

Symmetry in Integer Programming

François Margot, Carnegie Mellon University

August 2004

Abstract

This talk is an overview of techniques used in the Mathematical Programming Community to handle problems with symmetries. For problems modeled as integer linear programs, three essentially different and mutually exclusive approaches have been used: Reformulations, symmetry-breaking inequalities, and isomorphism pruning.

Reformulation is mostly used to handle problems where the symmetries come from the set of all permutations of a set of objects, inducing a permutation group on the variables. Some (or most) symmetries are removed from the formulations and additional branching strategies become available. The price to pay is that the reformulation has an exponential number of variables that must be handled by column generation techniques.

Adding symmetry breaking inequalities is probably the most natural way to try to handle symmetries. Several papers report strong improvements when using the best possible symmetry breaking set of inequalities. Methods for deriving such sets have been devised, but little is known about finding the most efficient set.

Isomorphism pruning is based on the theory of isomorphism-free backtracking procedure for enumeration problems. When coupled with a Branch-and-Cut algorithm, additional options for generating isomorphism-related cuts and setting variables are available. Improvements of several orders of magnitude for running time and enumeration tree size have been observed on highly symmetric problems.

Time permitting, a brief description of the data structures and algorithms used for handling group operations will be given.