Intro to linear programming

Emerson Melo

Indiana University

Linear Programming

- Linear programming is a subclass of convex optimization problems in which both the constraints and the objective function are linear functions.
- A linear program is an optimization problem of the form:

$$\begin{aligned} \text{minimize}\; & c^T x \\ \text{subject to}\; & Ax = b \\ & x \geq 0 \end{aligned}$$

  where $c \in \mathbb{R}^n, b \in \mathbb{R}^m$ and $A \in \mathbb{R}^{m \times n}$.
- This is called a linear program in standard form.

- Not all linear programs appear in this form but they can all be rewritten in this form using simple transformations.

- Linear programming is truly about solving systems of linear inequalities.
- In this sense, the subject natural follows the topic of the end of our last lecture, namely, that of solving systems of linear equations.

Example 1: Transportation

- m plants , s_i supply of plant i.
- n warehouses.
- d_j demand of warehouse j
- All plants produce product A (in different quantities) and all warehouses need product A (also in different quantities).
- The cost of transporting one unit of product A from i to j is c_{ij}.

- We want to minimize the total cost of transporting product A while still fulfilling the demand from the warehouses and without exceeding the supply produced by the plants.
- Decision variables: x_{ij}, quantity transported from i to j
- The objective function to minimize: $\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$
- The constraints are:
  - Not exceed the supply in any factory: $\sum_{j=1}^{n} x_{ij} \leq s_i, \forall i = 1, \ldots, m$
  - Fulfill needs of the warehouses: $\sum_{i=1}^{n} x_{ij} \geq d_j, \forall j = 1, \ldots, n$
  - Quantity transported must be nonnegative: $x_{ij} \geq 0, \forall i, j$

Example 2: Scheduling

- A hospital wants to start weekly nightshifts for its nurses.
- The goal is to hire the fewest number of nurses possible.
- There is demand d_j for nurses on days $\mathrm{j} = 1, \ldots 7$.
- Each nurse wants to work 5 consecutive days.
- How many nurses should we hire?

- The decision variables here will be $x_1, \ldots, x_7$, where x_j is the number of nurses hired on day j.
- The objective is to minimize the total number of nurses:

$$\sum_{j=1}^{7} x_j$$

- The constraints take into account the demand for each day but also the fact that the nurses want to work 5 consecutive days.
- This means that if the nurses work on day 1 , they will work all the way through day 5.

- Constraints:

$$
\begin{aligned}
x_1 + \quad\quad\quad\quad\quad\quad x_4 + x_5 + x_6 + x_7 &\geq d_1 \\
x_1 + x_2 + \quad\quad\quad\quad\quad\quad x_5 + x_6 + x_7 &\geq d_2 \\
x_1 + x_2 + x_3 + \quad\quad\quad\quad\quad\quad + x_6 + x_7 &\geq d_3 \\
x_1 + x_2 + x_3 + x_4 \quad\quad\quad\quad\quad\quad + x_7 &\geq d_4 \\
x_1 + x_2 + x_3 + x_4 + x_5 &\geq d_5 \\
x_2 + x_3 + x_4 + x_5 + x_6 &\geq d_6 \\
x_3 + x_4 + x_5 + x_6 + x_7 &\geq d_7
\end{aligned}
$$

- Naturally, we would want $x_1, \ldots, x_7$ to be positive integers as we do not want to get fractions of nurses.
- Such a constraint results in an IP (integer program).
- The obvious LP relaxation is the following: impose only nonnegativity constraints $x_1, \ldots, x_7 \geq 0$.

History of Linear Programming

- Solving systems of linear inequalities goes at least as far back as the late 1700 s, when Fourier invented (a pretty inefficient) solution technique, known today as the "Fourier-Motzkin" elimination method.
- In 1930s, Kantorovich and Koopmans brought new life to linear programming by showing its widespread applicability in resource allocation problems. They jointly received the Nobel Prize in Economics in 1975.
- Von Neumann is often credited with the theory of "LP duality" (the topic of our next lecture).

- In 1947, Dantzig invented the first practical algorithm for solving LPs: the simplex method.
- This essentially revolutionized the use of linear programmin in practice. (Interesting side story: Dantzig is known for solving two open problems in statistics, mistaking them for homework after arriving late to lecture.
- In 1979 , Khachiyan showed that LPs were solvable in polynomial time using the "ellipsoid method". This was a theoretical breakthrough more than a practical one, as in practice the algorithm was quite slow.
- In 1984, Karmarkar developed the "interior point method", another polynomial time algorithm for LPs, which was also efficient in practice. Along with the simplex method, this is the method of choice today for solving LPs.

LP in alternate forms

- As mentioned before, not all linear programs appear in the standard form:

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

- Essentially, there are 3 ways in which a linear program can differ from the standard form:
  1. it is a maximization problem instead of a minimization problem,
  2. some constraints are inequalities instead of equalities: $a_i^T x \geq b_i$,
  3. some variables are unrestricted in sign.
- There are, however, simple transformations that reduce these alternative forms to standard form.

Solving an LP in two variables geometrically

- Let us consider the problem:

$$\begin{aligned} \max \cdot x_1 + 6x_2 \\ \text{s.t. } x_1 \leq 200 \\ x_2 \leq 300 \\ x_1 + x_2 \leq 400 \\ x_1, x_2 \geq 0 \end{aligned}$$

- We are trying to find the "largest" level set of the objective function that still intersects the feasible region.

All possibilities for an LP

- Infeasible case

$$\begin{aligned} \min\ & x_1 + x_2 \\ \text{s.t.}\ & x_1 + 2x_2 \leq 8 \\ & 3x_1 + 2x_2 \leq 12 \\ & x_1 + 3x_2 \geq 13 \end{aligned}$$

- The three regions in the drawing do not intersect : there are no feasible solutions.

- Unbounded case

$$\begin{aligned} \min. \quad & 2x_1 - x_2 \\ \text{s.t.} \quad & x_1 - x_2 \leq 1 \\ & 2x_1 + x_2 \geq 6 \end{aligned}$$

- The intersection of two regions is unbounded; we can push the objective function as high up as we want.

- Infinite number of optimal solutions

$$\begin{aligned} \min\ & 2x_1 + 2x_2 \\ \text{s.t. } & x_1 \leq 200 \\ & x_2 \leq 300 \\ & x_1 + x_2 \leq 400 \\ & x_1, x_2 \geq 0 \end{aligned}$$

- There is an entire "face" of the feasible region that is optimal. Notice that the normal to this face is parallel to the objective vector.

- Unique optimal solution

$$
\begin{aligned}
\min\ & x_1 + 6x_2 \\
\text{s.t. }\ & x_1 \le 200 \\
& x_2 \le 300 \\
& x_1 + x_2 \le 400 \\
& x_1, x_2 \ge 0
\end{aligned}
$$

Geometry of LP

- The geometry of linear programming is very beautiful.
- The simplex algorithm exploits this geometry in a very fundamental way.
- We'll prove some basic geometric results here that are essential to this algorithm.

Definition

The set $\{x \mid a^T x = b\}$ where a is a nonzero vector in $\mathbb{R}^n$ is called a hyperplane.

Definition

The set $\{x \mid a^T x \geq b\}$ where a is a nonzero vector in $\mathbb{R}^n$ is called a halfspace.

Definition

The intersection of finitely many half spaces is called a polyhedron.

- This is always a convex set:
- Halfspaces are convex (why?) and intersections of convex sets are convex (why?).

Definition

A set $S \subset \mathbb{R}^n$ is bounded if $\exists K \in \mathbb{R}_+$such that $||x|| \leq K, \forall x \in S$.

Definition

A bounded polyhedron is called a polytope.

Definition

A point x is an extreme point of a convex set P if it cannot be written as a convex combination of two other points in P. In other words, there does not exist $y, z \in P, y, z \neq x$, and $\lambda \in [0,1]$ such that $x = \lambda y + (1-\lambda)z$.

- Alternatively, $x \in P$ is an extreme point if
  $x = \lambda y + (1-\lambda)z, \quad y, z \in P,\ \lambda \in [0,1] \Rightarrow x = y$ or $x = z$.
- Extreme points are always on the boundary, but not every point on the boundary is extreme.

Definition

Consider a set of constraints

$$\begin{aligned} a_i^T x &\geq b_i, \quad i \in M_1 \\ a_i^T x &\leq b_i, \quad i \in M_2 \\ a_i^T x &= b_i, \quad i \in M_3 \end{aligned}$$

Given a point $\bar{x}$, we say that a constraint i is tight (or active or binding) at $\bar{x}$ if $a_i^T \bar{x} = b_i$;

- Equality constraints are tight by definition.

Definition

Two constraints are linearly independent if the corresponding $a_j's$ are independent.

Definition

A point $x \in \mathbb{R}^n$ is a vertex of a polyhedron P, if

(i) it is feasible ($x \in P$),

(ii) $\exists n$ linearly independent constraints that are tight at x.

- You may be wondering if extreme points and vertices are the same thing. The theorem below establishes that this is indeed the case.
- Note that the notion of an extreme point is defined geometrically while the notion of a vertex is defined algebraically.
- The algebraic definition is more useful for algorithmic purposes and is crucial to the simplex algorithm. Yet, the geometric definition is used to prove the fundamental fact that an optimal solution to an LP can always be found at a vertex. This is crucial to correctness of the simplex algorithm.

Theorem (Equivalence of extreme point and vertex)

Let $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ *be a non-empty polyhedron with* $A \in R^{m \times n}$. *Let* $\bar{x} \in P$. *Then,* $\bar{x}$ *is an extreme point* $\Leftrightarrow$ $\bar{x}$ *is a vertex.*

Corollary

Given a finite set of linear inequalities, there can only be a finite number of extreme points.

Theorem (Equivalence of extreme point and vertex)

Let $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ be a non-empty polyhedron with $A \in R^{m \times n}$. Let $\bar{x} \in P$. Then, $\bar{x}$ is an extreme point $\Leftrightarrow$ $\bar{x}$ is a vertex.

Corollary

Given a finite set of linear inequalities, there can only be a finite number of extreme points.

Definition

A polyhedron contains a line if $\exists x \in P$ and $d \in \mathbb{R}^n, d \neq 0$, such that

$$x + \lambda d \in P, \forall \lambda \in \mathbb{R}$$

Theorem

Consider a nonempty polyhedron P. The following are equivalent:

(i) *P does not contain a line.*

(ii) *P has at least one extreme point.*

Corollary

Every bounded polyhedron has an extreme point.

Theorem (Optimality of extreme points)

Consider the LP:

$$\begin{aligned} &\min c^T x \\ &s.t.\ \ Ax \le b \qquad (P) \end{aligned}$$

Suppose P has at least one extreme point and there exists an optimal solution, then there exists an optimal solution which is at a vertex.

The Simplex Algorithm

- Consider some generic LP:

$$\begin{aligned} &\max c^T x \\ &\text{s.t. } Ax \leq b \\ &x \geq 0 \end{aligned}$$

- In a nutshell, this is all the simplex algorithm does:
    1. start at a vertex,
    2. while there is a better neighboring vertex, move to it.

Definition

Two vertices are neighbors if thev share n-1

- At every iteration of the simplex algorithm, we complete two tasks:
  1. Check whether the current vertex is optimal (if yes, we're done),
  2. If not, determine the vertex to move to next.

An example in two dimensions

$$
\begin{aligned}
\max \quad & x_1 + 6x_2 \\
\text{s.t.} \quad & x_1 \leq 200 \\
& x_2 \leq 300 \\
& x_1 + x_2 \leq 400 \\
& x_1 \geq 0 \\
& x_2 \geq 0
\end{aligned}
$$

- **Iteration 1:** The origin is feasible and all c_i are positive. Hence we can pick either x_1 or x_2 as the variable we want to increase. We pick x_1 and keep $x_2 = 0$.
- The origin corresponds to the intersection of (4) and (5). By increasing x_1, we are releasing (4).
- As we increase x_1, we must make sure we still satisfy the constraints.
- In particular, we must have $x_1 \leq 200$ (1) and $x_1 + x_2 \leq 400$ (3) which means $x_1 \leq 400$.
- Note that constraint (1) becomes tight before constraint (3) (recall that x_2 remains at zero as we increase x_1). Hence, the new vertex is at the intersection of (1) and (5), i.e., $D = (200, 0)$.

- Rewriting x_1 and x_2 in terms of y_1 , and y_2 , we get a new LP:

$$\begin{aligned}
&\max .200 - y_1 + 6y_2 \\
&\text{s.t.} \quad y_1 \geq 0 \\
&y_2 \leq 300 \\
&-y_1 + y_2 \leq 200 \\
&y_1 \leq 200 \\
&y_2 \geq 0
\end{aligned}$$

- Since the coefficient of y_2 is positive, we must continue.

- **Iteration 2:** The current vertex corresponds to the intersection of (1)(5).
- As the coefficient of y_2 is positive, we pick y_2 as the variable wevincrease, i.e., we release (5).
- The other constraints have to be satisfied, namely (2) and (3).
- This corresponds to $y_2 \leq 300$ and $y_2 \leq 200$.
- As $200 \leq 300$, constraint (3) is the one becoming tight next.

- The problem becomes:

$$
\begin{aligned}
\max .1400 + 5z_1 - 6z_2 \\
\text{s.t.} \quad z_1 &\geq 0 \\
z_1 - z_2 &\leq 100 \\
z_2 &\geq 0 \\
z_1 &\leq 200 \\
z_1 - z_2 &\geq -200
\end{aligned}
$$

Since coefficient of z_1 is positive, we must continue.

- **Iteration 3:** The current vertex is at the intersection (1) and (3).
- As the coefficient for z_1 is positive, we pick z_1 as the variable we will increase while keeping $z_2 = 0$.
- This means that we are releasing constraint (1).
- We have to meet all the constraints, limiting how much we can increase $z_1 : z_1 \leq 100, z_1 \leq 200$ and $z_1 \geq -200$. So (2) is the constraint becoming tight next.

- The problem becomes:

$$\begin{aligned} \max\, & 1900 - 5w_1 - w_2 \\ \text{s.t. } & w_1 - w_2 \leq 100 \\ & w_1 \geq 0 \\ & w_2 \geq 0 \\ & w_1 - w_2 \geq -100 \\ & w_1 \leq 300 \end{aligned}$$

- Both coefficients are negative. Hence we conclude that vertex B is optimal. The optimal value of our LP is 1900 .