CSE515: Advanced Algorithms
Lecture 8: Introduction to Linear Programming

Antoine Vigneron

Ulsan National Institute of Science and Technology

March 25, 2021

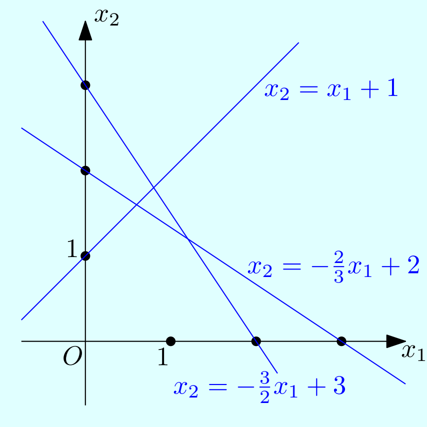
Introduction

- I will post assignment 1 tonight. It will be due in one week.
- Linear Programming is an important topic in optimization.
- It can be solved in polynomial time.
- In CSE515, I will present:
  - A practical algorithm, which runs in worst-case exponential time: The simplex algorithm.
  - A polynonial-time algorithm.
- In CSE520 Computational Geometry, I also give a very fast algorithm for linear programming with few variables.

Reference:

- Chapter 29 in Introduction to Algorithms by Cormen, Leiserson, Rivest and Stein.

Example

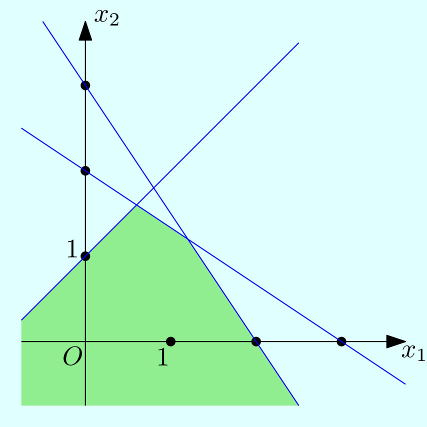


We want to find a point $(x_1, x_2) \in \mathbb{R}^2$ such that $x_1 + 2x_2$ is maximized and such that:

$$x_2 \leqslant x_1 + 1,$$
$$x_2 \leqslant -\frac{2}{3}x_1 + 2, \text{ and}$$
$$x_2 \leqslant -\frac{3}{2}x_1 + 3.$$

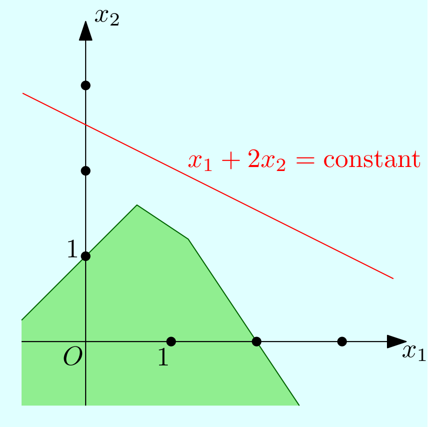
The inequalities are called the *constraints*, and $x_1 + 2x_2$ is the *objective function*.

Example



The optimal solution x^* should lie in the green region. This region is the *feasible region* of this linear program.

Example



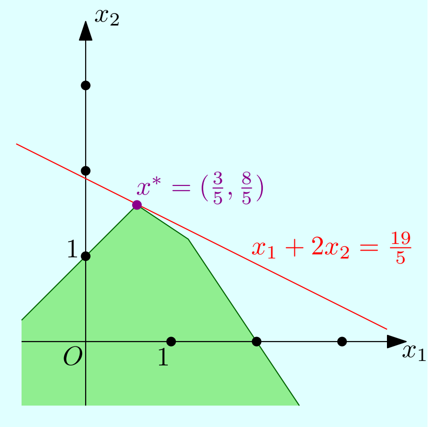
How to find the location of x^* in the feasible region?

Translate the line

$$x_1 + 2x_2 = \text{constant}$$

downward until it hits the feasible region. (See next slide.)

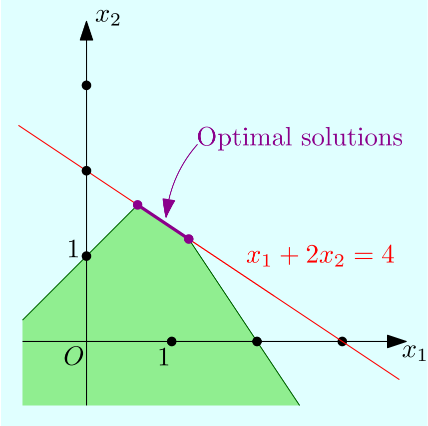
Example



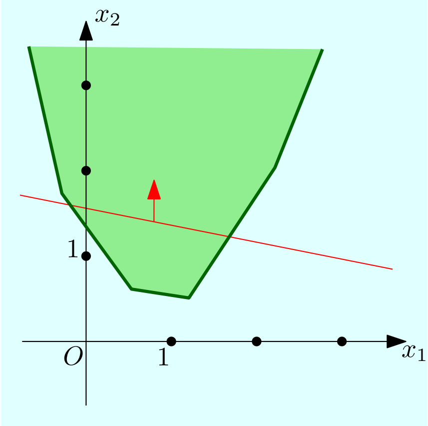
The optimal solution to our linear program is $x^* = \left(\frac{3}{5}, \frac{8}{5}\right)$.

The optimal solution here is unique. But a linear program could have no solution (if the feasible region is empty), an infinity of optimal solutions, or it could be unbounded (arbitrary large solutions).
See next slide.

Example



There are infinitely many optimal solutions, represented by the purple segment.



The objective function can take arbitrarily large values in the feasible region, so there is no optimal point.

General Formulation

A *linear program* (LP) can be written as follows:

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{n} c_j x_j \\ \text{subject to} \quad & \sum_{j=1}^{n} a_{ij} x_j \leqslant b_i, \qquad i = 1, \ldots, m \end{aligned}$$

Here, the mn real numbers a_{ij} are given, as well as the c_j's and the b_i's. The goal is to find the optimal value $x^* = (x_1^*, \ldots, x_n^*)$.

General Formulation

The vector $x = (x_1, \dots, x_n)$ is called the *variable*.

The *objective function* is the function $f : \mathbb{R}^n \to \mathbb{R}$ such that $f(x) = \sum_{j=1}^{n} c_j x_j$ for all x.

A *constraint* is an inequality $\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i$.

A point x that satisfies all the m constraints is a *feasible solution*.

- Otherwise, x is *infeasible*.
- The set of feasible points is the *feasible region*.

If there exists a feasible solution, the LP is said to be *feasible*. Otherwise the LP is *infeasible*.

General Formulation

In the formulation above, we can replace *maximize* with *minimize*, because maximizing $\sum_j c_j x_j$ is equivalent to minimizing $\sum_j (-c_j) x_j$.

We can also replace $\leqslant$ with $\geqslant$, because $\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i$ is equivalent to $\sum_{j=1}^{n} (-a_{ij}) x_j \geqslant -b_i$.

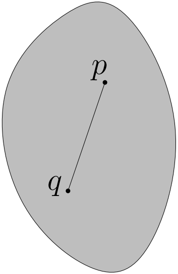
So this is a linear program:

$$\begin{array}{ll} \text{minimize} & x_1 - x_2 + 2x_3 \\ \text{subject to} & x_1 + x_2 + x_3 \leqslant 3, \\ & -x_1 + x_2 - x_3 \geqslant 4, \\ & x_1 + 3x_2 + x_3 = -2, \\ & x_1 - 3x_2 - x_3 \leqslant 0. \end{array}$$

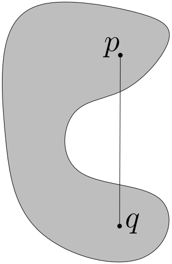
Convexity

Definition

A set $\mathcal{C} \subset \mathbb{R}^d$ is *convex* iff $\forall p, q \in \mathcal{C}$ the line segment $\overline{pq}$ is contained in $\mathcal{C}$.
In other words, $\lambda p + (1 - \lambda) q \in \mathcal{C}$ for all $p, q \in \mathcal{C}$ and $0 \leqslant \lambda \leqslant 1$.



Convex



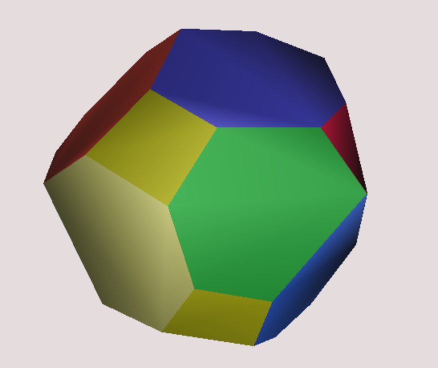
Non convex

- The intersection of a collection of convex sets is convex (proof?).

Geometry

- A constraint $a_{i1}x_1 + \cdots + a_{in}x_n \leqslant b_i$ corresponds to a halfspace in $\mathbb{R}^n$, where the coordinates are $x_1, \ldots, x_n$.
- The feasible region is the intersection of these halfspaces.
- A halfspace is convex. (Proof?)
- So the feasible region is convex.
- More precisely, it is a convex polytope.
- Intuitively, it explains why the problems is tractable: There is only one locally optimum region.

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



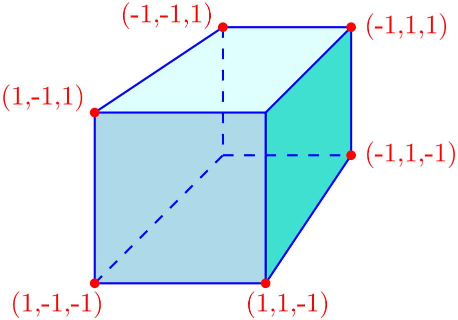
A convex polytope is an intersection of halfspaces

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

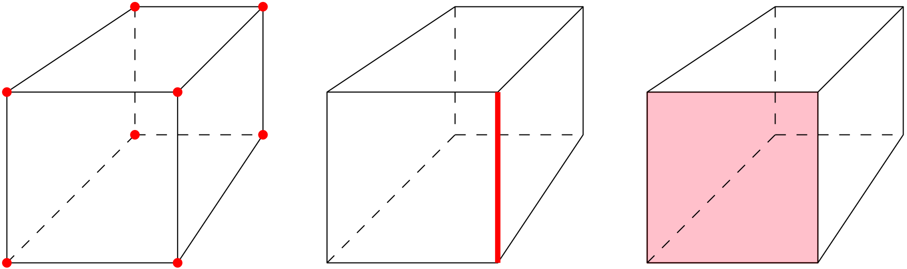
- For instance, the intersection of these 6 halfspaces

$$x \leqslant 1$$
$$x \geqslant -1$$
$$y \leqslant 1$$
$$y \geqslant -1$$
$$z \leqslant 1$$
$$z \geqslant -1$$

is a cube.



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



vertices | an edge | a facet

- Faces of a convex polytope in $\mathbb{R}^3$:
  - Vertices, edges and facets.

Example

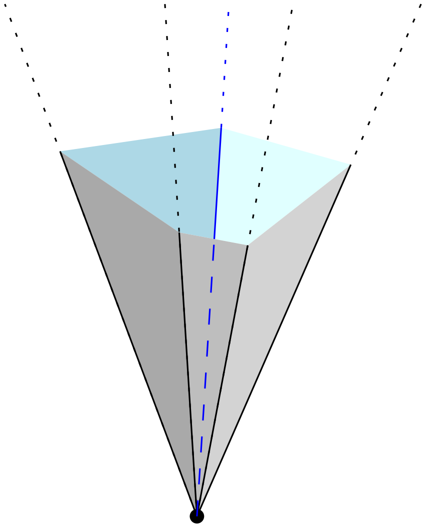
A cube has 8 vertices, 12 edges and 6 facets.

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

- A convex polytope can be unbounded.

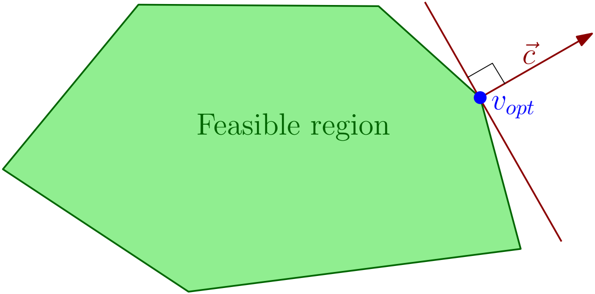
Example

A cone is unbounded.



Geometric Interpretation of Linear Programming

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

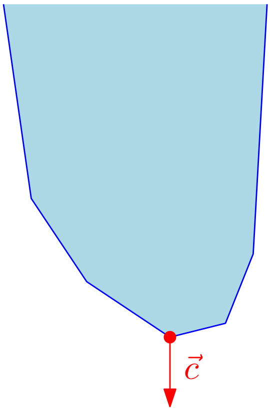


Geometric Interpretation of Linear Programming

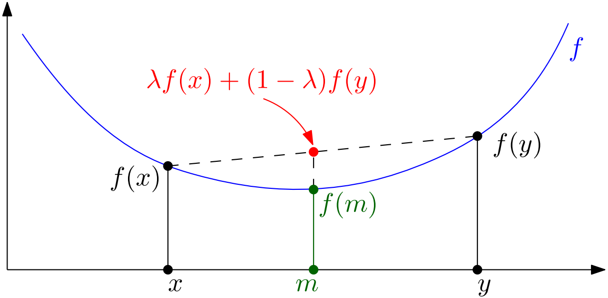
- If the LP is bounded, then (at least) one vertex of the feasible region is optimal.
- Each vertex is at the intersection of the halfplanes bounding n independent constraints.
- So we could solve the LP by going through all the vertices.
- Problem: There can be an exponential number of vertices.
  - Example?

Geometric Interpretation of Linear Programming

- After changing the basis, we can ensure that $\vec{c} = (c_1, c_2, \ldots, c_d) = (0, 0, \ldots, 0, -1)$
- So we want to find the lowest point in the feasibility region.
- We want to minimize a *convex function*.



Convex Functions



$m = \lambda x + (1-\lambda)y$

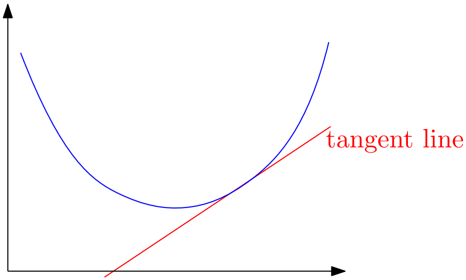
Definition

Let $\mathcal{D}$ be a subset of $\mathbb{R}^d$. We say that a function $f : \mathcal{D} \to \mathbb{R}$ is *convex* iff

$$f(\lambda x + (1-\lambda)y) \leqslant \lambda f(x) + (1-\lambda)f(y)$$

for all $x, y \in \mathcal{D}$ and $0 \leqslant \lambda \leqslant 1$.

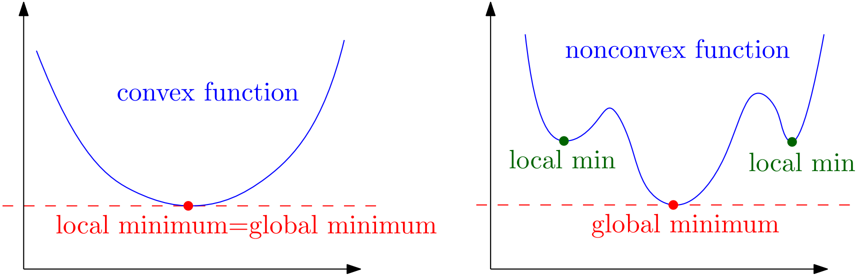
Convex Functions



- A convex function is above its tangent lines.
- In dimension $d \geqslant 2$, it is above its tangent planes/hyperplanes.

Convex Functions

- Any *local minimum* of a convex function is also a *global minimum*.



Convex Functions

- When the function is not convex, there may be many different local minima.
- Typically, this number is exponential in the dimension.
- This is why it is often difficult to minimize a nonconvex function.
- On the other hand, there are efficient algorithms for *Convex optimization*: minimizing a convex function over a convex set.
- Reason: We cannot get "stuck" in a local minimum.
- LP is a special case of convex optimization, so it is not surprising that it can be solved in polynomial time.

Example 1: Elections

Policy	Urban	Suburban	Rural
build roads	-2	5	3
gun control	8	2	-5
farm subsidies	0	0	10
gasoline tax	10	0	-2

- Interpretation:
  - $ 1,000 spent on advertising to building roads loses 2,000 urban votes, and earns 5,000 suburban votes and 3,000 rural votes.
- The goal is to earn at least 50,000 urban votes, 100,000 suburban votes and 25,000 rural votes, while spending a minimum amount of money.

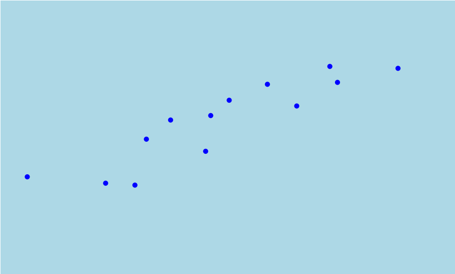
Exercise

Formulate this problem as a linear program.

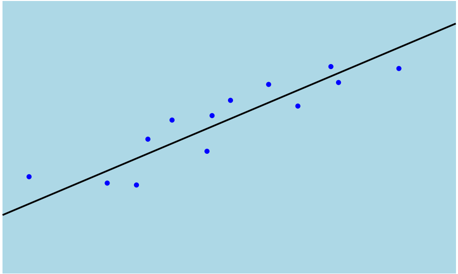
Example 2: Fitting a line

Problem

Find a best fitting line to a 2D point set.



INPUT

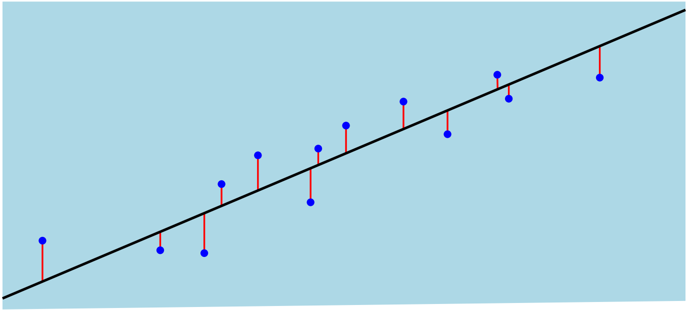


OUTPUT

- Input: points $(x_1, y_1), \ldots, (x_n, y_n)$
- Output: line $y = ax + b$

Example 2: Fitting a line

- Several criteria are possible.
- We use the following: Minimize the sum of the vertical distances.



- This is the sum of the red length in the figure.
- So we want to minimize $\sum_{i=1}^{n} |ax_i + b - y_i|$.

Example 2: Fitting a line

Exercise

Formulate this line-fitting problem as a linear program.

More Examples

Formulate the two problems below as linear programs:

Problem (Shortest path)

Given a directed graph $G(V, E)$*, each edge* $(u, v) \in E$ *having a length* $\ell(u, v) \geqslant 0$*, and two vertices* $s, t \in V$*, find the shortest path length from* s *to* t*.*

Problem (Maximum flow)

See Lecture 1.

Conclusion

- LPs are routinely solved in practice. Try for instance this online solver on Example 1: Simplex method tool.
- Maximum flow and shortest path can be solved through LPs, instead of specialized algorithms (i.e. Edmonds-Karp, Dijkstra).
  - Specialized algorithms are better in theory (better time bounds)
  - In practice it may not be the case.
- If we add to a LP the constraints that $x_1, \dots, x_n \in \mathbb{Z}$, it becomes an *integer program*.
  - Integer programming is **NP**-complete.
- If we allow a quadratic objective function, it becomes a *quadratic program*. This is **NP**-hard in general.
- Quadratic constraints also make the problem **NP**-hard.