CSC 373 - Algorithm Design, Analysis, and Complexity **Summer 2016**

Lalla Mouatadid

Linear Programming: Introduction

A bit of a historical background about linear programming, that I stole from Jeff Erickson's lecture notes:

"Linear programming was used implicitly by Fourier in the early 1800s, but it was first formalized and applied to problems in economics in the 1930s by Leonid Kantorovich. Kantorivich's work was hidden behind the Iron Curtain (where it was largely ignored) and therefore unknown in the West. Linear programming was rediscovered and applied to shipping problems in the early 1940s by Tjalling Koopmans. The first complete algorithm to solve linear programming problems, called the *simplex method*, was published by George Dantzig in 1947. Koopmans first proposed the name "linear programming" in a discussion with Dantzig in 1948. Kantorovich and Koopmans shared the 1975 Nobel Prize in Economics "for their contributions to the theory of optimum allocation of resources". Dantzig did not; his work was apparently too pure. Koopmans wrote to Kantorovich suggesting that they refuse the prize in protest of Dantzig's exclusion, but Kantorovich saw the prize as a vindication of his use of mathematics in economics, which his Soviet colleagues had written off as "a means for apologists of capitalism".

Linear Programming is a powerful algorithmic tool that allows us to express a number of optimization problems in a simple framework. To build some intuition on how a linear program is constructed, we start with the following simple example:

A rancher is mixing two types of food for his cattle, brand X and brand Y. Each serving requires 60 grams of protein and 30 grams of fat. Brand X has 15gr of protein and 10gr of fat and costs 80 cents per unit. Brand Y has 20gr of protein and 5gr of fat and costs 50 cents per unit. How much of each brand should the rancher buy in order to minimize his cost?

We can express this optimization problem in a more mathematical form as follows: We first need to define what it is we are optimizing. In this case, we want to minimize the rancher's costs $\mathcal{C}$. To formally define $\mathcal{C}$, we introduce two new variables: x_1 to represent the number of units of brand X and x_2 the number of units of brand Y. We can thus express the cost as follows:

$$\text{minimize } \mathcal{C} = 0.8x_1 + 0.5x_2 \tag{1}$$

Since each x_1 unit costs 80cents, and x_2 costs 50cents. We call (1) the **objective function**. The goal is to determine the values of x_1 and x_2 that will minimize (1). We have a few **constraints** imposed by the problem (and the cattle ...) that we need to satisfy.

First, we can't buy a negative number of x_1 or x_2 units, so

$$x_1, x_2 \geq 0 \tag{2}$$

Second, we need to satisfy the fat and protein constraints:

$$15x_1 + 20x_2 \geq 60 \qquad \text{We need at least 60gr of protein} \tag{3}$$

$$10x_1 + 5x_2 \geq 30 \qquad \text{and at least 30gr of fat} \tag{4}$$

And that's it! So if we put it all together, we can express our problem as follows:

$$\begin{aligned} \text{minimize } & \mathcal{C} = 0.8x_1 + 0.5x_2 \\ \text{subject to } & 15x_1 + 20x_2 \geq 60 \\ & 10x_1 + 5x_2 \geq 30 \\ & x_1, x_2 \geq 0 \end{aligned}$$

What's special about this representation? The objective function is **linear**, and the constraints are **linear**. It's a linear program, LP for short. In order to solve it, we will think about the LP in terms of matrices and vectors. Out objective function is a minimization of a dot product of two vectors:

$$c = (0.8, 0.5) \text{ and } x = (x_1, x_2)$$

So we can rewrite the objective function as:

$$\min_x c \cdot x$$

Similarly, we can express each constraint as a dot product where:

$$\begin{aligned} a_1 \cdot x &\geq 60 \\ a_2 \cdot x &\geq 30 \\ a_1 &= (15, 20) \\ a_2 &= (10, 5) \end{aligned}$$

Or, we can collect all the a_i vectors into a matrix A and the constants into a b vector as follows:

$$\begin{aligned} Ax &\leq b \\ \text{where } A = (a_1, a_2)^T \text{ and } b &= (60, 30)^T \end{aligned}$$

And so our LP becomes:

$$\begin{aligned} \min_x \ & c \cdot x \\ \text{subject to } & Ax \geq b \\ & x \geq 0 \end{aligned}$$

It turned out in fact that every linear program can be converted into the above form known as the canonical form of the LP [1]. It is indeed a more compact expression of:

$$\begin{aligned} \text{minimize} & \sum_{j=1}^{d} c_j x_j \\ \text{subject to } & \sum_{j=1}^{d} a_{ij} x_j \geq b_i \text{ for every } i = 1...n \\ & x_j \geq 0 \text{ for every } j = 1...d \end{aligned}$$

Formally, we define linear programming as follows:

Input: A linear program with d variables (d for dimension usually) and m constraints. **Problem**: A setting to each of the variables so that all the constrainsts are satisfied and the objective function is optimized.

Notice that the problem could either be a minimization (example above) or a maximization problem.

LPs have a nice geometric representation that we will exploit to solve them.

[1] sometimes, called the standard form

The Geometry of Linear Programming:

A vector x that satisfies the constraints is in fact just a point $x \in \mathbb{R}^d$. We say that a point x is **feasible** if it satisfies the constraints of the LP. The set of all feasible points is called the **feasible region** of the linear program.This feasible region has a nice geometric structure that helps us solve the LP.

In the example above, consider a constraint $a_1 \cdot x \geq b_1$; the corresponding linear function $a_1 \cdot x = b_1$ defines a line in 2D and thus splits the plan into two regions. The inequality $\geq b_1$ just indicates which side of the plane a feasible solution should lie on. Therefore, given all the constraints, compacted in $Ax \geq b$, we create a feasible region in the plane where every side of the region satisfies some constraint.

This same geometric interpretation applies to higher dimensions. In general, any linear equation on d variables defines a hyperplane in $\mathbb{R}^d$[2]. Every hyperplane divides $\mathbb{R}^d$ into two halfspaces, where the set of points in each halfspace satisfies a linear inequality.

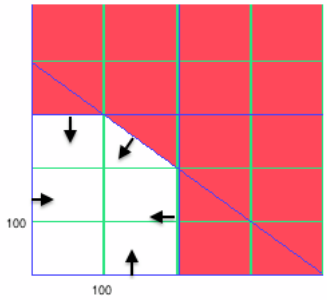
The feasible points are precisely the set of points that satisfy all the linear inequalities, i.e. all the constraints of the LP. So a feasible region id just the intersection of the halfspaces and the hyperplanes, where the halfspace satisfy the strict inequality constraints, and the hyperplanes satisfy the equality constraints.

Recall that that a polyhedron is the intersection of a finite number of hyperplanes and halfpsaces. Convince yourself that every polygon is convex.

We next go back to an example in 2D, since it is easier to visualize. Consider the following LP:

$$\text{maximize } x_1 + 6x_2 \tag{5}$$
$$\text{subject to } x_1 \leq 200 \tag{6}$$
$$x_2 \leq 300 \tag{7}$$
$$x_1 + x_2 \leq 400 \tag{8}$$
$$x_1, x_2 \geq 0 \tag{9}$$

Plotting all the constraints, we get the white feasible region:



The feasible region is shown in white.

So what does it mean to solve this LP? In essence, we are looking for a value p (think p for profit since it is a maximization problem), such that $p = x_1 + 6x_2$ is maximized. This is just the equation of a line with slope $\frac{-1}{6}$. For different values of p, we can "move" the line $x_1 + 6x_2 = p$ as far as possible while still touching the feasible region. The optimum solution will be the very last feasible point that the line sees; and I claim that this must be a vertex of the polygon. (Why?).

[2]We get a line in 2D, a plane in 3D, etc.

When you see why, you might wonder what if the profit line $p = x_1 + 6x_2$ is parallel to an edge of the polygon? Then the optimum solution is not unique but we are guaranteed the existence of optimum vertex! So how do we find this optimum vertex? We could try all the vertices of the polygon, and and pick the one that maximizes p, but we don't need to!

In 1947, George Dantzig introduced the Simplex Method. I recommend you read the textbook on how Simplex works in detail, we will look at the geometric interpretation of what the algorithm does, which basically is:

- Start at a vertex on the polygon.
- Repeatedly look for an adjacent vertex of better objective value.
- If we reach a vertex v that has no better neighbours, then the algorithm declares v to be optimal and halts.

Applying this to the example above, we start at the origin $(0, 0)$ (a vertex of the polygon) and compute $p = 0 + 6 \cdot 0 = 0$, so at $(0, 0)$ our profit is $p = 0$. Moving to the next vertex in the polygon, we have the point $(200, 0)$ which gives us $200 + 6 \cdot 0 = 200$, at $(200, 200)$ we get $p = 1400$, at $(100, 300)$ we get $p = 1900$ and at $(0, 300)$ we get $p = 1800 < 1900$. Therefore by the algorithm, the vertex lying in point $(100, 300)$ is the optimum vertex, and $p = 1900$.

Why does this local test imply global optimality? For one, the polygon is convex. Now think of the profit line going through $(100, 300)$: Since all the neighbours of the vertex in $(100, 300)$ lie below the profit line, the rest of the feasible polygon must also lie below this line, and thus $(100, 300)$ is optimal.

Simplex just returns the optimal solution: $(x_1, x_2) = (100, 300)$ and $p = 1900$, but how do we check this answer? We could maybe try to find a tight upperbound on p just by looking at our constraints? For instance, constraints (6) and (7) tell us:

$$x_1 + 6x_2 \leq 200 + 6 \cdot 300 \tag{10}$$
$$= 2000 \qquad \text{this means, } p \text{ can't be more than 2000.} \tag{11}$$

Another set of constraints, say (7) and (8) tell us:

$$5x_2 + x_1 + x_2 \leq 5 \cdot 300 + 400 \tag{12}$$
$$x_1 + 6x_2 \leq 1900 \tag{13}$$

This set of constraints gives us a tighter bound on p; but we know that $p = 1900$ using Simplex. We say that triple $(0, 5, 1)$ is a **certificate of optimality**. Consider the inequality above (12): We put it together by taking 0 of the first constraint (6), 5 of the second constraint (7) and 1 of the third constraint (8), so the triple (0, 5, 1) applied on the set of constraints gives us the optimal profit. The question is then how to find this certificate.

The goal is to find a triple (y_1, y_2, y_3) that, when applied to the set of constraints (3 constraints in the above example, hence a vector $y = (y_1, y_2, y_3) \in \mathbb{R}^3$), gives us a tight upperbound. That is:

$$y_1 \cdot (x_1 \leq 200)$$
$$y_2 \cdot (x_2 \leq 300)$$
$$y_3 \cdot (x_1 + x_2 \leq 400)$$

Clearly all the y_i's have to be positive, otherwise multiplying by a negative constant would not maintain the

inequalities. Expanding and summing over the inequalities above, we get:

$$\begin{aligned} y_1x_1 + y_2x_2 + y_3x_1 + y_3x_2 &\leq 200y_1 + 300y_2 + 400y_3 \\ (y_1 + y_3)x_1 + (y_2 + y_3)x_2 &\leq 200y_1 + 300y_2 + 400y_3 \\ \text{and } y_1, y_2, y_3 &\geq 0 \end{aligned}$$

We want to maximize the profit $p = x_1 + 6x_2$, so we want the left hand side of the above inequality to be $x_1 + 6x_2 = (y_1 + y_3)x_1 + (y_2 + y_3)x_2$ so that the right hand side is an upperbound. We achieve that by setting $(y_1 + y_3) = 1, (y_2 + y_3) = 6$; in fact we could even set $(y_1 + y_3) \geq 1, (y_2 + y_3) \geq 6$. Therefore our upper bound becomes:

$$\begin{aligned} x_1 + 6x_2 &\leq 200y_1 + 300y_2 + 400y_3 \\ y_1, y_2, y_3 &\geq 0 \\ y_1 + y_3 &\geq 1 \\ y_2 + y_3 &\geq 6 \end{aligned}$$

Clearly we can assign large values to y_1, y_2, y_3 and still satisfy all the above constraints, for instance $y_1 = y_2 = y_3 = 1000$ satisfy all the constraints and is an upperbound to $x_1 + 6x_2$. The goal however is to find a upper bound as tight as possible; what does this mean? It means we want to find values of y_1, y_2, y_3 that minimize $200y_1 + 300y_2 + 400y_3$ while satisfying all the constraints. Ah! Ah! This is just a minimization problem! Another linear program! #stillexcited.

$$\begin{aligned} \text{minimize } & 200y_1 + 300y_2 + 400y_3 && (14) \\ \text{subject to } & y_1, y_2, y_3 \geq 0 && (15) \\ & y_1 + y_3 \geq 1 && (16) \\ & y_2 + y_3 \geq 6 && (17) \end{aligned}$$

So if we can solve this new LP, then we'll get the best upper bound on our original LP. In fact, if we can find a pair of feasible solutions to the two LP that are equal, then they must be optimal! (Why?) We call the original LP the **primal** LP, and this new one the **dual** LP. For our example, we know that both $(x_1, x_2) = (100, 300)$ and $(y_1, y_2, y_3) = (0, 5, 1)$ give us a value of 1900, so this must be optimal. This is not by coincidence, it is called the Duality Theorem:

Theorem 1. *If an LP has a bounded optimum, then so does its dual and the two optimum values coincide.*

In general, we can go from a primal LP to its dual by a "mechanical" translation; essentially swapping the constraints and the variables:

$$\begin{aligned} \max\ & c \cdot x \\ \text{subject to } & Ax \leq b \\ & x \geq 0 \end{aligned}$$

We extract its dual as follow:

$$\begin{aligned} \min\ & y \cdot b \\ \text{subject to } & yA \geq c \\ & y \geq 0 \end{aligned}$$

The theorem above is better known as the **Fundamental Theorem of Linear Programming**, and it says:

Theorem 2. *A linear program has an optimal solutin x^* iff its dual has an optimal solution y^* such that $c \cdot x^* = y^* A x^* = y^* \cdot b$.*

And we have already encountered this when we did the Max Flow/Min Cut theorem.

The weaker version of this theorem, is the known as the **Weak Duality Theorem**, and it says:

Theorem 3. *If x is a feasible[3] solution for a canonical LP, and y a feasible solution for its dual, then $c \cdot x \leq yAx \leq y \cdot b$.*

Clearly if the LP has an unbounded solution, the dual is infeasible.

Notice however that the primal LP is not always a maximizing problem. What's the dual of the following LP?

$$\begin{aligned} \text{minimize } & 7x_1 + x_2 + 5x_3 \\ \text{subject to } & x_1 - x_2 + 3x_3 \geq 10 \\ & 5x_1 + 2x_2 - x_3 \geq 6 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

[3]Notice that the condition on x is feasibility only, not necessarily optimality, hence this weaker version.