Constraint qualification failure in action

Hassan Hijazi, Leo Liberti

PII: S0167-6377(16)30033-5
DOI: http://dx.doi.org/10.1016/j.orl.2016.05.006
Reference: OPERES 6098

To appear in: *Operations Research Letters*

Received date: 23 July 2014
Revised date: 24 May 2016
Accepted date: 24 May 2016

Constraint Qualification Failure in Action

Hassan Hijazi[a,*], Leo Liberti[b]

[a]*The Australian National University, Data61-CSIRO, Canberra ACT 2601, Australia*
[b] *CNRS, 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)