

## Multি-row approaches to cutting plane generation

Laurent Poirier (ULg)

December 18, 2012

This thesis focuses on the use of cutting-plane techniques to improve general-purpose mixed-integer linear programming solvers.

The first topic covered here is a fast separation method for two-row cuts. Two-row cuts are intersection cuts from two rows of a simplex tableau describing the LP relaxation of the problem. This type of cuts recently gathered a lot of attention from the scientific community following a paper by Andersen, Louveaux, Weismantel and Wolsey describing the facets of the underlying two-row model and providing an intuitive geometric classification the the derived cuts. The specificity of the approach adopted here is that it does not rely on an "infinite relaxation" point of view and generate intersection cuts from fixed lattice-free sets. Instead, given a fractional point, it aims at always finding a most violated facet-defining inequality for the two-row model. This can be achieved by optimizing over the polar set of the integer hull of the model. A fast way of performing this is provided, by means of a polyhedron that is equivalent to the polar for that purpose, but has a more compact representation. Moreover, a row-generation algorithm is developed in order to avoid the costly computations of integer hulls of two-dimensional cones. An implementation of the resulting algorithm performs separation of two-row cuts in a few milliseconds on average, on the standard MIPLIB 3 and 2003 testsets.

While this two-row separator is quick, the measurements of the computational usefulness of the cuts do not yield satisfactory results. Since all the cuts generated are facet-defining, this might suggest that the underlying two-row models are too weak. This observation prompted the second part of this thesis, an attempt to evaluate the strength of various multi-row relaxations, on small instances, using a generic separator. To that end, a separator is developed, which is able to compute facet-defining inequalities from arbitrary (yet reasonably small) mixed-integer sets. A row-generation approach is again adopted, but this time the slave part consists in the resolution of a mixed-integer problem instead of a closed-form oracle. Some interesting computational tricks are developed, in order to speedup the inherently hard computations.