

Constraint Qualification Failure in Action

Hassan Hijazi^{a,*}, Leo Liberti^b

^aThe Australian National University, Data61-CSIRO, Canberra ACT 2601, Australia

^bCNRS, LIX, Ecole Polytechnique, 91128, Palaiseau, France

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. A solution is proposed to avoid solvers' failure.

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 algorithms [12, 15, 2]. For more complex disjunctions, especially in convex Mixed-Integer Nonlinear Programs (MINLPs), the disjunctive programming approach [7, 10] 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 [5] and [13], there appear mathematical programs involving decision variables which represent step counters in an abstract computer description. Unboundedness in these frameworks amounts to a proof of non-termination of the abstract computer. Artificially bounding these variables deeply changes the significance of the mathematical program. In [8], Guan et al. use unbounded on/off constraints to model support vector machines.

Two main reformulation techniques exist for disjunctions in mathematical programming. There is the “big-M” approach, introducing large constants allowing to enable/disable a given constraint, and the convex hull-based formulations, aiming at defining the convex hull of each disjunction.

*Corresponding author

Email address: hassan.hijazi@anu.edu.au (Hassan Hijazi)

In Section 2, we present the convex hull formulation and show that it is not suitable in this framework. A new lifted second-order cone formulation is proposed in Section 3, and constraint qualification issues are presented in Section 4.

2. Unbounded disjunctions

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

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

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

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

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 $\text{conv}(S)$ 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 [3] (Section 3). An illustration is given in Figure 1. □

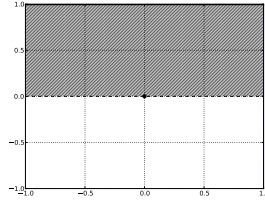


Figure 1: The convex hull $\text{conv}(\Gamma_0 \cup \Gamma_1)$ is the shaded region.

Lemma 1 indicates that the convex hull approach for unbounded disjunctions leads to a non-algebraic description of the feasible region.

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

Proposition 1. *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 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 . \square

Notice that given $\gamma \geq 0$ and $z \geq 0$, 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 [16, 1], and can be written as

$$4y^2 + (\gamma - z)^2 \leq (\gamma + z)^2, \text{ with } \gamma + z \geq 0.$$

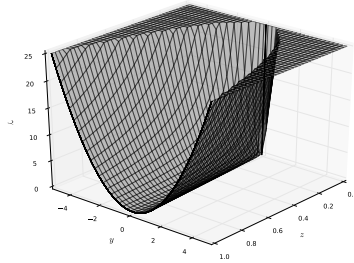


Figure 2: The surface $\gamma z = y^2$.

4. Constraint qualification

For an extensive study of constraint qualifications, we refer the reader to the excellent survey by Wang et al. in [18]. We consider the program,

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

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

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

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

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.

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

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

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

\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^* , 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 [14]). This happens when the algebraic description of feasible directions differs from the geometric one. KKT optimality conditions [11] 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 \mathcal{F} be the set given by:

$$\begin{cases} 4y^2 + (\gamma - z)^2 \leq (\gamma + z)^2 \\ z \leq 0 \\ \gamma \in \mathbb{R}^+ \wedge y \in \mathbb{R} \wedge z \in [0, 1] \end{cases}$$

The set \mathcal{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 $z \leq 0$.

Proposition 2. *Points in \mathcal{F} are not regular with respect to any constraint qualification.*

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

$$\hat{\mathbf{x}} = \begin{pmatrix} y & z & \gamma \\ 0 & 0 & \gamma_0 \end{pmatrix}$$

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

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

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

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

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

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

where $\text{cl}(\cdot)$ denotes the closure operator. If $\gamma_0 > 0$, the same reasoning applies, with $\mathcal{G}(\hat{\mathbf{x}}) = \{\mathbf{d} \in \mathbb{R}^3 \mid d_2 = 0\}$ and $\mathcal{T}(\hat{\mathbf{x}}) = \{\mathbf{d} \in \mathbb{R}^3 \mid d_1 = d_2 = 0\}$. Based on [6], $\mathcal{T}(\hat{\mathbf{x}}) = \mathcal{G}(\hat{\mathbf{x}})$ is a necessary and sufficient condition for optimal points to be KKT. Since this is not satisfied, the proof is completed. \square

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 by Guan et al. in [8] while modeling on/off constraints for support vector machines using perspective relaxations. The authors state the following “We believe that the failure of the nonlinear solvers is due to a failure of a constraint qualification”. In the next section, we present a purified model exhibiting similar behaviors.

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

Consider the following program,

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

its SOCP reformulation is defined as,

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

By constraining $z = 0$, we have $y^2 \leq 0$, implying $y = 0$ and $x \geq 4$, therefore, the optimal objective value is 1 with $x^* = 0$ and $z^* = 1$. (MISOCP) was given to CPLEX 12.6 [4], and GUROBI 6.5 [9], representing state-of-the-art commercial solvers. Both fail to find the optimal solution. CPLEX returns an “unrecoverable failure”, and GUROBI reports an optimal solution of 0. This is mainly due to the fact that branching on z , generates two subproblems ($z \leq 0$ and $z \geq 1$), one of which is irregular, as underlined in Proposition 2.

4.2. Further analysis

In order to confirm the failure’s root, we propose to solve the continuous relaxation of (MISOCP) after simulating a branching on the binary variable, i.e., $z \leq 0$. This leads to the following model:

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

As expected, CPLEX returns “primal-dual infeasible; objective 4.12”, and GUROBI reports “suboptimal; objective 3.17e-06”. Note that the open source interior-point solver IPOPT 3.12.2 [17] exhibits a similar behavior, converging to an infeasible point with an optimal objective of “14.37”.

5. Patch

The degeneracy described in this note applies for general on/off rotated second-order cone constraints of the form:

$$\lambda z \geq \sum_{i \in N} y_i^2, \quad z \in \{0, 1\}, \quad \lambda \in \mathbb{R}^+, \quad y_i \in \mathbb{R}, \quad \forall i \in N. \quad (2)$$

The ‘‘off’’ version of this constraint creates an irregularity as both the constraint and its derivative are equal to zero. Recall that $\mathcal{G}(\hat{\mathbf{x}})$ denotes the cone of locally constrained directions at $\hat{\mathbf{x}}$, defined as $\mathcal{G}(\hat{\mathbf{x}}) = \{\mathbf{d} \in \mathbb{R}^n \mid \forall i \in A(x^*) \nabla g_i(\hat{\mathbf{x}})^\top \mathbf{d} \leq 0\}$. Let us emphasize that if $i \in A(x^*)$, i.e. $g_i(\hat{\mathbf{x}}) = 0$, and if $\nabla g_i(\hat{\mathbf{x}}) = 0$, the constraint $g_i(\hat{\mathbf{x}})^\top \mathbf{d} \leq 0$ is always true, leading to an information loss when defining the set of feasible directions, which may imply $\mathcal{T}(\hat{\mathbf{x}}) \subset \mathcal{G}(\hat{\mathbf{x}})$.

Given the special structure of the rotated second-order constraints (2), we propose to exploit the following property,

Proposition 3. *if $z = 0$, constraint (2) is equivalent to the set of equations,*

$$y_i = 0, \quad \forall i \in N. \quad (3)$$

Note that adding constraints (3) to (SOCP) fixes the degeneracy issue and all solvers converge to the optimal solution with an objective value of 16. For the general mixed-integer model, we look at two cases.

5.1. Bounded case

If the y_i variables are bounded, or can artificially be bounded, i.e., $y_i^l \leq y_i \leq y_i^u$, it is easy to enforce constraints (3) by introducing the following redundant inequalities,

$$y_i^l z \leq y_i \leq y_i^u z, \quad \forall i \in N. \quad (4)$$

Formally, we can prove the following result:

Let \mathcal{F}^b be the set given by,

$$\begin{cases} 4y^2 + (\gamma - z)^2 \leq (\gamma + z)^2 \\ z \leq 0, \quad y^l z \leq y \leq y^u z, \\ \gamma \in \mathbb{R}^+ \wedge y \in \mathbb{R} \wedge z \in [0, 1] \end{cases}$$

Proposition 4. $\mathcal{T}(\hat{\mathbf{x}}) = \mathcal{G}(\hat{\mathbf{x}}), \quad \forall \hat{\mathbf{x}} \in \mathcal{F}^b$.

Proof. Based on the notations and the analysis developed in the proof of Proposition 2, consider a feasible point $\hat{\mathbf{x}} \in \mathcal{F}^b$:

$$\hat{\mathbf{x}} = \begin{pmatrix} y & z & \gamma \\ 0 & 0 & \gamma_0 \end{pmatrix}$$

Note that both constraints $y^l z \leq y$ and $y \leq y^u z$ are active. if $\gamma_0 = 0$, we have,

$$\mathcal{T}(\hat{\mathbf{x}}) = \mathcal{G}(\hat{\mathbf{x}}) = \{\mathbf{d} \in \mathbb{R}^3 \mid d_1 = d_2 = 0, \quad d_3 \geq 0\}.$$

Similarly, if $\gamma_0 > 0$,

$$\mathcal{T}(\hat{\mathbf{x}}) = \mathcal{G}(\hat{\mathbf{x}}) = \{\mathbf{d} \in \mathbb{R}^3, \mid d_1 = d_2 = 0\}$$

□

Note that Guan et al. [8] propose to replace constraints (2) with (4) and ask the question “is the suggested way to approximate the conic constraints using the big-M method the best?”. This note suggests that adding both set of constraints can break the degeneracy issue while strengthening the continuous relaxation of the mixed-integer model. Using notation from [8], adding the constraint $w_i^2 \leq u_i z_i$ to model (P1) ([8] p.4) should dramatically improve its continuous relaxation, as

$$w_i^2 < w_i^2 / z_i, \quad 0 < z_i \leq 1.$$

Thus swapping w_i^2 for u_i in the objective function of (P1), and adding constraints $w_i^2 \leq u_i z_i$ should increase the lower bound and penalize fractional values of z_i .

5.2. Unbounded case

If one is unable to introduce artificial bounds on the variables, a patch needs to be implemented in the branch and bound scheme of mixed-integer nonlinear solvers. This can be done by pre-identifying such disjunctive second-order cone constraints and dynamically adding the linear inequalities (3) when branching on the “off” case. Such an implementation is ongoing work.

6. Conclusion

Constraint qualification failure can lead to irregular situations where optimal solutions do not satisfy the KKT system. Under such circumstances, interior point methods may fail to converge. In mixed-integer programming, branching is performed by introducing linear constraints fixing or bounding a discrete variable. While this approach seems harmless in the linear case, it might produce degeneracy in nonlinear systems. We propose two simple patches which can help avoid solvers’ failure.

Acknowledgement

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

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] 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.
- [4] Cplex. IBM ILOG Cplex software. <http://www.ibm.com/>.
- [5] 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. Rodríguez-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.
- [6] F. Gould and J. Tolle. A necessary and sufficient qualification for constrained optimization. *SIAM Journal on Applied Mathematics*, 20(2):164–172, 1971.
- [7] 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.
- [8] W. Guan, A. Gray, and S. Leyffer. Mixed-Integer Support Vector Machine. NIPS Workshop on Optimization for Machine Learning, 2009.

- [9] Gurobi. Gurobi software. <http://www.gurobi.com/>.
- [10] H. Hijazi, P. Bonami, G. Cornuéjols, and A. Ouorou. Mixed-integer nonlinear programs featuring “on/off” constraints. *Computational Optimization and Applications*, 52(2):537–558, 2012.
- [11] H.W. Kuhn and A.W. Tucker. Nonlinear programming. pages 481–492. Proceedings of 2nd Berkeley Symposium. Berkeley: University of California Press., 1951.
- [12] A. Letchford. On disjunctive cuts for combinatorial optimization. *Journal of Combinatorial Optimization*, 5:299–315, 2001. 10.1023/A:1011493126498.
- [13] L. Liberti and F. Marinelli. Mathematical programming: Turing completeness and applications to software analysis. *Journal of Combinatorial Optimization*, to appear.
- [14] J. P. Penot. A new constraint qualification condition. *Journal of Optimization Theory and Applications*, 48:459–468, 1986.
- [15] M. Perregaard. *Generating Disjunctive Cuts for Mixed Integer Programs*. PhD thesis, Carnegie Mellon University, 2003.
- [16] 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.
- [17] 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.
- [18] 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.