CMPUT 675: Topics in Algorithms and Combinatorial Optimization (Fall 2009)

Lecture 6,7: Linear Programming, Simplex method

Lecturer: Mohammad R. Salavatipour **Scriber:** Yining Wang

Date: Sept 22 and 24, 2009

This lecture starts with some basic notions. Then we study the linear programming and polytopes.

1 Linear Programming

A linear programming problem can be formalized as follows: given a set of variables $x_1, ..., x_n$, the objective is to optimze a linear function of these variables subject to a set of linear contraints. For example,

$$\begin{array}{rlll} \text{min} & \sum_{j=1}^n c_j x_j & & \\ \text{subject to} & & & \\ & \sum_{j=1}^n a_{ij} x_j \geq b_i & \text{for} & i = 1, \cdots, m \\ & x_j \geq 0 & \text{for} & j = 1, \cdots, n. \end{array}$$

or in other form:

$$\begin{array}{ll} \text{min} & c^T x \\ \text{subject to} & \\ & Ax \geq b \\ & x \geq 0. \end{array}$$

A *feasible* solution is any **x** satisfying $Ax \geq b$. A linear program (LP) is said to be a *feasible LP* if it has at least one feasible solution.

Definition 1.1 *A linear program is unbounded (from below) if for any $\alpha \in \mathbb{R}$, there exists a feasible solution **x** such that $c^T x < \alpha$.*

An LP can be stated in many different (equivalnet) forms. Suppose the objective function is a maximization. Then:

1. One can transform a maximization problem into a minimiation problem and vice versa by changing the sign of variables in the objective fucntion:

$$\text{max } c^T x \leftrightarrow \text{min } -c^T x$$

2. If the linear constraints are equalities then it can be transformed into two constraints with inequality:

$$a_i^T x = b \rightarrow a_i^T x \geq b \text{ and } a_i^T x \leq b$$

3. By introducing slack variables, one can express inequalities as equalities. For example, We can replace the constraint $a_i^T x \leq b_i$ with $a_i^T x + s_i = b_i$ for a new variable s_i and add the constraint $s_i \geq 0$. It is easy that any feasible solution satisfying the first one corresponds to a unique solution satisfying the second one and vice-versa,

4. We can switch between non-positivity and non-negativity constraints:

$$x_j \leq 0 \rightarrow -x_j \geq 0n$$

5. Restricting x's sign: if there is no sign restriction on x we can write $x_j = x_j^+ - x_j^-$ and require that $x^+j, x_j^- \geq 0$ and replace all x_j's.

Using these rules, we can transform any LP in the canonical form:

$$\begin{array}{ll} \text{min} & c^T x \\ \text{subject to} & \\ & Ax \geq b \\ & x \geq 0. \end{array}$$

to an LP in the standard form:

$$\begin{array}{ll} \text{min} & c^T x \\ \text{subject to} & \\ & Ax = b \\ & x \geq 0. \end{array}$$

Example 1: Consider the following LP:

$$\begin{array}{rrcl} \max & 2x_1 + x_3 & & \\ s.t. & x_1 + x_2 + x_3 & \leq & 4 \\ & x_1 & \leq & 2 \\ & x_3 & \leq & 3 \\ & 3x_2 + x_3 & \leq & 3 \\ & x_1, x_2, x_3 & \geq & 0 \end{array} \tag{1}$$

A feasible solution for this LP is $(2, 0, 2)$.

Note that if there are constraints in an LP that are linearly dependent on other ones then they are redundant as they can be obtained from the others. So we can make the following assumption:

Assumption 1: The rank of the $m \times n$ matrix A in the linear program is m (full rank), i.e. we have m linearly independent columns of A ($m \leq n$).

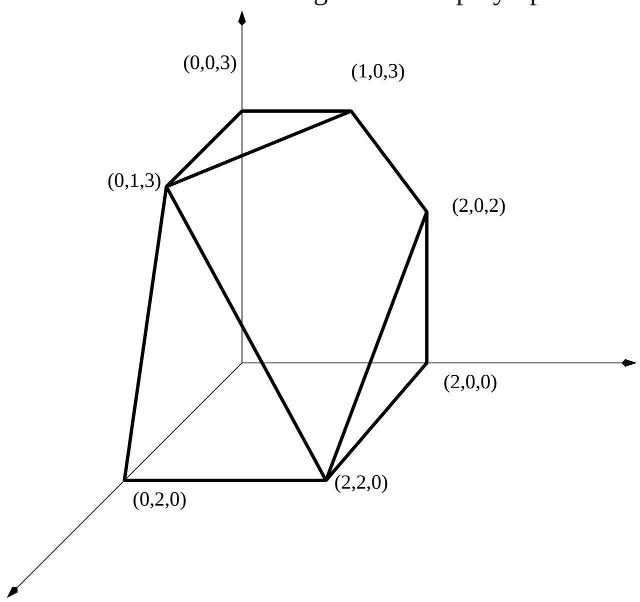
Assumption2: any linear program we deal with is feasible.

Definition 1.2 *A basis of A is a linearly independent collection of m columns of A, i.e., a non-singular submatrix $B = [A_{j_1}, A_{j_2}, ..., A_{j_m}]$, where B is an $m \times m$ matrix. A basic solution corresponding to B is a vector $x \in \mathbb{R}^n$ with the following properties:*

- *$x_j = 0$ if $A_j \notin B$;*
- *x_{j_k} = the kth component of $B^{-1}b$ for $k = 1, \cdots, m$.*

To find a basic solution, can be found by:

Figure 1: The polytope defined by the constraints of LP(1)



1. Find a set B of m linearly independent columns of A;
2. All x's corresponding to columns not in B are set to zero;
3. Solve the remaining m equations to find remaining m varialbes.

Basic feasible solutions (bfs) are very important in study of LPs. It can be proved that if a linear program is feasible then there is a basic feasible solution. From now on we assume we are talking about LP's whose solution set is non-empty (i.e. there is at least one feasible solution and therefore at least one bfs). It can be proved that for every bfs, there is an objective function such that the bfs is the unique optimum solution to the LP with that objective function:

Lemma 1.3 *Let x be basic feasible solution to $Ax = b$, for $x \geq 0$. Then there is a vector c such that x is the unique optimal solution to*

$$\begin{array}{rl} \min & c^T x \\ \textit{s.t.} & Ax = b \\ & x \geq 0. \end{array}$$

There is a strong relation between LP's and polytopes (defined formally below). The study of LP's is equivalent to study of the corresponding polytopes (or polyhedrons). Consider the vector space in $\mathbb{R}^n$. A linear subspace of $\mathbb{R}^n$ is a subset of it closed under vector addition and scalar multiplication.

Definition 1.4 *A hyperplane in $\mathbb{R}^n$ is a set of points $\{x \in \mathbb{R}^n \colon a_1x_1 + \cdots + a_nx_n = b\}$*

Every hyperplane defines two half spaces: $Ax \geq b$ and $Ax \leq b$. A body defined as the interesection of a collection of half-spaces is a polyhedron:

Definition 1.5 *A polyhedron is a convex body defined by a collection of half spaces. A polytope is a bounded polyhedron.*

Definition 1.6 *A set of points $\{a_1, \ldots, a_k \in \mathbb{R}^n\}$ are linearly independent if $\sum_{i=1}^{k} \lambda_i a_i = 0$ implies $\lambda_i = 0$ for all $1 \leq i \leq k$.*

Definition 1.7 *A linear subspace of* $\mathbb{R}^n$ *is a set* $S = \{x \in \mathbb{R}^n \colon a_i x_i + \cdots + a_n x_n = 0, 1 \le i \le n\}$. *An affine subspace is obtained from a linear subspace* S *by translating by a vector* b: $A = \{x + b \colon x \in S\}$.

Definition 1.8 *A set of points* $\{a_1, \ldots, a_k \in \mathbb{R}^n\}$ *are affinely independent if* $\sum_{i=1}^{k} \lambda_i a_i = 0$ *and* $\sum_{i=1}^{k} \lambda_i = 0$ *imply* $\lambda_i = 0$ *for all* i.

Dimension of a polyhedron P is the maximum number of affinely independent points in P minus 1.

Examples:

- A single point has dimension = 0;
- A line segment has dimension = 1;
- Any set of $k \le n+1$ points in $\mathbb{R}^n$ has dimension of at most $k-1$.
- Dimension of a set F defined by

$$\begin{aligned} Ax &= b \\ x &\ge 0. \end{aligned}$$

  with A being an $m \times n$ matrix is at most $n - m$.

Let P be a convex polytope in $\mathbb{R}^n$ and HS be a half-space defined by a hyperplane H. If $f = P \cap HS$ belongs to H, then f is a face of P.

Definition 1.9 *A facet is a face of dimension* $n-1$. *A vertex is a face of dimension 0 (point). An edge is a face of dimension 1 (line).*

Note that the hyperplane defining a facet corresponds to a defining half-space of P but the converse might not be true.

A vertex can be aslo defined equivalently as:

Definition 1.10 x *is a vertex in* P *if there does not exist* $y \ne 0$ *such that* $x + y, x - y \in P$.

One can bound the number of faces of a polytope using the following theorem:

Theorem 1.11 *Let* $A \in \mathbb{R}^{m \times n}$. *Then any face of* $P = \{x \in \mathbb{R}^n : Ax \le b\}$ *corresponds to the set of solutions to*

$$\sum_j a_{ij} x_j = b_i, i \in I$$

$$\sum_j a_{ij} x_j \le b_i, i \notin I$$

for some $I \subseteq \{1, \cdots, m\}$.

Proof: Exercise. ■

Corollary 1.12 *The number of non-empty faces of* P *is at most* 2^m.

Theorem 1.13 *Let* x^* *be a vertex of* $P = \{x \in \mathbb{R}^n \colon Ax \le b\}$. *Then, there is a set* $I \subseteq \{1, \ldots, m\}$ *such that* x^* *is the unique solution to* $\sum_j a_{ij} x_j = b_j, \quad \forall i \in I$.

Proof: Given a vertex x^* of P, define I as

$$I = \{i \colon \sum j a_{ij} x_j = b_i\}. \tag{2}$$

So for every $i \notin I$ we must have:

$$\sum_j a_{ij} b_j < b_i. \tag{3}$$

From the previous theorem we know that x^* is uniquely defined by:

$$\begin{aligned} \sum_j a_{ij} x_j &= b_i, \quad \forall i \in I \\ \sum_j a_{ij} x_j &\leq b_i, \quad \forall i \notin I \end{aligned}$$

If there is another solution x' which is a solution to Eq. (2), then $(1-\epsilon)x^* + \epsilon x'$ for sufficiently small ϵ, satisfies all those in eq. (2) and eq. (3). It contradicts the assumption that x^* is a vertex solution. ■

2 Linear Programs and Polytopes

Polytopes can be defined in 3 different ways:

1. The convex hull of a finite set of points (which will contain the vertices of the polytope),
2. As the intersection of a finite number of half spaces as long as the intersection is non-empty,
3. Algebraic version:

$$\begin{aligned} Ax &= b \\ x &\geq 0. \end{aligned}$$

The previous theorem shows that the vertices of the polytop correspond to bfs of the corresponding LP. Consider a polytope P defined by $Ax = b, \quad x \geq 0$. Let N be the set of non-basic variables and let B be the set of basic variables. We can partition x and accordingly A into $\{x_B, x_N\}$ and $\{A_B, A_N\}$. Then, a vertex x^* can be obtained by setting non-basic variables to zero and solving the remaining equations. In other words, we have $A_B x_B + A_N x_N = b$; we set $x_N = 0$. It follows that $A_B x_B = b$. We solve this to obtain values of basic variables (this set of equations must have unique solution), which means that A_B must have full rank (i.e. $rank(A_B) = |B|$). Since we assume A itself has full rank (i.e. $rank(A) = m$), we have $|B| \leq m$ and A_B is a $m \times m$ non-singular matrix. Recall that this set B is a basis and x is a bfs. Thus all vertices are bfs and vice versa. Note however that different bases may lead to the same vertex as there are different ways of extending A_B to a $m \times m$ matrix.

For example, consider the LP1 from Example 1:

$$\begin{aligned} \max \quad & 2x_1 + x_3 & & \\ s.t. \quad x_1 + x_2 + x_3 &\leq 4 & \quad (4) \\ x_1 &\leq 2 & \quad (5) \\ x_3 &\leq 3 & \quad (6) \\ 3x_2 + x_3 &\leq 3 & \quad (7) \\ x_1, x_2, x_3 &\geq 0 & \quad (8) \end{aligned}$$

Then, introducing slack variables x_4, x_5, x_6, x_7 for equations (4),(5),(6),and (7),respectively, we obtain the following LP in standard form: $Ax = b$ where:

$$b = \begin{bmatrix} 4 \\ 2 \\ 3 \\ 6 \end{bmatrix}, \qquad A = \left[\begin{array}{ccc|cccc} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right].$$

Then $B = \{A_1, A_2, A_3, A_6\}$ and $B' = \{A_1, A_2, A_4, A_6\}$ are to bases. and both correspond to the bfs $(2, 2, 0, 0, 0, 3, 0)$. For B if we set $x_4 = x_5 = x_7 = 0$ we obtain vertex $(2, 2, 0)$ and for B' we set $x_3 = x_5 = x_7 = 0$ which again implies the same solution corresponding to vertex $(2, 2, 0)$.

In general, whether in canonical or standard form, when equation $Ax = b$ holds we say the constraint is tight. From the discussion above, we have that a bfs is the unique solution to a set of n linearly independent tight constraints.

Theorem 2.1 *If $\min\{c^T x : Ax = b, \quad x \geq 0\}$ is finite, then there is an optimal solution that is a vertex.*

Proof: Consider an optimal solution x and suppose it is not a vertex. Then $\exists y \neq 0$ such that $x+y, x-y \in P$. It follows that $A(x + y) = b$ and $A(x - y) = b$, which implies that $Ay = 0$. Without loss of generality, assume that $c^T y \leq 0$ (if needed, take $-y$). For the case that $c^T y = 0$, since $y \neq 0$ and $c^T y = c^T(-y) = 0$ there must be an index j such that $y_j < 0$. In that case we choose y for which this index j exists. Consider $x + \alpha y$ for some $\alpha > 0$. Then, $c^T(x + \alpha y) = c^T x + c^T(\alpha y) \leq c^T x$ because of $c^T y \leq 0$.

Case 1: Suppose there exists an index j such that $y_j < 0$. Consider $x + \alpha y$. When $\alpha \to \infty$, the jth component of $y \to -\infty$ and so does the j'th component of $x + \alpha y$, but we assumed the minimum the polytop is finite. So let $\alpha = \max_{j:\, y_j<0} \frac{x_j}{-y_j}$ and k denote the value satisfying $\alpha = \frac{x_k}{-y_k}$, which is the largest value of α such that $x + \alpha y \geq 0$. Then $A(x + \alpha y) = Ax + A(\alpha y) = Ax = b$, which implies $x + \alpha y \in P$ and has one more component zero (which is the k'th component i.e. $(x_k + \alpha y_k)$). Also $c^T(x + \alpha y) = c^T x + c^T \alpha y \leq c^T x$.

Case 2: Suppose $y_j \geq 0$ for all j. Then we must have $c^T y < 0$ (or otherwise we have an index j with $y_j < 0$). Consider $x + \alpha y$. Since $A(x + \alpha y) = Ax + \alpha Ay = Ax = b$, and $x + \alpha y \geq x \geq 0$, $x + \alpha y$ is a feasible solution. On the other hand $c^T(x + \alpha y) = c^T x + \alpha c^T y$ which goes to $-\infty$ if α grows (because $c^T y < 0$), implying that P is unbounded which contradicts the assumption. Therefore this case cannot happen.

Case 1 can happen at most n times. By induction, we eventually find a vertex solution. ■

Here is an alternative proof of the above theorem.

Proof: We know that any point $x \in P$ is a convex combination of its vertices (by definition of a polytope). So if x^* is an optimum non-vertex solution and $x_1, \ldots, x_k$ are vertices then there are α_i, $1 \leq i \leq k$ with $\sum_i \alpha_i = 1$ and $x = \sum_{i=1}^k \alpha_i x_i$. Say x_j is the vertex with the smallest $c^T x_i$ value. Then $c^T x = \sum_{i=1}^k \alpha_i c^T x_i \geq c^T x_j \sum_{i=1}^k \alpha_i = c^T x_j$. So x_j is optimum too. ■

3 Solving LP's: Simplex Algorithm

Simplex is the first algorithm developed for solving LP. Although it has exponential running time in worst case it is the most practical algorithm. Is similar to Gaussian elimination for solving systems of equalities The general idea of the algorithm is to start from a bfs and move from one vertex to another one which has a better objective value. This is called pivoting. Repeat this procedure until all your neighbours have worse value (i.e. at a locally optimum vertex). Since the polytop is convex the locally optimum must be globally optimum as well. There are many ways to select the next vertex and for every variation there is an example for which the algorithm would take exponential time.

For ease of exposition we present the algorithm through an example. Consider the following LP:

$$\begin{array}{lrcl}
\max & 3x_1 + x_2 + 2x_3 & & \\
& x_1 + x_2 + 3x_3 & \leq & 30 \\
& 2x_1 + 2x_2 + 5x_3 & \leq & 24 \\
& 4x_1 + x_2 + 2x_3 & \leq & 36 \\
& x_1, x_2, x_3 & \geq & 0
\end{array}$$

We transform the LP into standard form by introducing slack variables.

$$\begin{array}{lrcl}
\max & Z & = & 3x_1 + x_2 + 2x_3 \\
& x_4 & = & 30 - x_1 - x_2 - 3x_3 \\
& x_5 & = & 24 - 2x_1 - 2x_2 - 5x_3 \\
& x_6 & = & 36 - 4x_1 - x_2 - 2x_3 \\
& x_1, \ldots, x_6 & \geq & 0
\end{array}$$

A basic solution we start with is all non-basic variables set to zero $(0, 0, 0, 30, 24, 36)$. This gives an objective value of $z = 0$ Next we select a non-basic var with positive co-efficient in the objective function and increase it as much as we can. Here, say we select x_1. If $x_1 \geq 30$ then $x_4 \leq 0$, if $x_1 \geq 12$ then $x_5 \leq 0$, and if $x_1 \geq 9$ then $x_6 \leq 0$; so the maximum amount we can increase is 9 and it will make the last constraint "tight". We increase x_1 by this amount and re-write the constraints by switching the role of x_1 and x_6:
$x_1 = 9 - \frac{x_2}{4} - \frac{x_3}{2} - \frac{x_6}{4}$
Rewriting all constraints we obtain:

$$\begin{array}{lrcl}
\max & Z & = & 27 + \dfrac{x_2}{4} + \dfrac{x_3}{2} - \dfrac{3x_6}{4} \\
& x_1 & = & 9 - \dfrac{x_2}{4} - \dfrac{x_3}{2} - \dfrac{x_6}{4} \\
& x_4 & = & 21 - \dfrac{3x_2}{4} - \dfrac{5x_3}{2} + \dfrac{x_6}{4} \\
& x_5 & = & 6 - \dfrac{3x_2}{2} - 4x_3 + \dfrac{x_6}{2} \\
& x_1, \ldots, x_6 & \geq & 0
\end{array}$$

This is a pivoting operation. The new basic solution obtained is $(9, 0, 0, 21, 6, 0)$ with $z = 27$. Next, we choose x_3 and increase it; the maximum amount we can do is given by constraint for x_5: it can increase to $3/2$ without violating any of the constraints. We then re-write the constraints again:

$$
\begin{array}{lrcl}
\max & Z & = & \frac{111}{4} + \frac{x_2}{16} - \frac{x_5}{8} - \frac{11x_6}{16} \\
 & x_1 & = & \frac{33}{4} - \frac{x_2}{16} + \frac{x_5}{8} - \frac{5x_6}{16} \\
 & x_3 & = & \frac{3}{2} - \frac{3x_2}{8} - \frac{x_5}{4} + \frac{x_6}{8} \\
 & x_4 & = & \frac{69}{4} + \frac{3x_2}{16} + \frac{5x_5}{8} - \frac{x_6}{16} \\
 & x_1, \ldots, x_6 & \geq & 0
\end{array}
$$

The basic solution becomes $(\frac{33}{4}, 0, \frac{3}{2}, \frac{69}{4}, 0, 0)$ The only way to increase Z is to increase x_2; the tightest constraint is the second one and it gives a bound of $x_2 \leq 4$. We obtain the following:

$$
\begin{array}{lrcl}
\max & Z & = & 28 - \frac{x_3}{6} - \frac{x_5}{6} - \frac{2x_6}{3} \\
 & x_1 & = & 8 + \frac{x_3}{6} + \frac{x_5}{6} - \frac{x_6}{3} \\
 & x_2 & = & 4 - \frac{8x_3}{3} - \frac{2x_5}{3} + \frac{x_6}{3} \\
 & x_4 & = & 18 - \frac{x_3}{2} + \frac{x_5}{2} \\
 & x_1, \ldots, x_6 & \geq & 0
\end{array}
$$

We cannot increase Z anymore, so we are at an optimum solution which is $(8, 4, 0, 18, 0, 0)$ with $z = 28$.

In general form, we start from a bfs. Suppose x_B, x_N are the basic and non-basic variables. Then we have

$$
\begin{array}{rcl}
\min c_B x_B + c_N x_N & & \\
s.t. \quad A_B x_B + A_N x_N & = & b \\
x & \geq & 0
\end{array}
$$

Note that $x_B = A^{-1}b - A_B^{-1} A_N x_N$ and the total cost of the solution is

$$
\begin{array}{rcl}
c^T x & = & c_B x_B + c_N x_N \\
 & = & c_B(A_B^{-1} b - A_B^{-1} A_N x_N) + c_N x_N \\
 & = & c_B A_B^{-1} b + (c_N - c_B A_B^{-1} A_N) x_N.
\end{array}
$$

Let us consider $d_N = c_N - c_B A_B^{-1} A_N$ as the reduced cost. If we can find an index $j \in N$ such that $d_j < 0$ then increasing x_j will decrease the total cost. We can increase x_j until a variable of x_B becomes zero; we obtain a new bfs. Doing this we move (pivot) from one bfs to another bfs. The running time of the algorithm is basically the number of moves from bfs to another bfs.

Every known rule for deciding the next non-basic variable to change has a counter-example showing that the worst case running time can be exponential. One natural questionis: what is the length of the shortest vertex-to-vertex path between two vertices of a convex polytope in which every two consecutive vertices are neighbours (by an edge). There is a famous (and wide open) conjecture:

Hirsch Conjecture: If we have m hyperplanes in d-dimensional Euclidean space has diameter no more than $n - d$.

This conjecture is known to be true for $d < 4$. The best known upper bound is sub-exponential as a function of n and d (so not even a polynomial bound is known let alone linear). Even if this conjecture is true it does not imply that the Simplex algorithm can find a sequence of pivoting operations that corresponds to a shortest path as such a path may not be monotone (in terms of the objective value over the vertices of the path).

Simplex algorithm was developed in 1947 (by Dantzig). In 1979 the first polynomial time algorithm for solving LP's was developed by Khachyan, named Ellipsoid algorithm. We will talk about that algorithm in the next week.

References

1. Combinatorial Optimization, Schrijver, (Volume 1) Springer-Verlag, 2003.

2. Combinatorial Optimization: Algorithms and Complexity, by Christos H. Papadimitriou, Kenneth Steiglitz, Dover Publications, 1998.

3. Lecture notes by M.X. Goemans for Combinatorial Optimization, 2007 http://www-math.mit.edu/ goemans/18433S07/18433.html

4. Introduction to Algorithms, by Cormen, Leiserson, Rivest, and Stein, McGraw Hill, 2001.