Editorial

Reformulation techniques in mathematical programming

The idea for a special issue of DAM on Reformulation Techniques in Mathematical Programming was proposed to one of the editors (LL) by Peter Hammer during the EURO 2006 conference in Reykjavik. Peter’s interest in reformulation techniques (that acted like symbolic algorithms on the mathematical expressions of a given problem) specially referred to the reformulation of pseudo-boolean problems, and it seemed he had plans for work in that direction. We would therefore like to dedicate this issue to the memory of Peter Hammer, whom we miss dearly.

When shifted from pseudo-boolean to general mathematical programming, reformulation techniques become somewhat less pervasive but still fundamental, as the papers in this issue show. Mathematical programming can be seen as a language for defining optimization problems in terms of a set of parameters, variables, objective function(s) and constraints. A rather effective although mathematically imprecise definition for the concept of “reformulation” is given in the online Mathematical Programming Glossary as follows: “obtaining a new formulation $Q$ of a problem $P$ that is in some sense better, but equivalent to a given formulation”. The papers in this issue also show that the phrase in some sense better, but equivalent definitely takes on a vast range of hues in the ideas people have about reformulations. Reformulations may change any combination of parameters, variables, objective(s) and constraints in a mathematical program. The fact that a reformulation must in some sense be equivalent to the problem that originated it is also subject to different interpretations. The Reformulation-Linearization Technique (RLT) [5] and the symbolic reformulation in [6] are both designed to obtain a tight bound for the objective function value of Mixed-Integer Nonlinear Programming (MINLP) problems. The factorable standard form for Nonlinear Programming (NLP) [4] was introduced largely because it offered scope for the automatic construction of a convex relaxation [7]. On the other hand, [1] proposes a formal definition of the term reformulation, based on polynomial-time reductions in worst-case complexity theory. We informally interviewed some researchers in mathematical programming about their own interpretation of what a reformulation actually is, the outcome of this exercise being that there is much disagreement in struggling towards a “catch-all” definition that would correctly classify as reformulations all transformations that people feel should really be called reformulations. So the fundamental problem of defining reformulation in precise terms in a practically useful way remains open (some possible answers are provided in [2,3]).

This issue contains papers that either propose new reformulations in mathematical programming (Billionnet et al., Gueye and Michelon, Hansen and Meyer, Sherali and Adams) or employ reformulation techniques to improve the state of the art in theoretical or applied fields (Ben-Amor et al., Damaschke, Frangioni and Gendron, Frank, Jaumard et al., Liberti et al., Lucena et al.). Eleven papers out of twenty-one submissions were accepted for publication.

References


© In memory of Peter Hammer (1936–2006).

Guest Editors
Leo Liberti
Nelson Maculan

E-mail addresses: liberti@lix.polytechnique.fr (L. Liberti), maculan@cos.ufrj.br (N. Maculan).

30 September 2008
Available online 3 December 2008