Constraint Qualification Failure in Second-Order Cone Formulations of Unbounded Disjunctions

Hassan Hijazi, Leo Liberti

Abstract

This note presents a theoretical analysis of disjunctive constraints featuring unbounded variables. In this framework, classical modeling techniques, including big-M approaches, are not applicable. We introduce a lifted second-order cone formulation of such on/off constraints and discuss related constraint qualification issues.

Keywords: mixed-integer nonlinear programming, disjunctive programming, second-order cone programming, on/off constraints, constraint qualification

1. Introduction

Disjunctions represent a key element in mixed-integer programming. One can start with basic disjunctions coming from the discrete condition imposed on integer variables, e.g. \((z = 0) \lor (z = 1)\), then consider more complex disjunctions of the form \((z = 0 \land x \geq 0) \lor (z = 1 \land f(x) \leq 0)\).

In mixed-integer linear programming, years of research have been devoted to study disjunctive cuts based on basic disjunctions in Branch & Cut algorithms [13, 16, 2]. For more complex disjunctions, especially in convex Mixed-Integer Nonlinear Programs (MINLPs), the disjunctive programming approach [8] consists of automatically reformulating each disjunction, with the concern of preserving convexity.

In most real-life applications, decision variables are naturally bounded, or artificial bounds can be enforced without losing relevant solutions. There are, however, some cases where unbounded variables are necessary. In both [6] and [14], there appear mathematical programs involving decision variables which represent step counters in an abstract computer description. Unboundedness in these directions amounts to a proof of non-termination of the abstract computer. Artificially bounding these variables deeply changes the significance of the mathematical program, which implies that these variables have to remain unbounded. The original motivation behind this work was inspired by indicator constraints appearing in large mixed-integer programs.

1Financial support by grants: Digiteo Emergence “PASO”, Digiteo Chair 2009-14D “RMNCCO”, Digiteo Emergence 2009-55D “ARM” is gratefully acknowledged.

NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.
arising in this context. Commercial solvers were unable to return optimal solutions when using conic reformulations of such on/off constraints, which initiated the current line of study, trying to identify the simplest model exhibiting similar irregularities. To the best of our knowledge, this is a first attempt to characterize constraint qualification failure in solutions of second-order cone programs at a sufficiently tractable level.

Given a disjunctive or indicator constraint, two main reformulation techniques exist in the literature. On the one hand, the big-M approach introduces large constants allowing to enable/disable the corresponding constraint. On the other hand, convex hull-based formulations aim at defining the convex hull of each disjunction. In Section 2, we present the convex hull formulation and show that it is not suitable in this framework, ruling out mixed-integer linear approaches. A new lifted second-order cone formulation is proposed in Section 3, and constraint qualification issues are discussed in Section 4.

2. Unbounded disjunctions

Given \((x, z) \in \mathbb{R}^n \times \{0, 1\}\), we consider general disjunctions of the form:

\[
(z = 0 \land f(x) \leq 0) \lor (z = 1 \land x \in \mathbb{R}^n)
\]  

(\(\star\))

The general constraint \(f(x) \leq 0\) can be factored out by introducing an artificial variable \(y \in \mathbb{R}\). (\(\star\)) becomes:

\[
(z = 0 \land f(x) \leq y \land y = 0) \lor (z = 1 \land f(x) \leq y \land y \in \mathbb{R})
\]

Now, we only need to model the union \(\Gamma_0 \cup \Gamma_1\), where \(\Gamma_0 = \{(z, y) \mid z = 0 \land y = 0\}\) and \(\Gamma_1 = \{(z, y) \mid z = 1 \land y \in \mathbb{R}\}\).

For any set \(S\), let \(\text{conv}(S)\) denote the convex hull of \(S\).

**Lemma 1.** Let \(\Gamma_c = \{(y, z) \in \mathbb{R}^2 \mid 0 < z \leq 1\}\). Then \(\text{conv}(\Gamma_0 \cup \Gamma_1) = \Gamma_0 \cup \Gamma_c\).

**Proof.** Refer to [4] (Section 3). An illustration is given in Figure 1.

![Figure 1: The convex hull \(\text{conv}(\Gamma_0 \cup \Gamma_1)\) is the shaded region.](image)

Lemma 1 indicates that the convex hull approach for unbounded disjunctions leads to a non-algebraic description of the feasible region and thus is not useful in mathematical programming.
**MIP representability.** Determining if a set is MIP representable is a fundamental property implying that the set can be exactly characterized by a mixed-integer linear program. Jeroslow and Lowe [11] introduce the following necessary condition for a set to be MIP representable:

**Proposition 1.** [11] If \( S \subseteq \mathbb{R}^n \) is MIP representable, then \( S \) is closed and \( \text{conv}(S) \) is a polyhedron.

Based on this result, we can guarantee that the set \( \Gamma_0 \cup \Gamma_1 \) is not MIP representable since its convex hull is not a polyhedron. This rules out the big-M approach, or other mixed-integer linear formulations.

3. **A second-order cone lifted formulation**

In order to have an algebraic description of the disjoint regions, we perform a further lifting step in variable \( \gamma \), introducing set \( \Gamma \):

\[
\Gamma = \begin{cases} 
\gamma z \geq y^2, \\
\gamma \geq 0, \\
z \in \{0, 1\}, \ y \in \mathbb{R}, \ \gamma \in \mathbb{R}.
\end{cases}
\]

Let \( \text{proj}_{(z,y)}(\Gamma) \) denotes the projection of \( \Gamma \) on the \((z,y)\) subspace.

**Proposition 2.** We have that \( \text{proj}_{(z,y)}(\Gamma) = \Gamma_0 \cup \Gamma_1 \).

**Proof.** For \( z = 0 \), the constraint \( \gamma z \geq y^2 \) forces \( y = 0 \), which corresponds to the definition of \( \Gamma_0 \). For \( z = 1 \), since \( \gamma \geq 0 \), the constraint \( \gamma \geq y^2 \) becomes redundant, thus \( y \) can take any value in \( \mathbb{R} \) matching the definition of \( \Gamma_1 \).

Notice that the constraint \( \gamma z \geq y^2 \) defines a rotated second-order cone in \( \mathbb{R}^2 \) (see Figure 2). It is therefore second-order cone representable [17, 1], since it can be written as \( 4y^2 + (\gamma - z)^2 \leq (\gamma + z)^2 \). On the one hand, if \( z = 1 \), this region is delimited by the parabola corresponding to the equation \( y = \gamma^2 \), where \( y \) is unbounded. On the other hand, if \( z = 0 \) the curve converges to the vertical axis defined by the system \( \{z = 0, \ y = 0, \ \gamma \geq 0\} \).

4. **Constraint qualification**

For an extensive study of constraint qualifications, we refer the reader to the excellent survey by Wang et al. in [19]. We consider the program,

\[
\begin{align*}
\min \quad & f(x) \\
\text{subject to} \quad & g_i(x) \leq 0 \\
& x \in \mathbb{R}^n,
\end{align*}
\]

where \( G = \{1, \ldots, m\} \), and all functions are assumed to be convex and differentiable.

Let \( \mathcal{F} \) denote the set of feasible points

\[
\mathcal{F} = \{x \in \mathbb{R}^n \mid \forall i \in G \ g_i(x) \leq 0\}.
\]
Figure 2: The surface $\gamma z = y^2$.

Given a feasible point $\hat{x}$, $A(\hat{x})$ denotes the corresponding set of active constraints

$$A(\hat{x}) = \{ i \in G \mid g_i(\hat{x}) = 0 \},$$

and $\mathcal{D}$ represents the cone of feasible directions at $\hat{x}$:

$$\mathcal{D}(\hat{x}) = \{ d \in \mathbb{R}^n \mid \exists T > 0, \forall t \in [0, T] \quad \hat{x} + td \in \mathcal{F} \}.$$  

$\mathcal{D}$ is a subset of the cone of tangent directions at $\hat{x}$, denoted $\mathcal{T}(\hat{x})$. Since $\mathcal{F}$ is a convex set, a direction is tangent to $\mathcal{F}$ at $\hat{x}$ iff it is representable as the limit of a sequence of feasible directions.

$$\mathcal{T}(\hat{x}) = \left\{ d \in \mathbb{R}^n \mid d = \lim_{k \to \infty} d_k, \quad d_k \in \mathcal{D}(\hat{x}) \right\}$$

Finally, define $\mathcal{G}(\hat{x})$ to be the cone of locally constrained directions at $\hat{x}$

$$\mathcal{G}(\hat{x}) = \left\{ d \in \mathbb{R}^n \mid \forall i \in A(\hat{x}) \quad \nabla g_i(\hat{x})^\top d \leq 0 \right\}.$$  

$\mathcal{G}$ can be seen as a linear algebraic description of the set of feasible directions. Nonlinear optimization algorithms, and precisely interior point methods, base their proof of convergence on constraint qualification conditions. In order to reach a minimum point $x^\ast$, the latter should satisfy some regularity conditions. This is mainly due to the fact that the locally constrained cone at a given point, may be different from the set of tangent directions (see [15]). This happens when the algebraic description of feasible directions differs from the geometric one. KKT optimality conditions [12] are based on the locally constrained cone and are no longer necessary if the latter does not coincide with the geometric definition.

In the following, we prove that this is the case for the second-order cone formulation introduced previously.
Let $F$ be the set given by:

$$
\begin{align*}
4y^2 + (\gamma - z)^2 & \leq (\gamma + z)^2 \\
z & = 0 \\
\gamma & \in \mathbb{R}^+ \land y \in \mathbb{R} \land z \in [0, 1]
\end{align*}
$$

The set $F$ represents the feasible region of a typical lower bounding relaxation occurring in a Branch-and-Bound (BB) algorithm on the binary variables $z$, along a branch $z = 0$.

**Proposition 3.** Points in $F$ are not regular with respect to any constraint qualification.

**Proof.** Consider a feasible point $x_0 \in F$:

$$
\begin{pmatrix}
y \\
z \\
\gamma
\end{pmatrix}_0 = \begin{pmatrix}
0 \\
0 \\
\gamma_0
\end{pmatrix}
$$

If $\gamma_0 = 0$, the locally constrained cone of $F$ at $x_0$ is defined as

$$
\mathcal{G}(x_0) = \{d \in \mathbb{R}^3 \mid d_2 = 0, d_3 \geq 0\}
$$

The cone of feasible directions at $x_0$ is defined as

$$
\mathcal{D}(x_0) = \{d \in \mathbb{R}^3 \mid d_1 = d_2 = 0, d_3 \geq 0\}
$$

Note that, since $F$ is convex,

$$
\mathcal{T}(x_0) = \text{cl}(\mathcal{D}(x_0)) = \mathcal{D}(x_0) \implies \mathcal{T}(x_0) \neq \mathcal{G}(x_0),
$$

where $\text{cl}(\cdot)$ denotes the closure. If $\gamma_0 > 0$, the same reasoning applies, with $\mathcal{G}(x_0) = \{d \in \mathbb{R}^3 \mid d_2 = 0\}$ and $\mathcal{T}(x_0) = \{d \in \mathbb{R}^3 \mid d_1 = d_2 = 0\}$. Based on [7], $\mathcal{T}(x_0) = \mathcal{G}(x_0)$ is a necessary and sufficient condition for optimal points to be KKT. Since this weakest possible constraint qualification is not satisfied, the proof is completed.

This is a negative result indicating that all derivative based algorithms may not converge to the unique global optimal solution, even though the feasible region is convex. This has been observed in practice on the example below.

### 4.1. A breach in state-of-the-art solvers?

In order to evaluate the second-order cone formulation in practice, we consider the following program:

$$
\begin{align*}
\text{min} & \quad x^2 + z \\
\text{s.t.} & \quad x - 4 \geq 0 \text{ if } z = 0, \\
& \quad x \geq 0, \\
& \quad x \in \mathbb{R}, z \in [0, 1]
\end{align*}
$$

(1)

Its SOCP reformulation is defined as,

$$
\begin{align*}
\text{min} & \quad x^2 + z \\
\text{s.t.} & \quad x - 4 \geq y, \\
& \quad 4y^2 + (\gamma - z)^2 \leq (\gamma + z)^2, \\
& \quad \gamma \geq 0, x \geq 0, \\
& \quad (x, y, \gamma) \in \mathbb{R}^3, z \in [0, 1]
\end{align*}
$$

(2)
By constraining $z = 0$, we have $4y^2 \leq 0$, implying $y = 0$ and $x = 4$, therefore, the optimal solution value is 1 with $x^* = 0$ and $z^* = 1$. Program (2) was given to Cplex 12.6 [10], Gurobi 5.6 [9], and Bonmin 1.5 [3], representing state-of-the-art solvers. All fail to find the optimal solution. Cplex returns an “unrecoverable failure”, Gurobi reports an optimal solution of 16, and Bonmin claims the problem is infeasible. This is mainly due to the fact that branching on $z$, generates two subproblems ($z = 0$ and $z = 1$), one of which is irregular as underlined in Proposition 3. This is further confirmed by relaxing the binary condition in Model (2) and posting the constraint $z \leq 0$. On the resulting continuous program, Cplex 12.6 [10] converges to an infeasible point and Ipopt [18] reaches its number of iterations limit. Note that Gurobi is unable to solve the root relaxation of Model (2), returning “NaN”.

5. Conclusion

Constraint qualification failure can lead to irregular situations where optimal solutions do not satisfy the KKT system. Under such circumstances, interior point methods, relying on the latter system, may fail to converge. In mixed-integer programming, branching is performed by introducing linear equations which fix (or bound) a subset of the discrete variables. While this approach seems harmless in the linear case, it might produce degeneracy in nonlinear systems.

References


