RC25480 (WAT1407-003) July 8, 2014
Mathematics

IBM Research Report

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

Hassan L. Hijazi
NICTA
7 London Circuit
Canberra ACT 2601
Australia
and
The Australian National University
Canberra ACT 0200
Australia

Leo Liberti
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598
USA

Research Division
Almaden – Austin – Beijing – Cambridge – Dublin - Haifa – India – Melbourne - T.J. Watson – Tokyo - Zurich

Constraint qualification failure in second-order cone formulations of unbounded disjunctions

Hassan L. Hijazi · Leo Liberti

July 7, 2014

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. $(z = 0) \vee (z = 1)$, then consider more complex disjunctions of the form $(z = 0 \wedge x \geq 0) \vee (z = 1 \wedge f(x) \leq 0)$. In mixed-integer linear programming, years of research have been devoted to study disjunctive cuts based on basic disjunctions in Branch & Cut (BC) algorithms [11,14,2]. 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 can at least be bounded by a very slack bound without losing any interesting solutions. There are, however, some cases where unbounded variables are necessary. In both [6] and [12], there appear mathematical programs involving decision variables

Hassan L. Hijazi
NICTA - 7 London Circuit - Canberra ACT 2601 Australia
The Australian National University - Canberra ACT 0200 Australia
E-mail: hassan.hijazi@nicta.com.au

Leo Liberti
IBM TJ Watson Research Center, Yorktown Heights, 10598 NY, USA
LIX, Ecole Polytechnique, 91128, Palaiseau, France
E-mail: leoliberti@us.ibm.com,liberti@lix.polytechnique.fr

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 should really be unbounded.

Two main reformulation techniques exist for disjunctions in mathematical programming. The "big-M" approach introduces large constants allowing to enable/disable a given constraint. Convex hull-based formulations, on the other hand, aim at defining the convex hull of each disjunction.

In this short note, we show that neither of the two approach is suitable for expressing on/off constraints featuring unbounded variables.

2 Unbounded disjunctions

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

$$(z = 0 \wedge f(\mathbf{x}) \leq 0) \vee (z = 1 \wedge \mathbf{x} \in \mathbb{R}^n) \tag{$\star$}$$

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

$$(z = 0 \wedge f(\mathbf{x}) \leq y \wedge y = 0) \quad \vee \quad (z = 1 \wedge f(\mathbf{x}) \leq y \wedge y \in \mathbb{R})$$

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

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

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

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

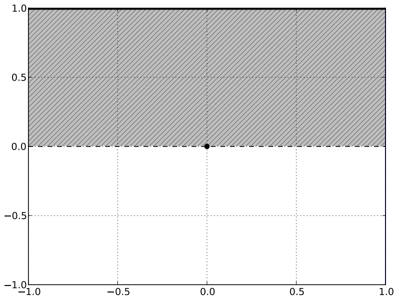

Fig. 1 The convex hull conv($\Gamma_0 \cup \Gamma_1$) is the shaded region.

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.

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 γ, introducing set Γ:

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

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

Proposition 1 *We have that* $\mathsf{proj}_{z,y}(\Gamma) = \Gamma_0 \cup \Gamma_1$.

Proof For $z = 0$, the constraint $\gamma z \geq y^2$ forces $y = 0$, which corresponds to the definition of Γ_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 Γ_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 [15,1], 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 $\gamma = y^2$, where y is unbounded. On the other hand, if $z = 0$ the curve converges to the vertical axis defined by the system $\{z = 0, \ y = 0, \ \gamma \geq 0\}$.

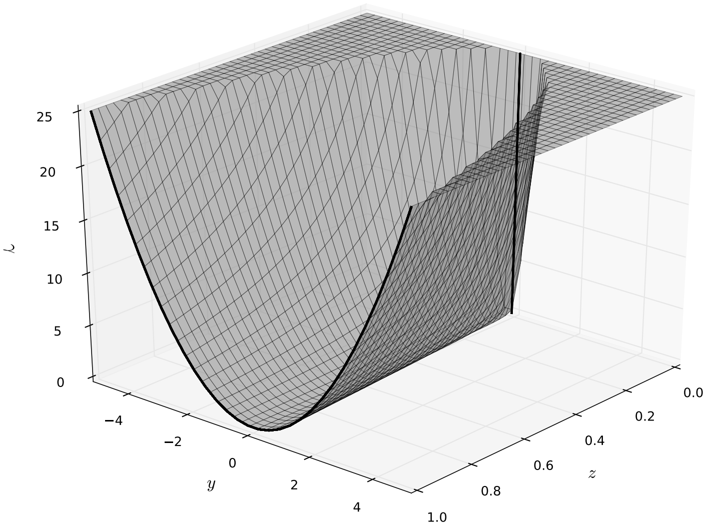

Fig. 2 The surface $\gamma z = y^2$.

4 Constraint qualification

For an extensive study of constraint qualifications, we refer the reader to the excellent survey by Wang et al. in [16]. We consider the program,

$$\begin{aligned} \min\ & f(\mathbf{x}) \\ \forall i \in G\ & g_i(\mathbf{x}) \le 0 \\ & \mathbf{x} \in \mathbb{R}^n, \end{aligned} \tag{P}$$

where $G = \{1, \dots, m\}$, and all functions are assumed to be convex and differentiable.

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

$$\mathcal{F} = \{x \in \mathbb{R}^n \mid \forall i \in G\ g_i(\mathbf{x}) \le 0\}.$$

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

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

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

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

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

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

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

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

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^*, 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 [13]). In the following, we prove that this is the case for the second-order cone formulation introduced previously.

Let $\mathcal{F}$ be the set given by:

$$\begin{cases} 4y^2 + (\gamma - z)^2 \le (\gamma + z)^2 \\ \gamma \ge 0 \wedge 0 \le z \le 1 \wedge y \in \mathbb{R} \\ z = 0 \end{cases}$$

The set $\mathcal{F}$ is the interesting part of 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 2 *Points in $\mathcal{F}$ are not regular with respect to any constraint qualification.*

Proof Consider a feasible point $\mathbf{x}_0 \in \mathcal{F}$:

$$\begin{array}{ccccccc} & & & \mathrm{y} & \mathrm{z} & \gamma & \\ \mathbf{x}_0 & = & (& 0 & 0 & \gamma_0 &) \end{array}$$

If $\gamma_0 = 0$, the locally constrained cone of $\mathcal{F}$ at $\mathbf{x}_0$, is defined as

$$\mathcal{G}(\mathbf{x}_0) = \{\mathbf{d} \in \mathbb{R}^3 \mid d_2 = 0,\ d_3 \geq 0\}$$

The cone of feasible directions at $\mathbf{x}_0$ is defined as

$$\mathcal{D}(\mathbf{x}_0) = \{\mathbf{d} \in \mathbb{R}^3 \mid d_1 = d_2 = 0,\ d_3 \geq 0\}$$

Note that, since $\mathcal{F}$ is convex,

$$\mathcal{T}(\mathbf{x}_0) = \mathsf{cl}\left(\mathcal{D}(\mathbf{x}_0)\right) = \mathcal{D}(\mathbf{x}_0) \implies \mathcal{T}(\mathbf{x}_0) \neq \mathcal{G}(\mathbf{x}_0),$$

where $\mathsf{cl}(\cdot)$ denotes the closure. If $\gamma_0 > 0$ the same reasoning applies, with $\mathcal{G}(\mathbf{x}_0) = \{\mathbf{d} \in \mathbb{R}^3 \mid d_2 = 0\}$ and $\mathcal{T}(\mathbf{x}_0) = \{\mathbf{d} \in \mathbb{R}^3 \mid d_1 = d_2 = 0\}$. Based on [7], $T(\mathbf{x}_0) = \mathcal{G}(\mathbf{x}_0)$ 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 simple program:

$$\begin{aligned} &\min\ x^2 + z \\ &\text{s.t.}\ \ x - 4 \geq 0 \text{ if } z = 0, \\ &\qquad x \geq 0, \\ &\qquad x \in \mathbb{R},\ z \in \{0,1\} \end{aligned} \tag{1}$$

and implement its SOCP reformulation in AMPL [5]:

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

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 2.

5 Discussion and Future Perspectives

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

1. F. Alizadeh and D. Goldfarb. Second-order cone programming. *Mathematical Programming*, 95:3–51, 2001.
2. E. Balas. Disjunctive programming. In Michael Jünger, Thomas M. Liebling, Denis Naddef, George L. Nemhauser, William R. Pulleyblank, Gerhard Reinelt, Giovanni Rinaldi, and Laurence A. Wolsey, editors, *50 Years of Integer Programming 1958-2008*, pages 283–340. Springer Berlin Heidelberg, 2010.
3. P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuejols, 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.
4. M. Conforti, G. Cornuéjols, and G. Zambelli. Polyhedral approaches to mixed integer linear programming. In *50 Years of Integer Programming 1958-2008*, pages 343–385. Springer Berlin Heidelberg, 2010.
5. R. Fourer, D. M. Gay, and B. W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming.* The Scientific Press, 1993.
6. E. Goubault, S. Le Roux, J. Leconte, L. Liberti, and F. Marinelli. Static analysis by abstract interpretation: a mathematical programming approach. In A. Miné and E. Rodriguez-Carbonell, editors, *Proceeding of the Second International Workshop on Numerical and Symbolic Abstract Domains*, volume 267(1) of *Electronic Notes in Theoretical Computer Science*, pages 73–87. Elsevier, 2010.
7. F. Gould and J. Tolle. A necessary and sufficient qualification for constrained optimization. *SIAM Journal on Applied Mathematics*, 20(2):164–172, 1971.
8. I. E. Grossmann and S. Lee. Generalized convex disjunctive programming: Nonlinear convex hull relaxation. *Computational Optimization and Applications*, 26:83–100, 2003. 10.1023/A:1025154322278.
9. Gurobi. Gurobi software. http://www.gurobi.com/.
10. IBM. *ILOG CPLEX 12.2 User's Manual.* IBM, 2010.
11. A. Letchford. On disjunctive cuts for combinatorial optimization. *Journal of Combinatorial Optimization*, 5:299–315, 2001. 10.1023/A:1011493126498.
12. L. Liberti and F. Marinelli. Mathematical programming: Turing completeness and applications to software analysis. *Journal of Combinatorial Optimization*, to appear.
13. J. P. Penot. A new constraint qualification condition. *Journal of Optimization Theory and Applications*, 48:459–468, 1986.
14. M. Perregaard. *Generating Disjunctive Cuts for Mixed Integer Programs.* PhD thesis, Carnegie Mellon University, 2003.
15. M. S.Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284(1-3):193 – 228, 1998. International Linear Algebra Society (ILAS) Symposium on Fast Algorithms for Control, Signals and Image Processing.
16. Z. T. Wang, S. C. Fang, and W. X. Xing. On constraint qualifications: motivation, design and inter-relations. *Journal of Industrial and Management Optimization*, 9(4):983–1001, 2013.