Algorithmic Game Theory and Applications

Lecture 5: Introduction to Linear Programming

Kousha Etessami

"real world example": the diet problem

- You are a fastidious eater. You want to make sure that every day you get enough of each vitamin: vitamin 1, vitamin 2,...., vitamin m.
- You are also frugal, and want to spend as little as possible.
- There are n foods available to eat: food 1, food 2, ..., food n.
- Each unit of food j has $a_{i,j}$ units of vitamin i.
- Each unit of food j costs c_j.
- Your daily need for vitamin i is b_i units.
- Assume you can buy each food in fractional amounts. (This makes your life <u>much</u> easier.)
- How much of each food would you eat per day in order to have all your daily needs of vitamins, while minimizing your cost?

A Linear Programming Example

Find $(x, y) \in \mathbb{R}^2$ so as to: Maximize $2x + y$

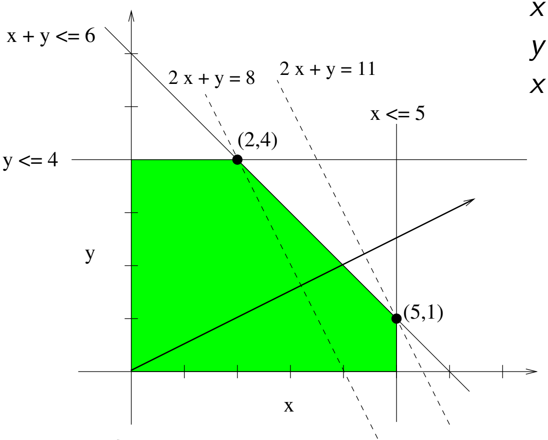
Subject to conditions ("constraints"):

$x + y \leq 6;$

$x \leq 5;$

$y \leq 4;$

$x, y \geq 0;$



Much of this simple "geometric intuition" generalizes nicely to higher dimensions. (But be very careful! Things get complicated very quickly!)

The General Linear Program

Definition: A *Linear Programming* or *Linear Optimization* problem instance $(f, \texttt{Opt}, C)$, consists of:

1. A linear objective function $f : \mathbb{R}^n \mapsto \mathbb{R}$, given by:
   $f(x_1, \ldots, x_n) = c_1\, x_1 + c_2\, x_2 + \ldots + c_n\, x_n + d$
   where we assume the coefficients c_i and constant d are rational numbers.
2. An optimization criterion: $\texttt{Opt} \in \{\texttt{Maximize}, \texttt{Minimize}\}$.
3. A set (or "system") $C(x_1, \ldots, x_n)$ of m *linear constraints*, or linear inequalities/equalities, $C_i(x_1, \ldots, x_n)$, $i = 1, \ldots, m$, where each $C_i(x)$ has form:

   $$a_{i,1}\, x_1 + a_{i,2}\, x_2 + \ldots + a_{i,n}\, x_n \;\Delta\; b_i$$

   where $\Delta \in \{\leq, \geq, =\}$, and where $a_{i,j}$'s and b_i's are rational numbers.

What does it mean to solve an LP?

For a constraint $C_i(x_1, \ldots, x_n)$, we say vector $v = (v_1, \ldots, v_n) \in \mathbb{R}^n$ *satisfies* $C_i(x)$ if, plugging in v for the variables $x = (x_1, \ldots, x_n)$, the constraint $C_i(v)$ holds true.

For example, $(3, 6)$ satisfies $-x_1 + x_2 \leq 7$.

$v \in \mathbb{R}^n$ is called a *solution* to a system $C(x)$, if v satisfies every constraint $C_i \in C$. I.e., $C_1(v) \wedge \ldots \wedge C_m(v)$ is true.

Let $K(C) \subseteq \mathbb{R}^n$ denote the set of all solutions to the system $C(x)$. We say C is **feasible** if $K(C)$ is not empty.
An *optimal solution*, for $\texttt{Opt} = \texttt{Maximize}$, is some $x^* \in K(C)$ such that:

$$f(x^*) = \max_{x \in K(C)} f(x)$$

(respectively, $f(x^*) = \min_{x \in K(C)} f(x)$, for $\texttt{Opt} = \texttt{Minimize}$)).

Given an LP problem $(f, \texttt{Opt}, C)$, our goal in principle is to find an "optimal solution".

What does it mean to solve an LP?

For a constraint $C_i(x_1, \ldots, x_n)$, we say vector $v = (v_1, \ldots, v_n) \in \mathbb{R}^n$ *satisfies* $C_i(x)$ if, plugging in v for the variables $x = (x_1, \ldots, x_n)$, the constraint $C_i(v)$ holds true.

For example, $(3, 6)$ satisfies $-x_1 + x_2 \leq 7$.

$v \in \mathbb{R}^n$ is called a *solution* to a system $C(x)$, if v satisfies every constraint $C_i \in C$. I.e., $C_1(v) \wedge \ldots \wedge C_m(v)$ is true.

Let $K(C) \subseteq \mathbb{R}^n$ denote the set of all solutions to the system $C(x)$. We say C is **feasible** if $K(C)$ is not empty.
An *optimal solution*, for $\texttt{Opt} = \texttt{Maximize}$, is some $x^* \in K(C)$ such that:

$$f(x^*) = \max_{x \in K(C)} f(x)$$

(respectively, $f(x^*) = \min_{x \in K(C)} f(x)$, for $\texttt{Opt} = \texttt{Minimize}$)).

Given an LP problem $(f, \texttt{Opt}, C)$, our goal in principle is to find an "optimal solution". **Oops!!** There may not be an optimal solution!

Things that can go wrong

Two things can go wrong when looking for an optimal solution:

1. There may be no solutions at all!
   I.e., C is not feasible, i.e., $K(C)$ is empty. Consider:

   $$\underline{\textit{Maximize}}\ x$$
   $$\underline{\textit{Subject to:}}\ x \le 3 \textit{ and } x \ge 5$$

2. $\max / \min_{x \in K(C)}\ f(x)$ may not exist (!), because $f(x)$ is <u>unbounded</u> above/below in $K(C)$. Consider:

   $$\underline{\textit{Maximize}}\ x$$
   $$\underline{\textit{Subject to:}}\ x \ge 5$$

So, we have to revise our goals to handle these cases.

Note: If we allowed <u>strict</u> inequalities, e.g., $x < 5$, there would have been yet another problem:

$$\underline{\textit{Maximize}}\ x$$
$$\underline{\textit{Subject to:}}\ x < 5$$

The LP Problem Statement

Given an LP problem instance $(f, \texttt{Opt}, C)$ as input, output one of the following three:

1. "The problem is Infeasible."
2. "The problem is Feasible But Unbounded."
3. "An Optimal Feasible Solution (OFS) exists.
   One such optimal solution is $x^* \in \mathbb{R}^n$.
   The optimal objective value is $f(x^*) \in \mathbb{R}$."

Oops!! It seems yet another thing could go wrong: What if every optimal solution $x^* \in \mathbb{R}^n$ is irrational?
How can we "output" irrational numbers?
Likewise, what if the Opt value $f(x^*)$ is irrational?

Fact: This problem never arises. The above three answers cover all possibilities, and furthermore, as long as all our coefficients and constants are rational, if an OFS exists, a rational OFS x^* exists, and the optimal value $f(x^*)$ is also rational. (We will learn why later.)

Simplified forms for LP problems

1. In principle, we need only consider Maximization, because:
$$\min_{x \in K} f(x) = -\max_{x \in K} -f(x)$$
(either side is unbounded if and only if both are.)
2. We only need an objective function $f(x_1, \ldots, x_n) = x_i$, for some x_i, because we can:
Introduce new variable x_0. Add new constraint $f(x) = x_0$ to constraints C. Make the new objective "Optimize x_0".
3. Don't need "$=$" constraints: $\alpha = \beta \Leftrightarrow (\alpha \leq \beta \wedge \alpha \geq \beta)$.
4. Don't need "$\alpha \geq b$", where $b \in \mathbb{R}$: $\alpha \geq b \Leftrightarrow -\alpha \leq -b$.
5. We can constrain every variable x_i to be $x_i \geq 0$:
Introduce two variables x_i^+, x_i^- for each variable x_i.
Replace each occurence of x_i by $(x_i^+ - x_i^-)$, and add the constraints $x_i^+ \geq 0$, $x_i^- \geq 0$.
(**N.B.** can't do both (2.) and (5.) together.)

A lovely but terribly inefficient algorithm for LP

Input: LP instance $(x_0, \mathtt{Opt}, C(x_0, x_1, \ldots, x_n))$.

1. **For** $i = n$ downto 1
   a. Rewrite each constraint involving x_i as $\alpha \leq x_i$, or as $x_i \leq \beta$. (One of the two is possible.) Let these be:
   $\alpha_1 \leq x_i, \ldots, \alpha_k \leq x_i$; $x_i \leq \beta_1, \ldots, x_i \leq \beta_r$
   (Retain these constraints, H_i, for later.)
   b. Remove H_i, i.e., all constraints involving x_i. Replace with constraints: $\{\alpha_j \leq \beta_l \mid j = 1, \ldots, k, \& l = 1, \ldots, r\}$.
2. Only x_0 (or no variable) remains. All constraints have the forms $a_j \leq x_0$, $x_0 \leq b_l$, or $a_j \leq b_l$, where a_j's and b_l's are constants. It's easy to check "feasibility" & "boundedness" for such a one(or zero)-variable LP, and to find an optimal x_0^* if one exists.
3. Once you have x_0^*, plug it into H_1. Solve for x_1^*. Then use x_0^*, x_1^* in H_2 to solve for x_2^*, ..., use $x_0^*, \ldots, x_{i-1}^*$ in H_i to solve for x_i^*. ... then $x^* = (x_0^*, \ldots, x_n^*)$ is an optimal feasible solution.

remarks on the lovely algorithm

- This algorithm was first discovered by Fourier (1826). Rediscovered in 1900's, by Motzkin (1936) and others.
- It is called *Fourier-Motzkin Elimination*, and can be viewed as a generalization of *Gaussian Elimination*, used for solving systems of linear equalities.
- Why is Fourier-Motzkin so inefficient? In the worst case, if every variable x_i is involved in every constraint, each iteration of the "For loop" squares the number of constraints. So, toward the end we could have roughly m^{2^n} constraints!!
- Let's recall Gaussian Elimination (GE). It is much nicer and does not suffer from this explosion.
- In 1947, Dantzig invented the celebrated **Simplex Algorithm** for LP. It can be viewed as a much more refined generalization of GE. Next time, Simplex!

more remarks

Immediate Corollaries of Fourier-Motzkin:

Corollary 1: The three possible “answers” to an LP problem do cover all possibilities.
(In particular, unlike “Maximize x; $x < 5$”, If an LP has a “Supremum” it has a “Maximum”.)

Corollary 2: If an LP has an OFS, then it has a rational OFS, x^*, and $f(x^*)$ is also rational.
Proof: We used only addition, multiplication, & division by rationals to arrive at the solution. □

further remarks

Although Fourier-Motzkin is bad in the worst case, it can still be quite useful. It can be used to remove redundant variables and constraints. And its worst-case behavior may in some cases not arise in practice.

Generalizations of Fourier-Motzkin are used in some tools (e.g., [Pugh,'92]) for solving "Integer Linear Programming", where we seek an optimal solution x^* not in $\mathbb{R}^n$, but in $\mathbb{Z}^n$. ILP is a **much harder** problem! (**NP**-complete.)

For ordinary LP however, Fourier-Motzkin can't compete with Simplex.

- **Food for Thought:** Think about what kinds of clever heuristics and hacks you could use during Fourier-Motzkin to keep the number of constraints as small as possible. E.g., In what order would you try to eliminate variables? (Clearly, any order is fine, as long as x_0 is last.)