf{y} \leq \mathbf{x} - (1-z)\mathbf{l}^0, \\ z\mathbf{l}^1 \leq \mathbf{y} \leq z\mathbf{u}^1, \\ 0 < z \leq 1. \end{array} \right\}.$$

□

Lemma 1 shows that only n additional variables are needed to explicitly describe the convex hull. One can notice that these variables appear in the perspective function of f and the constraints: $\mathbf{x} - (1-z)\mathbf{u}^0 \leq \mathbf{y} \leq \mathbf{x} - (1-z)\mathbf{l}^0$ and $z\mathbf{l}^1 \leq \mathbf{y} \leq z\mathbf{u}^1$. We observe that, if we consider only the last two sets of constraints, Fourier-Motzkin elimination can be applied in a straightforward way to eliminate $\mathbf{y}$ leading to the constraints $z\mathbf{l}^1 + (1-z)\mathbf{l}^0 \leq \mathbf{x} \leq z\mathbf{u}^1 + (1-z)\mathbf{u}^0$. Nevertheless, the projection becomes harder to describe when the nonlinear constraint in Γ is taken into account.

Next, we show how the $\mathbf{y}$ variables can be projected out in the case where the function f is isotone or order preserving (see definition below).

Definition 1 Let $f : E \to \mathbb{R}, \ E \subseteq \mathbb{R}^n$.

- f is *independently increasing* (resp. *decreasing*) on coordinate i if for all $\mathbf{x} \in dom(f)$ and $\lambda > 0$ such that $\mathbf{x}+\lambda e_i \in dom(f)$, where e_i is ith unit vector of the standard basis, we have $f(\mathbf{x}+\lambda e_i) \geq f(\mathbf{x})$ (resp. $f(\mathbf{x} + \lambda e_i) \leq f(\mathbf{x})$).
- We say that f is *independently monotone* on coordinate i if it is independently increasing or independently decreasing on the ith coordinate.

– f is *isotone* if it is independently monotone on every coordinate.

Example 1 Consider the following functions:

1. $f(x_1, x_2, x_3) = e^{(2x_1 - x_2)} + x_3$, $(x_1, x_2, x_3) \in \mathbb{R}^3$, f is independently increasing on coordinate 1 and 3, independently decreasing on coordinate 2, therefore it is an isotone function.
2. $f(x, y) = x^4 + y^2$, $(x, y) \in \mathbb{R}^2$, the variation of f depends on the sign of the variables, f is not an isotone function.
3. $f(\mathbf{x}) = \sum_{i=1}^{n} \frac{1}{c_i - x_i}$, where $x_i \in]-\infty, c_i]$ for $i = 1, \ldots, n$. Since f is a sum of one-variable increasing functions, it is an isotone function.

Additively separable functions which are sums of one-variable monotone functions are a commonly encountered case of isotone functions.

Theorem 2 ([15]) *Let $f : E \to \mathbb{R}$, $E \subseteq \mathbb{R}^n$, be an isotone closed convex function, with J^1 (resp. J^2) the set of indices on which f is independently increasing (resp. decreasing),*

$$\Gamma_0 = \{\ (\mathbf{x},\ z) \in \mathbb{R}^n \times \mathbb{B} : z = 0,\ \mathbf{l}^0 \le \mathbf{x} \le \mathbf{u}^0\},$$
$$\Gamma_1 = \{\ (\mathbf{x},\ z) \in \mathbb{R}^n \times \mathbb{B} : z = 1,\ f(\mathbf{x}) \le 0,\ \mathbf{l}^1 \le \mathbf{x} \le \mathbf{u}^1\} \ne \emptyset.$$

Then $\boldsymbol{conv}\ (\boldsymbol{\Gamma_0} \cup \boldsymbol{\Gamma_1}) = \boldsymbol{cl(\Gamma^{'})}$, *where*

$$\Gamma^{'} = \left\{ \begin{array}{ll} (\mathbf{x}, z) \in \mathbb{R}^{n+1} : & \\ z q_S\left(\mathbf{x}/z\right) \le 0, & \forall S \subset \{1, 2, ..., n\}, \\ z\mathbf{l}^1 + (1-z)\mathbf{l}^0 \le \mathbf{x} \le z\mathbf{u}^1 + (1-z)\mathbf{u}^0, & \\ 0 < z \le 1 & \end{array} \right\},$$

$q_S = (f \circ h_S)$, and $h_S(\mathbb{R}^n \times]0,1] \to \mathbb{R}^n)$ is defined by

$$(h_S(\mathbf{x}, z))_i = \begin{cases} l_i^1 & \forall i \in S \cap J_1, \\ u_i^1 & \forall i \in S \cap J_2, \\ x_i - \frac{(1-z)u_i^0}{z} & \forall i \in J_1, i \notin S, \\ x_i - \frac{(1-z)l_i^0}{z} & \forall i \in J_2, i \notin S. \end{cases}$$

Proof. *We prove that $cl(\Gamma^{'})$ is the projection onto the $(\mathbf{x}, z)$ space of $cl(\Gamma)$, the set defined in Lemma 1.*

1. *We show that $(\mathbf{x}, z) \in cl(\Gamma^{'}) \Rightarrow \exists \mathbf{y} \in \mathbb{R}^n$ s.t. $(\mathbf{x}, \mathbf{y}, z) \in cl(\Gamma)$.*

   For any given point $(\mathbf{x}, z) \in \Gamma^{'}$, $z \ne 0$, let $\mathbf{y} \in \mathbb{R}^n$ be defined as follows:

   $$y_i = \max\{z l_i^1,\ x_i - (1-z)u_i^0\},\ \forall i \in J^1 \text{ and } y_i = \min\{z u_i^1,\ x_i - (1-z)l_i^0\},\ \forall i \in J^2.$$

   One can see that $\exists S \subset \{1, 2, ..., n\}$ such that $h_S(\mathbf{x}/z) = \mathbf{y}/z$. Having $z q_S\left(\mathbf{x}/z\right) = z f\left(h_S(\mathbf{x}/z)\right) \le 0$ in $\Gamma^{'}$, we deduce that $z f(\mathbf{y}/z) \le 0$. All other constraints in Γ are satisfied by definition, leading to $(\mathbf{x}, \mathbf{y}, z) \in \Gamma$. Now consider the remaining points $(\mathbf{x}, 0) \in cl(\Gamma^{'})$. There exists a sequence of points $(\mathbf{x}^k, z^k) \in \Gamma^{'}$ such that $\lim_{k\to\infty}(\mathbf{x}^k, z^k) = (\mathbf{x}, 0)$. Defining $\mathbf{y}^k = \mathbf{y}\ \forall k \in \mathbb{N}$, we immediately get $(\mathbf{x}^k, \mathbf{y}^k, z^k) \in \Gamma$ and $\lim_{k\to\infty}(\mathbf{x}^k, \mathbf{y}^k, z^k) = (\mathbf{x}, \mathbf{y}, 0)$. This proves that $(\mathbf{x}, \mathbf{y}, 0) \in cl(\Gamma)$.

2. *We show that* $(\mathbf{x},\mathbf{y},z)\in cl(\Gamma)\Rightarrow(\mathbf{x},z)\in cl(\Gamma^{'})$.

   Let $(\mathbf{x},\mathbf{y},z)$ *be a point in* Γ $(z\neq 0)$*. By definition of* Γ *and functions* $h_S(\mathbf{x})$*, we have* $\forall S\subset\{1,2,...,n\}$
$$\frac{y_i}{z}\geq\max\left\{l_i^1,\ \frac{x_i}{z}-\frac{(1-z)u_i^0}{z}\right\}\geq(h_S(\mathbf{x}/z))_i,\ \forall i\in J^1$$
   and
$$\frac{y_i}{z}\leq\min\left\{u_i^1,\ \frac{x_i}{z}-\frac{(1-z)l_i^0}{z}\right\}\leq(h_S(\mathbf{x}/z))_i,\ \forall i\in J^2.$$
   f *being an isotone function we have* $zf(h_S(\mathbf{x}/z))\leq zf(\mathbf{y}/z)\leq 0,\ \forall S\subset\{1,2,...,n\}$*. Finally, we notice that the constraints* $z\mathbf{l}^1+(1-z)\mathbf{l}^0\leq\mathbf{x}\leq z\mathbf{u}^1+(1-z)\mathbf{u}^0$ *are obtained by composing the last two set of constraints defining set* Γ*.*
   The extension to the closure is trivial.

□

Now, we are able to describe $conv\,(\Gamma_0\cup\Gamma_1)$ in (3) without having to deal with additional variables. Note that this formulation might include an exponential number of constraints. In the next corollary, we show that only one constraint for each disjunction is sufficient to build a valid formulation of our original MINLP defined in (2).

Corollary 1 *Let* $f:E\to\mathbb{R},\ E\subseteq\mathbb{R}^n$*, be an isotone closed convex function with* J^1 *(resp.,* J^2*) the set of indices on which* f *is independently increasing (resp. decreasing),*

$$\Gamma_0=\{\ (\mathbf{x},\ z)\in\mathbb{R}^n\times\mathbb{B}:z=0,\ \mathbf{l}^0\leq\mathbf{x}\leq\mathbf{u}^0\},$$
$$\Gamma_1=\{\ (\mathbf{x},\ z)\in\mathbb{R}^n\times\mathbb{B}:z=1,\ h(\mathbf{x})\leq 0,\ \mathbf{l}^1\leq\mathbf{x}\leq\mathbf{u}^1\}\neq\emptyset,$$
$$\Gamma''=\left\{\begin{array}{l}(\mathbf{x},z)\in\mathbb{R}^{n+1}:\\ zq_\emptyset\,(\mathbf{x},z)\leq 0,\\ z\mathbf{l}^1+(1-z)\mathbf{l}^0\leq\mathbf{x}\leq z\mathbf{u}^1+(1-z)\mathbf{u}^0,\\ 0<z\leq 1\end{array}\right\},$$

with $q_\emptyset=(f\circ h_\emptyset)$ *and* $h_\emptyset(\mathbb{R}^n\times]0,1]\to\mathbb{R}^n)$ *defined as*

$$(h_\emptyset(\mathbf{x},z))_i=\begin{cases}\frac{x_i-(1-z)u_i^0}{z} & \forall i\in J_1,\\ \frac{x_i-(1-z)l_i^0}{z} & \forall i\in J_2.\end{cases}$$

Then

1. $cl\,(\boldsymbol{\Gamma''})$ ***is a valid convex relaxation for conv***$(\boldsymbol{\Gamma_0}\cup\boldsymbol{\Gamma_1})$*;*
2. $cl\,(\boldsymbol{\Gamma''})\cap\{\boldsymbol{z}\in\{\mathbf{0},\mathbf{1}\}\}\equiv\boldsymbol{\Gamma_0}\cup\boldsymbol{\Gamma_1}$*.*

Proof.

1. $cl(\Gamma'')$ *is a valid convex relaxation of* $conv(\Gamma_0\cup\Gamma_1)$ *since it only contains a subset of the constraints defining the convex hull in Theorem 2.*

2. *For $z=1$, one can check that $\Gamma''\cap\{(\mathbf{x},z)\colon z=1\}\equiv\Gamma_1$. For $z=0$, we have $(\Gamma_0\cup\Gamma_1)\subseteq cl\,(\Gamma'')$ since $cl\,(\Gamma'')$ is a valid convex relaxation of $\Gamma_0\cup\Gamma_1$. Therefore intersecting with $\{(\mathbf{x},z):z=0\}$ gives $(\Gamma_0\cup\Gamma_1)\cap\{(\mathbf{x},z):z=0\}=\Gamma_0\subseteq cl\,(\Gamma'')\cap\{(\mathbf{x},z):z=0\}$. On the other hand, by definition of Γ'', when $z=0$, all the constraints of Γ_0 are satisfied in $cl\,(\Gamma'')$, that is $cl\,(\Gamma'')\cap\{(\mathbf{x},z):z=0\}\subseteq\Gamma_0$.*

□

2.1 Relationships To Previous Theoretical Results

Balas [4,5] was the first to introduce the explicit algebraic formulation of the convex hull of a union of polyhedra in a higher dimensional space. Generalizations and extensions have been established for unions of nonlinear convex sets [8,13,22]. Günlük and Linderoth [14] have characterized the convex hull of the union of a point and a convex set in the space of original variables. We show that this characterization is a special case of Lemma 1, obtained by fixing $\mathbf{l}^0=\mathbf{u}^0=0$ in the definition of this lemma:

$$conv(\Gamma_0\cup\Gamma_1)=cl\left\{\begin{array}{l}(\mathbf{x},\mathbf{y},z)\in\mathbb{R}^{2n+1}:\\ zf\left(\mathbf{y}/z\right)\le 0,\\ \mathbf{x}\le\mathbf{y}\le\mathbf{x},\\ z\mathbf{l}^1\le\mathbf{y}\le z\mathbf{u}^1,\\ 0<z\le 1.\end{array}\right\}\equiv cl\left\{\begin{array}{l}(\mathbf{x},z)\in\mathbb{R}^{n+1}:\\ zf\left(\mathbf{x}/z\right)\le 0,\\ z\mathbf{l}^1\le\mathbf{x}\le z\mathbf{u}^1,\\ 0<z\le 1.\end{array}\right\}.$$

which corresponds exactly to the result established in [14].

Technical issues. Let us note that some technical issues must be considered due to the fact that the formulations introduced previously involve topological closures of sets. In practice, when implementing perspective functions, one must avoid dividing by zero (for $z=0$). A first alternative is to use cutting plane methods. Instead of explicitly writing the convex hull formulas in the original model, valid cuts are generated to strengthen the formulation (see [8,22,12] for more details). A second alternative proposed in [13,20,10], is to approximate the constraints by adding epsilon values to the corresponding functions. We show in Section 3 that, for our application of interest, these difficulties can be avoided while still guaranteeing exact convex MINLP models. In the next section, we introduce our main application and use the above results in order to improve existing formulations.

3 Application: The Delay-Constrained Routing Problem

In this section, we study the application of the results obtained in Section 2 to a problem in telecommunications first introduced by Ben-Ameur and Ouorou [3] and revisited in [15]: the delay-constrained routing problem.

The constant rise of traffic in telecommunication induce that networks (or certain parts of them) become congested. One of the main consequences of this congestion is a delay in the communications that go through the congested parts of the network. Parallel to this congestion is the introduction of more and more real time services (voice on IP, video on demand, gaming,...) that can only operate properly if the delay of the communications is controlled. Managing these delays is therefore an important question. A common approach is to study routing problems under average end-to-end delay constraints, see [6,21,18]. Unfortunately, this approach is not adequate with real time services since it ignores the heterogeneous nature of real world networks. These services rather require the consideration of individual source-to-destination delays (i.e., packet delay from the source to all destinations) and need to be differentiated from delay-tolerant services. The delay-constrained routing model we consider takes into account an end-to-end delay guarantee for each type of service. This application can be formulated as a mixed-integer nonlinear program including "on/off" constraints.

3.1 Mathematical Models

Let G be a finite directed network,

1. Parameters:
   (a) V represents the set of vertices, E the set of arcs and K the set of commodities.
   (b) For each commodity $k \in K$, $v_k \in \mathbb{R}$ is the quantity of demand that needs to be routed and $\alpha_k \in \mathbb{R}$ the maximum delay.
   (c) Each commodity has a set of candidate paths denoted by $P(k) = \{P_k^1, P_k^2, ..., P_k^{n_k}\}$, each one of these corresponding to a different routing for commodity k. N represents the maximum authorized number of activated paths per commodity.
   (d) For each arc e, $c_e \in \mathbb{R}$ represents its capacity and $w_e \in \mathbb{R}$ its cost.
2. Variables:
   (a) We call ϕ_k^i the fraction of the k_{th} demand carried by its i_{th} path, $\phi_k^i \in [0, 1]$.
   (b) z_k^i is a binary variable indicating if path P_k^i is activated.
   (c) x_e denotes the total amount of flow crossing over arc e, $x_e \in \mathbb{R}$.

Initial mathematical model (P). The objective function (5) is to minimize the total routing cost on all used links. For each commodity k, constraint (6) ensures that the total flow routed is greater than 1 in order to satisfy demand. Constraints (7) define the variables x_e on each arc as the sum of all the flows passing through e. In (8), x_e is bounded by the capacity installed on the link. Constraint (9) represents the main "on/off" delay constraints: the delay guarantee associated to a given commodity must be satisfied on its candidate path if the latter is activated. As mentioned above, this model allows to fix a maximum number of active paths per commodity. This is established in (10). In (11), we link the indicator variables z_k^i to the ϕ_k^i variables. Finally, bounds on all variables are introduced in (12-14). Let us point out that if $N = 1$, i.e. only one path can be activated per commodity (mono-routing), the variables ϕ_k^i become redundant and can be replaced by the z_k^i variables. Ben-Ameur and Ouorou showed in [3] that as soon as one considers two candidate paths per commodity, the underlying feasibility problem (ignoring the objective function) is NP-complete.

$$
\begin{aligned}
\min \quad & \sum_{e \in E} w_e x_e && && (5)\\
\text{s.t.} \quad & \sum_{i=1}^{n_k} \phi_k^i \geq 1, && \forall k \in K, && (6)\\
& \sum_{k \in K} \sum_{P_k^i \ni l} \phi_k^i v_k \leq x_e, && \forall e \in E, && (7)\\
& x_e \leq c_e, \ \forall e \in E && && (8)\\
& \sum_{e \in P_k^i} \frac{1}{c_e - x_e} \leq \alpha_k, && \forall k \in K, \ \forall P_k^i \in P(k) \text{ if } z_k^i = 1, && (9)\\
& \sum_{P_k^i \in P(k)} z_k^i \leq N, && \forall k \in K, && (10)\\
& \phi_k^i \leq z_k^i, && \forall k \in K, \ \forall P_k^i \in P(k), && (11)\\
& z_k^i \in \{0,1\}, && \forall k \in K, \ \forall P_k^i \in P(k), && (12)\\
& \phi_k^i \in [0,1], && \forall k \in K, \ \forall P_k^i \in P(k), && (13)\\
& x_e \in \mathbb{R}, && \forall e \in E. && (14)
\end{aligned}
$$

The continuous relaxation of this model is obviously non-convex due to the presence of "on/off" constraints in (9). We will next introduce four different convex models equivalent to (3) and offering each a different continuous relaxation.

Big-M relaxation: (P_{big-M}). A classical convex relaxation of constraint (9) is the big-M relaxation:

$$
\begin{aligned}
\min \quad & \sum_{e \in E} w_e x_e \\
\text{s.t.} \quad & (6),\ (7),\ (8),\ (10),\ (11),\ (12-14),\\
& \sum_{e \in P_k^i} \frac{1}{c_e - x_e} \leq M - z_k^i (M - \alpha_k), \ \forall k \in K, \ \forall P_k^i \in P(k). \qquad \text{(9-a)}
\end{aligned}
$$

This formulation is exact if z_k^i is a binary variable, provided that the constant M is big enough. When $z_k^i = 0$, the constraint (9-a) is redundant, due to the big-M quantity on its right-hand-side; when $z_k^i = 1$, the big-M disappears leading to the original delay constraint formula. Since the validity of constraint (9-a) and the quality of the bound given by this formulation depends on the constant M, one has to compute it accurately. Ben-Ameur and Ouorou [3] pointed out that the flow on a given arc e always admits an upper bound u_e verifying $u_e < c_e$. If a link e is used in an activated path P_k^i, then one can write $\frac{1}{c_e - x_e} \leq \alpha_k - \sum_{e' \neq e} \frac{1}{c_{e'}}$. Based on these observations, one can easily deduce an upper bound α_e for the delay on each arc and therefore obtain an upper bound on the total delay generated on any given path. In other words, the big M constant can be replaced by $\alpha_k^i = \sum_{e \in P_k^i} \alpha_e$.

Higher space convex hull relaxation model (P_{high}). First, we present a model based on the state of the art in disjunctive programming using Theorem 1 (introduced in [8]).

$$\begin{aligned}
\min \ & \sum_{e\in E} w_e x_e \\
\text{s.t. } & (6),\ (7),\ (8),\ (10),\ (11),\ (12-14), \\
& \sum_{e\in P_k^i} \left(\frac{{z_k^i}^2}{z_k^i c_e - \lambda_e^{(1,i,k)}} \right) - z_k^i \alpha_k \le 0, && \forall k \in K,\ \forall P_k^i \in P(k), \\
& x_e = \lambda_e^{(0,i,k)} + \lambda_e^{(1,i,k)}, && \forall k \in K,\ \forall P_k^i \in P(k),\ \forall e \in P_k^i, \\
& 0 \le \lambda_e^{(0,i,k)} \le (1 - z_k^i) u_e^0, && \forall k \in K,\ \forall P_k^i \in P(k),\ \forall e \in P_k^i, \\
& 0 \le \lambda_e^{(1,i,k)} \le z_k^i u_e^1, && \forall k \in K,\ \forall P_k^i \in P(k),\ \forall e \in P_k^i.
\end{aligned}$$

Next, we use the results of Section 2 to introduce new formulations. To every path P_k^i corresponds a delay constraint (9) in (P) that is written: $f^{(i,k)}(\mathbf{x}) \le 0$ if $z_k^i = 1$, with $f^{(i,k)} : \mathbb{R}^n \to \mathbb{R}$, $f^{(i,k)}(\mathbf{x}) = \sum_{e\in P_k^i} \frac{1}{c_e - x_e} - \alpha_k$. The functions $f^{(i,k)}$ being closed convex functions, Lemma 1 applies leading to the following corollary.

Corollary 2 *Let*

$$\begin{aligned}
& f : \mathbb{R}_+^n \to \mathbb{R},\ f(\mathbf{x}) = \sum_{i=1}^{n} \left(\tfrac{1}{c_i - x_i} \right) - b,\ b \ge 0, \\
& \Gamma_0 = \{\ (\mathbf{x},\ z) \in \mathbb{R}^n \times \mathbb{B} : z = 0,\ \mathbf{0} \le \mathbf{x} \le \mathbf{u}\}, \\
& \Gamma_1 = \{\ (\mathbf{x},\ z) \in \mathbb{R}^n \times \mathbb{B} : z = 1,\ f(\mathbf{x}) \le 0, \mathbf{l} \le \mathbf{x} \le \mathbf{u}\} \textit{ non empty.}
\end{aligned}$$

Then $\boldsymbol{conv}\,(\boldsymbol{\Gamma_0} \cup \boldsymbol{\Gamma_1}) = \{(\mathbf{x}, z) \in \mathbb{R}^{n+1} \mid \exists \mathbf{y} \in \mathbb{R}^n \textit{ with } (\mathbf{x}, \mathbf{y}, z) \in \boldsymbol{cl(\Gamma)}\}$*, where*

$$\Gamma = \left\{ \begin{array}{l}
(\mathbf{x}, \mathbf{y}, z) \in \mathbb{R}^{2n+1} : \\
\sum_{i=1}^{n} \left(\frac{z^2}{z c_i - y_i} \right) - zb \le 0, \\
\mathbf{x} - (1-z)\mathbf{u} \le \mathbf{y} \le \mathbf{x}, \\
z\mathbf{l} \le y \le z\mathbf{u}, \\
0 < z \le 1.
\end{array} \right\}$$

As discussed previously, the values of the functions in the nonlinear constraints are not well defined in $z_k^i = 0$. In the following proposition, we suggest a new valid relaxation of the convex hull which overcomes this issue while still being exact for $z_k^i \in \{0, 1\}$.

Proposition 1 *Let*

$$\begin{aligned}
& f : \mathbb{R}_+^n \to \mathbb{R},\ f(\mathbf{x}) = \sum_{i=1}^{n} \left(\tfrac{1}{c_i - x_i} \right) - b,\ b \ge 0. \\
& \Gamma_0 = \{\ (\mathbf{x},\ z) \in \mathbb{R}^n \times \mathbb{B} : z = 0,\ \mathbf{0} \le \mathbf{x} \le \mathbf{u} < \mathbf{c}\}, \\
& \Gamma_1 = \{\ (\mathbf{x},\ z) \in \mathbb{R}^n \times \mathbb{B} : z = 1,\ f(\mathbf{x}) \le 0,\ \mathbf{l} \le \mathbf{x} \le \mathbf{u}\} \textit{ non empty.}
\end{aligned}$$

For $\epsilon > 0$, let

$$\Gamma^\epsilon = \left\{ \begin{array}{l} (\mathbf{x}, \mathbf{y}, z) \in \mathbb{R}^{2n+1} : \\ \sum_{i=1}^{n} \left(\frac{z^2}{zc_i - y_i + (1-z)\epsilon} \right) - zb \leq 0, \\ \mathbf{x} - (1-z)\mathbf{u} \leq \mathbf{y} \leq \mathbf{x}, \\ z\mathbf{l} \leq \mathbf{y} \leq z\mathbf{u}, \\ 0 \leq z \leq 1. \end{array} \right\}$$

Then

1. ***proj**$_{(\mathbf{x},z)}(\Gamma^\epsilon)$ **is a valid convex relaxation of conv**$(\Gamma_0 \cup \Gamma_1)$*
2. ***proj**$_{(\mathbf{x},z)}(\Gamma^\epsilon) \cap \{z \in \{0,1\}\} \equiv \Gamma_0 \cup \Gamma_1$.*

Proof. *First, we show that all the constraints of Γ^ϵ are convex. The only nonlinear constraint in Γ^ϵ is $g(\mathbf{y}, z) \leq 0$ with $g(\mathbf{y}, z) = \sum_{i=1}^{n} g_i(y_i, z) - zb$ and $g_i(y_i, z) = \frac{z^2}{zc_i - y_i + (1-z)\epsilon}$. The Hessian matrix of g_i is*

$$\mathcal{H}(g_i) = \begin{pmatrix} \frac{2(y_i-\epsilon)^2}{(z(c_i-\epsilon)-(y_i-\epsilon))^3} & \frac{-2z(y_i-\epsilon)}{(z(c_i-\epsilon)-(y_i-\epsilon))^3} \\ \frac{-2z(y_i-\epsilon)}{(z(c_i-\epsilon)-(y_i-\epsilon))^3} & \frac{2z^2}{(z(c_i-\epsilon)-(y_i-\epsilon))^3} \end{pmatrix}.$$

Having $y_i \leq zu_i \leq zc_i + (1-z)\epsilon, \ \forall i \in \{1,..,n\}$, one can check that $\mathcal{H}$ is positive semidefinite, that is the functions g_i are all convex, g being a sum of convex functions is convex. Note also that, since $\frac{z^2}{zc_i - y_i + (1-z)\epsilon} \leq \frac{z^2}{zc_i - y_i}$, the validity of the constraint is preserved.

Next we prove that the projection of Γ^ϵ on the $(\mathbf{x}, z)$-space contains both Γ_0 and Γ_1.

For $z = 0$ we have

$$\Gamma^\epsilon = \left\{ \begin{array}{l} (\mathbf{x}, \mathbf{y}, 0) \in \mathbb{R}^{2n+1} : \\ \mathbf{x} \leq \mathbf{u}, \ \mathbf{y} = 0, \\ \mathbf{x} \geq \mathbf{y} = 0, \end{array} \right\},$$

in this case $proj_{(\mathbf{x},z)}(\Gamma^\epsilon) = \Gamma_0$.

For $z = 1$ we have

$$\Gamma^\epsilon = \left\{ \begin{array}{l} (\mathbf{x}, \mathbf{y}, 1) \in \mathbb{R}^{2n+1} : \\ \sum_{i=1}^{n} \left(\frac{1}{c_i - y_i} \right) - b \leq 0, \\ \mathbf{x} = \mathbf{y}, \ \mathbf{l} \leq \mathbf{y} \leq \mathbf{u} \end{array} \right\},$$

in this case $proj_{(\mathbf{x},z)}(\Gamma^\epsilon) = \Gamma_1$.

□

Based on this proposition, we introduce a new convex MINLP equivalent to (P) which gives a tighter continuous relaxation than the big-M model.

Reduced convex hull relaxation model (P_{red}) We replace constraints (9) by the convex relaxations defined in Proposition 1:

$$
\begin{aligned}
\min \quad & \sum_{e \in E} w_e x_e \\
\text{s.t. } & (6),\ (7),\ (8),\ (10),\ (11),\ (12-14) \\
& \sum_{e \in P_k^i} \left(\frac{{z_k^i}^2}{z_k^i c_e - y_e^{(i,k)} + (1 - z_k^i)\epsilon} \right) - z_k^i \alpha_k \le 0, && \forall k \in K,\ \forall P_k^i \in P(k), \quad \text{(9-b)} \\
& x_e - (1 - z_k^i) u_e \le y_e^{(i,k)} \le x_e, && \forall k \in K,\ \forall P_k^i \in P(k),\ \forall e \in P_k^i, \\
& z_k^i l_e \le y_e^{(i,k)} \le z_k^i u_e, && \forall k \in K,\ \forall P_k^i \in P(k),\ \forall e \in P_k^i.
\end{aligned}
$$

If we denote by $n_{max} = \max_k \{n_k\}$ the maximum number of candidate paths for commodities, up to $|E| \times |K| \times n_{max}$ variables can be added in this new model. While the bound might be tighter than those obtained with the big-M formulation, this relaxation may be difficult to solve due to the large number of additional variables. The corollary below is a first step toward a tight model defined in the space of original variables.

Corollary 3 *Let*

$$
\begin{aligned}
& f : \mathbb{R}_+^n \to \mathbb{R},\ f(\mathbf{x}) = \sum_{i=1}^{n} \left(\tfrac{1}{c_i - x_i} \right) - b,\ b \ge 0, \\
& \Gamma_0 = \{\ (\mathbf{x},\ z) \in \mathbb{R}^n \times \mathbb{B} : z = 0,\ \mathbf{0} \le \mathbf{x} \le \mathbf{u} < \mathbf{c}\}, \\
& \Gamma_1 = \{\ (\mathbf{x},\ z) \in \mathbb{R}^n \times \mathbb{B} : z = 1,\ f(\mathbf{x}) \le 0,\ \mathbf{l} \le \mathbf{x} \le \mathbf{u}\} \textit{ non empty.}
\end{aligned}
$$

For $\epsilon > 0$, let

$$
\Gamma_r^\epsilon = \left\{ \begin{array}{l} (\mathbf{x}, z) \in \mathbb{R}^{n+1} : \\ \sum_{i=1}^{n} \left(\frac{z^2}{z c_i - x_i + (1-z)(u_i + \epsilon)} \right) - zb \le 0, \\ z\mathbf{l} \le \mathbf{x} \le \mathbf{u}, \\ 0 \le z \le 1. \end{array} \right\}.
$$

Then

1. ***Γ_r^ϵ is a valid convex relaxation for conv($\Gamma_0 \cup \Gamma_1$)***
2. **$\Gamma_r^\epsilon \cap \{z \in \{0, 1\}\} \equiv \Gamma_0 \cup \Gamma_1$**

Proof. *f being an isotone closed convex function, Corollary 1 applies leading to the following constraint: $\sum_{i=1}^{n} \left(\frac{z^2}{z c_i - x_i + (1-z) u_i} \right) - zb \le 0$. Since $\frac{z^2}{z c_i - x_i + (1-z)(u_i + \epsilon)} \le \frac{z^2}{z c_i - x_i + (1-z) u_i}$ the validity of this new constraint is guaranteed, convexity is also maintained since one can replace $(u_i + \epsilon)$ with v_i leading to the initial constraint definition. Replacing z in Γ_r^ϵ respectively by 0 and 1, one can check that the resulting sets are Γ_0 and Γ_1 respectively.* □

Projected convex hull relaxation model (P_{proj}). We replace constraints (9) by the convex relaxations defined in the previous corollary:

$$\min \sum_{e\in E} w_e x_e$$
$$\text{s.t. } (6),\ (7),\ (8),\ (10),\ (11),\ (12-14)$$
$$\sum_{e\in P_k^i}\left(\frac{{z_k^i}^2}{z_k^i c_e - x_e + (1-z_k^i)(u_e+\epsilon)}\right) - z_k^i\alpha_k \le 0, \quad \forall k\in K,\ \forall P_k^i \in P(k), \quad \text{(9-c)}$$
$$0 \le x_e \le u_e,\ \forall e\in E.$$

In P_{proj} we add only one nonlinear constraint for each "on/off" constraint. As shown by Corollary 1, this is enough to obtain a valid formulation. Of course a stronger formulation would be obtained if all (exponentially many) non-linear constraints describing the convex hull of each "on/off" constraint were added. The main reason we use only one constraint here is that the formulation with all nonlinear constraints was too heavy to solve with the continuous nonlinear solver at our disposal. A natural way to overcome this would be to add these constraints in a dynamic way, but nonlinear solvers do not offer the capabilities to do that easily at this time. We note that from a theoretical point of view, independently of the application, the constraint used in Corollary 3 dominates all the others for $(\mathbf{x}, z)$ satisfying $\mathbf{x} \ge z\mathbf{l^1} + (1-z)\mathbf{u^0}$. To evaluate the quality of the model in practice, we compared the lower bound given by the continuous relaxation of the extended formulation P_{red} (which should be equal to the one obtained by adding all the nonlinear constraints) to the one given by P_{proj}. The experiment showed that the quality of the bounds are very close with both formulations. This assesses the strength of the projected model P_{proj} for our application but we do not claim that this experimental finding is true in general.

We next compare the formulation P_{proj} to the one proposed in [3].

3.2 Relationship To Previous Research On This Application

Ben-Ameur and Ouorou [3] have introduced the following convex reformulation of the "on/off" delay constraint:

$$\sum_{e\in P_k^i}\left(\frac{{z_k^i}^2}{c_e - x_e}\right) - z_k^i\alpha_k \le 0,\ \forall k\in K,\ \forall P_k^i \in P(k). \tag{15}$$

In Proposition 2 we show that constraint (9-c) introduced in (P_{proj}) dominates (15).

Proposition 2 *Constraint (9-c) dominates constraint (15).*

Proof.

Constraints (9-c) dominates constraints (15) if and only if
$\sum_{e\in P_k^i}\left(\frac{{z_k^i}^2}{c_e-x_e}\right) - z_k^i\alpha_k \le \sum_{e\in P_k^i}\left(\frac{{z_k^i}^2}{z_k^i c_e - x_e + (1-z_k^i)(u_e+\epsilon)}\right) - z_k^i\alpha_k,\ \forall k\in K,\ \forall P_k^i \in P(k).$

By definition of u_e, $\forall e \in E$, one can write:

$$u_e + \epsilon \le c_e \Rightarrow u_e + \epsilon - c_e \le 0 \Rightarrow {z_k^i}^2(1 - z_k^i)(u_e + \epsilon) - {z_k^i}^2(1 - z_k^i)c_e \le 0$$

$$\Rightarrow {z_k^i}^2(1 - z_k^i)(u_e + \epsilon) - {z_k^i}^2 c_e + {z_k^i}^3 c_e \le 0 \Rightarrow {z_k^i}^2(1 - z_k^i)(u_e + \epsilon) - {z_k^i}^2 c_e + {z_k^i}^3 c_e + {z_k^i}^2 x_e - {z_k^i}^2 x_e \le 0$$

$$\Rightarrow {z_k^i}^2(z_k^i c_e - x_e + (1 - z_k^i)(u_e + \epsilon)) - {z_k^i}^2(c_e - x_e) \le 0 \Rightarrow \frac{{z_k^i}^2}{(c_e - x_e)} - \frac{{z_k^i}^2}{(z_k^i c_e - x_e + (1 - z_k^i)(u_e + \epsilon))} \le 0$$

$$\Rightarrow \frac{{z_k^i}^2}{(c_e - x_e)} \le \frac{{z_k^i}^2}{(z_k^i c_e - x_e + (1 - z_k^i)(u_e + \epsilon))}, \ \forall e \in E \Rightarrow \sum_{e \in P_k^i} \left(\frac{{z_k^i}^2}{c_e - x_e}\right) \le \sum_{e \in P_k^i} \left(\frac{{z_k^i}^2}{z_k^i c_e - x_e + (1 - z_k^i)(u_i + \epsilon)}\right)$$

$$\Rightarrow \sum_{e \in P_k^i} \left(\frac{{z_k^i}^2}{c_e - x_e}\right) - z_k^i \alpha_k \le \sum_{e \in P_k^i} \left(\frac{{z_k^i}^2}{z_k^i c_e - x_e + (1 - z_k^i)(u_i + \epsilon)}\right) - z_k^i \alpha_k.$$

□

In the next section, we compare numerically the four formulations of DCRP presented in this section.

4 Computational Experiments

In order to compare the four models presented in Section 3, we solved the corresponding formulations on different networks with different settings for the number of candidate paths and the number of paths that can be simultaneously used to route the traffic demands. We consider two types of networks: real world networks (denoted rdatax) and randomly generated networks (denoted adatax). Table 1 summarizes the main characteristics of these networks.

Table 1 Main networks characteristics

name	$\|V\|$	$\|E\|$	$\|K\|$
rdata1	60	280	100
rdata2	61	148	122
adata3	100	600	200
rdata4	34	160	946
rdata5	67	170	761
adata6	100	800	500

To solve the convex MINLP models, we used the open source solver Bonmin [7] (release 1.1.3, see `http://www.coin-or.org/Bonmin`) with CBC [11] as underlying mixed-integer linear programming (MILP) solver and Ipopt [23] for the nonlinear programming solver. The four models were coded in C++ to be solved by Bonmin and all tests were performed on an Intel Xeon 1.6 Ghz CPU.

Bonmin offers the possibility to choose one of five algorithms: a nonlinear programming based branch-and-bound [9], a pure outer approximation decomposition [10], a vanilla implementation of the Quessada-Grossmann branch-and-cut algorithm [20], a hybrid method [7] and a method based on extended cutting planes [24] (similar to the method proposed in [1]). Here we report results obtained with the hybrid method since it was consistently better than the others (with all four models) in preliminary experiments. First, we give a brief summary of the hybrid method.

The hybrid algorithm combines the two schemes [10, 20] of which the pure outer approximation decomposition and the Quessada Grossmann Algorithm occur as extreme cases. The basic idea of outer approximation is to build an MILP equivalent to the original MINLP, namely a polyhedral

Table 2 Results for $N = 1$ (mono-routing).

	P_{big-M}		P_{proj}		P_{high}		P_{red}	
	cpu	nodes	cpu	nodes	cpu	nodes	cpu	nodes
	2 candidate paths per commodity							
rdata1	**0.4**	0	**0.4**	0	35	0	2	0
rdata2	190	5193	**155**	1012	2997	21312)	1948	15129
adata3	144	335	**57**	0	206	556	159	84
rdata4	**3**	0	**3**	0	2040	11691	(1485	5845
rdata5	[∞]	14788	**251**	3357	1549	2793	[0.03%]	15697
adata6	**1065**	27991	1470	42244	[1%]	6831	[0.3%]	2499
	3 candidate paths per commodity							
rdata1	2.7	0	**2.4**	0	2.8	0	11.9	0
rdata2	25	0	**13**	0	994	3671	1396	7815
adata3	[0.28%]	157748	344	5097	722	3286	**312**	1124
rdata4	[0.001%]	79807	**1525**	50583	[0.04%]	28876	[0.1%]	22438
rdata5	[0.43%]	138618	**[0.03%]**	202122	[0.2%]	9472	[0.14%]	9461
adata6	[0.006%]	176413	**934**	19351	[0.6%]	16539	[0.06%]	4067
	10 candidate paths per commodity							
rdata1	568	13649	**231**	4762	1415	9263	1209	6485
rdata2	120	0	**66**	0	1527	1599	1555	2563
adata3	**534**	5118	644	14216	4866	8841	6684	11626
rdata4	[1.9%]	79807	[2.1%]	96212	[∞]	3409	**[1.8%]**	3156
rdata5	[∞]	37446	[∞]	30500	[∞]	747	[∞]	1568
adata6	[2.7%]	35520	**[1.5%]**	2680	[∞]	2642	[∞]	1001

outer approximation of the non-linear constraints of the problem. The hybrid algorithm constructs this outer approximation using two principles: a decomposition method which solves a sequence of MILPs [10] and a branch-and-cut approach. The decomposition method is ran at the beginning of the algorithm for a limited amount of time (120 seconds in our experiments). If optimality was not proved during this initial phase, the branch-and-cut method is started using the outer approximation of the feasible region obtained at the end of the decomposition phase.

In the experiments, we ran the different formulations on different networks, with different numbers of candidate paths and using different values for the parameter N (maximum number of path used by each commodity): $N = 1$ (mono-routing), $N = 2$ (bi-routing), $N = \infty$ (multi-routing). For each instance, we report the computing time to optimality and the number of nodes explored during the branch-and-cut phase with the four models. If the problem is solved during the initial outer approximation decomposition phase, we just report 0 nodes of branch-and-cut. If optimality is not reached within the time limit, the gap between the current best integer feasible solution and the continuous relaxation is displayed inside brackets, ∞ indicates that no integer feasible solution has been found after two hours of computing time. For each instance, the best computing time or the smallest gap is listed in bold characters.

Table 2 summarizes the results for the case $N = 1$, i.e. only one path per commodity can be activated at a time (mono-routing problems). The number of candidate paths per commodity is set to 2, 3 and 10 respectively. These paths are obtained by the k-shortest path algorithm proposed by Martins and Pascoal in [19]. For the P_{proj} model, the epsilon value (appearing in the perspective formulation) is set to 10^{-4}. Different values were tested, varying between 10^{-3} and 10^{-6}. Experiments showed no sensitivity to this parameter, neither on the optimal value nor on the global computing time.

Table 3 Results with $N = 2$ (bi-routing).

	P_{big-M} cpu	P_{big-M} nodes	P_{proj} cpu	P_{proj} nodes
	2 candidate paths per commodity			
rdata1	0.8	0	**0.4**	0
rdata2	**2.4**	0	10.9	0
adata3	47.6	0	**4.7**	0
rdata4	5.6	0	**3.4**	0
rdata5	**38.3**	0	50.4	0
adata6	**40.1**	0	42.4	0
	3 candidate paths per commodity			
rdata1	16.7	0	**2.9**	0
rdata2	129.2	18	**59**	0
adata3	291.5	760	**171.3**	615
rdata4	**154.9**	62	343.8	472
rdata5	[0.15%]	91430	**609.1**	5290
adata6	1579.8	8176	**747.7**	7056
	10 candidate paths per commodity			
rdata1	1909	56788	**399**	7846
rdata2	**28.8**	0	288.8	666
adata3	**[0.11%]**	67705	[0.13%]	77129
rdata4	[1.2%]	23984	**[0.6%]**	32939
rdata5	[2%]	11772	**[1.2%]**	16285
adata6	[0.7%]	6480	**[0.14%]**	22364

The projected convex hull model P_{proj} scores the best performance on these instances. P_{proj} solves 14 instances out of 18 while P_{big-M}, P_{high} and P_{red} only solve (resp.) 10, 11 and 10 instances. If we consider geometric means, P_{proj} is 2.1 times faster than P_{big-M}, 6.7 times faster than P_{high} and 6.3 times faster than P_{red}. The advantage in terms of number of nodes is comparable. Considering the four problems P_{big-M} is unable to solve, P_{proj} is at least one order of magnitude faster than P_{big-M} with at least 5 times fewer nodes to reach optimality (it is at least 5.45 and 7.17 times faster that P_{high} and P_{red}). In conclusion, even if P_{high} and P_{red} provide better lower bounds, their average time is penalized due to the increased number of variables. For problem adata3 with 3 candidate paths, one can see that optimality is reached with P_{red} in only 1124 nodes, while the big-M model explored 157748 nodes with a final nonzero gap. Let us emphasize that the projected model P_{proj} provides bounds almost as tight as the extended formulations, without having to deal with the inconvenience of large size problems.

We now consider the bi-routing case($N = 2$) where two paths can be activated for each commodity. From previous results on the mono-routing case, it appears clearly that P_{proj} is consistently better than P_{high} and P_{red} for all instances except one, where they are equivalent. Furthermore, since the bi-routing and the multiple-routing case involve adding new variables corresponding to fractions of demands (ϕ_k^i), these high dimensional relaxations would be even larger and more difficult to solve in these cases. For this reason, P_{high} and P_{red} were not implemented in the remaining of the experiments. Table 3 reports results obtained for instances with respectively 2, 3 and 10 candidate paths per commodity.

First, we note that bi-routing problems seem in general easier to solve than their mono-routing counterparts. For these problems P_{big-M} and P_{proj} solved respectively 13 and 14 instances. The instances with 2 candidate paths are all solved in less than one minute with both formulations and

Table 4 Results with $N = \infty$ (multiple-routing)

	P_{big-M}		P_{proj}	
	cpu	nodes	cpu	nodes
	3 candidate paths per commodity			
rdata1	2.7	0	**1.2**	0
rdata2	**10**	0	29.7	0
adata3	195.7	164	**50.5**	0
rdata4	**20.7**	0	43	0
rdata5	2503.1	34783	**362.8**	4024
adata6	1482.4	6083	**224.7**	326
	10 candidate paths per commodity			
rdata1	799.7	12633	**220.8**	1922
rdata2	**16.1**	0	24.8	0
adata3	[0.08%]	94194	**768.6**	5207
rdata4	[0.4%]	40820	**[0.04%]**	45492
rdata5	[1.2%]	16106	**5467.7**	17347
adata6	[0.7%]	5880	**5392**	23867

instances with 3 candidate paths seem also much easier than before. On average P_{proj} is still faster than P_{big-M} taking about 1761 secs versus 2235 secs.

We finally consider the multiple-routing case where all the paths in $P(k)$ can be activated ($N = \infty$). Table 4 reports results obtained for instances with 3 and 10 candidate paths per commodity (instances with 2 paths are similar in the bi-routing and multi-routing cases).

Multi-routing problems bring the same observations as the bi-routing case. For these problems P_{big-M} and P_{proj} solved respectively 8 and 11 instances out of 12. All instances with 3 paths per commodity are solved with both formulations. On average, for all instances, P_{proj} takes 1649 secs while P_{big-M} takes 2819 secs. For the instance not solved by both formulations, the final gap for P_{proj} is about one order of magnitude smaller than for P_{big-M}.

5 Conclusion

Disjunctive programming is one of the most successful approaches for constructing tight convex relaxations to mixed-integer programs. Its efficiency in the context of MILP has been demonstrated many times. The case of convex MINLPs is more difficult and presents several challenges. One of them is that typical approaches using extended formulations often lead to nonlinear programs that are too large to be tackled using state of the art softwares. Therefore, it is necessary to construct formulations in smaller spaces. In this work we looked closely at a specific setting: A disjunctive set defined by the union of an hyper-rectangle and a closed convex set featuring one nonlinear constraint $f(\mathbf{x}) \leq 0$. When the function f is isotone, we gave an explicit characterization of the convex hull in the space of original variables. This leads to formulations that are tighter than those obtained using more naive approaches (such as big-M constraints) and whose continuous relaxations are still solvable in a reasonable amount of time. Numerical experiences showed that these new formulations allow to solve instances with computational gains of up to one order of magnitude compared to classical models.

There are several challenges that naturally arise from this study. The first one is to be able to extend results on compact formulations to more general settings of unions of convex sets. Also, one would like to automatically detect and reformulate these disjunctions. Finally, let us emphasize that even in the case of polytopes, applying Theorem 2 yields a family of new MIP cuts [16] that would be worth exploring.

References

1. K. Abhishek, S. Leyffer, and J. T. Linderoth. FilMINT: An outer-approximation-based solver for nonlinear mixed integer programs. Preprint ANL/MCS-P1374-0906, Mathematics and Computer Science Division, Argonne National Laboratory, 2006.
2. S. Aktürk, A. Atamtürk, and S. Gürel. A strong conic quadratic reformulation for machine-job assignment with controllable processing times. Technical Report BCOL Research Report 07.01, Industrial Engineering & Operations Research, University of California, Berkeley, 2007.
3. W. Ben-Ameur and A. Ouorou. Mathematical models of the delay-constrained routing problem. *Algorithmic Operations Research*, 1(2):94–103, 2006.
4. E. Balas. Disjunctive programming. In *Annals of Discrete Mathematics 5: Discrete Optimization*, pages 3–51. North Holland, 1979.
5. E. Balas. Disjunctive programming and a hierarchy of relaxations for discrete optimization problems. *SIAM J. Alg. Disc. Math.*, 6:466–486, 1985.
6. D.P. Bertsekas and R.G. Gallager. *Data networks.* Prentice-Hall, Englewood Cliffs, 1987.
7. P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter. An algorithmic framework for convex mixed-integer nonlinear programs. *Discrete Optimization*, 5(2):186–204, 2008.
8. S. Ceria and J. Soares. Convex programming for disjunctive optimization. *Mathematical Programming*, 86:595–614, 1999.
9. R. J. Dakin. A tree search algorithm for mixed programming problems. *Computer Journal*, 8:250–255, 1965.
10. M. A. Duran and I. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36:307–339, 1986.
11. J. Forrest. CBC, 2004. Available from `http://www.coin-or.org/`.
12. A. Frangioni and C. Gentile. Perspective cuts for a class of convex 0-1 mixed-integer programs. *Mathematical Programming*, 106(2):225–236, 2006.
13. I. Grossmann and S. Lee. Generalized convex disjunctive programming: Nonlinear convex hull relaxation. *Computational Optimization and Applications*, 26:83–100, 2003.
14. O. Günlük and J. Linderoth. Perspective relaxation of mixed-integer nonlinear programs with indicator variables. *Lecture Notes in Computer Science*, 5035:1–16, 2008.
15. H. Hijazi, P. Bonami, G. Cornuéjols and A. Ouorou. Mixed Integer Non-Linear Programs featuring "On/Off" constraints: convex analysis and application. Proceedings of ISCO 2010 - International Symposium on Combinatorial Optimization *Electronic Notes in Discrete Mathematics*, 36:1153 – 1160, 2010.
16. H. Hijazi, Mixed Integer Non-Linear Optimization approaches for network design in Telecommunications. *Ph.D. Thesis, Université d'Aix Marseille*, November 2010.
17. J. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms*, volume 2. Springer-Verlag, Berlin Heidelberg New York, 1993.
18. J.G. Klincewicz and J. Schmitt. Incorporating qos into ip entreprise network design. *Telecommunication Systems*, 20(1,2):81–106, 2002.
19. E. Martins and M. Pascoal. A new implementation of yens ranking loopless paths algorithm. *4OR*, 1(2):121–133, 2003.
20. I. Quesada and I. E. Grossmann. An LP/NLP based branch–and–bound algorithm for convex MINLP optimization problems. *Computers and Chemical Engineering*, 16:937–947, 1992.
21. M. G. C. Resende and P. M. Pardalos. *Handbook of Optimization in Telecommunications.* Springer, 2006. Chapter 17 by J.G. Klincewicz.
22. R. Stubbs and S. Mehrotra. A branch-and-cut method for 0-1 mixed convex programming. *Mathematical Programming*, 86:515–532, 1999.
23. A. Wächter and L. T. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.
24. T. Westerlund and F. Pettersson. A cutting plane method for solving convex MINLP problems. *Computers and Chemical Engineering*, 19:s131–s136, 1995.