Optimization and Approximation	ENS Lyon
Lecture 2	
Lecturer: Alantha Newman	*September 20, 2016*

1 Geometry of Linear Programming

Given an $m \times n$ matrix $\mathbf{A}$, a vector $\mathbf{b} \in \mathbb{R}^m$ and a vector $\mathbf{c} \in \mathbb{R}^n$, recall that the *linear programming* problem is to find an optimal solution $\mathbf{x} \in \mathbb{R}^n$ for the following problem:

$$\begin{array}{rrcll} & \min \quad \sum_{i=1}^{n} c_i x_i & & & \\ \text{subject to:} & \sum_{i=1}^{n} a_{ji} x_i & \geq & b_j, & j : 1 \leq j \leq m, \\ & x_i & \geq & 0, & i : 1 \leq i \leq n. \end{array}$$

We remarked that we can assume a linear program has this format: it is a minimization problem, it uses only inequalities in the form $\geq$, and all variables are required to be nonnegative. We refer to this as *canonical form*. Finding a vector $\mathbf{x}$ that minimizes $\mathbf{c}'\mathbf{x}$ is equivalent to maximizing $-\mathbf{c}'\mathbf{x}$. Moreover, both $\leq$ and $=$ constraints can be expressed using $\geq$ constraints:

a. $\sum a_{ji} x_i \ \leq \ b_j \quad \rightarrow \quad \sum -a_{ji} x_i \ \geq -b_j$,

b. $\sum a_{ji} x_i \ = \ b_j \quad \rightarrow \quad \sum -a_{ji} x_i \ \geq -b_i$ and $\sum a_{ji} x_i \ \geq b_j$.

Finally, while an x_i variable may not necessarily be nonnegative, we can replace x_i by $x_i' - x_i''$ and require that $x_i', x_i'' \geq 0$. Thus, any linear program can be reduced to the linear programming problem where the input is given in the canonical form described above. (We will also say that a maximization problem with all inequalities in the form $\leq$ and all variables nonnegative is in canonical form.)

1.1 Polytopes

A polytope can provide a representation or a (sometimes compact) description of the solutions to a discrete optimization problem. Consider the perfect matching problem: given a graph $G = (V, E)$, find a subset of the edges $M \subset E$ such that each vertex in V is incident to exactly one edge in M. Let a matching M be represented by a vector or a point in $\{0, 1\}^E$, where there is an entry in the vector for each edge $e \in E$, and the value for that entry is a 1 if edge $e \in M$ and 0 otherwise. If we take the convex hull of all such points corresponding to the set of perfect matchings, then we have the *matching polytope*. Later on, we will see how this polytope can be used to efficiently find perfect matchings. Some basic definitions:

Definition 1. *A set $S \in \mathbb{R}^n$ is* convex *if for all $\mathbf{x}, \mathbf{y} \in S$, and any $\lambda \in [0, 1]$, $\lambda \mathbf{x} + (1 - \lambda)\mathbf{y} \in S$.*

Definition 2. *A* polytope *is the convex hull of a finite number of points in $\mathbb{R}^n$.*

A polytope is a convex set.

1.2 Polyhedra

A polytope is a polyhedron, and this connection yields an alternate definition and description of a polytope. The following geometric concepts are necessary to define and understand polyhedra.

Definition 3. *Let $\mathbf{a}$ be a nonzero vector in $\mathbb{R}^n$ and let b be a scalar.*

(a) The set $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{a}'\mathbf{x} = b\}$ *is called a* hyperplane.

(b) The set $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{a}'\mathbf{x} \geq b\}$ *is called a* halfspace.

Definition 4. *A* polyhedron *is equal to the intersection of a finite number of halfspaces.*

Like a polytope, a polyhedron is also a convex set. We say a polyhedron is *bounded* if it does not contain a line or a half-line. A bounded polyhedron is a polytope.

1.3 Characterizations of Linear Programs

We will refer to the following linear program as (Q) and to its corresponding polyhedron as P. (To be precise: P is the intersection of the halfspaces given by the constraints $\mathbf{A}\mathbf{x} \geq b$ and $\mathbf{x} \geq 0$.)

$$\begin{aligned} &\min \mathbf{c}'\mathbf{x} \\ \mathbf{A}\mathbf{x} &\geq \mathbf{b} \\ \mathbf{x} &\geq 0. \end{aligned} \tag{Q}$$

An n-tuple $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ is a *feasible* solution of (Q) if it satisfies the constraints of (Q). In this case, $\mathbf{x}$ belongs to the polyhedron P. If $\mathbf{x}^*$ minimizes the objective function, then $\mathbf{x}^*$ is an *optimal* solution. We call the set of feasible solutions the *domain* of (Q). This set is equal to the set of all vectors or points in $\mathbb{R}^n$ that belong to P. There are three possible outcomes for the linear program (Q):

(i) The domain of (Q) is empty. Then (Q) has no solution and the linear program is *infeasible*. (The polyhedron P is empty.)

(ii) The domain of (Q) is unbounded; either there exists an optimal solution or the objective function is $-\infty$. (The polyhedron P contains a line or a half-line (ray).)

(iii) The domain of (Q) is bounded and non-empty. Then there exists an optimal solution. (The polyhedron P is non-empty but does not contain a line or half-line.)

Let's look at some examples. For (i), it is relatively easy to devise a set of constraints whose corresponding polyhedron is empty: consider the halfspaces $\{x \geq 2, -x \geq 2\}$. In case (ii), consider the generalized diet problem from Lecture 1. In this problem, we want to find a "good" diet that minimizes the number of calories. Suppose we wanted to find a good diet that maximizes the number of calories. This objective function as well as the corresponding polyhedron is unbounded. A specific example for case (iii) is shown in Figure 1.

A Question to Ponder: Let us consider two problems:

(P_1) Find a solution for (Q).

(P_2) Determine whether or not P is empty.

Are problems (P_1) and (P_2) equivalent? (Is there a polynomial-time reduction from (P_1) to (P_2)?)

1.4 Vertices of a Polyhedron

Consider the polyhedron shown in Figure 1. Notice that the optimal solution occurs at a "corner" of the polyhedron. There are several different ways of formally defining these special points.

Definition 5. *Let P be a polyhedron. A vector $\mathbf{x} \in P$ is a* vertex *of P if there exists some $\mathbf{c}$ such that $\mathbf{c}'\mathbf{x} < \mathbf{c}'\mathbf{y}$ for all $\mathbf{y} \in P$, $\mathbf{y} \neq \mathbf{x}$.*

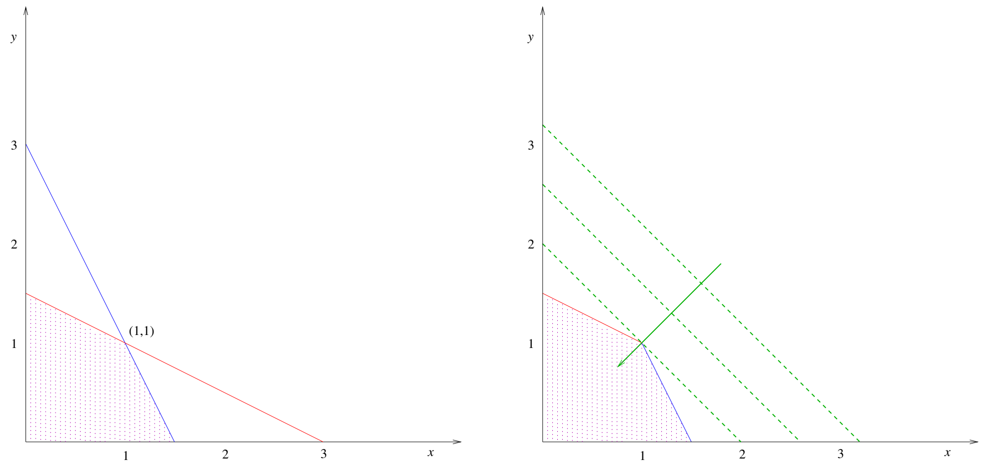

Figure 1: The polyhedron shown in these figures is the intersection of the halfspaces: $\{-x - 2y \geq -3, -2x - y \geq -3, x \geq 0, y \geq 0\}$. The objective function is $\min -x - y$. Consider the hyperplane $-x - y = z$ and slide it (i.e. increase the value of z) towards the polyhedron. The first point where it touches the polyhedron is $(1, 1)$, where $-x - y = -2$.

Another way to define a vertex: $\mathbf{x}$ is a vertex of P iff P is on one side of the hyperplane $\{\mathbf{y} \in \mathbb{R}^n | \mathbf{c}'\mathbf{y} = \mathbf{c}'\mathbf{x}\}$, which meets P only at the point $\mathbf{x}$.

Definition 6. *Let P be a polyhedron. A vector $\mathbf{x} \in P$ is an* extreme point *of P if we cannot find two vectors $\mathbf{y}, \mathbf{z} \in P$ ($\mathbf{y} \neq \mathbf{x}$, $\mathbf{z} \neq \mathbf{x}$) and a scalar $\lambda \in [0, 1]$ such that $\mathbf{x} = \lambda \mathbf{y} + (1 - \lambda)\mathbf{z}$.*

We will see later on that vertices and extreme points are the same.

1.5 Basic Feasible Solutions

The definitions of both vertices and extreme points are geometric; they are independent of the description of the polyhedron. Now, we will see an algebraic definition. For convenience, we denote the j^{th} row of the matrix $\mathbf{A}$ as $\mathbf{a}_j$. Suppose that the matrix $\mathbf{A}$ contains both equality and inequality constraints, i.e. it is not necessarily in canonical form.

Definition 7. *If a vector $\mathbf{x}^*$ satisfies $\mathbf{a}_j'\mathbf{x}^* = b_j$, then the corresponding constraint is* active *or* tight *at $\mathbf{x}^*$.*

Definition 8. *Let P be a polyhedron defined by inequality or equality constraints, and let $\mathbf{x}^* \in \mathbb{R}^n$.*

(a) The vector $\mathbf{x}^*$ is a basic solution *if:*

  (i) All equality constraints are active;

  (ii) Among the constraints that are active at $\mathbf{x}^*$, n of them are linearly independent.

(b) If $\mathbf{x}^* \in P$ and $\mathbf{x}^*$ is a basic solution, then we say that $\mathbf{x}^*$ is a basic feasible solution.

If more than n constraints are active at a point $\mathbf{x}^\star$, then we say that $\mathbf{x}^\star$ is *degenerate*. For now, let us assume that all basic feasible solutions in P are *non-degenerate*. (We will address issues that arise from degenerate solutions in the next lecture.) The following theorem will be proved in an exercise session.

Theorem 9. *Let P be a non-empty polyhedron and let $\mathbf{x}^* \in P$. Then the following are equivalent:*

(a) $\mathbf{x}^*$ *is a vertex;*

(b) $\mathbf{x}^*$ *is an extreme point;*

(c) $\mathbf{x}^*$ *is a basic feasible solution.*

A very useful fact is that if a linear program has an optimal solution with bounded value, there is an extreme point with this value. This allows us to restrict our search for an optimal solution to the (finite) set of extreme points. The proof of this theorem uses the following claim that will also be proved in an exercise session.

Lemma 10. *Suppose that the non-empty polyhedron P does not contain a line. Then P has at least one extreme point.*

Theorem 11. *Consider the linear programming problem of minimizing $\mathbf{c}'\mathbf{x}$ over a non-empty, polyhedron P that does not contain a line. Suppose there exists an optimal solution with bounded value. Then, there exists an optimal solution which is an extreme point of P.*

Proof. Let $\tau = \min_{\mathbf{x}\in P} \mathbf{c}'\mathbf{x}$, and let P_{opt} denote the set of all optimal solutions, i.e. $\{\mathbf{x} \in P \mid \mathbf{c}'\mathbf{x} = \tau\}$. P_{opt} is a polyhedron,‘ since it can be expressed as the intersection of a finite number of halfspaces. Furthermore, P_{opt} is non-empty by the assumption in the theorem statement. Since $P_{opt} \subseteq P$, and P contains no lines, then P_{opt} contains no lines. By Lemma 10, P_{opt} contains at least one extreme point.

Let $\mathbf{x}^*$ be an extreme point of P_{opt}. We will show that $\mathbf{x}^*$ is also an extreme point of P. Towards a contradiction, suppose that $\mathbf{x}^*$ is not an extreme point of P. Then, there exists $\mathbf{y}, \mathbf{z} \in P$, where $\mathbf{z} \neq \mathbf{x}^*, \mathbf{y} \neq \mathbf{x}^*$, and some $\lambda \in [0, 1]$ such that $\mathbf{x}^* = \lambda\mathbf{y} + (1-\lambda)\mathbf{z}$. It follows that $\lambda\mathbf{c}'\mathbf{y} + (1-\lambda)\mathbf{c}'\mathbf{z} = \mathbf{c}'\mathbf{x}^* = \tau$. Furthermore, since τ was the optimal value, $\mathbf{c}'\mathbf{y} \geq \tau$ and $\mathbf{c}'\mathbf{z} \geq \tau$. This implies that $\mathbf{c}'\mathbf{y} = \mathbf{c}'\mathbf{z} = \tau$, which implies that $\mathbf{y}, \mathbf{z} \in P_{opt}$. But this contradicts the fact that $\mathbf{x}^*$ is an extreme point of P_{opt}. Thus, $\mathbf{x}^*$ is an extreme point of P and is also optimal (since it belongs to P_{opt}). $\square$

2 Simplex Algorithm

We now introduce the simplex algorithm. Consider the following linear program in canonical form.

$$\begin{aligned} \max \quad & 5x_1 + 4x_2 + 3x_3 \\ \text{subject to:} \quad & 2x_1 + 3x_2 + x_3 \leq 5 \\ & 4x_1 + x_2 + 2x_3 \leq 11 \\ & 3x_1 + 4x_2 + 2x_3 \leq 8 \\ & x_1, x_2, x_3 \geq 0. \end{aligned} \tag{Q_1}$$

We can translate this linear program into *standard form* by introducing slack variables, $\{x_4, x_5, x_6\}$, one for each constraint. Let z denote the objective value. We obtain the following initial system, called a *dictionary*. These constraints are equivalent to those given in the original linear program.

$$\begin{aligned} x_4 &= 5 - 2x_1 - 3x_2 - x_3 \\ x_5 &= 11 - 4x_1 - x_2 - 2x_3 \\ x_6 &= 8 - 3x_1 - 4x_2 - 2x_3 \\ z &= 0 + 5x_1 + 4x_2 + 3x_3. \end{aligned} \tag{D_0}$$

Maximizing z under nonnegativity constraints subject to these new equations is therefore equivalent to solving (Q_1). In this form, by setting $x_1, x_2, x_3 = 0$ and $x_4 = 5, x_5 = 11, x_6 = 8$, we obtain a feasible solution.

Our goal is to increase z. What can we do? We can increase either x_1, x_2 or x_3 since each of these variables has a positive coefficient in the last equation (equal to z) in (D_0). Let us try to increase x_1. What are the constraints? (If we increase it too much, the solution may no longer be feasible.)

- $x_4 \geq 0 \;\Rightarrow x_1 \leq \frac{5}{2}$.
- $x_5 \geq 0 \;\Rightarrow x_1 \leq \frac{11}{4}$.
- $x_6 \geq 0 \;\Rightarrow x_1 \leq \frac{8}{3}$.

The first constraint (for x_4) is the most restrictive. So we assign $x_1 = \frac{5}{2}$, $x_2 = 0$, $x_3 = 0$ and we re-solve for x_4, x_5, x_6. Now we have a new (equivalent) dictionary in which we exchange the roles of x_1 and x_4. We say that x_1 is the *entering* variable and x_4 is the *leaving* variable. The choice of which new variable to include and which old variable to exclude is called a *pivot rule*. Now we obtain an updated dictionary.

$$
\begin{aligned}
x_1 &= \frac{5}{2} - \frac{x_4}{2} - \frac{3x_2}{2} - \frac{x_3}{2} \\
x_5 &= 1 - 2x_4 + 5x_2 \\
x_6 &= \frac{1}{2} + \frac{3x_4}{2} + \frac{x_2}{2} - \frac{x_3}{2} \\
z &= \frac{25}{2} - \frac{5x_4}{2} - \frac{7x_2}{2} + \frac{x_3}{2}.
\end{aligned} \tag{D_1}
$$

The nonnegative solutions for $\{x_1, x_2, \ldots, x_6\}$ in (D_1) are the same as those for dictionary (D_0). Thus, (D_1) is equivalent to (D_0) and therefore to (Q_1). The solution associated with (D_1) is obtained by letting so-called non-basic variables x_2, x_3, x_4 be zero, and solving for the values of the basic variables. (Here, all basic variables are non-zero, because we are assuming that each basic feasible solution is non-degenerate.) Now the only variable that can be used to increase z in (D_1) is x_3.

- $x_1 \geq 0 \;\Rightarrow x_3 \leq 5$.
- $x_5 \geq 0 \;\Rightarrow x_3 \leq \infty$.
- $x_6 \geq 0 \;\Rightarrow x_3 \leq 1$.

	x_1	x_2	x_3	x_4	x_5	x_6	z
(D_0)	0	0	0	5	11	8	0
(D_1)	$\frac{5}{2}$	0	0	0	1	$\frac{1}{2}$	$\frac{25}{2}$
(D_2)	2	0	1	0	1	0	13

Table 1: The solutions corresponding to each dictionary when the simplex algorithm is run on (Q_1).

After adding x_3 to the set of basic variables, and removing x_6, we obtain the following dictionary:

$$
\begin{aligned}
x_1 &= 2 - 2x_4 - 2x_2 + x_6 \\
x_5 &= 1 + 2x_4 + 5x_2 \\
x_3 &= 1 + 3x_4 + x_2 - 2x_6 \\
z &= 13 - x_4 - 3x_2 - x_6.
\end{aligned} \tag{D_2}
$$

Since all coefficients in last equation are now negative, the value of z cannot be increased further. The simplex algorithm terminates with the solution $(2,0,1,0,1,0)$. The solution for (Q_1) is $(2,0,1)$ with an optimal solution value of 13.

2.1 Visualizing Pivots

Let us consider the following linear program:

$$\begin{aligned} \max \quad & x+y \\ \text{subject to:} \quad & x-y \leq\ 1 \\ & -x+2y \leq\ 2 \\ & x,y \geq\ 0. \end{aligned} \tag{Q_2}$$

If we run the simplex algorithm, we can obtain the following three dictionaries associated with the solutions shown in Table 2.

$$\begin{aligned} x_3 &= 1-x_1+x_2 \\ x_4 &= 2+x_1-2x_2 \\ z &= 0+x_1+x_2. \end{aligned} \tag{D'_0}$$

$$\begin{aligned} x_1 &= 1-x_3+x_2 \\ x_4 &= 3-x_3-x_2 \\ z &= 1-x_3+2x_2. \end{aligned} \tag{D'_1}$$

$$\begin{aligned} x_1 &= 4-2x_3-x_4 \\ x_2 &= 3-x_3-x_4 \\ z &= 7-3x_3-2x_4. \end{aligned} \tag{D'_2}$$

	x_1	x_2	x_3	x_4	z
(D'_0)	0	0	1	2	0
(D'_1)	1	0	0	3	1
(D'_2)	4	3	0	0	7

Table 2: The solutions corresponding to each dictionary when the simplex algorithm is run on (Q_2).

Figure 2 depicts how the successive solutions move to adjacent vertices of the polyhedron corresponding to (Q_2). Since the simplex algorithm moves to a new basic feasible solution at each step, the number of steps is upper bounded by the number of basic feasible solutions or vertices in the associated polyhedron. A simple calculation shows that if a polyhedron is described by n variables and m constraints in canonical form, then after transformation to standard form, there are $2m+n$ constraints. Choosing n of these defines a basic feasible solution. The polyhedron therefore has at most $\binom{2m+n}{n}$ vertices.

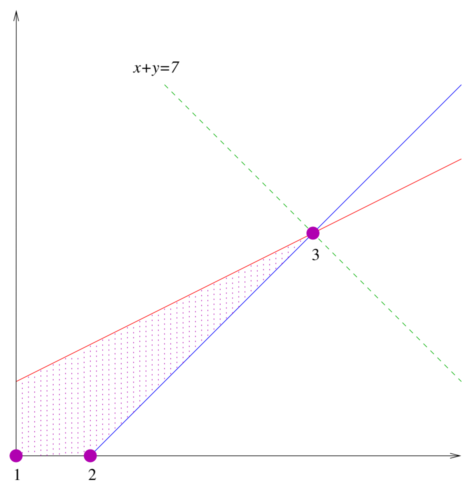

Figure 2: Solutions 1, 2 and 3 are associated with dictionaries (D'_0), (D'_1) and (D'_2). Each solution moves from one basic feasible solution to an adjacent basic feasible solution.

References

[BT97] Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to linear optimization.* Athena Scientific Belmont, MA, 1997.

[Van01] Robert J. Vanderbei. *Linear programming: Foundations and Extensions.* Kluwer Academic Publishers, Boston, MA, 2001.

These lecture notes are based on the following sources: lecture notes by Stéphan Thomassé from previous versions of the same course, Chapter 2 of [BT97] and Chapter 2 of [Van01].