COMPSCI 330: Design and Analysis of Algorithms	3/31/2016

Linear Programming

Lecturer: Debmalya Panigrahi — *Scribe: Tianqi Song*

1 Overview

In this lecture, we introduce linear programming. Linear programs are simply constrained optimization problems where all functions are linear. Even when limiting ourselves to linear functions, such problems are amazingly expressive. Many of the combinatorial problems we have seen until now can be expressed as linear program.[1]

2 Linear Programs

A linear program is a twist on the constraint satisfaction problem, which seeks an assignment of variables optimizing an *objective* subject to *constraints*.

$$\begin{aligned} \min / \max \quad & f(x) \\ \text{s.t.} \quad & g_1(x) \leq b_1 \\ & g_2(x) = b_2 \\ & g_3(x) \geq b_23 \\ & \cdots \end{aligned}$$

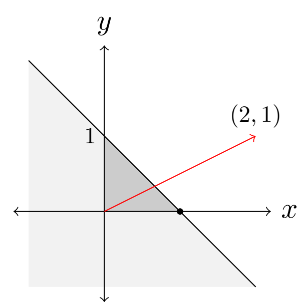
In linear programming, both objective and constraints are *linear functions* of variables.

$$f(x) = a_1x_1 + a_2x_2 + \cdots + a_nx_n$$

With n variables, we can visualize (the solutions of) any linear program as a *convex polyhedron* in $\mathbb{R}^n$.

Example 1. Consider the following linear program:

$$\begin{aligned} \max \quad & 2x + y \\ \text{s.t.} \quad & x + y \leq 1 \\ & x,\ y \geq 0 \end{aligned}$$



Each complete assignment of variables is a point in $\mathbb{R}^2$.

$$x = 0,\ y = 1 \rightarrow (0, 1)$$

[1] Some materials are from a note by Allen Xiao for COMPSCI 532 in Fall 2015.

The linearity of the objective function makes its coefficients appear as a direction in $\mathbb{R}^2$. The linear program asks to find a feasible point furthest in the direction of the objective. This can be evaluated as the dot product of both vectors, which produces the original objective expression.

$$(2,1)^\intercal(x,y) = 2x + y$$

The linearity of constraints causes them to appear as *half-spaces* in $\mathbb{R}^2$, bounded by *hyperplanes*. Points within a half-space "satisfy" the constraint, while points on the bounding hyperplane meet that constraint with equality. Points in the intersection of every half-spaces are *feasible* solutions. The intersection of the half-spaces forms a convex polyhedron. Thus, linear programming is a special case of convex programming.

Another way to interpret the objective is as the direction of "gravity". If we drop a ball from the inside of the feasible polyhedron, the point it stops at is the optimal. Of course, this suggests several possibilities for the solution. We might have a finite (bounded) solution on some intersection of constraints, an infinite (unbounded) solution if the polyhedron has no "bottom", and finally no solution if the feasible space is empty and we cannot initially place the ball. We will state these possibilities formally later.

Example 2. Maximum flow can be expressed as a linear program. Recall that $f(x,y)$ is the *net flow* across (x,y).

$$\begin{array}{lll} \max & \sum_{x \in V \setminus \{s\}} f(s,x) & \\ \text{s.t.} & f(x,y) \le u(x,y) & \forall (x,y) \in E \\ & \sum_{y} f(x,y) = 0 & \forall x \neq s,t \\ & f(x,y) + f(y,x) = 0 & \forall (x,y) \in E \end{array}$$

Linearity allows us to represent linear programs in a compact matrix form. We make a vector of variables x and objective coefficients c. Each constraint is a row of matrix A bounded by a value in b.

Definition 1. *Let $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$. The **canonical form** of a linear program is:*

$$\begin{array}{ll} \min & c^\intercal x \\ \text{s.t.} & Ax \ge b \\ & x \ge 0 \end{array}$$

Definition 2. *Let $x \in \mathbb{R}^n$, $A_1, A_2, A_3 \in \mathbb{R}^{m \times n}$, $b_1, b_2, b_3 \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, $x = (x_1\ x_2\ x_3)$. The **extended form** of a linear program is:*

$$
\begin{aligned}
\min \quad & c^\intercal x \\
\text{s.t.} \quad & A_1 x \geq b_1 \\
& A_2 x = b_2 \\
& A_3 x \leq b_3 \\
& x_1 \geq 0 \\
& x_2 \leq 0
\end{aligned}
$$

Definition 3. *Let $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$. The* ***standard form*** *of a linear program is:*

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

Claim 1. *Any linear program may be represented in canonical form.*

Proof. Where might a linear program deviate from canonical form? Consider a minimization problem with constraints A, where a_j is the jth row of A.

1. A variable x may be constrained to be negative ($x \leq 0$). We can perform a change of variables using $z = -x$. It follows that $z \geq 0$, which fits the canonical form sign constraint.

2. A variable x may not be constrained in sign at all. We can again perform a change of variables, this time using $z_1 - z_2 = x$, where $z_1, z_2 \geq 0$. Intuitively, we can represent any number as the difference of two nonnegative numbers.

3. A constraint may be in the $\leq$ direction rather than the $\geq$ direction.

   $$a_j x \leq b_j$$

   Negating the entire constraint gives us the correct direction.

   $$-a_j x \geq -b_j$$

   So we will replace the row a_j and b_j with their negatives.

4. A constraint may be with equality.

   $$a_j x = b_j$$

   Recall that equality holds when both $\leq$ and $\geq$ hold. We replace the equality constraint with two inequalities:

   $$
   \begin{aligned}
   & a_j x = b_j \\
   \implies \quad & (a_j x \geq b_j) \text{ and } (a_j x \leq b_j) \\
   \implies \quad & (a_j x \geq b_j) \text{ and } (-a_j x \geq -b_j)
   \end{aligned}
   $$

5. Although not applicable to canonical form, if we want to transform a $\geq$ constraint into an equality constraint, we can accomplish this by adding a slack variable $z \geq 0$:

$$
\begin{aligned}
& a_j x \geq b_j \\
\implies \quad & a_j x - z = b_j
\end{aligned}
$$

$\square$