Lecture 1 - The Basics of Linear Programming

Uri Feige

November 3, 2011, part of November 10

1 The linear algebra of linear programs

Many optimization problems can be formulated as linear programs. The main features of a linear program are the following:

- Variables are *real numbers*. That is, they are continuous rather than discrete.
- The objective function is a linear function of the variables. (Each variable effects the object function linearly, at a slope independent of the values of the other variables.)
- Constraints on the variables are linear.

One of the first recorded examples of a linear program is the so called *diet* problem. Assume that a persons diet is composed entirely of n foods. Assume that there are m nutrients, and the person is required to consume at least b_i units of nutrient i (for $1 \leq i \leq m$). Let a_{ij} denote the amount of nutrient i present in one unit of food j. Let c_i denote the cost of one unit of food item i. The problem is to design a diet of minimal cost that supplies at least the required amount of nutrients.

Using matrix and vector representation (in which vectors are column vectors, A_i is row i of matrix A given as its column vector, and x^T is the transpose of vector x and hence a row vector), we have the following linear program:

minimize $c^T x$
subject to
$Ax \geq b$
$x \geq 0$

Let us remark that one could add constraints bounding from above the amount of nutrients consumed (e.g., not too much fat in the diet).

(Modeling the diet problem as a linear program includes many hidden assumptions such as that the cost is linear, that there are no interdependencies in the nutritional values of various foods, that any fractional amount of food can be bought, etc. As these assumptions are only approximations of the true state of affairs, an optimal solution to the linear program can only be viewed as a guideline for selecting a desirable solution, and not as the ultimate solution.)

A solution satisfying all constraints is *feasible*. A feasible solution that also optimizes the objective function is *optimal*.

1.1 Fourier-Motzkin elimination

In the coming lectures we will see approaches for solving linear programs. But first, let us sketch an inefficient approach.

Consider first the case of finding a feasible solution. If all constraints were equalities rather than inequalities, we could use Gaussian elimination (and find a feasible solution in polynomial time). For inequalities we may perform a similar (but less efficient) procedure known as Fourier-Motzkin elimination. To eliminate a variable x_i, we bring all inequalities in which it participates into one of two forms: $x_i \leq \alpha_j$ and $x_i \geq \beta_j$, where for every $1 \leq j \leq m$, α_j and β_j are linear functions of the rest of the variables. We can now eliminate x_i, and replace all constraints that included it by the new constraints $\beta_j \leq \alpha_k$ (for $1 \leq j \leq m$, $1 \leq k \leq m$). Repeating this procedure for the other variables gives eventually a set of inequalities in one variable, that can easily be solved. Note that each variable elimination at most squares the number of constraints. Hence the complexity of the above procedure is $O(m^{2^n})$ in the worst case. (Note however that if we can in each elimination step choose a variable that appears in at most four inequalities, then the number of inequalities does not grow, and the procedure becomes polynomial.)

Given the ability to find feasible solutions, we can in principle also find near optimal solutions. This can be done by replacing the objective function by a constraint $c^T x \leq t$, and searching for feasible solution. By doing binary search on t (assuming that we have reasonable estimates for lower and upper bounds of t), we can find a solution arbitrarily close to the true optimum. For the specific case of using Fourier-Motzkin elimination, we can treat t as a variable and leave it last in the elimination order. This allows us to find the exact optimum.

We shall now develop some theory that will allow us to design more efficient algorithms for solving linear programs.

1.2 Forms

The linear program for the diet problem is in *canonical form*. In a linear program in *general form*, the constraints are linear but may involve inequalities of both types ($\leq$ and $\geq$), as well as equalities ($=$). Variables may be required to be nonnegative ≥ 0, or else be unconstrained. Another useful form of a linear program is the *standard form*:

minimize $c^T x$
subject to
$Ax = b$
$x \geq 0$

All forms are equivalent in terms of their expressive power, and it is simple to transform a linear program in general form to standard form and to canonical form.

To replace an unconstrained variable x_i by nonnegative variables, introduce to auxiliary variables x_i^+ and x_i^-, replace every occurrence of x_i by $x_i^+ - x_i^-$, and add the two nonnegativity constraints $x_i^+ \geq 0$ and $x_i^- \geq 0$.

To change a $\leq$ inequality $A_i^T x \leq b_i$ to a $\geq$ inequality, negate it, obtaining $-A_i^T x \geq -b_i$.

To change an equality $A_i^T x = b_i$ to inequalities, replace it by $A_i^T x \leq b_i$ and $A_i^T x \geq b_i$.

To change an inequality $A_i^T x \leq b_i$ to an equality, introduce a *slack variable* y_i, the inequality $A_i^T x + y_i = b_i$, and the nonnegativity constraint $y_i \geq 0$.

Note that the above transformations change a general linear program with n variables and m constraints to a linear program with $O(n+m)$ variables and constraints.

For linear programs in standard form, it is convenient to assume that the constraints (rows of the matrix A) are linearly independent. If the rows are not linearly independent, then it suffices to consider rows of A that constitute a basis for the row space (a maximal linearly independent set of row vectors). Either every solution that satisfies the constraints that correspond to the basis satisfies all constraints, or the LP is infeasible.

1.3 Basic feasible solutions

Consider an LP in standard form, with linearly independent constraints. Let B be a submatrix of A containing exactly m linearly independent columns. This is a *basis* of the column space of A. Let x_B be the set of basic variables corresponding to the columns of B. If $B^{-1}b \geq 0$, then the following is a basic feasible solution: the basic variables are set to $B^{-1}b$, and the nonbasic variables are set to 0. Clearly this solution is feasible. Note that it satisfies n linearly independent constraints with equality: the m constrains of $Ax = b$, and $n - m$ of the nonnegativity constraints. The other (nonnegativity) constraints are also satisfied, though not necessarily with equality.

Each basis gives at most one basic feasible solution. (It gives none if the condition $B^{-1}b \geq 0$ fails to hold.) Two different bases may give the same basic feasible solution, in which case the basic feasible solution is degenerate (more than $n-m$ variables are set to 0).

Basic feasible solutions play an important role in linear programs.

Lemma 1 *For every basic feasible solution, there is a choice of objective function $c^t x$ for which the bfs is the unique optimal solution.*

Proof: Consider an objective function for which $c_i = 0$ if x_i is positive in the bfs, and $c_i = 1$ otherwise. The objective function is nonnegative for every feasible solution (because of the nonnegativity constraints), and 0 for the particular bfs. As there is no other feasible solution with the same (or larger by containment) set of 0 variables (otherwise $B^{-1}b$ would have more than one solution), the bfs is the unique minimum. □

Lemma 2 *Every LP in standard form is either infeasible, or the optimal value is unbounded, or it has a basic feasible solution that is optimal.*

Proof: Assume that the LP is feasible and that its optimum is bounded from below. Let q be used to denote the value of the objective function. First we show that for every value of q, if there is a solution x of value q (namely, $c^t x = q$), then there is a basic feasible solution x^* of value at most q.

Let x^+ be the set of positive variables in x. If the columns in A corresponding to x^+ are linearly independent, then we can complete them to a basis giving x as a bfs. Hence it remains to deal with the case that the columns in A corresponding to x^+ are not linearly independent. Denote them by $A^1, \ldots, A^t$ and the values of the positive variables by $z_1, \ldots, z_t$. Then we have $\sum_{i=1}^t A^i z_i = b$. There is also a solution w to $\sum_{i=1}^t A^i x_i = 0$. Consider now the sign of $c^t w$. If $c^t w < 0$, then perturbing x by a positive multiple of w reduces the value of the objective function. As the value of the objective function is bounded from below, it must be the case that changing x by a multiple of w eventually causes a new nonnegativity constraint to becomes tight. This gives a feasible solution with lower value for the objective function and at least one more variable that is 0. If $c^t w > 0$, then the same effect is reached by perturbing x by a negative multiple of w. If $c^t w = 0$ then using the fact that w has at least one nonzero variable (and is 0 whenever x is), there is a multiple of w that we can add to x that causes one more variable to become 0, without changing the value of the objective function. Repeating the above process we must eventually reach some solution x^* for which the columns in A corresponding to $(x^*)^+$ are linearly independent, implying a bfs.

As there are only finitely many basic feasible solutions (at most $\binom{n}{m}$), there is one with minimum value, and this must be an optimum solution for the LP. □

Remark: The above proof technique also shows that every LP in standard form that has a feasible solution and has no infinite line must also have a basic feasible solution. (In the above proof, change x by a multiple of w in the direction where this line in bounded, until the first time a new nonnegativity constraint becomes tight.)

Recall Cramer's rule for solving $Bx = b$, where B is an invertible order n matrix. The solution is

$$x_j = \frac{\det B^j}{\det B}$$

for $1 \leq j \leq n$, where here B^j is the matrix B with column j replaced by b. If each entry in B and b is an integer with absolute value at most M, then each x_j is a rational number with numerator and denominator bounded by at most $M^n n!$. This can be used to show that the length of numbers involved in a basic feasible solution are polynomially related to the input size. (Moreover, it can be shown that when a system of linear equations is solved by Guassian elimination, the length of intermediate numbers produced by the algorithm is also polynomially related to the input size.)

Lemma 2 and its proof have the following algorithmic consequences:

1. Given a feasible solution of value q, one can find in polynomial time a basic feasible solution of value at most q.

2. In order to solve an LP optimaly, it suffices to consider only basic feasible solutions. As there are at most $\binom{n}{m}$ basic feasible solutions, we can solve LPs optimaly in this time.

1.4 An application – the Beck-Fiala theorem

Let us first remark that one can extend the notion of basic feasible solutions also to LPs that are not in standard form. Consider the case that among the constraints of the LP there are n constraints that are linearly independent. (For LPs in standard form, these can simply be taken to be the nonnegativity constraints.) Then a basic feasible solution is one that satisfies n linearly independent constraints with equality. Also here, every LP that is feasible and bounded has a bfs.

The existence of basic feasible solutions is a property that is used in many contexts. Here is a nice example.

There is a set S on n items. We are given a collection of subsets $S_i \subset S$, $1 \leq i \leq m$. The goal is to color the items red and blue, such that every subset is "nearly balanced", namely, has roughly the same number of red items as blue items. The (absolute value of the) maximum difference between number of red and blue items in a subset is called the *discrepancy* of the coloring. The Beck-Fiala theorem gives sufficient conditions for a coloring of low discrepancy to exist, regardless of the values of n and m.

Theorem 3 *If every item is contained in an most d subsets, then there is a coloring with discrepancy at most $2d - 1$.*

Proof: Think of 0 as denoting the color red and 1 as denoting the color blue. Think of x_j as a variable representing the color of item j. Let a_{ij} be 1 if set i contains item j, and 0 otherwise. Hence we seek a 0/1 solution satisfying $|\sum_j a_{ij}x_j - |S_i|/2| \leq d - 1/2$ for every set S_i.

Consider the following set of linear constraints:

1. For all $1 \leq j \leq n$, $x_j \geq 0$.
2. For all $1 \leq j \leq n$, $x_j \leq 1$.
3. For all $1 \leq i \leq m$, $\sum_{1 \leq j \leq n} a_{ij}x_j = |S_i|/2$.

This set has a feasible solution, namely, $x_j = 1/2$. We shall "round" this fractional solution to an integer solution. This rounding will introduce a slackness of at most $d - 1/2$ in the subset constraints.

As the set of constraints is feasible, it has a basic feasible solution. In a bfs, at least $n - m$ variables are set to either 0 or 1. Update the set of constraints by removing the integer variables, and updating the right hand side of every subset constraint by the sum of values removed from its left hand side. Some subset constraints are "small", in the sense that they have at most d noninteger variables left. For them, no matter how we set the remaining variables, the slackness introduced will be at most $d - 1/2$. (We used here the fact that the right hand side of each constraint is half-integer, and the right hand side of small constraints must lie between $1/2$ and $d - 1/2$.) Hence we can ignore the small constraints. So we are left with a set of residual constraints and a set of variables whose fractional values satisfy the residual constraints. As each residual constraint has more than d variables, and each variable appears in at most d residual constraints, then m', the number of residual constraints, must be smaller than n', the number of fractional variables. Again, moving to a bfs of the residual system of constraints, $n' - m' > 0$ variables receive integral value. Continuing with this process until no residual constraints are left, the theorem is proved. $\square$

We note that the proof (together with the proof of Lemma 2) gives a polynomial time algorithm for finding a coloring of low discrepancy.

1.5 An open question

It is quite easy to slightly strengthen the Beck-Fiala theorem. In the proof, call a set small if it has at most $d - 1$ fractional variables. Then for small sets the slackness will be at most $d - 3/2$, giving a discrepancy of at most $2d - 3$. Now it might not be the case that all sets eventually end up being small – all residual sets might contain exactly d fractional variables. In this case, rounding all remaining fractional variables to their nearest integer gives a slackness of at most $d/2$. Hence the discrepancy is at most $\max[2d-3, d]$. For d equals 1 and 2, this is best possible. (For $d = 1$, a single set with one item has discrepancy 1. For $d = 2$, three sets, each containing two out of three items, requires discrepancy 2.) For $d \geq 3$ we have that $2d - 3 \geq d$, and hence the discrepancy is at most $2d - 3$, a mild improvement over the Beck-Fiala theorem.

It is conjectured that under the terms of the Beck-Fiala theorem, there always is a coloring with discrepancy $O(\sqrt{d})$, offering a large improvement when d is large. If true, this bound would be best possible (up to constant multiplicative factors), as even for a collection of d random subsets of a set of d items, every coloring has a discrepancy of $\Omega(\sqrt{d})$. (This last statement can be proven by a straightforward probabilistic argument.)

2 The geometry of linear programs

In Section 1 we presented linear programs from a linear algebra point of view. The use of a geometric representation helps build intuition about linear programming. This is easiest to visualize in two dimensions (when there are only two variables, or only two constraints), and still manageable in three dimensions. Much of the low dimension intuition is also true in higher dimensions, and can be formalized and applied for linear programs with an arbitrary number of dimensions.

2.1 Two variables or two constraints

The constraints of a linear program with two variables in canonical form can be drawn as half-planes, and the feasible region as an intersection of half-planes. The objective function can be represented either as a vector pointing in its direction, or as equivalue lines.

The vertices will correspond to bfs for the linear program, when transformed to standard form (using slack variables). A degeneracy is the result of three constraints intersecting at a point on the boundary of the feasible region. (When transforming the LP from canonical form to standard form, this will give three slack variables that are 0, which will be larger than $n - m$.)

For linear programs in standard form, the two-dimensional representation above becomes uninteresting. There can be at most two equality constraints, and the feasible region degenerates to a point.

When there are two constraints in standard form, each column of the matrix A (which corresponds to a variable) can be viewed as a point in the plane. The vector b is another point in the plane. A feasible solution is two vectors that can be added (with nonnegative coefficients) to give the point b. (In canonical form, a feasible solution is a set of vectors that can be added to give a point in the quarter plane to the right and above b.) There is a degenerate solution if a single vector points in the direction of b. (More than $n - m$ variables can be set to 0.)

2.2 Some standard terminology

Convex combination, convex set, convex hull. A point $x \in R^n$ is a *convex combination* of points $x_1, \ldots, x_t$ if there are some $\lambda_i \geq 0$ with $\sum_{i=1}^{t} \lambda_i = 1$ such that $x = \sum_{i=1}^{t} \lambda_i x_i$. A set S is *convex* if for every two points $x, y \in S$, every convex combination of them (namely, their *convex hull*) is in S. The intersection of convex sets is a convex set.

Linear and affine vector spaces. A linear (vector) subspace in R^n is a set of vectors closed under addition and under multiplication by scalars. Equivalently, it is the set of solutions to a system of homogeneous linear equations $Ax = 0$. It is a convex set. The dimension of a linear space is the maximum number of linearly independent vectors that it contains. It can be shown that this is equal to n minus the rank (number of linearly independent rows) of the matrix A above. For a vector $b \in R^n$ and vector space $S \in R^n$, the set $\{b + s | s \in S\}$ is called an affine subspace. It has the same dimension as S. Equivalently, an affine subspace is the set of solutions to a set of linear equations $Ax = b$. The dimension of a set $T \in R^n$ is the dimension of the minimal affine subspace containing it. In particular, the dimension of the set of solutions to a linear program in standard form (with linearly independent constraints) is at most $n - m$.

Hyperplane, halfspace. For a vector $a \in R^n$, $a \neq 0$, and $b \in R$, the set of points x satisfying $a^T x = b$ is called a *hyperplane*. It is an affine subspace of dimension $n - 1$. The set of points satisfying $a^T x \geq b$ is a halfspace. Hyperplanes and halfspaces are convex. Observe that a hyperplane is the intersection of two halfspaces.

Polyhedron, polytope. The intersection of finitely many halfspaces is a *polyhedron*. It follows that the set of constraints of a linear program define a polyhedron which is the region of feasible solutions. It is a convex set. A polyhedron P that is *bounded* (namely, for some scalar c, $x^T x \leq c$ for all $x \in P$) is a *polytope*.

Face, facet, edge, vertex. Let P be a polytope of dimension d in R^n, HS a halfspace supported by hyperplane H. If $f = P \bigcap HS$ is contained in H and not equal to P, then f is a face of P and H is a supporting hyperplane or P. A *facet* is a face of dimension $d - 1$. An *edge* is a face of dimension 1. A *vertex* is a face of dimension 0. Equivalently, we can define a vertex of a polytope (or polyhedron) P as point $x \in P$ such that for some vector $c \in R^n$, $c^T x < c^t y$ for every $y \in P - \{x\}$. The equivalence between the two definitions can be seen by taking c to be the normal of H at x, pointing inwards towards the polytope.

Extreme point. An *extreme point* of a polyhedron P is a point $x \in P$ such that there are no other two points $y, z \in P$ such that x is a convex combination of them.

Recall that in Section 1.3 we defined basic feasible solutions for linear programs in standard form. This can be generalized to any linear program, where we define a BFS as a feasible solution for which n linearly independent constraints are tight (including nonnegativity constraints).

Lemma 4 *For a polyhedron P and a point $x \in P$, the following three statements are equivalent.*

1. *x is an extreme point of P.*
2. *x is a vertex of P.*
3. *x is a basic feasible solution.*

Proof: We shall prove the lemma for an LP in standard form. The proof for an LP in general form is left as homework.

BFS implies vertex. This is essentially Lemma 1 from Section 1.3.

Vertex implies extreme point. If w is a vertex, then for some vector c, $c^T x$ is uniquely minimized over P at w. If w were not extreme, then we could write $w = \lambda y + (1 - \lambda) z$ for some $y, z \in P - \{w\}$ and $0 < \lambda < 1$, and then for either y or z, their inner product with c would be smaller. A contradiction.

Extreme point implies BFS. We show that not BFS implies not extreme. Let $w = (w_1, \dots, w_n) \in P$ not be a BFS. Let x^+ be the set of nonzero variables in w, and let A^+ be the columns of A that correspond to the nonzero variables. The columns in A^+ are linearly dependent (as w is not a BFS). Hence there is a nonzero solution to $A^+ x^+ = 0$. Call it d^+. Let d be a vector that agrees with d^+ on the variables of x^+, and is zero elsewhere. Consider the points $y = w + \epsilon d$ and $z = w - \epsilon d$, where ϵ is chosen small enough to ensure that $y \geq 0$ and $z \geq 0$. Both $y, z \in P$, and $w = y/2 + z/2$, showing that w is not an extreme point. □

Another useful way of characterizing a polytope is as the convex hull of all its vertices.

Lemma 5 *For a polytope P of dimension d, every point in it is the convex hull of at most $d + 1$ of its vertices.*

Proof: The proof is by induction on d. The base case (dimension 0, a single vertex) is trivial. For the inductive step, we need the following facts:

1. Every polytope has a vertex. (We have seen something similar in Section 1.3 – the existence of basic feasible solutions for LPs in standard form.)
2. Every face of a polytope is a polytope of lower dimension. (The face satisfies a linear equality that is not satisfied by some points in P.)

Take an arbitrary point $x \in P$. Take an arbitrary vertex v of the polytope. Follow the line from v through x (this portion must lie entirely within the polytope) until is hits a face of P (which is must do, as the polytope in bounded) at a point z. Then x is a convex combination of v and z. By induction, z is a convex combination of at most d vertices, showing that w is a convex combination of at most $d + 1$ vertices. □