CMPUT 675: Approximation Algorithms **Fall 2014**

Lecture 9 (Sep 22): Linear Programming

Lecturer: Zachary Friggstad *Scribe: Zachary Friggstad*

9.1 Linear Programming Introduction

By now, you should realize the importance of efficiently computable *lower bounds* and *upper bounds* in approximation algorithms. Often, an approximation algorithm is devised by finding a related structure that we can efficiently compute and describing how to transform it into a feasible solution without inflating the cost too much.

Here are some examples we have seen:

- In VERTEX COVER, we compute a maximal matching. Taking both endpoints of each of these edges is a 2-approximation.
- In the TRAVELING SALESMAN PROBLEM, we used two lower bounds in Christofides' algorithm: the cost of a minimum spanning tree and twice the cost of a min-cost matching on any even-size subset of vertices.
- In MINIMIZING MAKESPAN ON IDENTICAL PARALLEL MACHINES, one lower bound was the efficiently-computable solution to the scaled running times. This was close to the true optimum essentially because we did not scale the running times by much.

There is certainly some art to this: we have to understand the problem structure well enough to identify efficiently computable structures that a) provide a bound on the optimum solution value and b) can be transformed into a feasible solution with comparable cost.

Linear programming is a versatile tool for obtaining such lower bounds. At a high level, we use linear programs (LPs) by first modelling the problem at hand by an integer program and then dropping the integrality requirements to obtain a linear program. Since this is a relaxation of the original integer programming model, the optimum linear programming solution immediately yields a lower or upper bound (depending on whether we are minimizing or maximizing the objective). The upshot is that we can efficiently solve linear programs in polynomial time. In general, the variables in an optimum LP solution will be fractional and the challenge is to round them to feasible integer solutions while not losing much in the objective.

9.1.1 Intro by Example: Minimum Weight Vertex Cover

Suppose $G = (V, E)$ is a graph and each $v \in V$ has a *cost* $c(v) \geq 0$. The goal is to find a vertex cover $U \subseteq V$ with minimum *cost*, not minimum size. This captures the classic VERTEX COVER problem from the first lecture by using $c(v) = 1$ for each vertex v.

Here is an integer programming formulation of the problem. We introduce a variable x_v for each $v \in V$. The idea is that setting $x_v = 1$ means we use v in the vertex cover and setting $x_v = 0$ means we do not use v in the vertex cover.

$$\begin{array}{lrcll} \text{minimize :} & \sum_{v \in V} c(v) \cdot x_v & & & \\ \text{subject to :} & x_u + x_v & \geq & 1 & \text{for each edge } (u,v) \in E \\ & x_v & \in & \{0,1\} & \text{for each vertex } v \in V \end{array}$$

This is an integer program. The last line says each x_v value should be either 0 or 1 which, as mentioned, corresponds to excluding or including v in the vertex cover. The second line asserts that for each edge $(u, v) \in E$, at least one of x_u or x_v should be 1. This is equivalent to saying at least one of u or v should be in the vertex cover. Finally, the first line says we should minimize the cost of the vertices v whose x_v value is set to 1. This is the same as minimizing the cost of the vertex cover.

Of course, this has not bought us much. Solving integer programs is still NP-hard. They key trick behind a vast number of approximation algorithms is to relax the integrality requirements of the integer program. This results in a linear program that we call the *LP relaxation* of the integer program.

$$\begin{array}{lrcll} \text{minimize :} & \sum_{v \in V} c(v) \cdot x_v & & & \\ \text{subject to :} & x_u + x_v & \geq & 1 & \text{for each edge } (u,v) \in E \\ & 0 \leq x_v & \leq & 1 & \text{for each vertex } v \in V \end{array}$$

We can solve this linear program in polynomial time. More on this later, but let us see what happens in this Minimum Weight Vertex Cover Problem.

Suppose the vector $\mathbf{x}^*$ is an optimal solution to the linear programming relaxation. We have $\sum_{v \in V} c(v) \cdot x_v^* \leq OPT$ because the minimum weight vertex cover corresponds naturally to a feasible solution to the linear program.

Let us form an integer solution $\mathbf{x}'$ by rounding $\mathbf{x}^*$. That is, set $x'_v = 1$ if $x_v^* \geq 1/2$, otherwise set $x'_v = 0$ if $x_v^* < 1/2$. Two things to observe:

- For every $v \in V$, $x'_v \leq 2x_v^*$. Therefore, $\sum_{v \in V} c(v) \cdot x'_v \leq 2 \cdot \sum_{v \in V} c(v) \cdot x_v^* \leq 2 \cdot OPT$.
- For every $(u, v) \in E$, since $x_u^* + x_v^* \geq 1$ (from the LP constraints), then one of x_u^* or x_v^* is $\geq 1/2$. Therefore, at least one of x'_u or x'_v is 1.

That is, $U = \{v \in V : x'_v = 1\}$ is a vertex cover of cost at most $2 \cdot OPT$. Simple!

This is more the exception than the norm. While LPs are used quite frequently in approximation algorithms it is hardly ever the case that rounding each variable off to the nearest integer gives a feasible solution with cost close to the LP optimum. Thus, the art is now in crafting a good rounding algorithm. In fact, in coming lectures we will see that coming up with a good LP relaxation can sometimes be an interesting challenge by itself!

9.2 General Theory

Here we summarize the basic theory of the Linear Programming problem we need to get started. A few more things will be mentioned as the semester rolls on. The claims about solving linear programs will not be prove here, but proofs of some of the structural results are in an appendix to these notes. These proofs are merely for the curious as understanding them is not a requirement for future material.

A *linear program* (LP) in *standard form* is presented as

$$\begin{array}{rrcl} \text{minimize :} & \mathbf{c}^T \cdot \mathbf{x} & & \\ \text{subject to :} & \mathbf{A} \cdot \mathbf{x} & \geq & \mathbf{b} \\ & \mathbf{x} & \geq & \mathbf{0} \end{array}$$

A more succinct representation: $\min\{\mathbf{c}^T \cdot \mathbf{x} : \mathbf{A} \cdot \mathbf{x} \geq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$.

Here, $\mathbf{x}$ is an n-dimensional column vector of variables $(x_1, \ldots, x_n)^T$, $\mathbf{A} = (a_{ji})$ is an $m \times n$ matrix, $\mathbf{b} = (b_j)$ is an m-dimensional column vector, and $\mathbf{c} = (c_i)$ is an n-dimensional column vector. The matrices/vectors $\mathbf{A}, \mathbf{b}$ and $\mathbf{c}$ are all known values (i.e. are not variables). The constraints $\mathbf{A} \cdot \mathbf{x} \geq \mathbf{b}$ mean $(\mathbf{A} \cdot \mathbf{x})_j \geq b_j$ for every $1 \leq j \leq m$. Similarly, $\mathbf{x} \geq 0$ means all variables must take nonnegative values. We say that n is the number of variables and m is the number of constraints.

Example 1:
Consider the following linear program in standard form.

$$\begin{array}{rrcl} \text{minimize :} & 3x_1 + x_2 & & \\ & & & \\ \text{subject to :} & 8x_1 + x_2 & \geq & 8 \\ & 6x_1 + 4x_2 & \geq & 24 \\ & 4x_1 + 7x_2 & \geq & 28 \\ & & & \\ & x_1, x_1 & \geq & 0 \end{array}$$

In matrix/vector notation, the LP is $\max\{\mathbf{c}^T \cdot \mathbf{x} : \mathbf{A} \cdot \mathbf{x} \geq \mathbf{b}, \mathbf{x} \geq 0\}$ where

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} 8 & 1 \\ 6 & 4 \\ 4 & 7 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 8 \\ 24 \\ 28 \end{pmatrix}, \text{ and } \mathbf{c} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$$

Visually, the set of feasible solutions to this LP lie in the space above the lines in Figure 9.1.

There is a unique optimum solution to this LP, namely $\mathbf{x}^* = \begin{pmatrix} 7/13 \\ 48/13 \end{pmatrix}$ which has value $3 \cdot \frac{7}{13} + \frac{48}{13} = \frac{69}{13}$. Clearly it is feasible, to see it is optimum, consider $\frac{17}{52}$ times the first constraint plus $\frac{5}{52}$ times the third constraint. This shows $3x_1 + x_2 \geq \frac{69}{13}$ for any feasible solution, so the proposed solution must be optimal.

In general, there is always a way to prove optimality of an LP solution by considering some combination of the constraints as in this example. We will discuss this later in the course when we talk about LP duality.

Finally, notice that in this example that the optimal solution is in one of the "corners" of the sketch in Figure 9.1. More specifically, two of the constraints are satisfied by $\mathbf{x}^*$ with equality. More on this below.

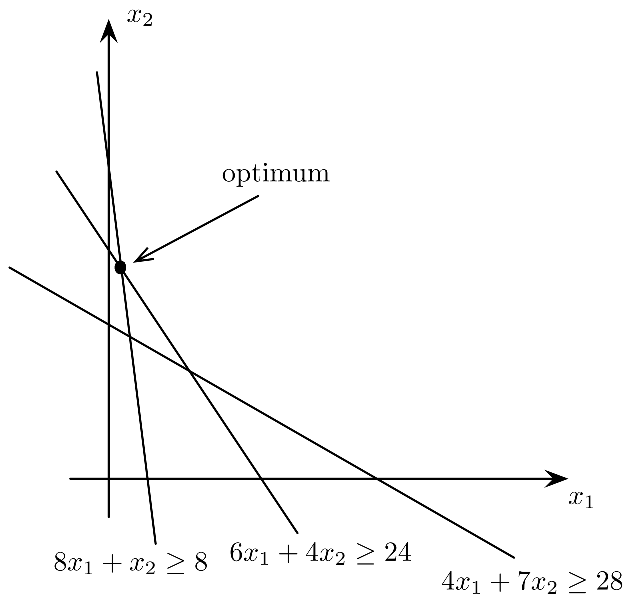

Figure 9.1: Sketch of the LP in the Example 1. The feasible solutions lie "above" all of the lines (by $\mathbf{A} \cdot \mathbf{x} \geq \mathbf{b}$) and in upper-right quadrant (by $\mathbf{x} \geq 0$).

The more general definition of a linear program is a bit more flexible. We may want to maximize the objective function, we can use $=$ or $\geq$ instead of $\leq$ for some of the individual constraints $(\mathbf{A} \cdot \mathbf{x})_j \geq b_j$, and the nonnegativity constraints are not necessary. It is a simple exercise to show that any LP according to this more general definition can be transformed into an "equivalent" LP in standard form. For example, maximize $\mathbf{c}^T \cdot \mathbf{x}$ is equivalent to minimize $(-\mathbf{c})^T \cdot \mathbf{x}$.

Assumption: For the rest of this general LP introduction, we will assume the LPs are in standard form. We will use many LPs that are not in this standard form in this course; everything discussed here generalizes in a simple way unless otherwise noted.

An assignment of values $\mathbf{x}^*$ to $\mathbf{x}$ is said to be *feasible* if $\mathbf{A} \cdot \mathbf{x}^* \geq \mathbf{b}$ and $\mathbf{x}^* \geq \mathbf{0}$. Linear programs can be classified in one of three ways.

- The linear program is *infeasible* if there is no feasible solution.
- The linear program is *unbounded* if for all values $\gamma \in \mathbb{R}$ there is some feasible solution $\mathbf{x}$ with $\mathbf{c}^T \cdot x \leq \gamma$.
- The linear program has a *optimum solution* if there is a feasible solution $\mathbf{x}^*$ such that for all other feasible solutions $\mathbf{x}'$, $\mathbf{c}^T \cdot \mathbf{x}^* \leq \mathbf{c}^T \cdot \mathbf{x}'$. Call such a solution $\mathbf{x}^*$ an *optimal solution.*

Theorem 1 *Every linear program is either infeasible, unbounded, or has an optimum solution.*

Note that Theorem 1 would not be true of we allowed strict inequalities, e.g. the one-dimensional problem $\min\{x : x > 1\}$ cannot be classified in any of these three ways.

9.2.1 Solving LPs

While much of the theory behind linear programming works when the given values $\mathbf{A}, \mathbf{b}, \mathbf{c}$ consist of arbitrary real numbers, we assume they are rational numbers as otherwise the issue of even representing and performing arithmetic with the values becomes quite complicated. One other parameter which we will rarely use beyond this section is the *bit complexity* of the linear program. This is the maximum number of bits used to represent any of the a_{ji}, b_j, c_i values in the input (a single rational number p/q is represented using $\sim \log_2 p + \log_2 q$ bits). Denote this quantity by Δ.

To *solve* a LINEAR PROGRAMMING instance means to correctly determine if it is infeasible, unbounded and has an optimal solution. In the case there is an optimal solution, such an optimal solution should be produced.

Theorem 2 *An instance of* LINEAR PROGRAMMING *can be solved in time* $\text{poly}(n, m, \Delta)$ *with the additional guarantee that if an optimal solution exists, then the optimal solution* $\mathbf{x}^*$ *returned by the algorithm is rational and can be described in* $\text{poly}(n, \Delta)$ *bits.*

We will not discuss how to solve LPs, Theorem 2 is really just there to let you know that they can be solved in polynomial time and that an optimum solution $\mathbf{x}^*$ can be stored using polynomially many bits. Interior point methods such as those described in [Y97] can be used to solve linear programs in polynomial time.

Note that the bit complexity of the optimum solution $\mathbf{x}^*$ does not depend on the number of constraints; this will be important later when we discuss how to solve LPs with exponentially many constraints that are described in an implicit manner.

9.3 Integrality Gaps

In this course, we often model an optimization problem as an integer program and then solve it's linear programming relaxation. We hope to get good approximations by rounding an optimum LP solution to an integer value. If we find one with cost within an α-factor of the LP optimum then this is an α-approximation.

One barrier to obtaining good approximations by comparing the cost of the solution versus the LP optimum is the following value.

Definition 1 *Let $\mathcal{P}$ be a linear program. Let OPT_{LP} be the optimum solution value to the $\mathcal{P}$ and let OPT_{IP} be the optimum* integer *solution to $\mathcal{P}$. The* integrality gap *of $\mathcal{P}$ is the ratio OPT_{IP}/OPT_{LP}.*

Sometimes we say things like the integrality gap is at most $O(\log n)$ for an LP relaxation of a particular problem. This means exactly what it sounds like. There are constants c, n_0 such that for any $n \geq n_0$ and any instance of the problem with size n (e.g. n vertices, n variables, n items, etc.), the integrality gap of the LP relaxation for that instance is at most $c \cdot \log_2 n$.

9.3.1 Vertex Cover Again

Recall the LP relaxation for MINIMUM WEIGHT VERTEX COVER from Section 9.1.1. We showed that any LP solution can be rounded to an integer solution with at most twice the cost of the LP solution. This shows $OPT_{IP} \leq 2 \cdot OPT_{LP}$ for this LP meaning the integrality gap is at most 2.

Can we do better? Is the integrality gap less than two? Yes, but not by much. It is, in fact, possible to show the integrality gap of this LP relaxation is at most $2 - \frac{2}{n}$ where $n = |V|$. This follows from an exercise question in Assignment 3.

This is essentially tight. Consider the complete graph $G = (V, E)$ with $c(v) = 1$ for each $v \in V$. The solution $x_v = \frac{1}{2}$ is feasible for the LP and has cost $n/2$, but the optimum integer solution has cost $n - 1$ as excluding any two vertices from the cover would result in some edge not being covered. In this instance, the integrality gap is at least

$$\frac{n-1}{n/2} = 2 - \frac{2}{n}.$$

So, if we want to find a better approximation for vertex cover we cannot rely solely on this LP for our lower bound.

9.3.2 Maximum Independent Set

What about MAXIMUM INDEPENDENT SET? In this problem, we have a graph $G = (V, E)$ and the goal is to find the largest $U \subseteq V$ such that no two $u, v \in U$ are adjacent.

A natural LP relaxation is the following

$$\begin{array}{llll} \text{maximize :} & \sum_{v \in V} x_v & & \\ \text{subject to :} & x_u + x_v \leq 1 & & \text{for each edge } (u,v) \in E \\ & x_v \in [0,1] & \text{for each vertex } v \in V & \end{array}$$

The idea is that in an integer solution, $x_v = 1$ means we include v is included in the independent set and $x_v = 0$ means v is excluded from the independent set. The objective function is then the size of the independent set and the constraints ensure at most one vertex is in the independent set for each edge.

This seems quite close to the relaxation we used for MINIMUM WEIGHT VERTEX COVER. However, the integrality gap can be quite bad.

Consider the complete graph $G = (V, E)$ (again). Setting $x_v = \frac{1}{2}$ is a feasible solution with value $n/2$, yet the optimum integer solution is 1 (since any two vertices are adjacent). Thus, the integrality gap is $\frac{1}{n/2} = \frac{2}{n}$ which is very bad!

9.4 Geometry of LPs

Definition 2 *A subset $\mathcal{P} \subseteq \mathbb{R}$ is said to be* convex *if for any two $\mathbf{x}, \mathbf{x}' \in \mathcal{P}$ and any $0 \leq \lambda \leq$, the point $\lambda\mathbf{x} + (1-\lambda)\mathbf{x}' \in \mathcal{P}$.*

That is, for any two points in $\mathcal{P}$ the line segment connecting these two points lies entirely in $\mathcal{P}$ as well.

Consider an LP $\min\{\mathbf{c}^T \cdot \mathbf{x} : \mathbf{A} \cdot \mathbf{x} \geq 0, \mathbf{x} \geq 0\}$.

Lemma 1 *The set of feasible solutions to an LP is convex.*

Proof. If $\mathbf{A} \cdot \mathbf{x} \geq \mathbf{b}$ and $\mathbf{A} \cdot \mathbf{x}' \geq \mathbf{b}$, then for any $0 \leq \lambda \leq 1$ we have

$$\mathbf{A} \cdot (\lambda\mathbf{x} + (1-\lambda)\mathbf{x}') = \lambda\mathbf{A} \cdot \mathbf{x} + (1-\lambda)\mathbf{A} \cdot \mathbf{x}' \geq \lambda \cdot \mathbf{b} + (1-\lambda) \cdot \mathbf{b} = \mathbf{b}.$$

Similarly, $\lambda\mathbf{x} + (1-\lambda)\mathbf{x}' \geq \mathbf{0}$. ■

Definition 3 *Let* $\overline{\mathbf{x}}$ *be a feasible solution. Call* $\overline{\mathbf{x}}$ *an* extreme point *if for any two feasible* $\mathbf{x}, \mathbf{x}'$ *and any* $0 \leq \lambda \leq 1$ *such that* $\overline{\mathbf{x}} = \lambda \mathbf{x} + (1-\lambda)\mathbf{x}'$ *it must be that either* $\overline{\mathbf{x}} = \mathbf{x}$ *or* $\overline{\mathbf{x}} = \mathbf{x}'$*. We say that* $\overline{\mathbf{x}}$ *is a* convex combination *of* $\mathbf{x}, \mathbf{x}'$*.*

That is, $\overline{\mathbf{x}}$ is an extreme point if the only line segments that pass through $\overline{\mathbf{x}}$ and are contained in the set of feasible LP solutions have $\overline{\mathbf{x}}$ as an endpoint.

Theorem 3 *If an LP in standard form has an optimal solution, then there is an optimal solution that is an extreme point of that LP.*

We emphasize that this is not necessarily true for LPs that are not in standard form. For example $\min\{x_1 + x_2 : x_1 + x_2 \geq 1\}$ has no extreme points yet the optimum solution value is clearly 1. However, it is enough to assume that $x_i \geq 0$ is a constraint for every variable x_i to ensure that there is an extreme point optimum.

Theorem 4 *Any extreme point has bit complexity* $\mathrm{poly}(n, \Delta)$*.*

Definition 4 *Let* $\overline{\mathbf{x}}$ *be a feasible LP solution. A constraint is said to be* tight *under* $\overline{\mathbf{x}}$ *if it holds with equality (e.g.* $(\mathbf{A} \cdot \overline{\mathbf{x}})_j = b_j$ *or* $\overline{x}_i = 0$ *for LPs in standard form).*

Definition 5 *Let* $\overline{\mathbf{x}}$ *be a feasible LP solution. Let* $\mathbf{A}[\overline{\mathbf{x}}]$ *be the matrix consisting of rows from* $\mathbf{A}$ *corresponding to constraints that are tight under* $\mathbf{x}$ *as well as rows of* $\mathbf{I}_n$ *corresponding to tight constraints under* $\mathbf{x} \geq \mathbf{0} \equiv \mathbf{I_n} \cdot \mathbf{x} \geq \mathbf{0}$ *(i.e. if* $x_i = 0$ *then include the i'th row of the identity matrix in* $\mathbf{A}[\overline{\mathbf{x}}]$*).*

An example illustrating this definition follows soon. First we present an alternative characterization of extreme points.

Theorem 5 *Let* $\overline{\mathbf{x}}$ *be a feasible LP solution. Then* $\overline{\mathbf{x}}$ *is an extreme point if and only if* $\mathrm{rank}(\mathbf{A}[\overline{\mathbf{x}}]) = n$ *where* n *is the number of variables.*

Example 2:

Consider the following LP:

$$\begin{array}{lrcl} \text{minimize :} & 3x_1 + 10x_2 + 5x_3 & & \\ & & & \\ \text{subject to} & x_1 + 2x_2 & \geq & 8 \\ & 5x_1 + 3x_2 + x_3 & \geq & 15 \\ & 4x_2 + 5x_3 & \geq & 20 \\ & 2x_1 + 8x_2 + 5x_3 & \geq & 36 \\ & & & \\ & x_1, x_2, x_3 & \geq & 0 \end{array}$$

Below are various feasible solution $\mathbf{x}'$ and their corresponding matrix of tight constraints $\mathbf{A}[\mathbf{x}']$.

Solution 1

$$\mathbf{x}' = \begin{pmatrix} 22/39 \\ 145/39 \\ 40/3 \end{pmatrix}, \quad \mathbf{A}[\mathbf{x}'] = \begin{pmatrix} 1 & 2 & 0 \\ 5 & 3 & 1 \\ 0 & 4 & 5 \\ 2 & 8 & 5 \end{pmatrix}$$

Every constraint apart from $\mathbf{x} \geq 0$ is tight, so each of these rows appears in the matrix. In fact, this is the optimal solution of the LP and has value 44. One can verify that $\text{rank}(\mathbf{A}[\mathbf{x}']) = 3$ so this is also an extreme point.

Solution 2

$$\mathbf{x}' = \begin{pmatrix} 8 \\ 0 \\ 4 \end{pmatrix}, \quad \mathbf{A}[\mathbf{x}'] = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 4 & 5 \\ 2 & 8 & 5 \\ 0 & 1 & 0 \end{pmatrix}$$

The first three rows of $\mathbf{A}[\mathbf{x}']$ are from the three tight constraints among the $\mathbf{A} \cdot \mathbf{x} \geq \mathbf{b}$ constraints (namely, the 1st, 3rd, and 4th constraint). Also, $x_2 \geq 0$ is a tight constraint under $\mathbf{x}'$ so we include row 2 of $\mathbf{I}_n$ in $\mathbf{A}[\mathbf{x}']$.

Solution 3

$$\mathbf{x}' = \begin{pmatrix} 0 \\ 6 \\ 0 \end{pmatrix}, \quad \mathbf{A}[\mathbf{x}'] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

None of the $\mathbf{A} \cdot \mathbf{x} \geq \mathbf{b}$ constraints are tight, only the two nonnegativity constraints $x_1 \geq 0, x_3 \geq 0$. Obviously $\textbf{rank}(\mathbf{A}[\mathbf{x}']) = 2$. By Theorem 5, we should be able to express $\mathbf{x}'$ as the convex combination of two other feasible solutions. We can express $\mathbf{x}'$ as half of $(0,5,0)^T$ and half of $(0,7,0)^T$, both of which are feasible (in fact, $(0,5,0)^T$ is an extreme point).

Solution 4

$$\mathbf{x}' = \begin{pmatrix} 167/39 \\ 145/78 \\ 98/39 \end{pmatrix}, \quad \mathbf{A}[\mathbf{x}'] = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 4 & 5 \\ 2 & 8 & 5 \end{pmatrix}$$

Now, $\mathbf{A}[\mathbf{x}']$ has 3 rows, but $\textbf{rank}(\mathbf{A}[\mathbf{x}']) = 2$. Letting $\mathbf{A}' := \mathbf{A}[\mathbf{x}']$ (for notational simplicity), we see that $\textbf{rank}(\mathbf{A}[\mathbf{x}']) < 3$ because $2 \cdot \mathbf{A}'_1 + \mathbf{A}'_2 = \mathbf{A}'_3$. By Theorem 5, this means we should be able to expression $\mathbf{x}'$ as a convex combination of two other feasible solutions. In fact, we can see that this $\mathbf{x}'$ is exactly half of the $\mathbf{x}$ values in **Solution 1** and half of the $\mathbf{x}$ values in **Solution 2**.

The following corollary is useful when the number of constraints is small with respect to the number of variables (and also in more general settings we will see later).

Corollary 1 *In any extreme point $\overline{\mathbf{x}}$, the* support $\{i : \overline{x}_i > 0\}$ *has size at most m.*

Proof. Since $\text{rank}(\mathbf{A}[\overline{\mathbf{x}}]) = n$ then there are at least n tight constraints. At least $n - m$ of these are of the form $x_i = 0$ for some $1 \leq i \leq n$. ■

References

Y97 Y. Ye, *Interior Point Algorithms. Theory and Analysis.* John Wiley & Sons, 1997.

A Some Proofs

These can be skipped. They are only for the curious.

Proof of Theorem 3.
Let $\mathbf{x}^*$ be an optimal solution that has the most tight constraints among all optimal solutions. Let $\mathbf{A}^* := \mathbf{A}[\mathbf{x}^*]$ be the matrix of these tight constraints (c.f. Definition 5) and let $\mathbf{b}^*$ denote the vector of b_i (or 0) values corresponding to these tight constraints. Thus, $\mathbf{A}^* \cdot \mathbf{x}^* = \mathbf{b}^*$.

We prove $\text{rank}(\mathbf{A}^*) = n$ by way of contradiction. Intuitively, this follows because if $\mathbf{A}^*$ does not have full rank then we can slide $\mathbf{x}^*$ along a line until it "hits" another constraint (i.e. a new constraint becomes tight) while ensuring the cost $\mathbf{c}^T \cdot \mathbf{x}^*$ does not increase, contradicting our choice of $\mathbf{x}^*$.

Suppose $\text{rank}(\mathbf{A}^*) < n$. Then there is some nonzero vector $\mathbf{z}$ such that $\mathbf{A}^* \cdot \mathbf{z} = \mathbf{0}$. Let i' be such that $z_{i'} \neq 0$ and, by negating $\mathbf{z}$ if necessary, assume $z_{i'} > 0$.

Note that for every $\xi > 0$ that $\mathbf{A}^* \cdot (\mathbf{x}^* \pm \xi \cdot \mathbf{z}) = \mathbf{b}^*$. Furthermore, for every constraint that is not tight under $\mathbf{x}^*$ we can choose ξ to be small enough, yet still positive, such that both $\mathbf{x}^* \pm \xi \cdot \mathbf{z}$ still satisfy this constraint (because there is some slack in the constraint under $\mathbf{x}^*$). So, there is in fact some $\xi > 0$ such that both $\mathbf{x}^* \pm \xi \cdot \mathbf{z}$ are feasible.

Choose ξ to be the *largest* value such that both $\mathbf{x}^* \pm \xi \cdot \mathbf{z}$ are feasible. This is well-defined because $z_{i'} > 0$ and the constraint $x_{i'} \geq 0$ is a constraint in the LP (by assumption) so $x^*_{i'} - \xi \cdot z_{i'} \geq 0$ means ξ cannot exceed $x^*_{i'}/z_{i'}$. By choosing the largest possible ξ, we have that one of $\mathbf{x}^* \pm \xi \cdot \mathbf{z}$ has more tight constraints than $\mathbf{x}^*$.

Next, if $\mathbf{c}^T \cdot \mathbf{z} \neq 0$ then one of $\mathbf{c}^T \cdot (\mathbf{x}^* \pm \xi \cdot \mathbf{z})$ is strictly less than $\mathbf{c}^T \cdot \mathbf{x}^*$ which contradicts optimality of $\mathbf{x}^*$. Thus, since $\mathbf{c}^T \cdot \mathbf{z} = 0$ then both $\mathbf{c}^T \cdot (\mathbf{x}^* \pm \xi \cdot \mathbf{z}) = \mathbf{c}^T \cdot \mathbf{x}^*$ meaning both are optimal solutions.

But then one of $\mathbf{x}^* \pm \xi \cdot \mathbf{z}$ is an optimal solution with more tight constraints than $\mathbf{x}^*$, a contradiction. Therefore, $\text{rank}(\mathbf{A}^*) = n$. ■

Proof of Theorem 5.
The proof of Theorem 3 shows that if $\text{rank}(\mathbf{A}^*[\overline{\mathbf{x}}]) < n$ then $\overline{\mathbf{x}}$ is not an extreme point (in the notation of the proof, $\mathbf{x}^*$ takes the place of $\overline{\mathbf{x}}$ and $\mathbf{x}^* \pm \xi\mathbf{z}$ are the endpoints of a line segment contained in the set of feasible solutions).

Now suppose $\overline{\mathbf{x}}$ is not an extreme point and that $\overline{\mathbf{x}} = \lambda\mathbf{x} + (1-\lambda)\mathbf{x}'$ where $\mathbf{x}, \mathbf{x}'$ are both feasible with $\overline{\mathbf{x}} \neq \mathbf{x}$ and $\overline{\mathbf{x}} \neq \mathbf{x}'$. Since $\mathbf{A}[\overline{\mathbf{x}}] \cdot \overline{\mathbf{x}} = \mathbf{b}^*$ (where $\mathbf{b}^*$ comes from the right hand side of the tight constraints) and since $\mathbf{x}, \mathbf{x}'$ are both feasible, then it must be that $\mathbf{A}[\overline{\mathbf{x}}] \cdot \mathbf{x} = \mathbf{A}[\overline{\mathbf{x}}] \cdot \mathbf{x}' = \mathbf{b}^*$ as well (i.e. if one of the constraints that was tight under $\overline{\mathbf{x}}$ was slack under, say, $\mathbf{x}$ then it would be violated under $\mathbf{x}'$).

Thus, $\mathbf{A}[\overline{\mathbf{x}}] \cdot (\overline{\mathbf{x}} - \mathbf{x}) = \mathbf{0}$. But $\overline{\mathbf{x}} - \mathbf{x} \neq \mathbf{0}$ so $\text{rank}(\mathbf{A}[\overline{\mathbf{x}}]) < n$. ■

Proof of Theorem 4.
Let $\overline{\mathbf{x}}$ be an extreme point, let $\mathbf{A}^* := \mathbf{A}[\overline{\mathbf{x}}]$ and let $\mathbf{b}^*$ denote the vector of b_i (or 0) values corresponding to the tight constraints of $\overline{\mathbf{x}}$. Now, $\mathbf{rank}(\mathbf{A}^*) = n$ by Theorem 5. By dropping some rows of $\mathbf{A}^*$ and $\mathbf{b}^*$ if necessary, assume $\mathbf{A}^*$ has exactly n rows. That is, we are assuming that $\mathbf{A}^*$ is an $n \times n$ matrix with full rank such that $\mathbf{A}^* \cdot \overline{\mathbf{x}} = \mathbf{b}^*$.

Let D be the product of all denominators of entries in $\mathbf{A}^*$ and $\mathbf{b}^*$. Replace $\mathbf{A}^*$ and $\mathbf{b}^*$ by $D\mathbf{A}^*$ and $D\mathbf{b}^*$ so that the resulting matrices contain only integers. Since D is the product of $n^2 + n$ integers, each having bit complexity at most Δ, then D has bit complexity at most $O(n^2 \cdot \Delta)$. So, the bit complexity of the new matrix and vector has only increased by a $\mathrm{poly}(n, \Delta)$-factor. We are now assuming $\mathbf{A}^*$ and $\mathbf{b}^*$ consist only of integer entries z satisfying $|z| \le 2^{\Delta'}$ for some $\Delta' = \mathrm{poly}(n, \Delta)$.

Let $\mathbf{A}^*(i : \mathbf{b}^*)$ denote the $n \times n$ matrix obtained by replacing column i in $\mathbf{A}^*$ by $\mathbf{b}^*$. Cramer's rule states

$$\overline{x}_i = \frac{\det(\mathbf{A}^*(i : \mathbf{b}^*))}{\det(\mathbf{A}^*)}.$$

Since the determinant of an integer matrix is an integer (this immediately follows from the Leibniz formula for determinants), we are left with proving that the determinant of an $n \times n$ integer matrix with entries having magnitude most $2^{\Delta'}$ is at most $2^{\mathrm{poly}(n, \Delta')}$ (the logarithm of this is the bit complexity).

Let M be such a matrix. By Hadamard's determinant bound (to justify the first inequality below),

$$\det(M) \le \prod_{i=1}^{n} ||M_i||_2 = \prod_{i=1}^{n} \sqrt{\sum_{i=1}^{n} M_{ij}^2} \le \prod_{i=1}^{n} (n 2^{2\Delta'}) = (n 2^{2\Delta'})^n.$$

This last quantity can be crudely bounded by $2^{n^2 2\Delta'}$. Thus, each numerator and denominator of each $\overline{x}_i$ value can be described using $O(n^2 \Delta') = \mathrm{poly}(n, \Delta)$ bits which is exactly what we wanted to show. ■

Theorem 2
We will not prove Theorem 2, but we simply note that the polynomial-time algorithms described in, say, [Y97] will return an extreme point optimal solution if an optimal solution exists (for LP in standard form). By Theorem 4, the bit complexity of this solution is at most $\mathrm{poly}(n, \Delta)$. The reduction from LPs in general form to LPs in standard form shows the bit complexity statement is true in general once we know it for LPs in standard form.

Proof of Theorem 1 (sketched).
It suffices to prove the statement for LPs in standard form. Suppose the LP is not infeasible and is not unbounded. That is, there is a feasible LP solution but that there is also a lower bound ℓ such that every feasible solution $\overline{\mathbf{x}}$ has $\mathbf{c}^T \cdot \overline{\mathbf{x}} \ge \ell$.

While Theorem 3 assumed that the LP had an optimal solution, it is easy to see that the proof technique shows something stronger: for every feasible solution $\overline{\mathbf{x}}$ there is some extreme point solution $\overline{\mathbf{x}}'$ such that $\mathbf{c}^T \cdot \overline{\mathbf{x}}' \le \mathbf{c}^T \cdot \overline{\mathbf{x}}$ At a high level, if $\overline{\mathbf{x}}$ is not extreme then we can slide it along some direction that does not increase the objective function value. If, in fact, the objective function value strictly decreases then another constraint must eventually become tight as we slide because the LP is not unbounded. Otherwise, one of the two directions we can slide along strictly decreases some x_i value so we are, again, guaranteed that another constraint will become tight as we slide. In either case, if we still do not have an extreme point after sliding until another constraint becomes tight, simply repeat this procedure.

Let $\mathbf{x}^*$ be an extreme point with the least objective function value among all extreme points. This is well defined because a) the feasibility of the LP, plus the above argument, shows that there is in fact some extreme point and b) an extreme point is characterized by a collection of n linearly independent tight constraints so there can only be finitely many extreme points (at most $\binom{m+n}{n}$).

The claim is that $\mathbf{x}^*$ is in fact an optimal solution to the LP. Otherwise, we could employ the above sketched argument to an even cheaper LP solution $\mathbf{x}'$ to find an even cheaper extreme point solution, contradicting the fact that $\mathbf{x}^*$ is the cheapest extreme point.

■