# Project Proposal: On the exact separation of mixed integer knapsack cuts

Xi He, Chenxin Ma

November 12, 2014

## 1 Introduction

In this project, we will focus on cutting plane for large Mixed Integer Programs (MIPs) in the context of Branch and Cut algorithms, which is studied in [2]. More specifically, we will study an efficient algorithm (KBB) for the Mixed Integer Knapsack Program (MIKP). Consider a positive integer $n$. For each $k \in \{1, ..., n\}$, let $a_k, c_k \in \mathbb{Q}$, $l_k \in \mathbb{Q} \cup \{-\infty\}$, and $u_k \in \mathbb{Q} \cup \{+\infty\}$. Let $b \in \mathbb{Q}$, and consider $I \subset \{1, ..., n\}$. The Mixed Integer Knapsack Problem (MIKP) can be described as:

$$\begin{aligned} &\max \sum_{i=1}^{n} c_i x_i \\ &s.t. \ \sum_{j=1}^{n} a_k x_k \leq b \\ &\qquad l_k \leq x_k \leq u_k, \quad \forall k \in \{1, 2, .., n\} \\ &\qquad x_k \in \mathbb{Z}, \forall k \in I, \end{aligned} \tag{1}$$

whose feasible set is polyhedron $K$. Most modern algorithms for solving Knapsack Problem are based either on branch-and-bound or dynamic programming. However, the most efficient codes seldom make explicit use of Linear Programming. 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 algorithm we implement is an LP-based branch-and-bound algorithm, which will either (a) proves MIKP is infeasible, (b) proves MIKP is unbounded or (c) finds an optimal solution to MIKP. According to the literature, the proposed algorithm performs quite well in practice, outperforming the general-use mixed integer programming solver CPLEX.

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.

## 2 Objectives

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 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.

## 3 Research plan

## 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.