CSE520: Computational Geometry
Lecture 12
Introduction to Linear Programming

Antoine Vigneron

Ulsan National Institute of Science and Technology

April 18, 2020

Outline

- Assignment 3 will be posted on Monday next week, due on Friday.
- No lecture next week (midterm week).

- This lecture is an introduction to linear programming.
- In next lecture, we will see an efficient algorithm for *fixed* dimension.
- It means $d = O(1)$ variables.
- Even without this restriction, there are polynomial-time algorithms.

Reference:

- Textbook Chapter 4.
- Dave Mount's lecture notes, lectures 8–10.

First Example

- A factory can make two types of products: X and Y
- A product of type X requires 10 hours of manpower, 4ℓ of oil, and $5m^3$ of storage.
- A product of type Y requires 8 hours, 2ℓ of oil and $10m^3$ storage.
- A product X can be sold \$200 and a product Y can be sold \$250.
- You have 168 hours of manpower available, as well as 60ℓ of oil and $150m^3$ of storage.
- How many products of each type should you make so as to maximize their total price?

Formulation

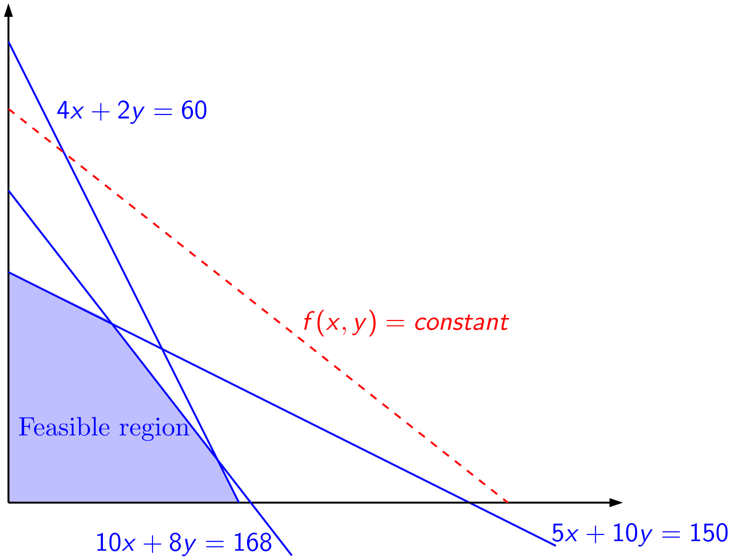
- x and y denote the number of products of type X and Y, respectively.
- Maximize the price

$$f(x,y) = 200x + 250y$$

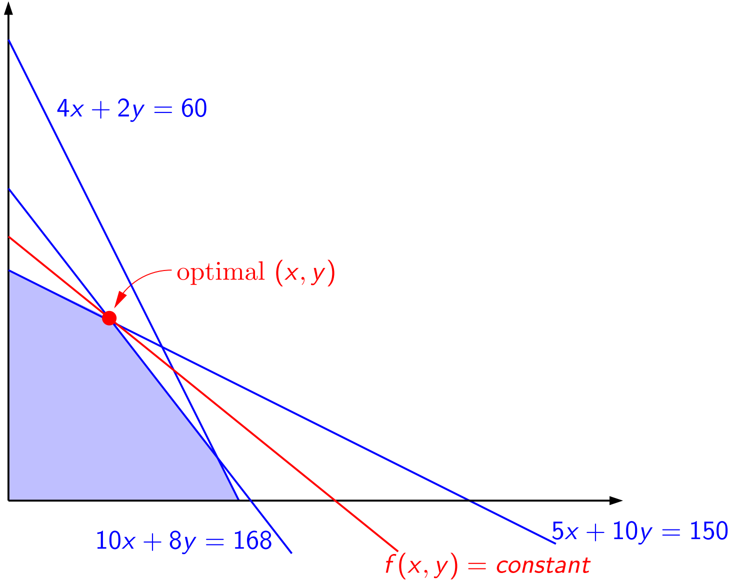
under the *constraints*

$$\begin{array}{rcrcr} -x & & & \leqslant & 0 \\ & & -y & \leqslant & 0 \\ 10x & + & 8y & \leqslant & 168 \\ 4x & + & 2y & \leqslant & 60 \\ 5x & + & 10y & \leqslant & 150. \end{array}$$

Geometric Interpretation



Geometric Interpretation



Solution

- From previous slide, at the optimum:
  - $x = 8$
  - $y = 11$.
- Luckily these are integers.
- So it is the solution to our problem.
- If we add the constraint that all variables are integers, we are doing *integer programming*.
  - We do not deal with it in CSE520.
  - We consider only linear inequalities, no other constraint.
- Our example was a special case where the linear program has an integer solution, hence it is also a solution to the integer program.

Problem Statement

- Maximize the *objective function*

$$f(x_1, x_2, \ldots, x_d) = c_1x_1 + c_2x_2 + \cdots + c_dx_d$$

subject to the *constraints*

$$\begin{array}{ccccc} a_{11}x_1 & + \cdots + & a_{1d}x_d & \leqslant & b_1 \\ a_{21}x_1 & + \cdots + & a_{2d}x_d & \leqslant & b_2 \\ \vdots & & \vdots & & \vdots \\ a_{n1}x_1 & + \cdots + & a_{nd}x_d & \leqslant & b_n \end{array}$$

- This is linear programming in dimension d.

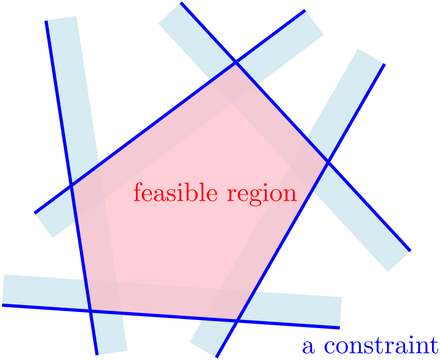
Problem Statement

- The goal could be to minimize f, because minimizing $f(x)$ is equivalent to maximizing $-f(x)$.
- Inequalities could be $\geqslant$ instead of $\leqslant$, because $a_{i1}x_1 + \cdots + a_{id}x_d \geqslant b_d$ is equivalent to $-a_{i1}x_1 - \cdots - a_{id}x_d \leqslant -b_d$
- So this is also a 3-dimensional linear program:
- Minimize $f(x_1, x_2, x_3) = 2x_1 - 3x_2 + 4x_3$ subject to

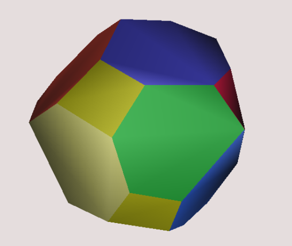
$$
\begin{array}{rcrcrcr}
2x_1 & + & x_2 & + & x_3 & \geqslant & 3 \\
5x_1 & - & x_2 & - & x_3 & \leqslant & -5 \\
-x_1 & + & 2x_2 & + & 3x_3 & \geqslant & 7 \\
3x_1 & + & x_2 & - & x_3 & \leqslant & 2
\end{array}
$$

Geometric Interpretation

- Each constraint represents a half-space in $\mathbb{R}^d$.
- The intersection of these half-spaces forms the *feasible region*.
- The feasible region is a *convex polyhedron* in $\mathbb{R}^d$.



Convex Polyhedra

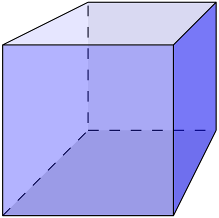


Definition (Convex polyhedron)

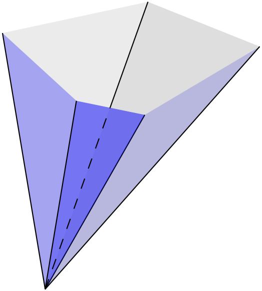
A convex polyhedron is the intersection of a finite number of half-spaces in $\mathbb{R}^d$.

- May also be called *convex polytope*.
- A convex polyhedron is not necessarily bounded.
- Special case: A convex polygon is a a bounded, convex polytope in $\mathbb{R}^2$.

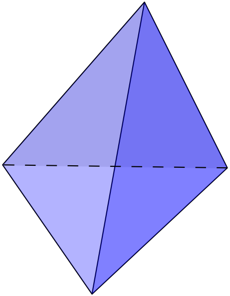
Convex Polyhedra in $\mathbb{R}^3$



a cube



a cone

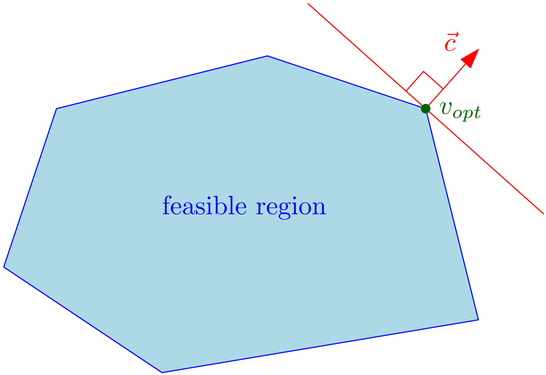


a tetrahedron

- Faces of a convex polyhedron in $\mathbb{R}^3$:
  - Vertices, edges and facets.
  - Example: A cube has 8 vertices, 12 edges and 6 facets.

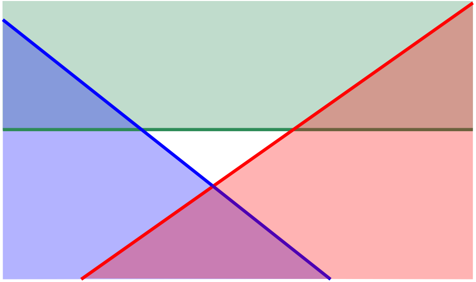
Geometric Interpretation

- Let $\overrightarrow{c} = (c_1, c_2, \dots, c_d)$.
- We want to find a point v_{opt} of the feasible region such that $\overrightarrow{c}$ is an outer normal at v_{opt}, if there is one.



Infeasible Linear Programs

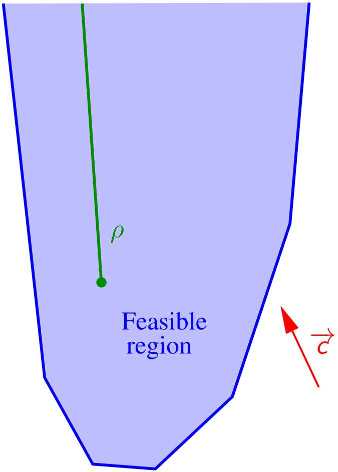
- The feasible region may be empty.



- In this case there is no solution to the linear program.
- The program is said to be *infeasible*.
- We would like to know when it is the case.

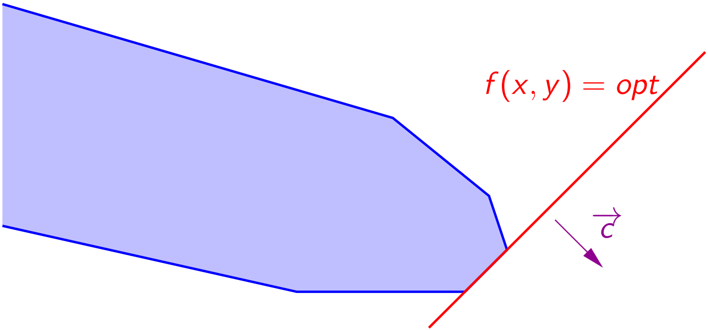
Unbounded Linear Programs

- The feasible region may be unbounded in the direction of $\overrightarrow{c}$.
- In this case, we say that the linear program is *unbounded*.
- Then we want to find a ray ρ in the feasible region along which f takes arbitrarily large values.



Degenerate Cases

- A linear program may have an infinite number of optimal solutions.

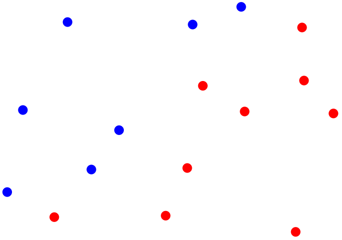


- In this case, we report only one solution.

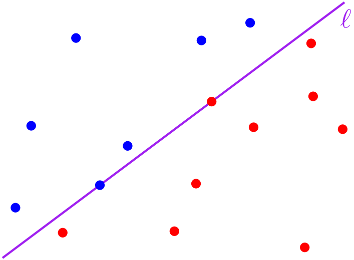
More Examples

- In practice many optimization problems are linear programs. (For instance in engineering, operations research)
- We now give two geometric applications of linear programming.
- With the algorithm presented in next lecture, they can be solved in *linear* time.

Separating Two Point Sets



input points



separating line ℓ

Problem (separating line)

Given a set $B = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ of m blue points and a set $R = \{(u_1, v_1), \ldots, (u_n, v_n)\}$ of n red points, does there exist a line ℓ such that B is on one side of ℓ and R is on the other side?

Separating Two Point Sets

- Without loss of generality, we assume that B is above ℓ and R is below. Let ℓ have equation $y = ax + b$.
- Then ℓ is a solution to our problem iff

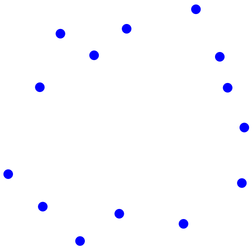
$$\begin{aligned} y_i &\geqslant ax_i + b && \text{for all } 1 \leqslant i \leqslant m, \text{ and} \\ v_i &\leqslant au_i + b && \text{for all } 1 \leqslant i \leqslant n. \end{aligned}$$

- We can rewrite it

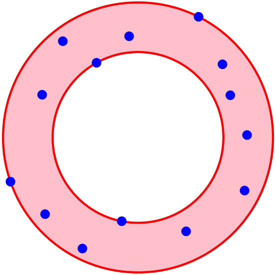
$$\begin{aligned} ax_i + b &\leqslant y_i && \text{for all } 1 \leqslant i \leqslant m, \text{ and} \\ -au_i - b &\leqslant -v_i && \text{for all } 1 \leqslant i \leqslant n. \end{aligned}$$

- This is a set of $m + n$ linear constraints on the two variables a and b.
- If we add an arbitrary objective function such as $f(a, b) = a$, we obtain a two-dimensional linear program.
- A line is a feasible solution to this program iff it is a separating line.
- So we just solve the separating line problem in $O(m + n)$ time by linear programming.

Smallest Enclosing Annulus



input points



smallest enclosing annulus

Problem (smallest enclosing annulus)

Given a set of n points $\{(x_1, y_1), \ldots, (x_n, y_n)\}$*, find a minimum area annulus that contains it.*

Smallest Enclosing Annulus

- We denote by (a, b) the center of the annulus, and by $r \leqslant R$ its two radiuses.
- So a point (x, y) is in the annulus iff

$$r^2 \leqslant (x-a)^2 + (y-b)^2 \leqslant R^2.$$

- The area of the annulus is $\pi(R^2 - r^2)$.
- So the problem can be reformulated as: Find a, b, r, R such that

$$r^2 \leqslant (x_i - a)^2 + (y_i - b)^2 \leqslant R^2 \quad \text{for all } i$$

  and $R^2 - r^2$ is minimum.

Smallest Enclosing Annulus

- It can be rewritten: Minimize $R^2 - r^2$ subject to

$$2x_i a + 2y_i b + r^2 - a^2 - b^2 \leqslant x_i^2 + y_i^2$$
$$2x_i a + 2y_i b + R^2 - a^2 - b^2 \geqslant x_i^2 + y_i^2$$

- We now introduce the change of variable $z = r^2 - a^2 - b^2$ and $t = R^2 - a^2 - b^2$. The problem becomes: Minimize $t - z$ subject to

$$2x_i a + 2y_i b + z \leqslant x_i^2 + y_i^2$$
$$2x_i a + 2y_i b + t \geqslant x_i^2 + y_i^2$$

- This is a 4-dimensional linear program with $2n$ constraints and variables a, b, z, t.

Smallest Enclosing Annulus

- We first solve this problem in $O(n)$ time by linear programming.
- Then we let $r = \sqrt{z + a^2 + b^2}$ and $R = \sqrt{t + a^2 + b^2}$.
- Conclusion: we can find a smallest enclosing annulus in *linear* time.
- The approach we used here was to *linearize* the problem:
- We started with a problem whose constraints and objective functions were non-linear.
- We made them linear using a change of variable.
- This approach does not apply to all optimization problems, but it is often worth trying.