Lifting, tilting and fractional programming revisited: a study on mixed integer linear sets

Daniel Espinoza[a], Ricardo Fukasawa[b], Marcos Goycoolea[c,*]

[a] *Universidad de Chile*
[b] *University of Waterloo, 200 University Ave. West, Waterloo, ON - Canada - N2L3G1*
[c] *Universidad Adolfo Ibañez*

Abstract

Lifting, tilting and fractional programming, though seemingly different, reduce to a common core optimization problem. This connection allows us to revisit key properties of lifting, tilting and fractional programming in a simple common framework, and extend known results from each of these problems to the other two.

Key words: Lifting, Tilting, Fractional programming, Mixed-integer programming

1. Introduction

Consider a nonempty mixed integer linear set $M = \{x \in \mathbb{R}^n : Ax \geq h,\ x_i \in \mathbb{Z}\ \forall i \in I\}$ where $A \in \mathbb{Q}^{m\times n}$, $h \in \mathbb{Q}^m$ and $I \subseteq \{1, \ldots, n\}$. Given $a, c \in \mathbb{Q}^n$ and $b, d \in \mathbb{Q}$ we are interested in solving

$$\max\{\lambda : (a^T x - b) - \lambda(c^T x - d) \geq 0\ \forall x \in M\}. \qquad (1)$$

that is, we would like to either determine an optimal value λ^* for this problem, or show that no such value exists because the problem is either infeasible or unbounded.

Though problem (1) seems to have a very particular structure, it is a common problem that appears in three seemingly different contexts: lifting, tilting and fractional programming. In this article we explore how these three problems are in fact special cases of (1), and present a general algorithm for solving (1) which extends the algorithms presented in [8] for lifting, in [10] for tilting, and in [7] for fractional programming. Though the connection between these three problems does not require a complex mathematical development, it allows us to show how results from one problem can extend known results for the other two.

We start by defining the three different problems in question. Throughout this section we will consider that P is a polyhedron and that Q is a proper face of P. Given an inequality $\alpha^T x \geq \beta$ which is valid for P, we say that Q is *defined* by this inequality if $Q = \{x \in P : \alpha^T x = \beta\}$. We can then define lifting and tilting as follows:

- **Lifting.** Given an inequality $a^T x \geq b$ which is valid for Q, *lifting* consists in obtaining an inequality $\bar{a}^T x \geq \bar{b}$ which is valid for P and such that $a^T x - b = \bar{a}^T x - \bar{b}$ for all $x \in Q$.
- **Tilting.** Assume P has dimension m and Q is defined by $c^T x \geq d$ and has dimension $k < m - 1$. *Tilting* consists in obtaining an inequality $\bar{c}^T x \geq \bar{d}$ which is both valid for P and such that it defines face Q' of P such that $Q \subsetneq Q'$.

These two problems are important in the context of generating cutting planes for general mixed-integer programs (MIP) or particular combinatorial optimization problems. In particular, one would be interested in the case when P is defined as the convex hull of M.

Finally, we introduce the fractional programming problem. Consider the mixed integer linear set defined as before, and assume $c^T x \neq d$ for all $x \in M$. Mixed integer linear fractional programming consists in solving

$$\inf\left\{\frac{a^T x - b}{c^T x - d} : x \in M\right\}. \qquad (2)$$

This is an important optimization problem arising in several classes of applications. For comprehensive surveys of methods and applications see Stancu-Minasian [22] and Schaible [20, 21].

The outline of the paper is as follows. In Section 2 we present an algorithm for solving Problem (1), and discuss key properties, including feasibility, finiteness and convergence. Afterwards we develop the connection between (1) and each of these problems and its consequences. Section 3 is devoted to Lifting, Section 4 to Tilting, and Section 5 to Fractional Programming.

2. The base problem

In this section we consider problem (1). We will assume throughout this section that $c^T x \geq d$ for all $x \in M$. Let

*Corresponding author
Email addresses: daespino@dii.uchile.cl (Daniel Espinoza), rfukasaw@math.uwaterloo.ca (Ricardo Fukasawa), marcos.goycoolea@uai.cl (Marcos Goycoolea)

$RM = \{r \in \mathbb{R}^n : Ar \geq 0\}$. We first characterize feasibility and unboundedness in Proposition 2.1 and 2.2.

Algorithm 1: Solving the base problem

Input: A nonempty mixed integer linear set M, vectors $a, c \in \mathbb{Q}^n$, and scalars $b, d \in \mathbb{Q}$.

Output: The solution λ^* to $\max\{\lambda : (a^T x - b) - \lambda(c^T x - d) \geq 0,\ \forall x \in M\}$, or proof of infeasibility or unboundedness.

1 $i \leftarrow 1$, $z_1 \leftarrow \min\{-(c^T x - d) : x \in M\}$.
2 Let x^1 be a minimizer or $r^1 \in RM : -c^T r^1 < 0$.
3 **if** $z_1 = 0$ **then**
4     $z \leftarrow \min\{(a^T x - b) : x \in M\}$.
5   **if** $z \geq 0$ **then**
6   Problem is unbounded. Stop.
7   Problem is infeasible. Stop.
8 **if** $z_1 = -\infty$ **then**
9   **while** $z_i < 0$ **do**
10   **if** $c^T r^i = 0$ **then**
11   Problem is infeasible. Stop.
12         $\lambda_{i+1} \leftarrow \frac{a^T r^i}{c^T r^i}$, $i \leftarrow i + 1$.
13         $z_i \leftarrow \min\{(a^T r) - \lambda_i(c^T r) : r \in RM, r \leq 1\}$.
14   Let r^i be a minimizer having value z_i.
15     $z_i \leftarrow \min\{(a^T x - b) - \lambda_i(c^T x - d) : x \in M\}$.
16   Let x^i be a minimizer having value z_i.
17 **while** $z_i < 0$ **do**
18   **if** $(c^T x^i - d) = 0$ **then**
19   Problem is infeasible. Stop.
20     $\lambda_{i+1} \leftarrow \frac{(a^T x^i - b)}{(c^T x^i - d)}$, $i \leftarrow i + 1$.
21     $z_i \leftarrow \min\{(a^T x - b) - \lambda_i(c^T x - d) : x \in M\}$.
22   Let x^i be a minimizer having value z_i.
23 $\lambda^* \leftarrow \lambda_i$.

Proposition 2.1. *Assume $M \neq \emptyset$ and that $c^T x \geq d$ for all $x \in M$ (note that this implies $c^T r \geq 0$ for all $r \in RM$). Problem* (1) *is feasible if and only if both*

$$(c^T x - d) = 0 \text{ implies } (a^T x - b) \geq 0,\ \forall x \in M, \quad \text{(3a)}$$

$$c^T r = 0 \text{ implies } ar \geq 0,\ \forall r \in RM. \quad \text{(3b)}$$

Note that if there exists $x \in M$ such that $(c^T x - d) = 0$ then condition (3a) *suffices for Problem* (1) *to be feasible, since in this case* (3a) *implies* (3b).

Proof. ($\Rightarrow$) If (3a) does not hold, then there exists $\bar{x} \in M$ such that $c^T \bar{x} = d$ and $a^T \bar{x} < b$, so for any $\lambda \in \mathbb{R}$, we have that $(a^T \bar{x} - b) - \lambda(c^T \bar{x} - d) = a^T \bar{x} - b < 0$. If (3b) does not hold, then let $\bar{r} \in RM$ be such that $c^T \bar{r} = 0$ and $a^T \bar{r} < 0$. Note that we may assume $\bar{r} \in \mathbb{N}^n$. Let $\bar{x} \in M$ be any arbitrary point. If (1) is feasible, then pick $\bar{\lambda}$ feasible for it. Let $x_\alpha := \bar{x} + \alpha \bar{r} \in M$ for all $\alpha \in \mathbb{Z}_+$. But then $c^T x_\alpha - d = c^T \bar{x} - d$ and $a^T x_\alpha - b = a^T \bar{x} - b + \alpha a^T \bar{r}$. But then $a^T x_\alpha - b = a^T \bar{x} - b + \alpha a^T \bar{r} \geq \bar{\lambda}(c^T \bar{x} - d) \iff \alpha \leq \frac{\bar{\lambda}(c^T \bar{x} - d) - a^T \bar{x} + b}{a^T \bar{r}}$ for all $\alpha \in \mathbb{Z}_+$, which is a contradiction.

($\Leftarrow$) We assume conditions (3a) and (3b) hold and show that there exists λ such that $(a^T x - b) - \lambda(c^T x - d) \geq 0$ for all $x \in M$. Let $\{x^1, \ldots, x^p\}$ and $\{r^1, \ldots, r^q\}$ be the extreme points and extreme rays of $conv(M)$. Given that condition (3a) holds, if $(a^T x^i - b) < 0$ then $(c^T x^i - d) > 0$ for $i = 1, \ldots, p$. Likewise, given that condition (3b) holds, if $a^T r^j < 0$ then $c^T r^j > 0$ for $j = 1, \ldots, q$. Then, there exists $\lambda < 0$ such that, $(a^T x^i - b) - \lambda(c^T x^i - d) \geq 0$ $\forall i = 1, \ldots, p$ and $a^T r^j - \lambda c^T r^j \geq 0$, $\forall j = 1, \ldots, q$

Since for any $x \in M$ there exist $\alpha \geq 0$ and $\beta \geq 0$ such that $x = \sum_i \alpha_i x^i + \sum_j \beta_j r^j$ and $\sum \alpha_i = 1$ the second implication follows. □

Proposition 2.2. *Assume that $c^T x \geq d$ for all $x \in M$. Problem* (1) *is unbounded if and only if*

$$(c^T x - d) = 0 \text{ and } (a^T x - b) \geq 0 \text{ for all } x \in M. \quad (4)$$

Proof. It is easy to see that condition (4) is sufficient for unboundedness, so we focus on necessity. Suppose that this condition does not hold. Either (i) there exists $x' \in M$ such that $(a^T x' - b) < 0$ and $(c^T x' - d) = 0$, or (ii) there exists $x' \in M$ such that $(c^T x' - d) > 0$. From Proposition 2.1 we have that in the first case the problem is infeasible. In the latter we have that every feasible solution λ satisfies $\lambda \leq -(a^T x' - b)/(c^T x' - d)$, implying boundedness. □

In Algorithm 1 we describe how to solve Problem (1). Observe that lines 17 - 22 are basically Dinkelbach's algorithm, lines 8 - 16 consider the case in which M is unbounded, and lines 1 - 7 consider the special case in which $(c^T x - d) = 0$ for all $x \in M$. To our knowledge, Chvátal et al. [5] are the first to consider rays in order to extend Dinkelbach's algorithm to unbounded sets. The proof of correctness of the algorithm is given in Theorem 2.3.

Theorem 2.3. *Algorithm 1 satisfies the following:*

(1) If the algorithm reaches step 6, then the problem is unbounded.
(2) If the algorithm reaches steps 7, 11, or 19, then the problem is infeasible.
(3) The sequence $\{\lambda_i\}$ is monotone decreasing.
(4) If the algorithm reaches steps 16 or 22, then z_i is finite and there exists an optimal solution x^i.
(5) If the algorithm reaches step 23, the value λ^* corresponds to the optimal solution value. Moreover, the algorithm has identified $x^* \in M$ such that $c^T x^* - d > 0$ and $\lambda^* = \frac{(a^T x^* - b)}{(c^T x^* - d)}$, or $r^* \in RM$ such that $c^T r^* > 0$ and $\lambda^* = \frac{(a^T r^*)}{(c^T r^*)}$.

Proof. (1) If the algorithm reaches step 6 it is because $(c^T x - d) = 0$ and $(a^T x - b) \geq 0$ for all $x \in M$. From Proposition 2.2 it follows that the problem is unbounded.

(2) If the algorithm reaches step 7, step 11 or step 19, then we either violate (3a) or (3b), so the result follows from Proposition 2.1.

(3) Since $(c^T x - d) \geq 0$ for all $x \in M$, it follows that $c^T r \geq 0$ for all ray r of $conv(M)$. Moreover, in step 12 we have that $c^T r^i > 0$ and $a^T r^i - \lambda_i c^T r^i < 0$. Thus, $\lambda_i > \frac{a^T r^i}{c^T r^i} = \lambda_{i+1}$. Likewise, in step 20 we have that $(c^T x^i - d) > 0$ and $(a^T x^i - b) - \lambda_i (c^T x^i - d) < 0$. Thus, $\lambda_i > \frac{(a^T x^i - b)}{(c^T x^i - d)} = \lambda_{i+1}$.

(4) Let i_o be the value of i at step 16. Note that $z_{i_o} > -\infty$ (since we exited the loop from step 9). It follows that $a^T r - \lambda_{i_o} c^T r \geq 0$ for all $r \in RM$. Since $c^T r \geq 0$ for all rays of $conv(M)$, and since the sequence λ_i is monotone decreasing, we also have that $a^T r - \lambda_i c^T r \geq 0$ for all $i \geq i_o$ and $r \in RM$. Thus z^i is finite for $i \geq i_o$ and the solution x^i obtained in step 16 exists. For step 22 the argument is analogous.

(5) Assume that the algorithm terminates on the i-th interation. Since it is clear that λ^* is feasible it suffices to show that every $\bar{\lambda} > \lambda^* = \lambda_i$ is infeasible. First, assume λ^* was obtained at step 12. This implies $\lambda^* = \frac{a^T r^{i-1}}{c^T r^{i-1}}$. In this case $a^T r^{i-1} - \lambda^* c^T r^{i-1} = 0$. Since $c^T r^{i-1} > 0$ it follows that $a^T r^{i-1} - \bar{\lambda} c^T r^{i-1} < 0$. This implies $\bar{\lambda}$ is not feasible. In fact, for any $x \in M$ we have, for α large enough, that $x' = x + \alpha r$ is such that $(a^T x' - b) - \lambda (c^T x - d) < 0$. Next, assume that λ^* was obtained in step 20. This implies $\lambda^* = \frac{(a^T x^{i-1} - b)}{(c^T x^{i-1} - d)}$, and in turn that $(a^T x^{i-1} - b) - \lambda^* (c^T x^{i-1} - d) = 0$. Given $(c^T x^{i-1} - d) > 0$ it follows that $(a^T x^{i-1} - b) - \bar{\lambda}(c^T x^{i-1} - d) < 0$. In both cases $\bar{\lambda}$ is infeasible. □

A natural concern regarding Algorithm 1 is the number of times it has to invoke its oracle for solving MIPs (steps 4, 15 and 21). The following proposition, while true in general, can give an indication as to the number of iterations required to terminate when $c^T x - d$ is integer. This proposition results from combining results of Schaible [19] and Easton and Gutierrez [8].

Proposition 2.4. *Consider an instance of Problem* (1). *Suppose that iterates x^i and x^{i+1} have been obtained from step 20 of Algorithm 1. If λ^{i+1} is not optimal, then,* $(c^T x^i - d) > (c^T x^{i+1} - d)$. *Analogously, suppose that iterates r^i and r^{i+1} have been obtained from step 12 of Algorithm 1. If λ^{i+1} is not optimal, then $c^T r^i > c^T r^{i+1}$.*

Proof. Observe that the following conditions hold:

$$(a^T x^i - b) - \lambda^{i+1}(c^T x^i - d) = 0, \quad (5)$$

$$(a^T x^{i+1} - b) - \lambda^{i+1}(c^T x^{i+1} - d) \leq 0, \quad (6)$$

$$(a^T x^{i+1} - b) - \lambda^i (c^T x^{i+1} - d) \geq (a^T x^i - b) - \lambda^i (c^T x^i - d). \quad (7)$$

Thus, (5) - (6) imply that:

$$(a^T x^{i+1} - b) - (a^T x^i - b) \leq \lambda^{i+1}((c^T x^{i+1} - d) - (c^T x^i - d)).$$

Putting this together with (7) we get that

$$(\lambda^{i+1} - \lambda^i)((c^T x^{i+1} - d) - (c^T x^i - d)) \geq 0.$$

Since $\lambda^i > \lambda^{i+1}$, it follows that $(c^T x^{i+1} - d) \leq (c^T x^i - d)$. We now prove that $(c^T x^i - d) \neq (c^T x^{i+1} - d)$. For this, define:

$$P_i = \{x \in P : (c^T x - d) = (c^T x^i - d)\}$$

Observe that,

$$x^i = \arg\min\{(a^T x - b) - \lambda^i (c^T x - d) : x \in P_i\} = \arg\min\{(a^T x - b) : x \in P_i\}.$$

From this, $(a^T x - b) - \lambda^{i+1}(c^T x - d) \geq 0$ for all $x \in P_i$. In fact, if $x \in P_i$ we have $(a^T x - b) - \lambda^{i+1}(c^T x - d) = (a^T x - b) - \lambda^{i+1}(c^T x^i - d) \geq (a^T x^i - b) - \lambda^{i+1}(c^T x^i - d) = 0$.

However, since λ^{i+1} is not optimal, we have $(a^T x^{i+1} - b) - \lambda^{i+1}(c^T x^{i+1} - d) < 0$. Thus, x^{i+1} is not in P_i. The case for rays is strictly analogous. □

Two more points regarding convergence. First, it is easy to see that if the oracle used for solving the MIPs over M returns a finite number of possible solutions in steps 1, 4, 15 and 21, then the algorithm terminates in a finite number of steps. Even if this is not the case, the convergence rate can easily be shown to be superlinear. The following proposition is a simple extension of a result in Schaible [19]:

Proposition 2.5. *Assume that λ^* is the last iterate obtained in step 12 of Algorithm 1, and let $r^* \in RM$ be such that $\lambda^* = \frac{a^T r^*}{c^T r^*}$. Then, for all $\lambda_i \neq \lambda^*$ obtained before step 17 of Algorithm 1, we have*

$$\frac{\lambda^* - \lambda^{i+1}}{\lambda^* - \lambda^i} \leq 1 - \frac{(c^T r^*)}{(c^T r^i)} < 1.$$

Analogously, assume that λ^* is the last iterate obtained in step 20 of Algorithm 1, and let $x^* \in M$ be such that $\lambda^* = \frac{(a^T x^* - b)}{(c^T x^* - d)}$. Then, for all $\lambda_i \neq \lambda^*$ obtained after step 17 of Algorithm 1, we have

$$\frac{\lambda^* - \lambda^{i+1}}{\lambda^* - \lambda^i} \leq 1 - \frac{(c^T x^* - d)}{(c^T x^i - d)} < 1.$$

Proof. Note that

$$(a^T x_i - b) - \lambda_i (c^T x_i - d) \leq (a^T x^* - b) - \lambda_i (c^T x^* - d)$$

implies

$$\frac{(a^T x_i - b)}{(c^T x_i - d)} - \lambda_i \leq \frac{(a^T x^* - b)}{(c^T x_i - d)} - \lambda_i \frac{(c^T x^* - d)}{(c^T x_i - d)}$$

Thus,

$$\lambda_{i+1} - \lambda^* = \frac{(a^T x_i - b)}{(c^T x_i - d)} - \frac{(a^T x^* - b)}{(c^T x^* - d)} \leq \frac{(a^T x^* - b)}{(c^T x_i - d)} - \frac{(a^T x^* - b)}{(c^T x^* - d)} + \lambda_i \left(1 - \frac{(c^T x^* - d)}{(c^T x_i - d)}\right) = \left(\frac{1}{(c^T x_i - d)} - \frac{1}{(c^T x^* - d)}\right)((a^T x^* - b) - \lambda_i (c^T x^* - d)) = \left(\frac{1}{(c^T x_i - d)} - \frac{1}{(c^T x^* - d)}\right)(\lambda^* - \lambda_i)(c^T x^* - d).$$

By observing $\lambda^* < \lambda_i$ (see Theorem 2.3) and dividing by $(\lambda_i - \lambda^*)$ the result follows from the fact that $(c^T x^i - d) > (c^T x^* - d)$ (see Proposition 2.4). The case of rays is strictly analogous. □

An alternative to using Algorithm 1 was proposed by Easton and Gutierrez [8]. Rather than enumerating a full branch-and-bound tree each time step 22 of Algorithm 1 is executed, they propose enumerating a single branch-and-bound tree with which to explore M. In their work they describe a branch-and-bound mechanism in which the objective function is iteratively changed. We present an equivalent scheme which works by adding cutting planes (and keeping the same objective function throughout).

Suppose we have a candidate solution $\lambda^i \in \mathbb{R}$ to problem (1) and consider the problem of finding $x \in M$ that minimizes $(a^T x - b) - \lambda^i (c^T x - d)$:

$$\begin{aligned} &\min z \\ &(a^T x - b) - \lambda^i (c^T x - d) \quad \leq z \\ &x \in M \end{aligned} \tag{8}$$

If we solve this problem with a branch-and-bound algorithm and in the course of branching identify a feasible solution $\bar{x} \in M$ such that $z(\bar{x}) = (a^T \bar{x} - b) - \lambda^i (c^T \bar{x} - d) < 0$ we clearly know that λ^i is not feasible for problem (1). Moreover, we know that if $c^T \bar{x} = d$, then problem (1) is infeasible (see Proposition 2.1). If this is not the case, we can simulate Algorithm 1 by adding a new cutting plane $(a^T x - b) - \lambda^{i+1}(c^T x - d) \leq z$, where $\lambda^{i+1} = \frac{a^T \bar{x} - b}{c^T \bar{x} - d}$. As before, we will have $\lambda^i > \lambda^{i+1} \geq \lambda^*$. Further, since $c^T x \geq d$ for all $x \in M$, we obtain a tighter cut than before. In fact, this cut dominates the previous one, and we can think that it replaces it. However, we can keep the same branch-and-bound tree (including generated cuts) and iterate from there, thus avoiding possible redundancies. We call this the *one-tree* algorithm.

Note that we can start this algorithm with the value of λ^i obtained from step 8 of Algorithm 1. Also, we could start it with $\lambda^1 = \frac{(a^T x^1 - b)}{(c^T x^1 - d)}$ where x^1 is the solution found at step 2. By the arguments presented in Theorem 2.3 it is easy to see that the one-tree algorithm is correct.

3. Lifting

Consider a mixed integer set M and a proper face Q of $conv(M)$ defined by $Q = \{x \in conv(M) : c^T x = d\}$, where $c^T x \geq d$ is a valid inequality for $conv(M)$. Assume that $a^T x \geq b$ is valid for Q, and that $Q \neq \emptyset$. The *lifting* problem consists in finding an inequality $\bar{a}^T x \geq \bar{b}$ that is valid for $conv(M)$, such that $(\bar{a}^T x - \bar{b}) = (ax - b)$ for all $x \in Q$, and such that $Q \subsetneq Q' = \{x \in conv(M) : \bar{a}^T x = \bar{b}\}$.

Lifting was first proposed by Gomory [11] and Pollateschek [16] in the context of group and set packing problems. Padberg [15] was the first to propose a computational method for lifting, and Wolsey [23] generalized the procedure as it is known today. Lifting has many important practical uses in general MIP. Crowder et al. [6] and Gu et al. [12, 13] have shown that lifted cover inequalities can significantly improve the performance of general-use MIP solvers.

The lifting problem can be solved by finding a value $\lambda \in \mathbb{R}$ such that $(a^T x - b) - \lambda(c^T x - d) \geq 0$ is valid for $conv(M)$. Given that $Q \neq \emptyset$ and that $(a^T x \geq b)$ is valid for Q, Proposition 2.1 ensures that the lifting problem is feasible. Given that $c^T x \geq d$ is assumed to be a valid inequality in the context of lifting, the larger the value of λ, the better the inequality obtained. The *optimal lifting* problem consists in finding the largest such value. Clearly, the optimal lifting problem reduces to a problem of type (1). Given that $Q \neq conv(M)$, Proposition 2.2 ensures that the optimal lifting problem is well-defined (e.g., not unbounded), and can be solved with Algorithm 1. Easton and Gutierrez [9] use a very similar algorithm to lift integer variables.

Computational implementations of optimal lifting have typically restricted to the case in which $c^T x \geq d$ corresponds to the bound constraint of an integer variable (e.g. constraints of the form $x_i \leq t$). They have proceeded by fixing x_i to each of the values it can take and solving several problems independently (eg, $\min\{ax : x \in M\ ,\ x_i = k\}$ for each value k in the domain of x_i) to determine the largest feasible coefficient.

It is when we move away from lifting pure integer variables that things become more interesting. First, it should be noted that lifting a constraint of the form $(c^T x - d)$ is theoretically the same as lifting a single variable. For this, add a slack variable $s = (c^T x - d)$ and lift from the face defined by $s \geq 0$ (Louveaux and Wolsey [14]). However, in practice it has not been so clear how this variable s should be lifted. In this way, using Algorithm 1 to solve the lifting problem is different to what is traditionally done for several reasons: First, because it can be used to lift continuous and/or unbounded variables (such as the slack variable of mixed integer constraint, as illustrated above), or to directly lift using arbitrary constraints. Second, because it can result in solving less MIPs to get the final lifting coefficient (due to improved convergence rates). There is also a third computationally attractive application.

In practical applications of lifting it is often the case that Q is described by several valid inequalities. That is, $Q = \{x \in conv(M) : c_i^T x = d_i \text{ for } i = 1, \ldots, k\}$, where $c_i^T x \geq d_i$ are valid inequalities of $conv(M)$ for $i = 1, \ldots, k$. Approaches for solving this typically rely on lifting each of these constraints one at a time (*sequential lifting*).

Easton and Gutierrez [9] observe the following. Let $\alpha_i > 0$ for $i = 1, \ldots, k$ and define $c = \sum_{i=1}^k \alpha_i c_i$ and $d = \sum_{i=1}^k \alpha_i d_i$. Clearly, $c^T x \geq d$ is valid for P. Further, if $x \in M$, then $c^T x = d$ implies $c_i^T x = d_i$ for $i = 1, \ldots, k$. Thus $Q = \{x \in conv(M) : c^T x = d\}$ and we can lift $a^T x \geq b$ by solving (1). This can be achieved with Algorithm 1 regardless of how the constraints $c_i^T x \geq d_i$ are defined.

4. Tilting and the facet procedure

Consider a mixed integer set M and a valid inequality $c^T x \geq d$ defining a proper face Q of $P := conv(M)$. We would like to obtain from $c^T x \geq d$ a facet-defining inequality of $conv(M)$ (or show that $c^T x \geq d$ is one). This can be done by what is called the *facet procedure* [1, 5]. At the heart of the facet procedure is the *tilting* problem, which is just an instance of (1).

We start by describing the facet procedure. Consider a set $Q_o = \{x_1, \ldots, x_q\} \neq \emptyset$ of affinely independent points in Q. These prove $dim(Q) \geq q - 1$. Consider also a set of linearly independent *implicit equations* $P_o^\perp = \{p_1, \ldots, p_r\}$ in $P^\perp := \{p \in \mathbb{R}^n : p^T x \text{ is constant}, \forall x \in M\}$ (i.e. the set of valid equations for $conv(M)$). These prove $dim(P) \leq n - r$. Finally, note that any point $\bar{x} \in M \setminus Q$ is a certificate that Q is a proper face of P. Observe that if $q + r = n$ then $c^T x \geq d$ must define a facet of P. The facet-procedure consists in increasing at each iteration the cardinality of Q_o or $P_o^\perp$ by using *tilting*. The *tilting* procedure receives as input: $c^T x \geq d$ defining a face Q of P, Q_o, $P_o^\perp$ and $\bar{x}$. It produces as output one of the following:

- A point $x_{q+1} \in M$ which is affinely independent of the points in Q_o, a valid inequality $\bar{c}^T x \geq \bar{d}$ for M such that $\bar{c}^T x_i = \bar{d}$ for $i = 1, \ldots, q+1$, and a new point $\bar{x}$ such that $\bar{c}^T \bar{x} > \bar{d}$, or,
- A point p_{r+1} in $P^\perp$ which is linearly independent of the points in $P_o^\perp$.

Observe that with these inputs and outputs we can iteratively repeat the tilting procedure and obtain a facet-defining inequality satisfying the required conditions in at most n iterations. At each round, tilting works as follows. It begins by finding a *valid tilting direction.* That is, a non-zero pair (v, w) such that

$$v^T p_i = 0 \qquad \forall p_i \in P_o^\perp \tag{9}$$
$$w - v^T x_i = 0 \qquad \forall x_i \in P_o \tag{10}$$
$$w - v^T \bar{x} = 0 \tag{11}$$

It is not difficult to see that $c^T x \geq d$ does not define a facet of P if and only if there is a nonzero solution to the above system of linear equations [5], which can easily be obtained by linear programming. Given a valid tilting direction (v, w), tilting continues by solving the problem,

$$\max\{\lambda : (v^T x - w) - \lambda(c^T x - d) \geq 0, \ \forall x \in M\} \tag{12}$$

Several outputs are possible if Algorithm 1 is used to solve problem (12) (we assume below that all rays are integer).

1. The problem is infeasible and the algorithm returns a point $x^* \in M$ such that $(c^T x^* - d = 0)$ and $(v^T x^* - w) < 0$. In this case let $(\bar{c}, \bar{d}) = (c, d)$ and $x_{q+1} = x^*$. The fact that $v^T x^* < w$ and $v^T x_i = w$ for all $i = 1, \ldots, q$ imply that x^* is affinely independent from all points in Q_o. Leave $\bar{x}$ unchanged.
2. The problem is infeasible and the algorithm returns a ray r^* of $conv(M)$ such that $(c^T r^* = 0)$ and $(v^T r^*) < 0$. In this case let $(\bar{c}, \bar{d}) = (c, d)$ and $x_{q+1} = x_q + r^*$. By the same argument above, x_{q+1} is affinely independent of all points in Q_o. Leave $\bar{x}$ unchanged.
3. The algorithm has optimal solution $\lambda^* = 0$. This implies $v^T x \geq w$ for all $x \in M$. In this case, solve
$$z = \max\{(v^T x - w) : x \in M\} \tag{13}$$
There are two possible sub-cases:
   (a) If $z = 0$ then $M \subseteq \{x : v^T x = w\}$, so define $p_{r+1} = v$. Condition (9) implies v is linearly independent of all vectors in $P_o^\perp$.
   (b) If $z > 0$ define $(\bar{c}, \bar{d}) = (v, w)$ and $x_{q+1} = \bar{x}$. If $z = +\infty$ let r be the ray proving unboundedness of (13), and redefine $\bar{x} = x_{q+1} + r$. Otherwise, redefine $\bar{x}$ to be the maximizer of problem (13).
4. The algorithm has optimal solution $\lambda^* \neq 0$. In this case let $(\bar{c}, \bar{d}) = (v - \lambda^* c, \ w - \lambda^* d)$. If the optimal solution of (12) is x^*, let $x_{q+1} = x^*$. If the optimizing solution is a ray r^* let $x_{q+1} = x_q + r^*$. By Theorem 2.3-(5), we have that $c^T x_{q+1} > d$ so x_{q+1} is affinely independent of $x_1, \ldots, x_q$. Leave $\bar{x}$ as it was before.

Observe that Proposition 2.2 and the fact that $c^T \bar{x} > d$ ensure that problem (12) is not unbounded. For a detailed proof and geometrical interpretation of the algorithm see [5]. Finally, note that if many inequalities are to be tilted over a same set $conv(M)$, one only needs to compute the points $P_o^\perp$ once, thus speeding up computations.

Algorithm 1 is very similar to the tilting algorithm of [5]. However, the results in Section 2 allows for a much more compact presentation of the facet procedure, as well as new convergence results and the one-tree algorithm.

5. Fractional Programming

Mixed integer linear fractional programming consists in solving problem (2). Throughout this section we will assume that this problem is well-defined; that is $c^T x \neq d$ for all $x \in M$. In the context of fractional programming it is usually assumed that $c^T x > d$ for all $x \in M$. This makes sense if M is a convex set, since otherwise there would exist x such that $c^T x = d$. However, if M is a mixed-integer set, we can have $c^T x \neq d, \forall x \in M$ and $c^T x^1 > d$, $c^T x^2 < d$ for $x^1, x^2 \in M$. Define then $M^+ = \{x \in M : c^T x \geq d\}$, and $M^- = \{x \in M : c^T x \leq d\}$ and suppose that we can separately solve

$$\inf \left\{ (a^T x - b)/(c^T x - d) : x \in M^+ \right\}, \tag{14a}$$
$$\inf \left\{ (a^T x - b)/(c^T x - d) : x \in M^- \right\}. \tag{14b}$$

Observe that if both (14a) and (14b) are infeasible, then (2) is infeasible. Else, if either (14a) or (14b) is unbounded, then (2) is unbounded. Else, if both (14a) and (14b) are feasible, then taking the minimum of the two solutions yields the optimal solution of (2). Finally, if only one of these two problems is feasible, then its solution is optimal for (2).

Next, observe that solving (14b) is equivalent to solving

$$\inf\left\{(-a^T x + b)/(-c^T x + d) : x \in M^-\right\}. \qquad (15)$$

However, solving (14a) and (15) is equivalent to solving two instances of (2), where $(c^T x - d) \geq 0$ for all $x \in M$. Thus, in what follows we assume that the condition holds.

This small trick may seem unimportant, but it extends the class of soluble fractional programming problems. Indeed, any fractional programming algorithm that works by solving linear relaxations would not be able to deal with problems where $c^T x \neq d$ for all $x \in M$, but where there exists $\bar{x} \in LP(M)$ with $c^T \bar{x} = d$.

Solving (2) consists in finding a minimizing solution $\bar{x}$, or showing that none exists. If no optimal solution exists, an algorithm for solving (2) should identify a sequence of feasible solutions in M that are asymptotically optimal or that prove the problem is unbounded. We now show how it is possible to obtain the solution to (2) by solving (1).

Suppose that Problem (1) admits an optimal solution λ^*. We can obtain an optimal (or asymptotically optimal) solution of (2) as follows:

1. If there exists an extreme ray $\bar{r}$ of $conv(M)$ such that $\lambda^* = \frac{a^T \bar{r}}{c^T \bar{r}}$ define the following asymptotically optimal sequence of solutions to (2). Let $\bar{x}$ be any feasible solution in M and, for $k \in \mathbb{N}$, define $x(k) = \bar{x} + k\bar{r}$. It is easy to see that $\lim_{k\to\infty} \frac{a^T x(k) - b}{c^T x(k) - d} = \frac{a^T \bar{r}}{c^T \bar{r}} = \lambda^*$. If solving Problem (1) with Algorithm 1, one will obtain $\bar{r}$ from step 2 of the algorithm, and $\bar{x}$ from step 16.
2. If there exists a solution $\bar{x} \in M$ such that $\lambda^* = \frac{a^T \bar{x} - b}{c^T \bar{x} - d}$, then $\bar{x}$ is a minimizer for Problem (2). If solving Problem (1) with Algorithm 1, one will obtain $\bar{x}$ from steps 16 or 22.

Because we are assuming that $(c^T x - d) \neq 0$ for all $x \in M$, it is clear that Problem (1) can never be unbounded. Further, because of the same reason, if Problem (1) is infeasible it can only be because there exists a ray $\bar{r}$ of $conv(M)$ such that $a^T \bar{r} < 0$ and $c^T \bar{r} = 0$. In this case it is easy to see that Problem (2) is unbounded. In order to obtain an unbounded sequence of minimizing solutions we proceed in a manner analogous to case 1 above.

Previous methodologies to solve mixed integer fractional programming problems require combining specialized simplex-like algorithms for fractional programming (see [4] and [2]) with branch-and-bound methods. In contrast, the presented methodology is easy to implement using existing commercial MIP solvers, thus taking advantage of all the significant advances of recent years. In addition, the one-tree algorithm promises to be a competitive new alternative that requires computational testing.

6. Acknowledgements

Marcos Goycoolea was partially funded by FONDECYT grant 11075028, ANILLO grant ACT-88 and Basal project CMM, Universidad de Chile. Daniel Espinoza was partially funded by ICM grant P05-004F.

References

[1] D. Applegate, R. E. Bixby, V. Chvátal, and W. Cook. TSP cuts which do not conform to the template paradigm. In *Computational Combinatorial Optimization, Optimal or Provably Near-Optimal Solutions [based on a Spring School]*, pages 261–304, London, UK, 2001. Springer-Verlag GmbH.

[2] E.B. Bajalinov. *Linear-Fractional Programming: Theory, Methods, Applications and Software*. Springer, 2003.

[3] E. Balas. Facets of the knapsack polytope. *Mathematical Programming*, 8:146–164, 1975.

[4] A. Cambini, S. Schaible, and C. Sodini. Parametric linear fractional programming for an unbounded feasible region. *Journal of global optimization*, 3:157–169, 1993.

[5] V. Chvátal, W. Cook, and D. Espinoza. Local cuts for mixed-integer programming. *Submitted*, 2009.

[6] H. Crowder, E.L. Johnson, and M. Padberg. Solving large-scale zero-one linear-programming problems. *Operations Research*, 31(5):803–834, 1983.

[7] W. Dinkelbach. On nonlinear fractional programming. *Management science*, 13:492–492, 1967.

[8] T. Easton and T. Gutierrez. Sequential and simultaneous uplifting of general integer variables. *Submitted to Mathematical Programming*, 2008.

[9] T. Easton and K. Hooker. Simultaneously lifting sets of binary variables into cover inequalities for knapsack polytopes. *Discrete Optimization*, 2007. doi:10.1016/j.disopt.2007.05.003.

[10] D. G. Espinoza. *On Linear Programming, Integer Programming and Cutting Planes*. PhD thesis, School of Industrial and Systems Engineering, Georgia Institute of Technology, March 2006.

[11] R. E. Gomory. Some polyhedra related to combinatorial problems. *Journal of Linear Algebra and its Applications*, 2:451–558, 1969.

[12] Z. Gu, G. L. Nemhauser, and M. W. P. Savelsbergh. Lifted cover inequalities for 0-1 integer programs: Computation. *INFORMS Journal on Computing*, 10:427–437, 1998.

[13] Z. Gu, G. L. Nemhauser, and M. W. P. Savelsbergh. Lifted cover inequalities for 0-1 integer programs. *Mahtematical Programming*, 85:437–467, 1999.

[14] Q. Louveaux and L.A. Wolsey. Lifting, superadditivity, mixed integer rounding and single node flow sets revisited. *Annals of operations research*, 153:47–77, 2007.

[15] M. W. Padberg. On the facial structure of set packing polyhedra. *Mathematical programming*, 5:199–215, 1973.

[16] M.A. Pollatschek. *Algorithms for finite weighed graphs*. PhD thesis, Faculty of Industrial and Management Engineering, Technion-Israel Institute of Technology, 1970.

[17] J.P.P. Richard, I.R. de Farias, and G.L. Nemhauser. Lifted inequalities for 0-1 mixed integer programming: Basic theory and algorithms. *Mathematical Programming*, 98:89–113, 2003.

[18] J.P.P. Richard, I.R. de Farias, and G.L. Nemhauser. Lifted inequalities for 0-1 mixed integer programming: superlinear lifting. *Mathematical Programming*, 98:115–143, 2003.

[19] S. Schaible. Fractional programming ii: on dinkelbach's algorithm. *Management science*, 22(8):868–873, 1976.

[20] S. Schaible. *Handbook of global optimization*, chapter Fractional programming, pages 495–608. Springer, 1995.

[21] S.Schaible and J. Shi. Recent developments in fractional programming: single ratio and max-min case. In *Proceedings of the 3rd International Conference in Nonlinear Analysis and Convex Analysis, W. Takahashi and T. Tanaka (Eds.)*, pages 2–11, 2004.

[22] I.M. Stancu-Minasian. *Fractional programming: theory, methods and applications*. Kluwer academic publishers, 1997.

[23] L.A. Wolsey. Facets and strong valid inequalities for integer programs. *Operations Research*, 24(2):367–372, 1976.