1 Outline

In this lecture, we will discuss the following topics.

- Methods for solving integer programming
- Chvátal-Gomory cuts
- Gomory's fractional cuts

2 Methods for solving integer programming

The most successful algorithmic frameworks for integer programming are the **branch-and-bound** and the **cutting-plane** methods. Let us discuss the outlines and the ideas behind the methods. Again, we consider an integer program

$$z^* = \max\left\{c^\top x + h^\top y : \ (x, y) \in S\right\} \tag{MILP$_0$}$$

where

$$S = \left\{(x, y) \in \mathbb{Z}^d \times \mathbb{R}^p : \ Ax + Gy \leq b\right\}.$$

The first common step is to solve the LP relaxation given by

$$z_0 = \max\left\{c^\top x + h^\top y : \ (x, y) \in P_0\right\} \tag{LP$_0$}$$

where

$$P_0 = \left\{(x, y) \in \mathbb{R}^d \times \mathbb{R}^p : \ Ax + Gy \leq b\right\}.$$

The modern software uses the **barrier method** and the **simplex method**[1]. Let z_0 be the optimal value of the LP relaxation, and assume that z_0 is finite. For rational data, z_0 being finite implies that the optimal value z^* of the integer program is also finite (we will see this later).

Let (x^0, y^0) be an optimal solution to the LP relaxation. What if $x^0 \in \mathbb{Z}^d$? Then $(x^0, y^0) \in S$. This implies the following

$$\begin{aligned}
\max\left\{c^\top x + h^\top y : \ (x, y) \in S\right\} &\leq \max\left\{c^\top x + h^\top y : \ (x, y) \in P_0\right\} \\
&= c^\top x^0 + h^\top y^0 \\
&\leq \max\left\{c^\top x + h^\top y : \ (x, y) \in S\right\}.
\end{aligned}$$

Here, the left-hand side and the right-most side coincide, so the equalities hold throughout, which implies that (x^0, y^0) is an optimal solution to the integer program!

In general, the LP relaxation does not necessarily return an integer solution, and x^0 may have some fractional components. Here, the branch-and-bound and the cutting-plane methods provide two natural strategies to deal with the situation where x^0 has some fractional component.

[1] https://www.gurobi.com/documentation/9.5/refman/choosing_the_right_algorit.html

2.1 Branch-and-bound method

Suppose that component x_j^0 is fractional for some $1 \leq j \leq d$. Then we know that

$$x_j \geq \lceil x_j^0 \rceil \quad \text{or} \quad x_j \leq \lfloor x_j^0 \rfloor.$$

Based on this, we define

$$S_1 = S \cap \left\{(x,y):\ x_j \geq \lceil x_j^0 \rceil\right\} \quad \text{and} \quad S_2 = S \cap \left\{(x,y):\ x_j \geq \lceil x_j^0 \rceil\right\},$$

and in fact, $S = S_1 \cup S_2$. Moreover, we create two subproblems

$$\max\left\{c^\top x + h^\top y:\ (x,y) \in S_1\right\}, \tag{MILP$_1$}$$

$$\max\left\{c^\top x + h^\top y:\ (x,y) \in S_2\right\}. \tag{MILP$_2$}$$

Here, the maximum of the optimal values of (MILP$_1$) and (MILP$_2$) would be the optimal value of the origirnal integer program (MILP$_0$).

Note that starting from (MILP$_0$), we have generated two subproblems (MILP$_1$) and (MILP$_2$). We can represent this as a tree structure as in Figure 26.1

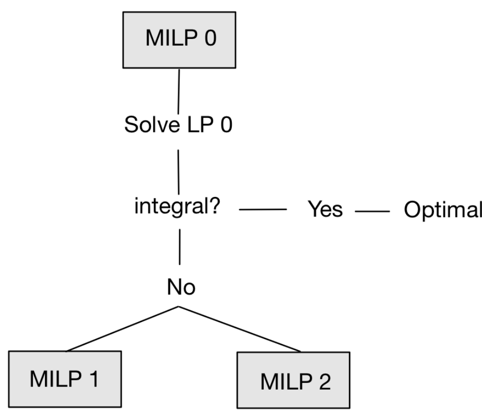

Figure 26.1: Generating two subproblems

For (MILP$_1$) and (MILP$_2$), we solve their LP relaxations,

$$\max\left\{c^\top x + h^\top y:\ (x,y) \in P_1\right\}, \tag{LP$_1$}$$

$$\max\left\{c^\top x + h^\top y:\ (x,y) \in P_2\right\} \tag{LP$_2$}$$

where

$$P_1 = P_0 \cap \left\{(x,y):\ x_j \geq \lceil x_j^0 \rceil\right\} \quad \text{and} \quad P_2 = P_0 \cap \left\{(x,y):\ x_j \leq \lfloor x_j^0 \rfloor\right\}.$$

- If (LP$_1$) is infeasible, then it means (MILP$_1$) is infeasible. Then we can remove it from the search tree (**Prune by infeasibility**).
- If (LP$_1$) returns an integral solution, it means (MILP$_1$) is solved. Then we keep the integral solution but remove (MILP$_1$) from the search tree (**Prune by integrality**).

- If (LP$_1$) returns a **fractional** solution while the value of (LP$_1$) is less than or equal to the value of the current best integral solution (possibly from solving (MILP$_2$)), then we remove (MILP$_1$) from the search tree (**Prune by value**).

- If (LP$_1$) returns a **fractional** solution while the value of (LP$_1$) is greater than the value of the current best integral solution, then we apply the **branching** procedure on a fractional component.

We apply the same rule for (MILP$_2$) and repeat the procedure for other subproblems remaining in the tree. The tree that this process generates is called the **branch-and-bound** tree.

2.2 Cutting-plane method

Remember our example of the two-dimensional mixed integer linear set. The MIR inequality is valid for the mixed-integer set but it is violated by the point $(\beta, 0)$. So, if we take the set of solutions satisfying the MIR inequalities, the point $(\beta, 0)$ is removed. Here, we also say that the inequality **cuts off** the point and that the point is **separated** from the mixed-integer set. In this sense, we say that the MIR inequality is a **cutting-plane** or a **cut**.

Likewise, we take the solution (x^0, y^0) that is optimal to the LP relaxation (LP$_0$), and we find a cutting-plane that separates (x^0, y^0) from S. Here, an inequality $\alpha^\top x + \gamma^\top y \leq \beta$ is a cutting-plane that separates (x^0, y^0) from S if

$$\alpha^\top x + \gamma^\top y \leq \beta \quad \forall (x, y) \in S \quad \text{and} \quad \alpha^\top x^0 + \gamma^\top y^0 > \beta.$$

Then we get a new relaxation

$$\max \left\{ c^\top x + h^\top y : \ (x, y) \in P_1 \right\}$$

where

$$P_1 = P_0 \cap \left\{ (x, y) : \ \alpha^\top x + \gamma^\top y \leq \beta \right\}.$$

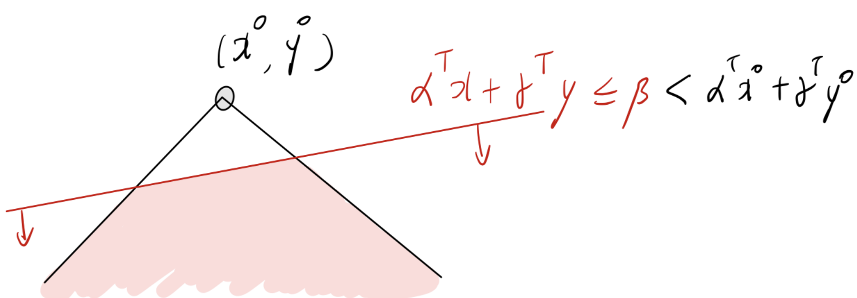

Figure 26.2: Applying a cutting-plane

Repeating the process, we obtain the following cutting-plane algorithm.

- Set $t = 0$.

- $P_0 = \left\{ (x, y) \in \mathbb{R}^d \times \mathbb{R}^p : \ Ax + Gy \leq b \right\}$ be the solution set of the LP relaxation (LP$_0$).

- Until we find an integral solution, repeat the following procedure.

3. Solve and obtain an optimal solution (x^t, y^t) to $\max\left\{c^\top x + h^\top y : (x,y) \in P_t\right\}$.
4. If (x^t, y^t) is integral, then we stop and (x^t, y^t) is the optimal integral solution.
5. If (x^t, y^t) is not integral, then find a cutting-plane $\alpha^\top x + \gamma^\top y \leq \beta$ that separates (x^t, y^t) and set
$$P_{t+1} = P_t \cap \left\{(x,y) : \ \alpha^\top x + \gamma^\top y \leq \beta\right\}$$
and $t \leftarrow t+1$.

2.3 Branch-and-cut method

The Branch-and-cut method is basically combining the branch-and-bound method and the cutting-plane algorithm. While running the branch-and-bound procedure, we may find and apply a cutting-plane that separates a fractional solution obtained from solving a subproblem.

The state-of-the-art integer programming software such as CPLEX and Gurobi implements the branch-and-cut method. They apply some specially designed branching rules and cut generation schemes.

One may want to use some problem specific cuts, such as subtour elimination inequalities for TSP and odd set inequalities for the matching problem. In that case, we can use the **(cut) callback** feature within CPLEX and Gurobi to apply user-defined cuts. Basically, the callback feature invokes a node in the branch-and-bound tree and apply the user-defined cut.

- CPLEX: https://www.ibm.com/docs/en/icos/22.1.1?topic=legacy-cut-callback
- Gurobi: https://www.gurobi.com/documentation/10.0/refman/cpp_cb_addcut.html

It is often the case that adding problem-specific cuts leads to a significant improvement in solution time.

How do you find problem specific cuts? One common way is to use the software **PORTA**.

- PORTA: https://porta.zib.de

Given a finite set of vectors as input, PORTA computes the convex hull. Conversely, given a set of linear inequalities, PORTA computes the **extreme points** and the **extreme rays**.

3 Chvátal-Gomory cuts

Starting with this lecture, we discuss **general-purpose cutting planes** for integer programming in-depth. What is a general-purpose cutting plane? Given a polyhedron $P = \{x \in \mathbb{R}^d_+ : Ax \leq b\}$ and the set of integer solutions $S = P \cap \mathbb{Z}^d$, we want to find an inequality $c^\top x \leq d$ such that

- $c^\top x \leq d$ is valid for S, i.e., every point $z \in S$ satisfies $c^\top z \leq d$,
- $c^\top x \leq d$ may be violated by some point in P.

Here, a general-purpose cutting plane is a prodcedure of generating such inequalities no matter which values of A and b are given. The first is the so-called **Chvátal-Gomory cuts**.

Ralph Gomory [Gom58] discovered the first finitely convergent cutting plane algorithm for solving pure integer linear programs. The cuts used within his algorithms are called **Gomory's fractional**

cuts. In fact, the terminology refers to a "procedure" of generating valid inequalities/cuts for a given (pure) integer program. That said, one may generate and apply Gomory's fractional cuts for any pure integer linear programs. For this reason, Gomory's fractional cuts are general-purpose cutting planes.

Vašek Chvátal [Chv73] later studied a cut-generation scheme that unifies some of the existing valid inequalities for combinatorial optimization problems such as the matching problem and the stable set problem. Although the initial focus was on combinatorial optimization problems, the cut-generation scheme can be applied to any pure integer linear programs. In the paper, Chvátal proved that the cuts obtained from his cut-generation scheme are essentially equivalent to Gomory's fractional cuts. For this reason, the generated cuts are now called **Chvátal-Gomory cuts**.

Chvátal-Gomory cuts are prevalent in the discrete optimization literature. Many fundamental classes of facet-defining inequalities for combinatorial optimization problems are Chvátal-Gomory cuts, e.g., odd set inequalities for the matching problem and odd circuit inequalities for the stable set problem. Chvátal-Gomory cuts are computationally effective for solving integer linear programs in practice [FL07, BCD+08], and Chvátal-Gomory cuts for nonlinear integer programs have also been studied [ÇI05].

Then, what are Chvátal-Gomory cuts? How do we obtain them? Today, we will see combinatorial, rounding, arithmetic, and geometric arguments to obtain Chvátal-Gomory cuts.

3.1 Combinatorial argument

Consider the matching problem. Let $G = (V, E)$ be an undirected graph, and the integer programming formulation for the matching problem is given as follows.

$$\begin{aligned} \max \quad & \sum_{e \in E} w_e x_e \\ \text{s.t.} \quad & \sum_{e \in \delta(v)} x_e \leq 1, \quad v \in V \\ & x_e \in \{0,1\}, \quad e \in E \end{aligned}$$

where

- w_e is the weight of edge $e \in E$,
- $\delta(v) = \{e \in E : v \text{ is adjacent to } e\}$.

Here, we want to generate valid inequalities for

$$S = \left\{ x \in \{0,1\}^E : \sum_{e \in \delta(v)} x_e \leq 1, \quad v \in V \right\}.$$

Let $U \subseteq V$ be a subset of the vertex set with an odd number of vertices. Then look at the set of edges that are fully contained in U. Then the following inequality is satisfied by S.

$$\sum_{e \in E(U)} x_e \leq \frac{|U| - 1}{2}$$

where $E(U)$ is the set of edges fully contained in U. We call this inequality an **odd-set inequality**. Why is the odd-set inequality valid? Note that the left-hand side $\sum_{e \in E(U)} x_e$ counts the maximum

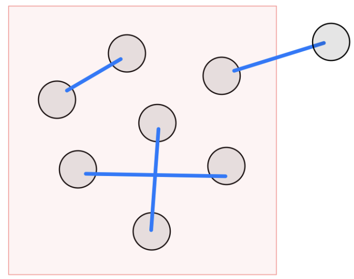

Figure 26.3: Odd cardinality subset

number of edges from $E(U)$ a maching can take. Here, a matching takes an edge in $E(U)$ means that two vertices in U are matched. Note that $|U|$ is odd, and by parity, at least one vertex always remains unmatched. Equivalently, at most $|U|-1$ vertices in U can be matched by a matching. Hence, $E(U)$ contains at most $(|U|-1)/2$ edges in a matching.

Chvátal [Chv73] developed the following systematic argument to derive odd-set inequalities. First, take the inequalities $\sum_{e\in\delta(v)} x_e \leq 1$ for all vertices $v \in U$, and sum them up:

$$\sum_{v\in U}\sum_{e\in\delta(v)} x_e \leq |U|.$$

Here, in the left-hand side, x_e appears twice if e is fully contained in u, and x_e appears once if only one of e's two ends is contained in U. In other words, x_e appears twice if $e \in E(U)$, and x_e appears once if $e \in \delta(U)$ where

$$\delta(U) = \{e \in E : \text{ one end of } e \text{ is in } U \text{ while the other end of } e \text{ is not in } U\}.$$

This implies that the inequality is equivalent to

$$2\sum_{e\in E(U)} x_e + \sum_{e\in\delta(U)} x_e \leq |U|.$$

Now let us divide each side by 2.

$$\sum_{e\in E(U)} x_e + \frac{1}{2}\sum_{e\in\delta(U)} x_e \leq \frac{|U|}{2}.$$

Here, as $x_e \geq 0$ for all $e \geq E$, the inequality implies that

$$\sum_{e\in E(U)} x_e + \lfloor\frac{1}{2}\rfloor\sum_{e\in\delta(U)} x_e = \sum_{e\in E(U)} x_e \leq \frac{|U|}{2}.$$

Finally, we know that the left-hand side is an integer, so rounding down the right-hand side still preserves validity.

$$\sum_{e\in E(U)} x_e \leq \lfloor\frac{|U|}{2}\rfloor = \frac{|U|-1}{2}.$$

As a result, we deduce the odd-set inequality for subset U.

In fact, the odd set inequalities describe the **matching polytope**.

Theorem 26.1 (Edmonds [Edm65]). *For the matching problem,*

$$\mathrm{conv}(S) = \left\{ x \in \mathbb{R}^E_+ : \sum_{e \in \delta(v)} x_e \leq 1, \quad \forall v \in V, \quad \sum_{e \in E(U)} x_e \leq \frac{|U|-1}{2}, \quad \forall U \subseteq V \text{ odd} \right\}.$$

Here, $\mathrm{conv}(S)$ *is called the* ***matching polytope****.*

3.2 Chvátal's integer rounding procedure

Given $v = (v_1, \ldots, v_d) \in \mathbb{R}^d$, let $\lfloor v \rfloor$ denote the vector $(\lfloor v_1 \rfloor, \ldots, \lfloor v_d \rfloor)$ obtained after rounding down each component of v. Let $P = \{x \in \mathbb{R}^d_+ : Ax \leq b\}$ and $S = P \cap \mathbb{Z}^d$. Here, Chvátal's rounding procedure is as follows.

1. Take $\lambda \in \mathbb{R}^m_+$ and $\lambda^\top Ax \leq \lambda^\top b$ valid for P.
2. Round down the left-hand side to obtain $\lfloor A^\top \lambda \rfloor^\top x \leq b^\top \lambda$. This inequality is valid because $\lambda^\top Ax \leq \lambda^\top b$ and $x \geq 0$.
3. Round down the right-hand side to obtain $\lfloor A^\top \lambda \rfloor^\top x \leq \lfloor b^\top \lambda \rfloor$. This inequality is valid because $x \in \mathbb{Z}^d$ and $\lfloor A^\top \lambda \rfloor \in \mathbb{Z}^d$.

In general, we consider polyhedron $P = \{x \in \mathbb{R}^d : Ax \leq b\}$ that is not necessarily contained in $\mathbb{R}^d_+$ and $S = P \cap \mathbb{Z}^d$. For general polyhedra, Chvátal's rounding procedure works as follows.

1. Take $\lambda \in \mathbb{R}^m_+$ such that $A^\top \lambda \in \mathbb{Z}^d$. Then $(A^\top \lambda)^\top x \leq b^\top \lambda$ is valid for P.
2. Round down the right-hand side to obtain $(A^\top \lambda)^\top x \leq \lfloor b^\top \lambda \rfloor$. This inequality is valid because $x \in \mathbb{Z}^d$ and $A^\top \lambda \in \mathbb{Z}^d$.

In fact, the rounding procedure for the nonnegative case, i.e., $P \subseteq \mathbb{R}^d_+$, is a special case. When P is given by $P = \{x \in \mathbb{R}^d_+ : Ax \leq b\}$, the constraint system is given by

$$Ax \leq b, \ -x \leq 0.$$

Recall that we took a multiplier vector λ to obtain the corresponding Chvátal-Gomory cut

$$\lfloor A^\top \lambda \rfloor^\top x \leq b^\top \lambda.$$

Note that this inequality is equivalent to

$$\lambda^\top Ax + (A^\top \lambda - \lfloor A^\top \lambda \rfloor)^\top (-x) \leq b^\top \lambda.$$

This is equivalent to take multipliers $(\lambda, A^\top \lambda - \lfloor A^\top \lambda \rfloor)$ for system $Ax \leq b, -x \leq 0$.

An inequality that can be obtained from Chvátal's rounding procedure is a Chvátal-Gomory cut. Note that the multiplier vector λ can be arbitrary. In particular, there are infinitely many choices for λ, thereby leading to infinitely many Chvátal-Gomory inequalities.

3.3 Geometric view

Let $P = \{x \in \mathbb{R}^d : \ Ax \leq b\}$ be a polyhedron and $S = P \cap \mathbb{Z}^d$ be the set of integer solutions in P. Let $c \in \mathbb{Z}^d$. Then

$$c^\top x \leq \max\left\{c^\top y : \ y \in P\right\}$$

is valid for P. Let $d \in \mathbb{Z}$ be defined as

$$d = \lfloor \max\left\{c^\top y : \ y \in P\right\} \rfloor.$$

Then it follows that

$$P \cap \{x \in \mathbb{R}^d : \ c^\top x \geq d + 1\} = \emptyset.$$

Moreover, $c^\top x \leq d$ is valid for $\mathrm{conv}(S)$.

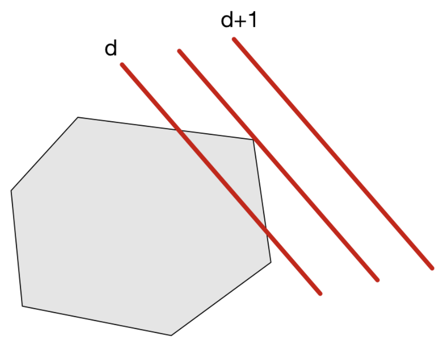

Figure 26.4: Geometric view of Chvátal-Gomory cuts

Proposition 26.2. *$c^\top x \leq d$ is a Chvátal-Gomory cut, and every Chvátal-Gomory cut can be obtained this way.*

Proof. Let $\beta = \max\{c^\top x : x \in P\}$. Then $\beta < d + 1$. Moreover, $c^\top x \leq \beta$ is valid for P. Then by Strong duality, there exists $\lambda \in \mathbb{R}^m_+$ such that $A^\top \lambda = c$ and $b^\top \lambda = \beta$. Therefore, the corresponding Chvátal-Gomory cut $(A^\top \lambda)^\top x \leq \lfloor b^\top \lambda \rfloor$ is equivalent to $c^\top x \leq \lfloor \beta \rfloor$.

Conversely, let $(A^\top \lambda)^\top x \leq \lfloor b^\top \lambda \rfloor$ be a Chvátal-Gomory cut. By construction, $(A^\top \lambda)^\top x \leq b^\top \lambda$ is valid for P, and thus

$$P \cap \{x \in \mathbb{R}^d : \ (A^\top \lambda)^\top x \geq \lfloor b^\top \lambda \rfloor + 1\} = \emptyset,$$

as required. $\square$

3.4 Modular arithmetic and Gomory's fractional cuts

Assume that

$$S = \left\{x \in \mathbb{Z}^d_+ : \ \sum_{j=1}^{d} a_j x_j = a_0\right\}.$$

Let p be a positive integer, and assume that

$$a_j = r_j + q_j p, \quad j = 0, 1, \ldots, d$$

where $0 \leq r_j < p$ and $q_j \in \mathbb{Z}$. Then $\sum_{j=1}^{d} a_j x_j = a_0$ can be rewritten as

$$\begin{aligned}\sum_{j=1}^{d} r_j x_j &= r_0 + \left(q_0 p - \sum_{j=1}^{d} q_j p\right)\\ &= r_0 + \left(q_0 - \sum_{j=1}^{d} q_j\right) p\end{aligned}$$

Then it follows that

$$S \subseteq \left\{x \in \mathbb{Z}_+^d : \sum_{j=1}^{d} r_j x_j = r_0 + kp \text{ where } k \in \mathbb{Z}\right\}.$$

Since $r_1, \ldots, r_d$ are all nonnegative and $x \geq 0$,

$$\sum_{j=1}^{d} r_j x_j \geq 0.$$

We know that if $x \in S$, then the left-hand side has a value of the form $r_0 + kp$. What is the smallest nonnegative value that $r_0 + kp$ can take? It is attained when $k = 0$, in which case the value is r_0. Therefore, it follows that

$$\sum_{j=1}^{d} r_j x_j \geq r_0$$

is valid for S. Remember that p is an arbitary positive integer. In particular, we may choose $p = 1$. Then the corresponding inequality is

$$\sum_{j=1}^{d} (a_j - \lfloor a_j \rfloor) x_j \geq a_0 - \lfloor a_0 \rfloor.$$

This is a **Gomory's fractional cut**.

Remark 26.3. To generate a Gomory's fractional cut for $S = \left\{x \in \mathbb{Z}_+^d : Ax \leq b\right\}$, we first add slack variables to make constraints equalities. More precisely, we add nonnegative variables s to have

$$Ax + s = b.$$

Then any equation

$$(A^\top \lambda)^\top x + \lambda^\top s = b^\top \lambda$$

can be used to produce a fractional cut.

Example 26.4. Consider the following system of constraints.

$$\begin{aligned}-x_1 + 2x_2 &\leq 4\\ 5x_1 + x_2 &\leq 20\\ -2x_1 - 2x_2 &\leq -7\\ x_1, x_2 &\in \mathbb{Z}_+\end{aligned}$$

By adding slack variables, we obtain

$$\begin{aligned}
-x_1 + 2x_2 + s_1 &= 4\\
5x_1 + x_2 + s_2 &= 20\\
-2x_1 - 2x_2 + s_3 &= -7\\
x_1, x_2 &\in \mathbb{Z}_+\\
s_1, s_2, s_3 &\geq 0
\end{aligned}$$

Let us multiply $\lambda = (-1/11, 2/11, 0)$ to the equality constraints. Then we deduce that

$$x_1 - \frac{1}{11}s_1 + \frac{2}{11}s_2 = \frac{36}{11}.$$

Then the corresponding Gomory's fractional cut is given by

$$\frac{10}{11}s_1 + \frac{2}{11}s_2 \geq \frac{3}{11}.$$

In the original space, this is equivalent to

$$\frac{10}{11}(4 + x_1 - 2x_2) + \frac{2}{11}(20 - 5x_1 - x_2) \geq \frac{3}{11}.$$

This inequality gives us

$$x_2 \leq \frac{7}{2}.$$

It is known that the Chvátal-Gomory cuts and Gomory's fractional cuts are essentially equivalent. In particular, every Chvátal-Gomory cut can be obtained as a Gomory's fractional cut, and every Gomory's fractional cut can be obtained as a Chvátal-Gomory cut [Chv73].

References

[BCD+08] P. Bonami, G. Cornuéjols, S. Dash, Matteo Fischetti, and Andrea Lodi. Projected Chvátal-gomory cuts for mixed integer linear programs. *Mathematical Programming*, 113:241–257, 2008. 3

[Chv73] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4(4):305–337, 1973. 3, 3.1, 3.4

[ÇI05] M. T. Çezik and G. Iyengar. Cuts for mixed 0-1 conic programming. *Mathematical Programming*, 104:179–202, 2005. 3

[Edm65] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965. 26.1

[FL07] Matteo Fischetti and Andrea Lodi. Optimizing over the first Chvátal closure. *Mathematical Programming*, 110:3–20, 2007. 3

[Gom58] R. E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64:275–278, 1958. 3