Mixed Integer NonLinear Programs featuring “On/Off ” constraints: convex analysis and applications

Hassan Hijazi, Pierre Bonami, Gérard Cornuéjols, Adam Ouorou

Mixed Integer NonLinear Programs featuring “On/Off” constraints: convex analysis and applications

Hassan Hijazi · Pierre Bonami · Gérard Cornuéjols · Adam Ouorou

October 2009

Abstract We call "on/off" constraint an algebraic constraint that is activated if and only if a corresponding boolean variable is turned "on" or equal to 1. Our main subject of interest is to derive tight convex formulations of Mixed Integer NonLinear Programs (MINLPs) featuring "on/off" constraints. We study the simple set defined by one "on/off" constraint with bounded variables. Using disjunctive programming, we introduce convex hull formulations of this set defined in higher dimensional spaces. Because the large number of variables in these formulations appears to be practically disadvantageous, we concentrate our efforts on defining explicit projections into lower spaces. When the functions defining the "on/off" constraint are order preserving or isotone, we show that the convex hull can be formulated in the space of original variables. This result applies in particular when the functions are additively separable, sum of one variable monotone functions.
As a direct application to our results, we present new formulations to a well-known telecommunication problem: routing several commodities subject to multiple delay constraints. While classical multi-commodity routing problems deal with only one design specification, usually a total queuing delay constraint, this model takes into account individual delays for each type of commodity, allowing operators to offer a so-called "differentiated quality of service". Numerical results on randomly generated and real world networks are presented to assess the efficiency of the new models.

Keywords mixed integer nonlinear programming, on/off constraints, disjunctive constraints, convex programming, routing problems, multiple delay constraints.

First and second authors are supported by ANR grant ANR06-BLAN-0375.

Third author is supported by NSF grant CMMI0653419, ONR grant N00014-03-1-0133 and ANR grant ANR06-BLAN-0375.

H. Hijazi
Orange Labs R&D/CORE-MCN, 38-40 rue du Général Leclerc, 92794 Issy-Les-Moulineaux cedex 9, France; and LIF, CNRS-Aix Marseille Université, Parc Scientifique et Technologique de Luminy, Marseille, 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;
and LIF, CNRS-Aix Marseille Universite, Parc Scientifique et Technologique de Luminy, Marseille, France.

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

1 Introduction

In recent years, Mixed Integer NonLinear Programming (MINLP) has been a very active area of research. A special case of interest is that of *convex MINLPs* where the objective function to minimize as well as the feasible region obtained by dropping the integrality requirements on variables is convex. For such problems, several algorithms have been developed [10,16,19,7] and have been implemented in solvers such as FilMINT [1] or Bonmin[6] which are able to solve problems of medium sizes.
In order to solve larger problems, a key element is to be able to produce convex continuous relaxations which should provide good bounds on the value of the underlying MINLP and be easily solvable by state of the art MINLP solvers. To the best of our knoweldge automatic procedures handling such task are yet to be devised.
In this work, motivated by an application in telecommunications, we study tight relaxations for certain types of convex MINLPs featuring "on/off" constraints. An "on/off" constraints is an algebraic constraint that has to be activated if and only if a corresponding 0-1 indicator variable is equal to 1. Given convex functions $g : \mathbb{R}^{n+1} \to \mathbb{R}^m$, $h : \mathbb{R}^{n+1} \to \mathbb{R}$ and $f^k : \mathbb{R}^n \to \mathbb{R}$, $\forall k \in \{1,\ 2, ..., K\}$, we are concerned with optimization problems that can be written as:

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

Each $f^k(x) \leq 0$ represents an "on/off" constraint, with z_k its corresponding indicator variable. The $g(x,z) \leq 0$ gather the remaining convex constraints. Bounds on variables are assumed to be finite.
Of course the difficulty of solving a problem like (1) lies in the presence of the "on/off" constraints. Such constraints are clearly non-convex since each one generates a feasible region defined by the union of two disjoint sets. Furthermore, reformulating these constraints is necessary in order to solve the problem with a standard convex MINLP solver.
A way to model (1) as a convex MINLP is to use so called Big-M constraints. Unfortunately the continuous relaxations resulting from such models, although compact and usually easy to solve, typically provide weak lower bounds.
Another way to formulate (1) is to use disjunctive programming (see [5,8,17,13]). Indeed, (1) can be reformulated as a disjunctive program as follows:

$$\begin{aligned} &\min\ h(x,z) \\ s.t.\ & g(x,z) \leq 0 \\ & (x, z_k) \in \Gamma_0^k \cup \Gamma_1^k,\ \forall k \in \{1,\ 2, ..., K\} \\ & \Gamma_0^k = \{\ (x,\ z_k) \in \mathbb{R}^n \times \mathbb{B} : z_k = 0,\ l_i \leq x_i \leq u_i,\ \forall i \in \{1,\ 2,\ ...,\ n\}\ \} \\ & \Gamma_1^k = \{\ (x,\ z_k) \in \mathbb{R}^n \times \mathbb{B} : z_k = 1,\ f^k(x) \leq 0,\ l_i \leq x_i \leq u_i,\ \forall i \in \{1,\ 2,\ ...,\ n\}\ \}. \end{aligned} \tag{2}$$

In Stubbs and Mehrotra (1999) [17], Ceria and Soares (1999) [8], as well as in Grossman and Lee (2003) [13], one can find the necessary tools for writing the algebraic formulation of $conv\left(\Gamma_0^k \cup \Gamma_1^k\right)$, the convex hull of $\Gamma_0^k \cup \Gamma_1^k$, in a higher-dimensional space, by introducing additional variables. Using these results, one can rewrite (2) as:

$$
\begin{aligned}
&\min\ h(x,z) \\
s.t.\ & g(x,z) \leq 0 \\
&(x,z_k) \in conv(\Gamma_0^k \cup \Gamma_1^k),\ \forall k \in \{1,\ 2,...,K\} \\
&\Gamma_0^k = \{\ (x,\ z_k) \in \mathbb{R}^n \times \mathbb{B} : z_k = 0,\ l_i \leq x_i \leq u_i,\ \forall i \in \{1,\ 2,\ ...,\ n\}\ \} \\
&\Gamma_1^k = \{\ (x,\ z_k) \in \mathbb{R}^n \times \mathbb{B} : z_k = 1,\ f^k(x) \leq 0,\ l_i \leq x_i \leq u_i,\ \forall i \in \{1,\ 2,\ ...,\ n\}\ \}. \\
&x \in \mathbb{R}^n, z_k \in \{0,1\},\ \forall k \in \{1,\ 2,...,K\}.
\end{aligned}
\tag{3}
$$

Clearly, (3) is a valid formulation for (1). Furthermore, its continuous relaxation is a continuous convex nonlinear program that is likely to provide relatively good bounds. Nevertheless, writing $conv(\Gamma_0^k \cup \Gamma_1^k)$ in higher dimensional spaces leads to a nonlinear program of large size, often difficult to solve.

Naturally, one would be interested in formulations for (3) that do not require additional variables. Günlük and Linderoth [14] showed in their recent work that when Γ_0^k is restricted to a single point, the convex hull can be formulated in the space of original variables, using the so called *perspective function*. 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 case where Γ_0^k is a hyper-rectangle given by finite bounds on the x variables. In Section 2, based on the work of [8], we start by studying the formulation of $conv\left(\Gamma_0^k \cup \Gamma_1^k\right)$ in extended spaces. Next, we define projections of this convex hull into lower dimentionnal spaces. Our main result is that under specific assumptions on the f^k functions, we introduce tight relaxations in the space of original variables.

In Section 3, we study a first application for this type of problems, introduced by Ben Ameur and Ouorou [3]: the Delay Constrained Routing Problem. This model gives guarantees on the individual delay for each type of commodity, allowing telecommunication operators the ability to provide a "differentiated quality of service". We propose new mathematical models for this problem based on the convex hull formulations introduced in Section 2. Finally, Section 4 reports computational results obtained on telecommunication instances, allowing to compare existing models to the new ones.

Given a set $\Gamma \in \mathbb{R}^n$, we denote by $cl(\Gamma)$ its topological closure and by $proj_{(x^1,...,x^j)}(\Gamma)$ its projection on the $(x^1,...,x^j)$ space. The effective domain of a function f, denoted $dom(f)$, is the set of points $x \in \mathbb{R}^n$ for wich $f(x) < +\infty$. Let f be a closed convex function, the constraint $f(x) \leq 0$ admits a unique closed extension $(cl\ f) \leq 0$, wich results from redefining f at points $x \notin dom(f)$ in such away that:
$\{x \in \mathbb{R}^n : (cl\ f)(x) \leq 0\} = cl(\{x \in \mathbb{R}^n : f(x) \leq 0\}$.

2 Convex hull of $\Gamma_0 \cup \Gamma_1$

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

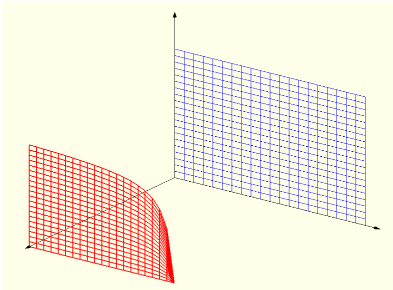

Fig. 1 The sets $\mathit{\Gamma}_0$ and $\mathit{\Gamma}_1$.

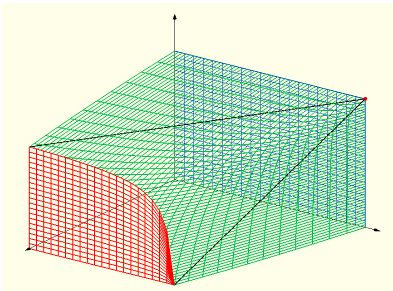

Fig. 2 $\mathrm{conv}(\mathit{\Gamma}_0 \cup \mathit{\Gamma}_1)$.

$$
\begin{aligned}
&\min_{(x,z)\in\mathbb{R}^2, z\in\mathbb{B}} h(x,z) \\
&s.t.\ (x,z) \in conv\,(\mathit{\Gamma}_0 \cup \mathit{\Gamma}_1) \\
&\quad \mathit{\Gamma}_0 = \{\ (x,\ z) \in \mathbb{R}^2 \times \mathbb{B} : z = 0,\ l_1 \le x_1 \le u_1,\ l_2 \le x_2 \le u_2\} \\
&\quad \mathit{\Gamma}_1 = \{\ (x,\ z) \in \mathbb{R}^2 \times \mathbb{B} : z = 1,\ \tfrac{1}{c_1 - x_1} + \tfrac{1}{c_2 - x_2} \le d,\ l_1 \le x_1 \le u_1,\ l_2 \le x_2 \le u_2\}
\end{aligned}
\tag{4}
$$

where $u_1 \le c_1$ and $u_2 \le c_2$.

Figure 1 gives a representation of the sets $\mathit{\Gamma}_0$ and $\mathit{\Gamma}_1$ in $\mathbb{R}^3$. $\mathit{\Gamma}_0$ is the rectangle on the right hand side of the figure and $\mathit{\Gamma}_1$ is the convex set on left hand side.
Figure 2 gives a representation of $conv(\mathit{\Gamma}_0 \cup \mathit{\Gamma}_1)$ in $\mathbb{R}^3$. An interesting remark is that the nonlinear constraint of $conv\,(\mathit{\Gamma}_0 \cup \mathit{\Gamma}_1)$ can be derived by using the perspective function of f.

The perspective function $\tilde{f} : \mathbb{R}^{n+1} \to \mathbb{R} \cup \{+\infty\}$ of f is defined by: $\tilde{f}(x,z) \equiv \begin{cases} zf(x/z) & \text{if } z > 0, \\ +\infty & \text{if } z \le 0. \end{cases}$

The closure of $\tilde{f}$, as introduced in [8], is shown in [15] (Proposition VI.2.2.2) to be defined as:

$$
(cl\ \tilde{f})(x,z) \equiv \begin{cases} zf(x/z) & \text{if } z > 0, \\ \lim_{z\to 0^+} zf(\tilde{x} - x + x/z) & \text{if } z = 0, \\ +\infty & \text{if } z < 0, \end{cases}
$$

where $\tilde{x}$ is an arbitrary point in the relative interior of $\mathrm{dom}(f)$.

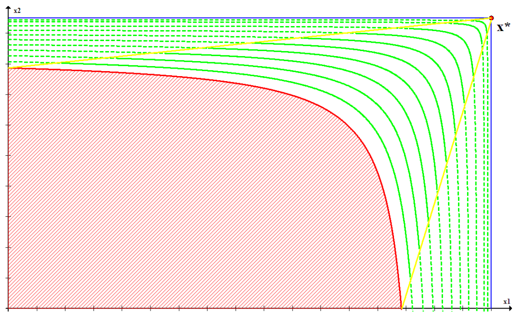

Fig. 3 $\mathrm{conv}(\mathit{\Gamma}_0 \cup \mathit{\Gamma}_1)$.

In our example, the nonlinear constraint of *conv* $(\mathit{\Gamma}_0 \cup \mathit{\Gamma}_1)$ is obtained by taking the perspective function of f from the point $x^* = (u_1, u_2)$, that is $\tilde{f}(x - (1-z)x^*, z) \leq 0$. Figure 3 plots the level curves of $\tilde{f}(x - (1-z)x^*, z)$ in $\mathbb{R}^3$.
For this specific example, *conv* $(\mathit{\Gamma}_0 \cup \mathit{\Gamma}_1)$ is written as follows:

$$conv\,(\mathit{\Gamma}_0 \cup \mathit{\Gamma}_1) = \left\{ \begin{array}{l} (x,\ z) \in \mathbb{R}^3 : \\ (cl\tilde{f})(x - (1-z)x^*,\ z) \leq 0 \\ l_1 \leq x_1 \leq u_1,\ l_2 \leq x_2 \leq u_2. \end{array} \right\} = \left\{ \begin{array}{l} (x,\ z) \in \mathbb{R}^3 : \\ cl \left(\frac{z}{zc_1 - x_1 + (1-z)u_1} + \frac{z}{zc_2 - x_2 + (1-z)u_2} - d \right) \leq 0 \\ l_1 \leq x_1 \leq u_1,\ l_2 \leq x_2 \leq u_2. \end{array} \right\}.$$

We now return to the general case dealing with one "on/off" constraint in $\mathbb{R}^n$. First, we recall a result of Ceria and Soarres [8] characterizing the convex hull of a union of closed convex sets: 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 following representation
$K^i \equiv \{x \in \mathbb{R}^n : G^i(x) \leq 0\}$, and $G^i : \mathbb{R}^n \to \mathbb{R}^m$ is a vector mapping whose components are closed convex functions.

Theorem 1 ([8]) *Let $I \equiv \{i : K^i \neq \emptyset\}$. If the set K is bounded below or above then $x \in P$ if and only if there exist vectors (λ_i, x^i), for every $i \in I$, such that the following nonlinear system is feasible*

$$\begin{aligned} & x = \sum_{i \in I} x^i \\ & \sum_{i \in I} \lambda_i = 1,\ \ i \in I, \\ & (cl\ \tilde{G}^i)(\lambda_i, x^i) \leq 0,\ \ i \in I, \\ & \lambda_i \geq 0,\ \ i \in I. \end{aligned}$$

where $(cl\ \tilde{G}^i)(\lambda_i, x^i)$ denotes the closure of the perspective mapping of G at (λ, x).

Based on this theorem, *conv* $(\mathit{\Gamma}_0 \cup \mathit{\Gamma}_1)$ can be formulated in a space of dimension $3n + 5$.
We note that applying this result in the formulation of (3), a total of $|K|.(2n+4)$ variables must be added. 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, we show that in our context, the convex hull explicit formula, corresponding to one "on/off" constraint, can be obtained by adding only n variables.

Lemma 1 *Let:*
$f : E \to \mathbb{R},\ E \subseteq \mathbb{R}^n$, *be a closed convex function,*
$\Gamma_0 = \{\ (x,\ z) \in \mathbb{R}^n \times \mathbb{B} : z = 0,\ l_i^0 \leq x_i \leq u_i^0,\ \forall i \in \{1, 2, ..., n\}\ \}$,
$\Gamma_1 = \{\ (x,\ z) \in \mathbb{R}^n \times \mathbb{B} : z = 1,\ f(x) \leq 0,\ l_i^1 \leq x_i \leq u_i^1,\ \forall i \in \{1, 2, ..., n\}\ \}$, *non empty, then:*

$$\boldsymbol{conv\ (\Gamma_0 \cup \Gamma_1) = \{(x, z)\ |\ \exists y \in \mathbb{R}^n,}\ \textit{s.t.}\ \boldsymbol{(x, y, z) \in cl(\Gamma)}\},$$

$$\textit{where } \Gamma = \left\{ \begin{array}{l} (x, y, z) \in \mathbb{R}^{2n+1} : \\ zf\,(y/z) \leq 0, \\ x_i - (1-z)u_i^0 \leq y_i \leq x_i - (1-z)l_i^0,\ \forall i \in \{1, 2, ..., n\}, \\ zl_i^1 \leq y_i \leq zu_i^1,\ \forall i \in \{1, 2, ..., n\}, \\ 0 < z \leq 1. \end{array} \right\}$$

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

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

Substituing $x^0 = x - x^1$ and $\lambda_0 = 1 - z$, we obtain:

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

In Lemma 1, we show that only n variables are needed to explicitly describe the convex hull. One can notice that these new variables appear in the perspective function of f as well as in: $x-(1-z)u^0 \leq y \leq x-(1-z)l^0$ and $zl^1 \leq y \leq zu^1$. We observe that, if we consider only the last two sets of constraints, the Fourrier-Motzkin elimination can be applied in a straightforward way leading to the constraints $zl^1+(1-z)l^0 \leq x \leq zu^1+(1-z)u^0$. Nevertheless, the projection becomes harder when taking into account the nonlinear constraint in Γ.

Next, we show that if the function f is order preserving or isotone (see definition below), the y variables can be entirely projected out.

Definition 1 Let $f: E \rightarrow \mathbb{R},\ E \subseteq \mathbb{R}^n$, f is *independently increasing* (resp. *decreasing*) on the ith coordinate if:
$\forall x=(x_1,x_2,...,x_i,...,x_n) \in dom(f),\ x'=(x_1,x_2,...,x_i',...,x_n) \in dom(f)\ s.t.\ x_i' \geq x_i \Rightarrow f(x') \geq$ (resp. $\leq$) $f(x)$.
We say that f is *independently monotone* on the ith coordinate if it is independently increasing or independently decreasing on this given coordinate.
f is *order preserving* if it is independently monotone on each and 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 order preserving 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 order preserving function.
3. $f(x)=\sum_{i=1}^{n} \frac{1}{c_i-x_i}$, 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 order preserving function.

Additively separable functions which are sum of one-variable monotone functions are commonly encountered order preserving functions.

Theorem 2 *Let:*

$f: E \rightarrow \mathbb{R},\ E \subseteq \mathbb{R}^n$, *be an order preserving closed convex function,*

J^1*(resp.* J^2*) be the set of indexes on wich* f *is independently increasing (resp. decreasing),*

$\Gamma_0=\{\ (x,\ z) \in \mathbb{R}^n \times \mathbb{B} : z=0,\ l_i^0 \leq x_i \leq u_i^0,\ \forall i \in \{1,2,...,n\}\ \}$,

$\Gamma_1=\{\ (x,\ z) \in \mathbb{R}^n \times \mathbb{B} : z=1,\ f(x) \leq 0,\ l_i^1 \leq x_i \leq u_i^1,\ \forall i \in \{1,2,...,n\}\ \}$, *non empty, then:*

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

$$where\ \Gamma^{'} = \left\{ \begin{array}{l} (x,z) \in \mathbb{R}^{n+1} : \\ zq_S\left(x/z\right) \leq 0,\ \forall S \subset \{1,2,...,n\}, \\ zl_i^1 + (1-z)l_i^0 \leq x_i \leq zu_i^1 + (1-z)u_i^0,\ \forall i \in \{1,2,...,n\}, \\ 0 < z \leq 1. \end{array} \right\},$$

$$with\ q_S = (f \circ h_S),\ h_S(\mathbb{R}^n \to \mathbb{R}^n)\ defined\ by\ (h_S(x))_i = \begin{cases} l_i^1\ \forall i \in S \cap J_1, \\ u_i^1\ \forall i \in S \cap J_2, \\ x_i - \frac{(1-z)u_i^0}{z}\ \forall i \in J_1, i \notin S, \\ x_i - \frac{(1-z)l_i^0}{z}\ \forall i \in J_2, i \notin S. \end{cases}$$

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

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

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

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

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

   Let (x,y,z) be a point in Γ $(z \neq 0)$. By definition of Γ and functions $h_S(x)$, we have $\forall S \subset \{1,2,...,n\}$

   $$\frac{y_i}{z} \geq \max\{l_i^1,\ \frac{x_i}{z} - \frac{(1-z)u_i^0}{z}\} \geq (h_S(x/z))_i,\ \forall i \in J^1$$

   and

   $$\frac{y_i}{z} \leq \min\{u_i^1,\ \frac{x_i}{z} - \frac{(1-z)l_i^0}{z}\} \leq (h_S(x/z))_i,\ \forall i \in J^2.$$

   f being an order preserving function one has $zf(h_S(x/z)) \leq zf(y/z) \leq 0,\ \forall S \subset \{1,2,...,n\}$. Finally, notice that the constraint $zl^1 + (1-z)l^0 \leq x \leq zu^1 + (1-z)u^0$ is obtained by composing the last two constraints defining set Γ.

   The extention to the closure is trivial.

This theorem gives a new formulation that can be also used to explicitly write $conv\left(\Gamma_0 \cup \Gamma_1\right)$ in (3). This formulation has the advantage that no additional variable is introduced. However, it has an exponential number of nonlinear constraints, up to $(|K|.(2^n-1))$. In the next corollary, we show that adding only one nonlinear constraint for each "on/off" constraint is sufficient to have a valid formulation for our MINLP problem (2).

Corollary 1 *Let:*

$f : E \to \mathbb{R},\ E \subseteq \mathbb{R}^n$, *be an order preserving closed convex function,*

J^1*(resp.* J^2*) be the set of indexes on wich* f *is independently increasing (resp. decreasing),*

$\Gamma_0 = \{\ (x,\ z) \in \mathbb{R}^n \times \mathbb{B} : z = 0,\ l_i^0 \leq x_i \leq u_i^0,\ \forall i \in \{1, 2, ..., n\}\ \}$,

$\Gamma_1 = \{\ (x,\ z) \in \mathbb{R}^n \times \mathbb{B} : z = 1,\ f(x) \leq 0,\ l_i^1 \leq x_i \leq u_i^1,\ \forall i \in \{1, 2, ..., n\}\ \}$, *non empty, then:*

1. $\boldsymbol{\Gamma''}$ *is a valid convex relaxation for* $\boldsymbol{conv(\Gamma_0 \cup \Gamma_1)}$
2. $\boldsymbol{\{(x, z) \in \mathbb{R}^n \times \mathbb{B} : (x, z) \in \Gamma''\} \equiv \Gamma_0 \cup \Gamma_1}$

$$\textit{where } \Gamma'' = \left\{ \begin{array}{l} (x, z) \in \mathbb{R}^{n+1} : \\ z(cl\ q')\,(x/z) \leq 0,\ \ s.t.: \\ zl_i^1 + (1-z)l_i^0 \leq x_i \leq zu_i^1 + (1-z)u_i^0,\ \ \forall i \in \{1, 2, ..., n\}, \\ 0 \leq z \leq 1. \end{array} \right\},$$

avec $q' = (f \circ h_\emptyset)$, $h_\emptyset(\mathbb{R}^n \to \mathbb{R}^n)$ *defined as* $(h_\emptyset(y))_i = \begin{cases} y_i - \frac{(1-z)u_i^0}{z} & \forall i \in J_1, \\ y_i - \frac{(1-z)l_i^0}{z} & \forall i \in J_2. \end{cases}$

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

Connections to earlier works: theory Balas [4,5] was the first to introduce the explicit algebraic formulation of convex hulls of union of polyhedra in higher dimensional spaces. Generalizations and extensions have been established for unions of nonlinear convex sets [8,13,17]. More specific cases have been studied, in particular, Günlük and Linderoth [14] have entirely characterized the convex hull of the union of a point and a convex set in the space of original variables. We show that this characterisation is a sepcial case of Lemma 1, it is actually obtained by fixing $l_0 = u_0 = 0$ in the lemma's definition:

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

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

The fact that all the formulations we have introduced include topological closures of sets presents some technical issues. Practically speaking, when implementing the perspective mapping of f, one must find a way to bypass the division by zero. A first alternative is to use cutting planes methods. Instead of explicitly writing the convex hull formulas in a mathematical program, valid cuts are generated to strengthen the formulation (see [8,17,12] for more details). A second alternative proposed in [13,16,10], is to approximate the functions by adding epsilons to the corresponding constraints, avoiding therefore divisions by zero. In this work, our goal being to write exact convex formulations of "on/off" MINLPs, 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 results shown in this section in order to improve existing formulations.

3 Application: The Delay Constrained Routing Problem

In this section, we study an application occuring in Telecommunications, first introduced by Ben Ameur and Ouorou [3]. Modern telecommunication services include group-based real-time applications, such as online games and video conferencing which are very sensitive to transmission delays. These types of applications require consideration of both source-to-destination delay (i.e., packet delay from the source to all destinations) and inter-destination delay variation (i.e., the difference in packet delay from the source to different destinations) constraints. Commonly, routing problems are studied under mean end-to-end delay constraints. This approach is not appropriate when dealing with these new services since it ignores the heterogeneous nature of real worl networks. The models we consider take into account a delay for each type of commodity as well as a limit on the authorized number of active paths per commodity. Assuming that the transmission delay through a link depends on its capacity as well as on the traffic carried through, we next show how delay constraints can be integrated into routing problems. This application can be formulated by a mixed integer non-linear program including "on/off" constraints fitting perfectly into our hypotheses. We note that the model introduced in [3] is a particular case of ours, where the number of authorized paths is equal to one (mono-routing).

3.1 Mathematical models

Notations 1 *Let G be a finite oriented network,*

1. *Parameters:*
   - *V represents the set of vertices, E the set of arcs and K the set of commodities.*
   - *d_k denotes the k_{th} commodity, $v_k \in \mathbb{R}$ the quantity that need to be routed and α_k the delay guaranty corresponding to this demand, $\alpha_k \in \mathbb{R}$.*
   - *Each commodity has a set of candidate paths denoted by $P(k) = \{P_k^1, P_k^2, \ldots, P_k^{n_k}\}$, each one of these corresponding to a different routing for d_k. N represents the maximum authorized number of activated paths per demand.*
   - *For each arc e, $c_e \in \mathbb{R}$ represents its capacity and $w_e \in \mathbb{R}$ its cost.*
2. *Variables:*

- *We call ϕ_k^i the fraction of the k_{th} demand carried out by its i_{th} path, $\phi_k^i \in [0,1]$.*
- *z_k^i constitutes the binary variables indicating if path P_k^i is activated.*
- *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 any commodity k, in (6) the fraction of routed demand must be greater than 1 in order to guarantee the satisfaction of all demands. 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. (9) represent the main "on/off" delay constraints: the delay guarantee associated to a given commodity must be satisfied on its candidate path if and only 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 the fact that if $N = 1$, that is if only one path can be activated per commodity (i.e. mono-routing), the variables ϕ_k^i becomes 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.

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

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) having different continuous relaxations.

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

$$\min \sum_{e \in E} w_e x_e$$

$$s.t.\ (6),\ (7),\ (8),\ (10),\ (11) \text{ and } (12-14)$$

$$\sum_{e \in P_k^i} \frac{1}{c_e - x_e} \leq M - z_k^i (M - \alpha_k),\ \forall k \in K,\ \forall P_k^i \in P(k). \tag{9-a}$$

This formulation is exact if z_k^i is a binary variable and provided that constant M is big enough. When $z_k^i = 0$, the constraint (9-a) is redundant, due to the big-M quantity on the right side, when $z_k^i = 1$, the big-M disappears leading to the original delay constraint formula. Since the the quality of the bound corresponding to this formulation depends on the constant M, one has to compute it efficently. Ben Ameur and Ouorou [3] pointed out the fact 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 $\frac{1}{c_e - x_e} \leq \alpha_k - \sum_{e' \neq e} \frac{1}{c_{e'}}$. On the other hand, if the arc is not used at all, the corresponding delay remains lower than $\frac{1}{c_e}$. 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 $\alpha_k^i = \sum_{e \in P_k^i} \alpha_e$. Indeed, by definition, constraints $\sum_{e \in P_k^i} \frac{1}{c_e - x_e} \leq \sum_{e \in P_k^i} \alpha_e\ \forall k \in K,\ \forall P_k^i \in P(k)$ will always be satisfied.
Next, we introduce new formulations to this model, based on results established in Section 2. To every path P_k^i corresponds a delay constraint (9) in (P) that is written: $f^{(i,k)}(x) \leq 0$ if $z_k^i = 1$, with $f^{(i,k)} : \mathbb{R}^n \to \mathbb{R}$, $f^{(i,k)}(x) = \sum_{e \in P_k^i} \frac{1}{c_e - x_e} - \alpha_k$. The $f^{(i,k)}$ functions being closed convex functions, Lemma 1 can directly be applied leading to the following corollary.

Corollary 2 *Let:*

$f : \mathbb{R}_+^n \to \mathbb{R},\ f(x) = \sum_{i=1}^{n} \left(\frac{1}{c_i - x_i} \right) - b,\ b \geq 0.$

$\Gamma_0 = \{\ (x,\ z) \in \mathbb{R}^n \times \mathbb{B} : z = 0,\ 0 \leq x_i \leq u_i,\ \forall i \in \{1, 2, ..., n\}\ \},$

$\Gamma_1 = \{\ (x,\ z) \in \mathbb{R}^n \times \mathbb{B} : z = 1,\ f(x) \leq 0,\ l_i \leq x_i \leq u_i,\ \forall i \in \{1, 2, ..., n\}\ \}$, *non empty, then:*

$$\boldsymbol{conv\ (\Gamma_0 \cup \Gamma_1) = \{(x, z) \mid \exists y \in \mathbb{R}^n,\ } s.t.\ \boldsymbol{(x, y, z) \in cl(\Gamma)\}},$$

$$\textit{where } \Gamma = \left\{ \begin{array}{l} (x, y, z) \in \mathbb{R}^{2n+1} : \\ \sum_{i=1}^{n} \left(\frac{z^2}{z c_i - y_i} \right) - zb \leq 0, \\ x_i - (1 - z) u_i \leq y_i \leq x_i,\ \forall i \in \{1,\ 2,\ ...,\ n\}, \\ z l_i \leq y_i \leq z u_i,\ \forall i \in \{1,\ 2,\ ...,\ n\}, \\ 0 < z \leq 1. \end{array} \right\}$$

Proof f being a closed convex function, Lemma 1 applies, the above statement is obtained by replacing function f by its explicit expression.

As discussed previously, one can see that the nonlinear functions are undefined when $z_k^i = 0$. 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:*

$f : \mathbb{R}_+^n \rightarrow \mathbb{R},\ f(x) = \sum\limits_{i=1}^{n} \left(\frac{1}{c_i - x_i}\right) - b,\ b \geq 0.$

$\Gamma_0 = \{\ (x,\ z) \in \mathbb{R}^n \times \mathbb{B} : z = 0,\ 0 \leq x_i \leq u_i < c_i,\ \forall i \in \{1, 2, ..., n\}\ \},$

$\Gamma_1 = \{\ (x,\ z) \in \mathbb{R}^n \times \mathbb{B} : z = 1,\ f(x) \leq 0,\ l_i \leq x_i \leq u_i,\ \forall i \in \{1, 2, ..., n\}\ \}$, *non empty, and*

$$\Gamma^\epsilon = \left\{ \begin{array}{l} (x, y, z) \in \mathbb{R}^{2n+1} : \\ \sum\limits_{i=1}^{n} \left(\frac{z^2}{zc_i - y_i + (1-z)\epsilon}\right) - zb \leq 0, \\ x_i - (1-z)u_i \leq y_i \leq x_i,\ \forall i \in \{1,\ 2,\ ...,\ n\}, \\ zl_i \leq y_i \leq zu_i,\ \forall i \in \{1,\ 2,\ ...,\ n\}, \\ 0 \leq z \leq 1. \end{array} \right\},\ \textit{then}$$

1. $\boldsymbol{proj_{(x,z)}(\Gamma^\epsilon)}$ *is a valid convex relaxation for* $\boldsymbol{conv(\Gamma_0 \cup \Gamma_1)}$
2. $\boldsymbol{\{(x, z) \in \mathbb{R}^n \times \mathbb{B} : (x, z) \in proj_{(x,z)}(\Gamma^\epsilon)\} \equiv \Gamma_0 \cup \Gamma_1}$

Proof First, we show that all constraints of Γ^ϵ are convex, next we prove that the projection of the latter on the (x, z)-space contains both Γ_0 and Γ_1.
The only nonlinear constraint in Γ^ϵ is $g(y, z) \leq 0$ with $g(y, z) = \sum_{i=1}^{n} g_i(y_i, z) - b$ 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 g_i functions are all convex, g being a sum of convex functions is therefore convex. On the other hand, having $\frac{z^2}{zc_i - y_i + (1-z)\epsilon} \leq \frac{z^2}{zc_i - y_i}$, the validity of these constraints is preserved.
Let us now consider the projection of Γ^ϵ on the (x, z) space:

For $z = 0$ we have

$$\Gamma^\epsilon = \left\{ \begin{array}{l} (x, y, 0) \in \mathbb{R}^{2n+1} : \\ x_i \leq u_i,\ y_i = 0,\ \forall i \in \{1,\ 2,\ ...,\ n\}, \\ x_i \geq y_i,\ \forall i \in \{1, 2, ..., n\}, \end{array} \right\},$$

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

For $z = 1$ we have

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

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

Based on this proposition, we introduce a new convex MINLP equivalent to (P) wich 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:

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

Let Nb_{max} be the maximum number of candidate paths per commodity, up to $|E| \times |K| \times Nb_{max}$ variables can be added in this new model. While bounds corresponding to this model 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 bellow corollary represents a first step toward a tight model defined in the space of original variables.

Corollary 3 *Let:*

$f : \mathbb{R}_+^n \to \mathbb{R},\ f(x) = \sum_{i=1}^{n} \left(\frac{1}{c_i - x_i} \right) - b,\ b \geq 0.$

$\Gamma_0 = \{\ (x,\ z) \in \mathbb{R}^n \times \mathbb{B} : z = 0,\ 0 \leq x_i \leq u_i < c_i,\ \forall i \in \{1, 2, ..., n\}\ \},$

$\Gamma_1 = \{\ (x,\ z) \in \mathbb{R}^n \times \mathbb{B} : z = 1,\ f(x) \leq 0,\ l_i \leq x_i \leq u_i,\ \forall i \in \{1, 2, ..., n\}\ \}$, *non empty, and*

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

1. $\boldsymbol{\Gamma_r^\epsilon}$ *is a valid convex relaxation for* $\boldsymbol{conv(\Gamma_0 \cup \Gamma_1)}$
2. $\boldsymbol{\{(x,z) \in \mathbb{R}^n \times \mathbb{B} : (x,z) \in \Gamma_r^\epsilon\} \equiv \Gamma_0 \cup \Gamma_1}$

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

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

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

In Corollary 1, we decide to add only one nonlinear constraint in the formulation, knowing that an exponentinal number of nonlinear constraints can be generated (see Theorem 2). In our computationnal testing, we noticed that adding only this specific nonlinear constraint is sufficient to have good bounds for the considered MINLP, and that generating some of the other nonlinear constraints only slows down the nonlinear relaxation algorithm.

Finally, below, we present a model based on the state of the art formulations of convex hulls in disjunctive programming, found in [8,13,17].

Higher space convex hull relaxation model: (P_{high}) This formulation uses directly the theorem introduced in [8] without taking into account results of Lemma 1.

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

In the next subsection the four models are compared on real world networks in order to evaluate the efficiency of each.

Connections to earlier works: application Ben Ameur and Ouorou [3] have introduced the following convex reformulation of the "on/off" delay constraint:

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

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
$\sum_{e\in P_k^i}\left(\frac{{z_k^i}^2}{c_e-x_e}\right) - z_k^i\alpha_k \leq \sum_{e\in P_k^i}\left(\frac{{z_k^i}^2}{z_k^i c_e - x_e + (1-z_k^i)(u_e+\epsilon)}\right) - z_k^i\alpha_k,\ \forall k\in K,\ \forall P_k^i\in P(k).$

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

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

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

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

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

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

4 Computational testings

We compare the four models presented in Section 3 on real world networks (denoted rdatax) as well as on randomly generated networks (denoted adatax), with up to 100 vertices and 1000 commodities. All models are implemented in C++ and solved with Bonmin (release 1.1.3) [6] an open source convex MINLP solver (see `http://www.coin-or.org/Bonmin`). The time limit for Bonmin is set to 2 hours. The underlying MILP solver used is CBC [11] and the nonlinear programming solver is Ipopt [18]. All tests were performed on an Intel Xeon 1.6 Ghz CPU.
Bonmin offers the possibility to choose one of five solution algorithms: a nonlinear programming based Branch & Bound [9], an Outer Approximation decomposition [10], and three branch-and-cut algorithms based on the Quessada Grossmann algorithm [16], a vanilla implementation of this algorithm, a hybrid method including a preliminary phase of Outer Approximation Decomposition and periodically adding outer approximation cuts, and finally, a method based on adding Extended Cutting Plane cuts [19] (similar to the method proposed in [1]). Here we report results obtained with the hybrid method since it appeared to be consistently better than the others (with all four models) in preliminary experiments. In the following tables, we compare the computing time to optimality and the number of nodes developped in the branch and bound with the four models, results are reported in tables with the following form: **(cpu time ; number of nodes)**. 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

Table 1 Mono-routing analysis, 2 paths per commodity

	$\|V\|$	$\|E\|$	$\|K\|$	P_{bigM}	P_{proj}	P_{high}	P_{red}
rdata1	60	280	100	**(0.4 ; 0)**	**(0.4 ; 0)**	(35 ; 0)	(2 ; 0)
rdata2	61	148	122	(190 ; 5193)	**(155 ; 1012)**	(2997 ; 21312)	(1948 ; 15129)
adata3	100	600	200	(144 ; 335)	**(57 ; 0)**	(206 ; 556)	(159 ; 84)
rdata4	34	160	946	**(3 ; 0)**	**(3 ; 0)**	(2040 ; 11691)	(1485 ; 5845)
rdata5	67	170	761	([∞] ; 14788)	**(251 ; 3357)**	(1549 ; 2793)	([0.03%] ; 15697)
adata6	100	800	500	**(1065 ; 27991)**	(1470 ; 42244)	([1%] ; 6831)	([0.3%] ; 2499)

Table 2 Mono-routing analysis, 3 paths per commodity.

	$\|V\|$	$\|E\|$	$\|K\|$	P_{bigM}	P_{proj}	P_{high}	P_{red}
rdata1	60	280	100	(2.7 ; 0)	**(2.4 ; 0)**	(2.8 ; 0)	(11.9 ; 0)
rdata2	61	148	122	(25 ; 0)	**(13 ; 0)**	(994 ; 3671)	(1396; 7815)
adata3	100	600	200	([0.28%] ; 157748)	(344; 5097)	(722 ; 3286)	**(312 ; 1124)**
rdata4	34	160	946	([0.001%] ; 79807)	**(1525 ; 50583)**	([0.04%] ; 28876)	([0.1%] ; 22438)
rdata5	67	170	761	([0.43%] ; 138618)	**([0.03%] ; 202122)**	([0.2%] ; 9472)	([0.14%] ; 9461)
adata6	100	800	500	([0.006%] ; 176413)	**(934 ; 19351)**	([0.6%] ; 16539)	([0.06%] ; 4067)

Table 3 mono-routing analysis, 10 paths per commodity.

	$\|V\|$	$\|E\|$	$\|K\|$	P_{bigM}	P_{proj}	P_{high}	P_{red}
rdata1	60	280	100	(568 ; 13649)	**(231 ; 4762)**	(1415 ; 9263)	(1209 ; 6485)
rdata2	61	148	122	(120 ; 0)	**(66 ; 0)**	(1527 ; 1599)	(1555 ; 2563)
adata3	100	600	200	**(534 ; 5118)**	(644 ; 14216)	(4866 ; 8841)	(6684 ; 11626)
rdata4	34	160	946	([1.9%] ; 79807)	([2.1%] ; 96212)	([∞] ; 3409)	**([1.8%] ; 3156)**
rdata5	67	170	761	([∞] ; 37446)	([∞] ; 30500)	([∞] ; 747)	([∞] ; 1568)
adata6	100	800	500	([2.7%] ; 35520)	**([1.5%] ; 2680)**	([∞] ; 2642)	([∞];1001)

that no integer feasible solution has been found after two hours of computing times. For each instance, the best computing time or the smallest gap is listed in bold characters.
In the following tables, we report results obtained for different networks and different parameters using the hybrid algorithm in Bonmin. In tables 1, 2 and 3, we consider the case when N = 1, that is 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 in tables 1, 2 and 3.

First, let us point out the fact that having zero node explored in the Branch & Bound means that the problem has been solved during the Outer Approximation Decomposition [10] initial phase of the hybrid algorithm (it means in no case that the initial continuous relaxation is integer feasible). The main observation is that the model based on the projected convex hull P_{proj} gives the best performance on these instances. P_{proj} solves 14 instances out of 18 while P_{bigM}, P_{high} and P_{red} solve respectively 10, 11 and 10 instances. If we consider geometric means, P_{proj} is 2.1 times faster than P_{bigM}, 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. If we look only at the four problems which are solved by P_{proj} but not solved by P_{bigM}, P_{proj} is at least one order of magnitude faster than P_{bigM} with at least 5 times less nodes to reach optimality (it is at least 5.45 and 7.17 times faster that P_{high} and P_{red}). One can conclude that even if P_{high} and P_{red} provide better continuous relaxations and usualy require less nodes, they remain slower due to the increased number of variables. Problem adata3 in Table 2 gives a good illustration of this: one can see that optimality is reached with P_{red} in only 1124 nodes, where the Big-M model explored 157748 nodes without reaching the optimum. However, P_{high} and P_{red} often remain slower to solve since they have an important number of additional variables. On the same problem, the continuous relaxation of P_{red} is solved in 3.5 secs while for P_{bigM} it takes only 0.5 secs. The projected model P_{proj} is able to give bounds almost as tight as the extended formulations, without having to deal with

Table 4 Bi-routing analysis, 2 paths per commodity.

	$\|V\|$	$\|E\|$	$\|K\|$	P_{bigM}	P_{proj}
rdata1	60	280	100	(0.8 ; 0)	**(0.4 ; 0)**
rdata2	61	148	122	**(2.4 ; 0)**	(10.9 ; 0)
adata3	100	600	200	(47.6 ; 0)	**(4.7 ; 0)**
rdata4	34	160	946	(5.6 ; 0)	**(3.4 ; 0)**
rdata5	67	170	761	**(38.3 ; 0)**	(50.4 ; 0)
adata6	100	800	500	**(40.1 ; 0)**	(42.4 ; 0)

Table 5 Bi-routing analysis, 3 paths per commodity.

	$\|V\|$	$\|E\|$	$\|K\|$	P_{bigM}	P_{proj}
rdata1	60	280	100	(16.7 ; 0)	**(2.9 ; 0)**
rdata2	61	148	122	(129.2 ; 18)	**(59 ; 0)**
adata3	100	600	200	(291.5 ; 760)	**(171.3 ; 615)**
rdata4	34	160	946	**(154.9 ; 62)**	(343.8 ; 472)
rdata5	67	170	761	([0.15%] ; 91430)	**(609.1 ; 5290)**
adata6	100	800	500	(1579.8 ; 8176)	**(747.7 ; 7056)**

Table 6 Bi-routing analysis, 10 paths per commodity.

	$\|V\|$	$\|E\|$	$\|K\|$	P_{bigM}	P_{proj}
rdata1	60	280	100	(1909 ; 56788)	**(399 ; 7846)**
rdata2	61	148	122	**(28.8 ; 0)**	(288.8 ; 666)
adata3	100	600	200	**([0.11%] ; 67705)**	([0.13%] ; 77129)
rdata4	34	160	946	([1.2%] ; 23984)	**([0.6%] ; 32939)**
rdata5	67	170	761	([2%] ; 11772)	**([1.2%] ; 16285)**
adata6	100	800	500	([0.7%] ; 6480)	**([0.14%] ; 22364)**

the inconvenience of large size problems (still on the same instance the continuous relaxation of P_{proj} is solved in 0.9 secs).
We now consider the bi-routing case, maximum two paths can be activated per commodity, in order to route fractions of the demand ($N = 2$).
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 but 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. Tables 4, 5 and 6 reports results obtained for instances with respectively 2, 3 and 10 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_{bigM} and P_{proj} solved respectively 13 and 14 instances. Instances having 2 paths per commodity seem very easy to solve (they are all solved in less than one minute with both formulations) and instances with 3 path seem much easier than before. On average P_{proj} is still faster than P_{bigM} taking about 1761 secs versus 2235 secs.
We finally consider the multiple-routing case where all the paths in $P(k)$ can be activated ($N = \infty$). Tables 7 and 8 reports respectively results obtained for instances with 3 and 10 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_{bigM} and P_{proj} solved respectively 8 and 11 instances out of 12. Now, all instances with 3 paths per commodity seem quite easy to solve with both formulations. On average, for all

Table 7 Multiple-routing analysis, 3 paths per commodity.

	$\|V\|$	$\|E\|$	$\|K\|$	P_{bigM}	P_{proj}
rdata1	60	280	100	(2.7 ; 0)	**(1.2 ; 0)**
rdata2	61	148	122	**(10 ; 0)**	(29.7 ; 0)
adata3	100	600	200	(195.7 ; 164)	**(50.5 ; 0)**
rdata4	34	160	946	**(20.7 ; 0)**	(43 ; 0)
rdata5	67	170	761	(2503.1 ; 34783)	**(362.8 ; 4024)**
adata6	100	800	500	(1482.4 ; 6083)	**(224.7 ; 326)**

Table 8 Multiple-routing analysis, 10 paths per commodity

	$\|V\|$	$\|E\|$	$\|K\|$	P_{bigM}	P_{proj}
rdata1	60	280	100	(799.7 ; 12633)	**(220.8 ; 1922)**
rdata2	61	148	122	**(16.1 ; 0)**	(24.8 ; 0)
adata3	100	600	200	([0.08%] ; 94194)	**(768.6 ; 5207)**
rdata4	34	160	946	([0.4%] ; 40820)	**([0.04%] ; 45492)**
rdata5	67	170	761	([1.2%] ; 16106)	**(5467.7 ; 17347)**
adata6	100	800	500	([0.7%] ; 5880)	**(5392 ; 23867)**

instances, P_{proj} takes 1649 secs while P_{bigM} takes 2819 secs. For the instance not solved by both formulations, P_{proj} final gap is about one order of magnitude smaller than the P_{bigM} one.

5 Conclusion

Studying and writing explicit formulations of union of convex sets is a main topic in disjunctive programming. While many works deal with finding the explicit formulation of convex hulls for unions of general convex sets, we looked closely at specific two set cases: the union of a hyper-rectangle and a closed convex set defined by one nonlinear constraint. On one hand, we have established formulations in reduced dimensional spaces, on the other hand, we showed that when the nonlinear function defining the convex set is order preserving, we can give an explicit characterization of the convex hull in the space of original variables. Using these results, we could propose new formulations for "The Delay Constrained Routing Problem", an up-to-date telecommunication problem. Numerical testings showed that the new formulations allow to solve new instances with gains up to one order of magnitude of computing time compared to the classical models. Our results can be directly applied to any MINLP featuring on/off constraints, leading to more efficient mathematical models and hopefully better computing times.

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. 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.
7. P. Bonami, G. Cornuéjols, A. Lodi, and F. Margot. A feasibility pump for mixed integer nonlinear programs. *Mathematical Programming*, 119(2):331–352, 2009.
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. J. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms*, volume 2. Springer-Verlag, Berlin Heidelberg New York, 1993.
16. 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.
17. R. Stubbs and S. Mehrotra. A branch-and-cut method for 0-1 mixed convex programming. *Mathematical Programming*, 86:515–532, 1999.
18. 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.
19. T. Westerlund and F. Pettersson. A cutting plane method for solving convex MINLP problems. *Computers and Chemical Engineering*, 19:s131–s136, 1995.