Sidharth Ghoshal

August 9, 2015

A cutting plane Tree Algorithm for Integer Programming

Consider the following problem

We note that this problem can be reduced to a problem of just satisfying:

where M and b are a different set of matrices. Suppose we have an oracle O that is able to quickly (meaning in polynomial time) determine if the system above has a solution, and retrieve exactly one of those solutions.

Then consider the following procedure

CurrentMax = objective value of solution to:

obtained by O

CurrentMin = objective value of solution to:

obtained by O

While matrix A is feasible as an LP:

currentExp = ½(CurrentMin + CurrentMax)

query O to solve:

If there is a solution:

CurrentSolution = thisSolution

currentMin = objectiveValue + 1

If there isn’t a solution:

currentMax = currentExp – 1

This is effectively a routine for binary search that makes use of the satisfiability Oracle. We note that it takes strictly less than where StartingMax is the first value of currentMax, for this loop to end.

Now the natural question that arises is, how to build the oracle? An implementation of an Oracle via some different cutting plane algorithms is described below:

Problem: determine quickly whether the system

Has an integer solution, and if it does, find it.

Some intuition and highlighting of procedure:

Assume the system has no inequality of the form

Suppose we without regard to any other aspect of the system decide to maximize

We will obtain some optimal solution such that

Now if U is an integer point then we are completely good to go and have solved the problem at hand. On the other hand if it isn’t we can always apply a Gomory’s cut to the point to exclude it from the solution. But notice the following:

since is not included in the system if is not integral then we can add the cut

Furthermore suppose that was integral but U wasn’t integral. Since the cut was not in the original system it definitely would have maximized at a specific point on polyhedron and thus by introducing gomory’s cut on that point, we can be certain that if we maximize again it must be the case that the new solution is such that

And thus if it isn’t already integral we most certainly could add the constraint

Whereas

We can declare this procedure as introducing a “rounding cut”. The idea is to find an objective hyper plane that isn’t parallel to any of the current hyper planes in the system, AND consists solely of integer coefficients. So that in maximizing it, we either:

1. Discover an Integer Point
2. Determine that the objective as non-integral value so we can add the rounding cut
3. Determine the objective is non-integral, but has integral value, so utilize Gomory’s cut to remove it and then add a rounding cut.

We note that the use of Gomory’s cut however is redundant since if the original hyper plane wasn’t parallel to any in the system then it definitely was maximized at some vertex of the system and thus it isn’t necessary at all to employ Gomory’s cut we can just subtract 1 from the objective and round again:

However like most cutting schemes this suffers from one critical issue:

Optimizations:

Inheriting Cuts

O(2n) level cuts

consider the folowing