Introduction to Linear and Combinatorial Optimization (ADM I)

Martin Skutella

TU Berlin

Winter 2019/20

Chapter 1: Introduction

Optimization Problems

Generic optimization problem

Given: set X, function $f : X \to \mathbb{R}$
Task: find $x^* \in X$ maximizing (minimizing) $f(x^*)$, i.e.,

$$f(x^*) \geq f(x) \quad \big(f(x^*) \leq f(x)\big) \qquad \text{for all } x \in X.$$

- An x^* with these properties is called optimal solution (optimum).
- Here, X is the set of feasible solutions, f is the objective function.

Short form:

$$\begin{aligned} \text{maximize} \quad & f(x) \\ \text{subject to} \quad & x \in X \end{aligned}$$

or simply: $\max\big\{f(x) \mid x \in X\big\}$.

Problem: Too general to say anything meaningful!

Convex Optimization Problems

Definition 1.1.

Let $X \subseteq \mathbb{R}^n$ and $f : X \to \mathbb{R}$.

a X is convex if for all $x, y \in X$ and $0 \leq \lambda \leq 1$ it holds that

$$\lambda \cdot x + (1 - \lambda) \cdot y \in X.$$

b f is convex if X is convex and for all $x, y \in X$ and $0 \leq \lambda \leq 1$,

$$\lambda \cdot f(x) + (1 - \lambda) \cdot f(y) \geq f\big(\lambda \cdot x + (1 - \lambda) \cdot y\big).$$

c If f is convex, $\min\big\{f(x) \mid x \in X\big\}$ is a convex optimization problem.

Note: $f : X \to \mathbb{R}$ is called concave if $-f$ is convex.

Local and Global Optimality

Definition 1.2.

Let $X \subseteq \mathbb{R}^n$ and $f : X \to \mathbb{R}$.
$x' \in X$ is a local optimum of the optimization problem $\min\{f(x) \mid x \in X\}$ if there is an $\varepsilon > 0$ such that

$$f(x') \le f(x) \qquad \text{for all } x \in X \text{ with } \|x' - x\|_2 \le \varepsilon.$$

Theorem 1.3.

For a convex optimization problem, every local optimum x is a (global) optimum.

Proof: Assume that $f(x^*) < f(x)$. Then for each $\lambda \in (0, 1]$ we get:

$$f\big(\lambda \cdot x^* + (1-\lambda)\cdot x\big) \le \lambda \cdot f(x^*) + (1-\lambda)\cdot f(x) < f(x).$$

But $\lambda \cdot x^* + (1-\lambda)\cdot x$ converges to x for $\lambda \to 0$, a contradiction! $\square$

Optimization Problems Considered in this Course:

$$\begin{array}{ll} \text{maximize} & f(x) \\ \text{subject to} & x \in X \end{array}$$

- $X \subseteq \mathbb{R}^n$ polyhedron, f linear function
  $\longrightarrow$ linear optimization problem (in particular convex)
- $X \subseteq \mathbb{Z}^n$ integer points of a polyhedron, f linear function
  $\longrightarrow$ integer linear optimization problem
- X related to some combinatorial structure (e. g., graph)
  $\longrightarrow$ combinatorial optimization problem
- X finite (but usually huge) or countably infinite
  $\longrightarrow$ discrete optimization problem

Discrete Optimization Problem: Trivial Solution Strategy

Given: finite set X (feasible solutions), objective function $f : X \to \mathbb{R}$

Task: find $x \in X$ maximizing (or minimizing) $f(x)$

Historical remark by William R. Pulleyblank about the 1960's:

Problems were finite or infinite, and once a problem was known to be finite there were no algorithmic questions to be asked because it was all over.

I remember when I took my first combinatorics class from the distinguished combinatorialist Eric Milner, and there was a point where we were talking about a theorem, and I said *"how would you find one of these?"*.

And he looked at me with a kind look, but the sort of look a parent gives a child when he says something sort of stupid.

He simply said *"But Bill, it's finite"*, and I said *"Oh! of course, it's finite."*

[From: M. Jünger et al.: Combinatorial Optimization (Edmonds Festschrift), 2003]

Discrete Optimization Problem: Trivial Solution Strategy

Given: finite set X (feasible solutions), objective function $f : X \to \mathbb{R}$

Task: find $x \in X$ maximizing (or minimizing) $f(x)$

Trivial solution strategy

1. choose some $x_0 \in X$;
2. for all $x \in X$: if $f(x) > f(x_0)$, then $x_0 := x$;
3. output x_0;

Running time: $O(|X| \cdot F)$ where F is time to evaluate f at $x \in X$

Problem

Usually, X is not explicitly but only implicitly given.
$\Longrightarrow$ $|X|$ might be huge (exponential) compared to input size.

Minimum Spanning Tree (MST) Problem

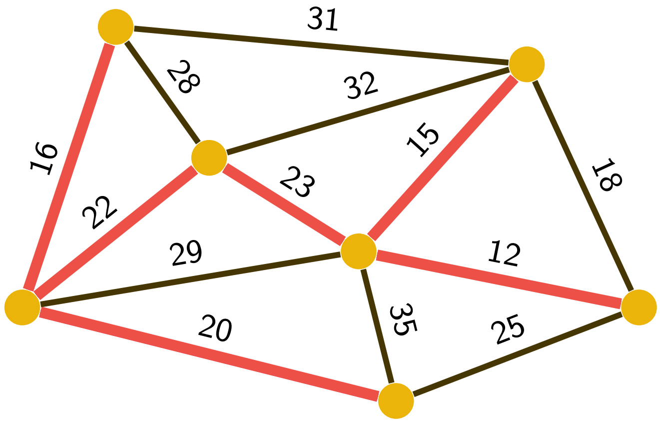
Given: undirected graph $G = (V, E)$, cost function $c : E \to \mathbb{R}_{\geq 0}$;

Task: find connected subgraph of G containing all nodes in V with minimum total cost

That is, $X = \{E' \subseteq E \mid E' \text{ connects all nodes in } V\}$ and $f : X \to \mathbb{R}$ is given by:

$$f(E') := \sum_{e \in E'} c(e)$$

Example:



Minimum Spanning Tree (MST) Problem

Given: undirected graph $G = (V, E)$, cost function $c : E \to \mathbb{R}_{\geq 0}$;

Task: find connected subgraph of G containing all nodes in V with minimum total cost

That is, $X = \{E' \subseteq E \mid E' \text{ connects all nodes in } V\}$ and $f : X \to \mathbb{R}$ is given by:

$$f(E') := \sum_{e \in E'} c(e)$$

Remarks.

- Notice that there always exists an optimal solution without cycles.
- A connected graph without cycles is called a tree.
- A subgraph of G containing all nodes in V is called spanning.

Theorem 1.4 (Cayley's formula).

The number of spanning trees of a complete graph on n nodes is n^{n-2}. □

Shortest Path Problem

Given: directed graph $D = (V, A)$, arc costs c_a, $a \in A$,
start node $s \in V$, destination node $t \in V$;

Task: find s-t-path of minimum cost in D (if one exists)

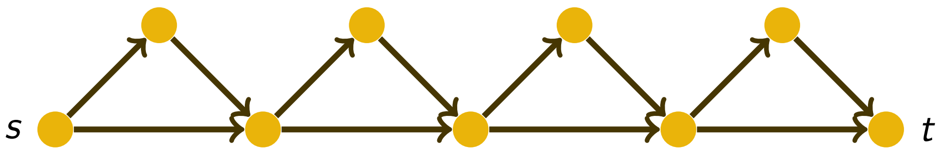
That is, $X = \{P \subseteq A \mid P \text{ is } s\text{-}t\text{-path in } D\}$ and $f : X \to \mathbb{R}$ is given by:

$$f(P) := \sum_{a \in P} c_a$$

Remark.

Note that the finite set of feasible solutions X is only implicitly given by D. This holds for all interesting problems in combinatorial optimization!

Example: digraph with $2n + 1$ nodes, $3n$ arcs, and 2^n s-t-paths.

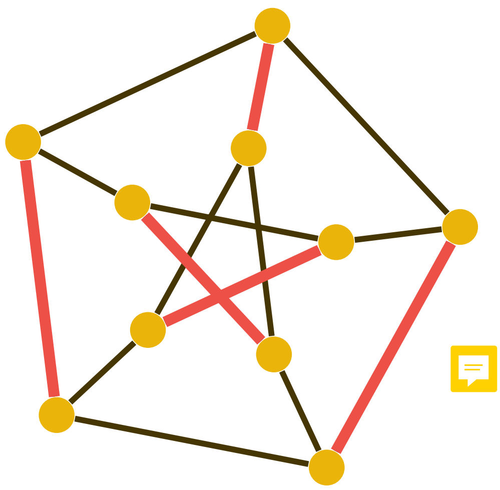


Maximum Weighted Matching Problem

Given: undirected graph $G = (V, E)$, weight function $w : E \to \mathbb{R}$.

Task: find matching $M \subseteq E$ with maximum total weight.

($M \subseteq E$ is a matching if every node is incident to at most one edge in M.)



Maximum Weighted Matching Problem

Given: undirected graph $G = (V, E)$, weight function $w : E \to \mathbb{R}$.

Task: find matching $M \subseteq E$ with maximum total weight.

($M \subseteq E$ is a matching if every node is incident to at most one edge in M.)

Formulation as an integer linear program (IP):

Variables: $x_e \in \{0, 1\}$ for $e \in E$ with interpretation: $x_e = 1 \iff e \in M$

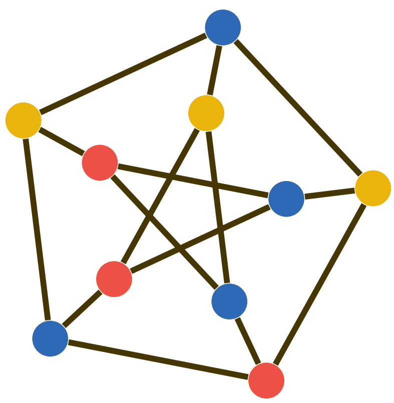
$$
\begin{array}{lll}
\text{maximize} & \sum_{e \in E} w(e) \cdot x_e & \\
\text{subject to} & \sum_{e \in \delta(v)} x_e \leq 1 & \text{for all } v \in V, \\
 & x_e \in \{0, 1\} & \text{for all } e \in E.
\end{array}
$$

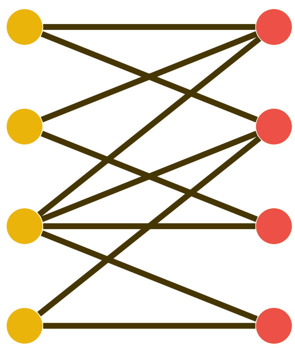
IP: Linear program where all variables may only take integer values.

Minimum Node Coloring Problem

Given: undirected graph $G = (V, E)$

Task: color the nodes of G such that adjacent nodes get different colors; use a minimum number of colors



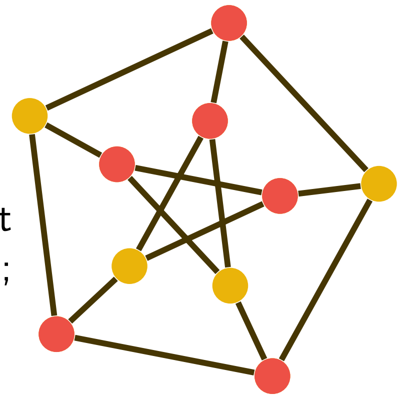


bipartite

Definition 1.5.

A graph whose nodes can be colored with two colors is called bipartite.

Minimum Weighted Node Cover Problem



Given: undirected graph $G = (V, E)$,
weight function $w : V \to \mathbb{R}_{\geq 0}$

Task: find $U \subseteq V$ of minimum total weight such that every edge $e \in E$ has at least one endpoint in U; such a set U is called node cover.

Formulation as an integer linear program (IP):

Variables: $x_v \in \{0, 1\}$ for $v \in V$ with interpretation: $x_v = 1 \iff v \in U$

$$
\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{v \in V} w_v \cdot x_v & \\
\text{subject to} & x_v + x_{v'} \geq 1 & \text{for all } e = \{v, v'\} \in E, \\
 & x_v \in \{0, 1\} & \text{for all } v \in V.
\end{array}
$$

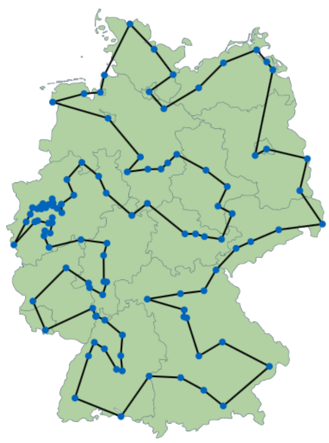
Traveling Salesperson Problem (TSP)

Given: complete graph K_n on n nodes, length function $\ell : E(K_n) \to \mathbb{R}$;

Task: find a Hamiltonian circuit of minimum total length

(A Hamiltonian circuit visits every node exactly once.)

Examples:



Formulation as an integer linear program? (later!)

Minimum Cost Flow Problem

Given: directed graph $D = (V, A)$, with arc capacities $u : A \to \mathbb{R}_{\geq 0}$, arc costs $c : A \to \mathbb{R}$, and node balances $b : V \to \mathbb{R}$

Interpretation:

- nodes $v \in V$ with $b(v) > 0$ ($b(v) < 0$) have demand (supply) and are called sinks (sources)
- the capacity $u(a)$ of arc $a \in A$ limits the amount of flow that can be sent through arc a.

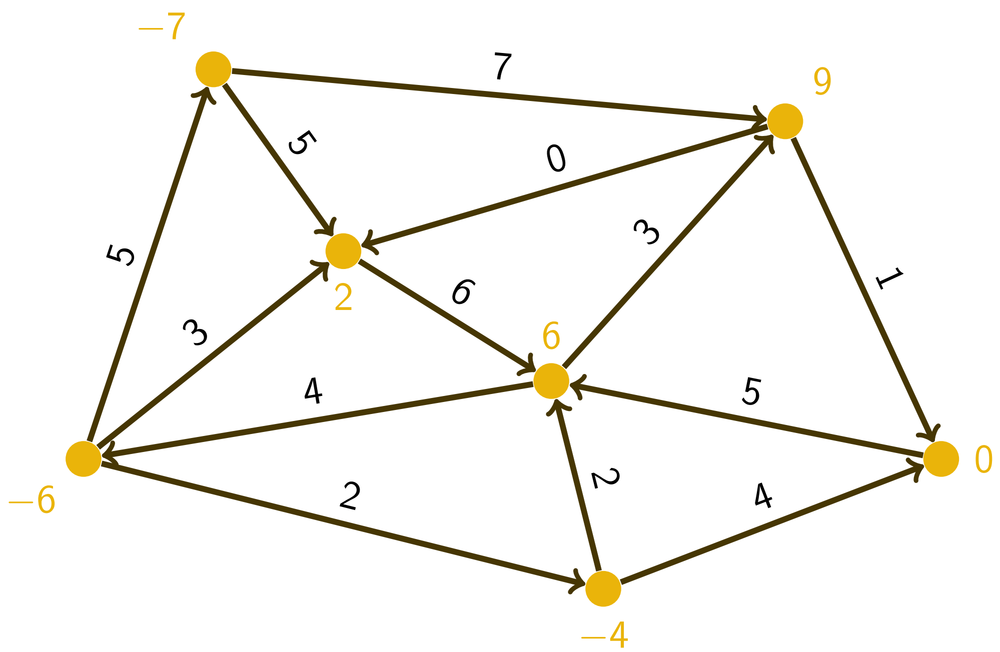
Task: find a flow $x : A \to \mathbb{R}_{\geq 0}$ obeying arc capacities and satisfying all supplies and demands, that is,

$$\begin{aligned} 0 \leq x(a) &\leq u(a) && \text{for all } a \in A, \\ \sum_{a \in \delta^-(v)} x(a) - \sum_{a \in \delta^+(v)} x(a) &= b(v) && \text{for all } v \in V, \end{aligned}$$

such that x has minimum cost $c(x) := \sum_{a \in A} c(a) \cdot x(a)$.

Minimum Cost Flow Problem (Cont.)

Example: flow satisfying given supplies and demands



Minimum Cost Flow Problem (Cont.)

Formulation as a linear program (LP):

$$\begin{aligned}
\text{minimize} \quad & \sum_{a \in A} c(a) \cdot x(a) && && (1.1)\\
\text{subject to} \quad & \sum_{a \in \delta^-(v)} x(a) - \sum_{a \in \delta^+(v)} x(a) = b(v) && \text{for all } v \in V, && (1.2)\\
& x(a) \le u(a) && \text{for all } a \in A, && (1.3)\\
& x(a) \ge 0 && \text{for all } a \in A. && (1.4)
\end{aligned}$$

- Objective function given by (1.1). Set of feasible solutions:

$$X = \{x \in \mathbb{R}^A \mid x \text{ satisfies (1.2), (1.3), and (1.4)}\} \ .$$

- Notice that (1.1) is a linear function of x and (1.2) – (1.4) are linear equations and linear inequalities, respectively $\longrightarrow$ linear program

Minimum Cost Flow with Fixed Cost

Fixed costs $w : A \to \mathbb{R}_{\ge 0}$.

If arc $a \in A$ shall be used (i.e., $x(a) > 0$), it must be bought at cost $w(a)$.

Add variables $y(a) \in \{0, 1\}$ with $y(a) = 1$ if arc a is used, 0 otherwise.

This leads to the following mixed-integer linear program (MIP):

$$\begin{aligned}
\text{minimize} \quad & \sum_{a \in A} c(a) \cdot x(a) + \sum_{a \in A} w(a) \cdot y(a) && \\
\text{subject to} \quad & \sum_{a \in \delta^-(v)} x(a) - \sum_{a \in \delta^+(v)} x(a) = b(v) && \text{for all } v \in V,\\
& x(a) \le u(a) \cdot y(a) && \text{for all } a \in A,\\
& x(a) \ge 0 && \text{for all } a \in A.\\
& y(a) \in \{0, 1\} && \text{for all } a \in A.
\end{aligned}$$

MIP: Linear program where some variables may only take integer values.

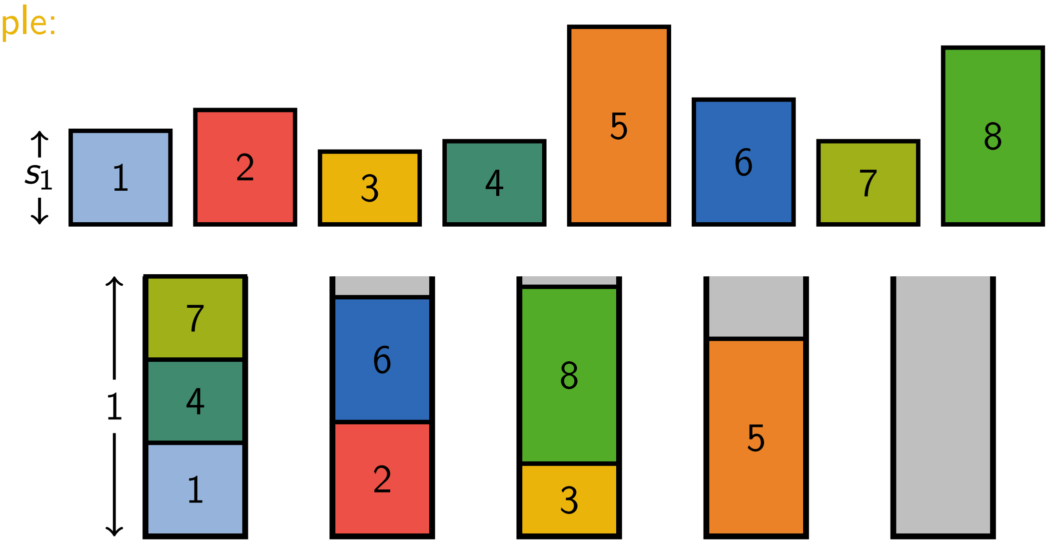
Bin Packing Problem

Given: n items with positive sizes $s_1, \dots, s_n \leq 1$

Task: pack the items into a minimum number of unit-size bins, i.e.,

$$\text{partition } \{1, \dots, n\} = \dot{\bigcup_{j=1}^{k}} I_j \text{ with minimum } k \text{ and } \sum_{i \in I_j} s_i \leq 1 \text{ for all } j$$

Example:



Bin Packing Problem

Given: n items with positive sizes $s_1, \dots, s_n \leq 1$

Task: pack the items into a minimum number of unit-size bins, i.e.,

$$\text{partition } \{1, \dots, n\} = \dot{\bigcup_{j=1}^{k}} I_j \text{ with minimum } k \text{ and } \sum_{i \in I_j} s_i \leq 1 \text{ for all } j$$

Formulation as an integer linear program (IP) with variables:

$x_{ij} \in \{0,1\}$ with interpretation: $\quad x_{ij} = 1 \iff$ item i in bin j

$y_j \in \{0,1\}$ with interpretation: $\quad y_j = 1 \iff$ bin j non-empty

$$\begin{array}{lll}
\text{minimize} & \sum_{j=1}^{n} y_j & \\
\text{subject to} & \sum_{j=1}^{n} x_{ij} = 1 & \text{for all } i = 1, \dots, n, \\
& \sum_{i=1}^{n} x_{ij} \leq y_j & \text{for all } j = 1, \dots, n, \\
& x_{ij}, y_j \in \{0,1\} & \text{for all } i, j = 1, \dots, n.
\end{array}$$

Knapsack Problem

Given: n items with positive values $v_1, \ldots, v_n$ and weights $w_1, \ldots, w_n$, knapsack of capacity W

Task: find subset $I \subseteq \{1, \ldots, n\}$ with $\sum_{i \in I} w_i \leq W$ and $\sum_{i \in I} v_i$ maximum

Formulation as an integer linear program (IP):

Variables: $x_i \in \{0, 1\}$ for $i = 1, \ldots, n$ interpretation: $x_i = 1 \iff i \in I$

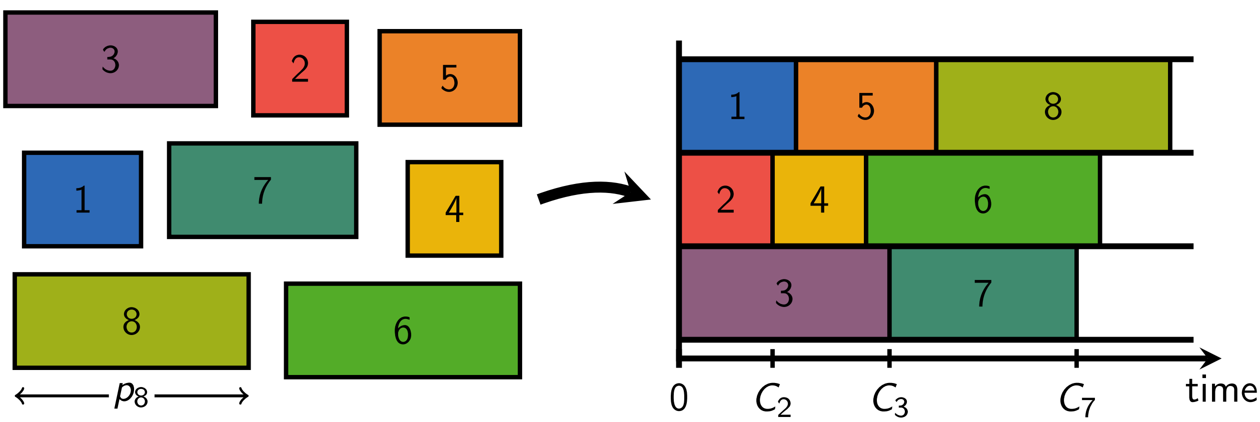
$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} v_i \cdot x_i \\ \text{subject to} \quad & \sum_{i=1}^{n} w_i \cdot x_i \leq W \\ & x_i \in \{0, 1\} \qquad\qquad \text{for } 1 = 1, \ldots, n. \end{aligned}$$

Parallel Machine Scheduling

Given: n jobs $j = 1, \ldots, n$, processing times $p_j > 0$, weights $w_j > 0$

Task: schedule jobs on m parallel machines; minimize $\sum_j w_j C_j$

Example: scheduling on 3 parallel machines



Formulation as an integer linear program (IP)?

Typical Questions

For a given optimization problem:

- How to find an optimal solution?
- How to find a feasible solution?
- Does there exist an optimal/feasible solution?
- How to prove that a computed solution is optimal?
- How difficult is the problem?
- Is there an *efficient algorithm* with "small" worst-case running time?
- How to formulate the problem as a (mixed integer) linear program?
- Is there a useful special structure of the problem?

Solving MIPs: Mathematical Progress vs. Faster Hardware

Mixed-Integer Linear Program (MIP)

variables $x \in \mathbb{R}^n$, parameters $c \in \mathbb{Q}^n$, $b \in \mathbb{Q}^m$, $A \in \mathbb{Q}^{m \times n}$

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \geq b \\ & x_j \in \mathbb{Z} \quad \text{for certain } j \end{aligned}$$

Bob Bixby's question (2015): Which option is faster?

Option 1: Solve a MIP with 2015 software on a 1991 computer

Option 2: Solve a MIP with 1991 software on a 2015 computer

Info: computer speed increased by factor ≈ 3500

But: Option 1 is another ≈ 300 times faster!

Preliminary Outline of the Course

- linear programming and the simplex algorithm
- geometric interpretation of the simplex algorithm
- LP duality, complementary slackness
- sensitivity analysis
- basic theory of polyhedra
- efficient algorithms for minimum spanning trees, shortest paths
- efficient algorithms for maximum flows and minimum cost flows
- complexity of linear programming and the ellipsoid method
- large-scale linear programming
- ...

Literature on Linear Optimization (not complete)

- D. Bertsimas, J. N. Tsitsiklis, *Introduction to Linear Optimization*, Athena, 1997.
- V. Chvatal, *Linear Programming*, Freeman, 1983.
- G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, 1998 (1963).
- M. Grötschel, L. Lovász, A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*. Springer, 1988.
- J. Matoušek, B. Gärtner, *Using and Understanding Linear Programming*, Springer, 2006.
- M. Padberg, *Linear Optimization and Extensions*, Springer, 1995.
- A. Schrijver, *Theory of Linear and Integer Programming*, Wiley, 1986.
- R. J. Vanderbei, *Linear Programming*, Springer, 2001.

Literature on Combinatorial Optimization (not complete)

- R. K. Ahuja, T. L. Magnanti, J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice-Hall, 1993.
- W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, A. Schrijver, *Combinatorial Optimization*, Wiley, 1998.
- L. R. Ford, D. R. Fulkerson, *Flows in Networks*, Princeton University Press, 1962.
- M. R. Garey, D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, 1979.
- B. Korte, J. Vygen, *Combinatorial Optimization: Theory and Algorithms*, Springer, 2002.
- C. H. Papadimitriou, K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Dover Publications, reprint 1998.
- A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, Springer, 2003.

Notation: Numbers, Vectors, Matrices

Numbers

- set of integers $\mathbb{Z}$
- set of rational numbers $\mathbb{Q}$
- set of real numbers $\mathbb{R}$
- set of non-negative numbers $\mathbb{Z}_{\geq 0}$, $\mathbb{Q}_{\geq 0}$, $\mathbb{R}_{\geq 0}$
- set of positive numbers $\mathbb{Z}_{>0}$, $\mathbb{Q}_{>0}$, $\mathbb{R}_{>0}$

Column Vectors in $\mathbb{R}^n$ (or $\mathbb{Q}^n$, $\mathbb{Z}^n$)

- i-th unit vector $e_i \in \mathbb{R}^n$, $i = 1, \dots, n$
- zero vector $0 \in \mathbb{R}^n$

$(m \times n)$-Matrix $A \in \mathbb{R}^{m \times n}$

- entry in row i and column j: A_{ij}
- j-th column: A_j

Notation: Graphs and Digraphs

Graph $G = (V, E)$ (undirected graph)

- finite node set V
- finite edge set $E \subseteq \{\{v, w\} \mid v, w \in V,\ v \neq w\}$
- for $v \in V$, $\delta(v) := \{e \in E \mid v \in e\}$ (incident edges)
- for $S \subseteq V$, $\delta(S) := \{e \in E \mid e \cap S \neq \emptyset \text{ and } e \cap (V \setminus S) \neq \emptyset\}$ (cut induced by S)
- for $S \subseteq V$, $\gamma(S) := \{e \in E \mid e \subseteq S\}$

Digraph $D = (V, A)$ (directed graph)

- finite node set V
- finite arc set $A \subseteq V \times V$
- for $v \in V$, $\delta^+(v) := A \cap \big(\{v\} \times (V \setminus \{v\})\big)$ (outgoing arcs), $\delta^-(v) := A \cap \big((V \setminus \{v\}) \times \{v\}\big)$ (incoming arcs)
- for $S \subseteq V$, $\delta^+(S) := A \cap \big(S \times (V \setminus S)\big)$, $\delta^-(S) := A \cap \big((V \setminus S) \times S\big)$ (directed cuts induced by S)
- for $S \subseteq V$, $\gamma(S) := A \cap \big(S \times S\big)$