LEHIGH UNIVERSITY

418 FINAL PROJECT

On the exact separation of mixed integer knapsack cuts

Author:
Chenxin Ma, Xi He

Supervisor:
Prof. Ted Ralphs

Fall 2014

Contents

1 Introduction

Consider a positive integer n, letting $b \in \mathbb{Q}$, $a \in \mathbb{Q}^n$, $l \in \{\mathbb{Q} \cup \{-\infty\}\}^n$, $u \in \{\mathbb{Q} \cup \{+\infty\}\}^n$ and $I \subset N := \{1, ..., n\}$. The Mixed Integer Knapsack Set is defined as

$$K = \{x \in \mathbb{R}^n : ax \leq b, l \leq x \leq u, x_i \in \mathbb{Z}, \forall i \in I\}. \tag{1}$$

Furthermore, if we have $c \in \mathbb{Q}^n$ and assume l_i is finite for each $i \in N$, then the Mixed Integer Knapsack Problem (MIKP) can be described as:

$$\max\{cx : x \in K\} \tag{2}$$

In this project, we concern two algorithms developed from [2]. The first one is to derive strongly valid inequalities for mixed integer knapsack set (1), *i.e.*, for a given point $x^\star \in \mathbb{Q}^n$, we focus on

- identifying an inequality (π, π_0) which is valid for $\mathrm{conv}(K)$ (equivalent valid for K) and violated by $x^\star$.
- proving no such inequality exists

This is of great practical importance since by obtaining valid inequalities for (1), we can also obtain valid inequalities for general Mixed Integer Programming (MIP) problems. Suppose $D \in \mathbb{Q}^{m \times n}$ and $d \in \mathbb{Q}^n$, the mixed integer set can be described as

$$P = \{x \in \mathbb{R}^n : Dx \leq d, l \leq x \leq u, x_i \in \mathbb{Z}, \forall i \in I\}. \tag{3}$$

Then if (a, b) in (1) can be obtained as a nonnegative linear combination of row from (D, d), then $P \in K$ and we can find that any inequality which is valid for K will also be valid for P. Actually, the valid inequality is called as $knapsack\ inequality$ or $knapsack\ cut$ of P.

The basic idea is to transform the problem of getting a valid inequality to a linear separation problem:

$$\begin{aligned} LP_1 : \min & \sum_{i=1}^{n} u_i + v_i \\ s.t.\ & \pi x^k - \pi_0 \leq 0, && \forall k \in \{1, 2, .., q\} \\ & \pi x^r \leq 0, && \forall k \in \{1, 2, .., t\} \\ & \pi x^* - \pi_0 = 1 \\ & \pi + u - v = 0 \\ & u \geq 0, v \geq 0, \end{aligned}$$

where $\{x^1, x^2, ..., x^q\}$ and $\{r^1, r^2, ..., r^l\}$ represent the extreme points and rays of $conv(K)$. The following steps can be applied to speed up the procedure

- Eliminating variables from LP_1
- Fixing variables and lifting back again

The second one is an efficient branch and bound algorithm with both domination branching and reduced-cost bound improvement (KBB) for the Mixed Knapsack Integer Problem (2). Most modern algorithms for solving Knapsack Problem are based either on branch-and-bound or on dynamic programming. However, the most efficient codes seldom make explicit use of Linear Programming. Here we will develop an LP-based branch and bound approach, and three strategies are used to speedup solving the problem:

- Detecting unbounded solutions and Preprocessing,
- Branch and bound (depth-first-search strategy), by using variable reduced-cost information to improve variable bounds at each node of the tree,
- Domination (To use the lexicographic and cost dominance to significantly reducing the search space).

The algorithm exploits dominance conditions, which differs from traditional linear-programming based algorithms by allowing feasible solutions to be pruned during the branching phase. The idea is that feasible solutions will only be pruned if either (a) they are not optimal (cost-domination-criteria), or (b) if they are optimal, but somewhere else in the tree it is known that there is another optimal solution. In the computational study, we will try to see how domination plays an role in this algorithm. In fact, according to the literature, the proposed algorithm performs quite well in practice, outperforming the general-use mixed integer programming solver CPLEX.

2 Research plan

We will implement the algorithm in [2], which contains:

- a simple eight-step pre-processing procedure in order to reduce an instance of MIKP to another, equivalent instance of MIKP which is easier to solve.
- a depth-first-search branch and bound algorithm which always branches on the unique fractional variable. We use a simple linear programming algorithm, a variation of Dantzigs algorithm [1], which runs in linear time by taking advantage of the fact that variables are sorted by decreasing efficiency.
- using dominance to improve the branch and bound search. In fact, lexicographic and cost dominance allow us to disregard feasible solutions that are not the unique lexicographically smallest optimum solution, hence significantly reducing the search space.

After implementing it, we will conduct extensive computational experiments. The goal is to reproduce the improvement that the new algorithm brings in [2]. Moreover, we will try to combine different branching strategies (see [3]) and bound improvement to compare the different versions of the algorithm. If possible, we will take a glimpse to use this knapsack cut generation procedure to solve some classes of mixed integer programming problems.

References

[1] George B Dantzig. Discrete-variable extremum problems. *Operations research*, 5(2):266–288, 1957.

[2] Ricardo Fukasawa and Marcos Goycoolea. On the exact separation of mixed integer knapsack cuts. *Mathematical programming*, 128(1-2):19–41, 2011.

[3] Jeffrey T Linderoth and Ted K Ralphs. Noncommercial software for mixed-integer linear programming. *Integer programming: theory and practice*, 3:253–303, 2005.