

Lecture 1

Lecturer: David P. Williamson

Scribe: Ebad Ahmed

Much of the course will be devoted to linear programming (LP), the study of the optimization of a linear function of several variables subject to linear equality and inequality constraints. Here “programming” should be understood in the sense of planning — more like TV programming than computer programming — and linear refers to the types of functions involved.

There are many forms such a problem can take. We start with a (column) vector $x \in \mathbf{R}^n$ of *decision variables*. We want to maximize a linear *objective function* $c^T x$ for $c \in \mathbf{R}^n$ subject to linear inequalities $Ax \leq b$ for $A \in \mathbf{R}^{m \times n}$, $b \in \mathbf{R}^m$. The inequality is componentwise; if a_j is the j th row of A , and b_j is the j th component of b , then we want to have $a_j x \leq b_j$ for $j = 1, \dots, m$. Any decision vector x for which $Ax \leq b$ is called *feasible* (or a *feasible solution*). We call the set

$$Q := \{x \in \mathbf{R}^n : Ax \leq b\}$$

of points satisfying all the constraints the *feasible region* or *feasible set*. For any feasible solution x , $c^T x$ is the *value* of x . A feasible solution x^* is *optimal* if it has the maximum value (if it exists) among all feasible solutions, and $c^T x^*$, for x^* optimal, is the value of the linear program. Note that all our vectors are columns, and that a subscripted letter could be a component of a vector (like b_j) or itself a vector (like a_j). We write:

$$\begin{array}{ll} \max & c^T x \\ & Ax \leq b, \end{array}$$

or sometimes just $\max(c^T x : Ax \leq b)$.

We will study the following things:

- What is the geometry of the feasible region?
- What form do optimal solutions take? How can we know if a solution is optimal?
- How can we efficiently find an optimal solution?

Let’s consider a concrete example.

Example 1 (*Product Mix*): The Marie-Antoinette bakery makes high-end bread and cakes. Each loaf requires 3 pounds of flour and 2 hours of oven time, while each cake requires just 1 pound of flour but 4 hours of oven time. There are 7 pounds of flour and 8 hours of oven time available, and all other ingredients are in ample supply. (Note that this is a very small operation, and the oven can handle only one bakery product at a time!) Each loaf and each cake makes a \$5 profit. How many loaves and how many cakes should be made to maximize profit?

If we let x_1 and x_2 denote the numbers of loaves and cakes made (our decision variables), then the objective function is $5x_1 + 5x_2$, to be maximized. The flour constraint is $3x_1 + 1x_2 \leq 7$, while the oven constraint is $2x_1 + 4x_2 \leq 8$. Are these all the constraints? No: the numbers of loaves and cakes cannot be negative, so we get

$$\begin{array}{rclcl} \max_x & 5x_1 & + & 5x_2 & \\ & 3x_1 & + & 1x_2 & \leq 7, \\ & 2x_1 & + & 4x_2 & \leq 8, \\ & x_1 \geq 0, & & x_2 \geq 0. & \end{array}$$

We might argue that x_1 and x_2 should be integers, but this makes our problem an integer linear programming problem, which is potentially much harder to solve. So for now we allow x_1 and x_2 to take on any real values. This might be a reasonable approximation for a problem instance of more realistic size: perhaps x_j is the number of batches (say of 100 loaves or 100 cakes) made, so fractions are possible.

Our problem above is of the form $\max\{c^T x : Ax \leq b\}$, where $A = \begin{bmatrix} 3 & 2 & -1 & 0 \\ 1 & 4 & 0 & -1 \end{bmatrix}^T$ (note that the rows of A give the coefficients of the constraints), $c = \begin{pmatrix} 5 \\ 5 \end{pmatrix} = (5, 5)^T = (5; 5)$ and $b = (7, 8, 0, 0)^T = (7; 8; 0; 0)$.

We can solve such a small problem graphically, by drawing the feasible region in \mathbf{R}^2 :

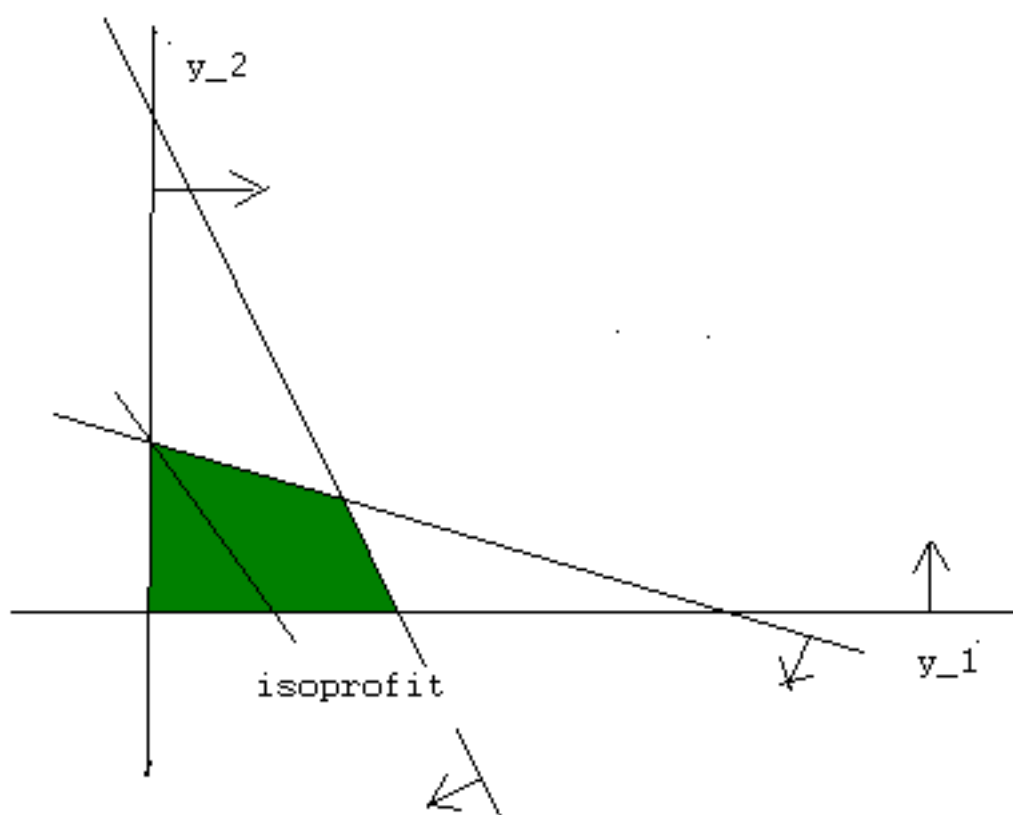


Figure 1: The feasible region and an isoprofit line for Example 1.

We can also draw the line $5x_1 + 5x_2 = 10$; parallel lines show points of equal profit. There are clearly an infinite number of feasible points (by contrast, there are only finitely many *integer* feasible points: so why is integer linear programming harder than LP??). But moving the “isoprofit” line up as much as possible, we see that $(2; 1)$ looks like a good point; it gives a profit of \$15.

This is pretty convincing: but can we get an *algebraic* proof that this is optimal, which might

work even if we can't draw a picture? Yes! *Any* feasible point must satisfy the two constraints

$$\begin{array}{rcl} 3x_1 & + & 1x_2 \leq 7, \\ 2x_1 & + & 4x_2 \leq 8, \end{array}$$

so satisfies their sum: $5x_1 + 5x_2 \leq 15$. But we also have a feasible point, $x = (2; 1)$, which gives objective function value 15: so it must be optimal!

Let us modify our example a bit. What if the profit per loaf becomes \$7 and per cake \$4? The objective function is now $7x_1 + 4x_2$. Adding the constraints no longer works, but we could take positive multiples of them first:

$$\begin{array}{rcl} 2 & \times & 3x_1 + 1x_2 \leq 7, \\ + & 1/2 & \times \quad 2x_1 + 4x_2 \leq 8, \\ \hline & & 7x_1 + 4x_2 \leq 18, \end{array}$$

and the feasible point $(2; 1)$ gives exactly \$18 revenue, so is still optimal.

What if the objective function becomes $1x_1 + 4x_2$? Simple algebra suggests

$$\begin{array}{rcl} -2/5 & \times & 3x_1 + 1x_2 \leq 7, \\ + & 11/10 & \times \quad 2x_1 + 4x_2 \leq 8, \\ \hline & & 1x_1 + 4x_2 \leq 6?? \end{array}$$

Is this valid? No!! Multiplying an inequality by a negative number changes its sense, and we can't then add the resulting inequalities.

Instead, we can proceed as follows:

$$\begin{array}{rcl} 1 & \times & 2x_1 + 4x_2 \leq 8, \\ + & 1 & \times \quad -1x_1 \leq 0, \\ \hline & & 1x_1 + 4x_2 \leq 8, \end{array}$$

and $x = (0; 2)$ is feasible and gives objective function value \$8.

Let's generalize this discussion. For each constraint $a_j x \leq b_j$, we want to multiply it by $y_j \geq 0$, so that we have

$$\begin{array}{rcl} y_1 & \times & (a_1 x \leq b_1) \\ y_2 & \times & (a_2 x \leq b_2) \\ & & \vdots \\ + & y_m & \times \quad (a_m x \leq b_m) \\ \hline & & c^T x \leq y^T b \end{array}$$

So we generate an upperbound of $\sum_j y_j b_j = y^T b$. Also, for each x_i , we want to have exactly c_i copies of x_i . Thus we want

$$\begin{array}{rcl} y_1 a_{11} + y_2 a_{21} + \cdots + y_m a_{m1} & = & c_1 \\ y_1 a_{12} + y_2 a_{22} + \cdots + y_m a_{m2} & = & c_2 \\ & & \vdots \\ y_1 a_{1n} + y_2 a_{2n} + \cdots + y_m a_{mn} & = & c_n, \end{array}$$

or $A^T y = c$. Then by the same arguments as above we have that for any feasible x , $c^T x \leq y^T b$. We summarize this argument in the following lemma.

Lemma 1 Let y be a column vector that satisfies $y \geq 0$ and $A^T y = c$, then for all x that satisfy $Ax \leq b$ we have that $c^T x \leq y^T b$.

Proof: We know that $b_j \geq a_j x$ for all j . Multiplying this inequality by the non-negative y_j , we get that $y_j b_j \geq y_j (a_j x)$. Adding these constraints up for all j , we get

$$\begin{aligned} \sum_{j=1}^m y_j b_j &\geq \sum_{j=1}^m y_j (a_j x) \\ &= \sum_{j=1}^m y_j \sum_{i=1}^n a_{ji} x_i \\ &= \sum_{i=1}^n \left(\sum_{j=1}^m y_j a_{ji} \right) x_i \\ &= \sum_{i=1}^n c_i x_i. \end{aligned}$$

More compactly,

$$y^T b \geq y^T (Ax) = (y^T A)x = (A^T y)^T x = c^T x.$$

□

Our goal is to derive the best upper bound for the linear program that can be derived this way, i.e., the upper bound $\alpha = y^T b$ that has the smallest value of the problem. We can write this as another linear program: $\min(y^T b : A^T y = c, y \geq 0)$. This is the *dual* linear program. The original linear program is called the *primal*. The lemma above then implies what is known as the weak duality theorem:

Theorem 2 (Weak Duality) The maximum in the primal LP has value less than or equal to the minimum in the dual LP.

Note that it is convenient to denote that if a maximization problem has no feasible solution then its value is $-\infty$, and if a minimization problem has no feasible solution that its value is $+\infty$. The main theorem of linear programming (known as the strong duality theorem) states that the max is equal to the min, with one possible exception: it is possible that the max is $-\infty$ and the min is $+\infty$, i.e., that neither the primal nor the dual has any solution at all.

In the examples we worked above, it was always the case that the value of the primal was *equal* to the value of the dual. Were we just lucky? No, whenever the primal or the dual have a feasible solutions, this is always the case. This is called *strong duality* and is the fundamental theorem in linear programming. It is possible that neither the primal nor the dual have feasible solutions.

Theorem 3 (Strong Duality) If either the primal or dual has a feasible solution, then the value of the primal LP equals the value of the dual LP.

The proof of the strong duality theorem is nowhere as simple as that of the weak duality theorem, and one of the things we will study in this course is how to go about proving strong duality.