1 Introduction

1.1 Optimization Problems

Optimization Problems

Generic optimization problem

Given: feasible set X, objective function $f : X \longrightarrow \mathbb{R}$

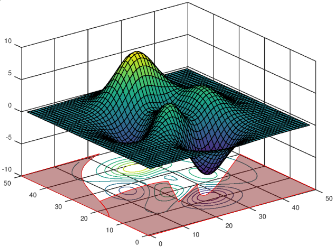
Task: find minimum $x^* \in X$ of f, i.e.,

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

- a maximum of f is a minimum of $-f$
- minima and maxima are called optima
- short forms

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

$$\max\{f(x) \mid x \in X\}$$



Optimization Problems

Generic optimization problem

Given: feasible set X, objective function $f : X \longrightarrow \mathbb{R}$

Task: find minimum $x^* \in X$ of f, i.e.,

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

- a maximum of f is a minimum of $-f$
- minima and maxima are called optima
- short forms

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

$$\max\{f(x) \mid x \in X\}$$

Problem: Too general to say anything meaningful!

Examples

- Examples of decision variables (x)
  - Size of some device
  - Model parameters
  - Amount invested in different assets
  - Input of a dynamical system
- Examples of objective functions (f)
  - Minimize costs / Maximize profits
  - Minimize error / Maximize fit
  - Minimize energy / power consumption
  - Minimize risk
  - Maximize fairness between different agents
- Example of constraints (X)
  - Prior knowledge
  - Physical limitations
  - Budget

Convex Optimization Problems

Definition 1.1 (Convexity)

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

a X is convex if

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

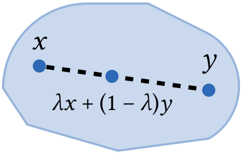
for all $x, y \in X$ and $\lambda \in [0, 1]$.

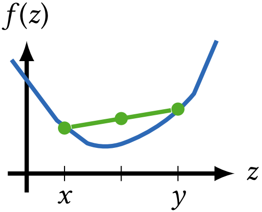
b f is convex if X is convex and

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

for all $x, y \in X$ and $\lambda \in [0, 1]$.

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





- f is called concave if $-f$ is convex

Definition 1.2 Let $X \subseteq \mathbb{R}^n$ and $f : X \longrightarrow \mathbb{R}$.
$x' \in X$ is a local optimum of the optimization problem $\min\{f(x) \mid x \in X\}$ if there is an $\varepsilon > 0$ such that

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

Theorem 1.3 For a convex optimization problem, every local optimum x is a (global) optimum.

Proof:

- let x be local optimum, for a contradiction assume $f(x^*) < f(x)$ for some $x^* \in X$
- for each $\lambda \in (0, 1]$ we have

$$f\Big(\underbrace{\lambda x^* + (1-\lambda)x}_{\in X}\Big) \leq \lambda f(x^*) + (1-\lambda)f(x) < f(x).$$

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

Optimization Problems 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 (LP)

- $X \subseteq \mathbb{Z}^n$ integer points of a polyhedron, f linear function
  $\longrightarrow$ integer linear optimization problem (IP)

- $Y \subseteq \mathbb{R}^n$ polyhedron, $X = Y \cap (\mathbb{R}^{n_1} \times \mathbb{Z}^{n_2})$, f linear function
  $\longrightarrow$ mixed integer linear optimization problem (MIP)

- 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

Trivial Solution Strategy

Given: finite set X of feasible solutions, objective function $f : X \longrightarrow \mathbb{R}$

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

William R. Pulleyblank about the 1960's

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

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.

Trivial Solution Strategy

Given: finite set X of feasible solutions, objective function $f : X \longrightarrow \mathbb{R}$

Task: find $x \in X$ 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.

1 Introduction

1.2 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 $\mathbb{N} := \mathbb{Z}_{>0}$, $\mathbb{Q}_{>0}$, $\mathbb{R}_{>0}$
- set of integers $[n] := \{1, \dots, n\}$

Vectors

- all vectors $x = (x_1, \dots, x_n)^\top$ are column vectors
- for a finite set V, we identify a function $x : V \longrightarrow \mathbb{R}$ with a vector $x \in \mathbb{R}^V$, $x(v)$ and x_v are used interchangeably
- for $U \subseteq V$, the incidence vector χ^U in $\mathbb{R}^V$ is defined as

$$\chi^U(s) = \begin{cases} 1 & \text{if } s \in U, \\ 0 & \text{otherwise} \end{cases}$$

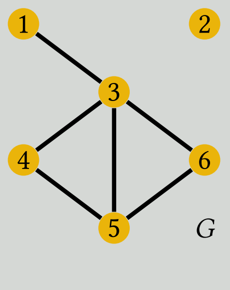
- $\mathbb{R}^n := \mathbb{R}^{\{1,\dots,n\}}$, $e_i := \chi^{\{i\}} := \chi^i$, $\mathbf{0} = \chi^\emptyset$

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

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

Definition 1.4 (Graph)
Tuple $G = (V, E)$ with

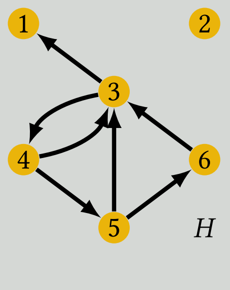
- finite node set V
- finite edge set $E \subseteq \{ e \subseteq V,\ |e| = 2 \}$



- $G = (V, E)$ is a graph with $V = \{1, 2, \ldots, 6\}$, $E = \{ \{1,3\}, \{3,4\}, \{3,5\}, \{3,6\}, \{4,5\}, \{5,6\} \}$

Definition 1.5 (Digraph)
Tuple $G = (V, A)$ with

- finite node set V
- finite arc set $A \subseteq V \times V$



- u and v are called tail and head of arc $(u, v) \in A$.
- $H = (V, A)$ is a digraph with $V = \{1, 2, \ldots, 6\}$, $A = \{ (3,1), (3,4), (4,3), (4,5), (5,3), (5,6), (6,3) \}$

Incidence

set of incident edges for node $v \in V$

- $\delta(v) := \{ e \in E \mid v \in e \}$
- e.g., $\delta(5) = \{ \{4,5\}, \{3,5\}, \{5,6\} \}$

cut induced by set of nodes $S \subseteq V$

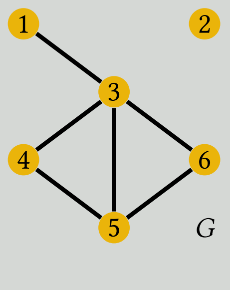
- $\delta(S) := \{ e \in E \mid e \cap S \neq \emptyset \text{ and } e \cap (V \setminus S) \neq \emptyset \}$
- e.g., $\delta(\{3,4,5\}) = \{ \{1,3\}, \{3,6\}, \{5,6\} \}$

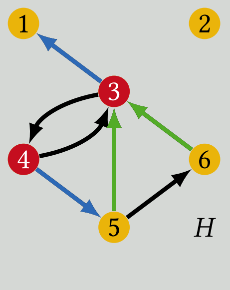
set of outgoing/incoming arcs of node $v \in V$

- $\delta^+(v) := A \cap (\{v\} \times V)$, $\delta^-(v) := A \cap (V \times \{v\})$
- e.g., $\delta^+(3) = \{ (3,1), (3,4) \}$, $\delta^-(3) = \{ (4,3), (5,3), (6,3) \}$

directed cuts induced by set of nodes $S \subseteq V$

- $\delta^+(S) := A \cap (S \times (V \setminus S))$, $\delta^-(S) := A \cap ((V \setminus S) \times S)$
- e.g., $\delta^+(\{3,4\}) = \{ (3,1), (4,5) \}$, $\delta^-(\{3,4\}) = \{ (5,3), (6,3) \}$





Walks, Paths, Cycles

- a walk P is a sequence

$$P = v_0, e_1, v_1, e_2, \dots, v_{k-1}, e_k, v_k$$

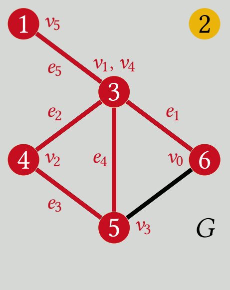
with $k \in \mathbb{N}$ and $e_i = \{v_{i-1}, v_i\} \in E$ for all $i = 1, \dots, k$

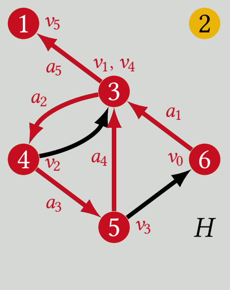
- a directed walk (or diwalk) P is a sequence

$$Q = v_0, a_1, v_1, a_2, \dots, v_{k-1}, a_k, v_k$$

with $k \in \mathbb{N}$ and $a_i = (v_{i-1}, v_i) \in A$ for all $i = 1, \dots, k$

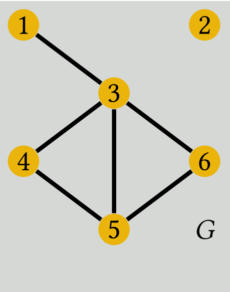
- we also call P a v_0-v_k-walk and Q a v_0-v_k-diwalk
- (di-)paths are (di-)walks with $v_i \neq v_j$ for all $i \neq j$
- closed (di-)walks are (di-)walks with $v_0 = v_k$
- (di-)cycles are closed (di-)walks with $k \geq 1$ and $v_i \neq v_j$ for all $0 \leq i < j < k$
- we say walk, path, cycle instead of diwalk, dipath, dicycle, when clear from context

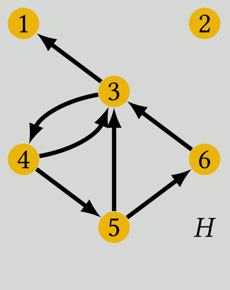




Connectedness

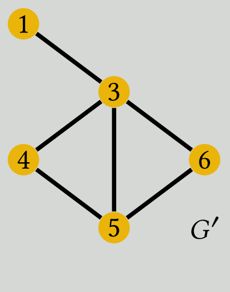
- an undirected graph is connected if there is a v-w-walk for all $v, w \in V$
  - e.g., G is not connected
  - e.g., G' is connected
- a digraph is connected if the underlying undirected graph obtained from ignoring the direction of each edge is connected
  - e.g., H is not connected
  - e.g., H' is connected
- a digraph is strongly connected if there is a v-w-diwalk for all $v, w \in V$
  - e.g. H' is not strongly connected
  - e.g. H'' is strongly connected

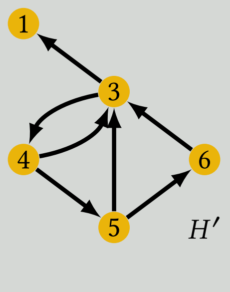




Connectedness

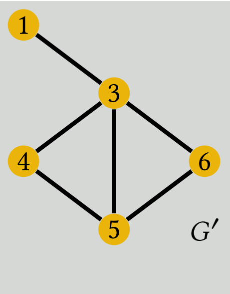
- an undirected graph is connected if there is a v-w-walk for all $v, w \in V$
  - e.g., G is not connected
  - e.g., G' is connected
- a digraph is connected if the underlying undirected graph obtained from ignoring the direction of each edge is connected
  - e.g., H is not connected
  - e.g., H' is connected
- a digraph is strongly connected if there is a v-w-diwalk for all $v, w \in V$
  - e.g. H' is not strongly connected
  - e.g. H'' is strongly connected

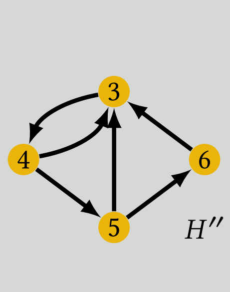




Connectedness

- an undirected graph is connected if there is a v-w-walk for all $v, w \in V$
  - e.g., G is not connected
  - e.g., G' is connected
- a digraph is connected if the underlying undirected graph obtained from ignoring the direction of each edge is connected
  - e.g., H is not connected
  - e.g., H' is connected
- a digraph is strongly connected if there is a v-w-diwalk for all $v, w \in V$
  - e.g. H' is not strongly connected
  - e.g. H'' is strongly connected





1 Introduction

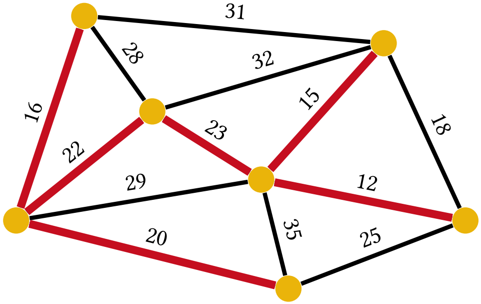
1.3 Examples

Minimum Spanning Tree Problem

Given: undirected graph $G = (V, E)$, edge costs $c_e \in \mathbb{R}_{\geq 0}$, $e \in E$

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

- $X = \left\{ E' \subseteq E \mid \bigcup_{e \in E'} = V \text{ and } G' = (V, E') \text{ is connected } \right\}$
- $f : X \longrightarrow \mathbb{R}$ is given by $f(E') := \sum_{e \in E'} c_e$



Minimum Spanning Tree Problem

Given: undirected graph $G = (V, E)$, edge costs $c_e \in \mathbb{R}_{\geq 0}$, $e \in E$

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

- $X = \left\{ E' \subseteq E \mid \bigcup_{e \in E'} = V \text{ and } G' = (V, E') \text{ is connected } \right\}$
- $f : X \longrightarrow \mathbb{R}$ is given by $f(E') := \sum_{e \in E'} c_e$

Remarks

- X is given implicitly by G
- there is always 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 1.6 (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 $c_a \in \mathbb{R}$, $a \in A$
start node $s \in V$, destination node $t \in V$

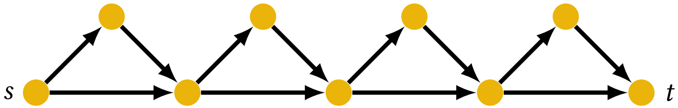
Task: find s-t-walk of minimum cost in D (if one exists)

- $X = \{ P \subseteq A \mid P \text{ is } s\text{-}t\text{-walk in } D \}$
- $f : X \longrightarrow \mathbb{R}$ is given by $f(P) := \sum_{a \in P} c(a)$

Remarks

- X is given implicitly by D
- X may be countably infinite

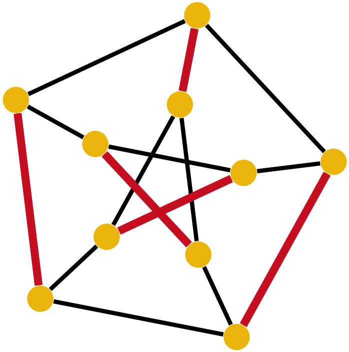
- digraph with $2n + 1$ nodes, $3n$ arcs, and 2^n s-t-paths



Maximum Weighted Matching Problem

Given: undirected graph $G = (V, E)$, edge weights $w_e \in \mathbb{R}$, $e \in E$.

Task: find matching $M \subseteq E$ with maximum total weight, i.e., every node is incident to at most one edge in M.



Maximum Weighted Matching Problem

Given: undirected graph $G = (V, E)$, edge weights $w_e \in \mathbb{R}$, $e \in E$.

Task: find matching $M \subseteq E$ with maximum total weight, i.e., every node is incident to at most one edge in M.

Formulation as an integer linear program (IP)

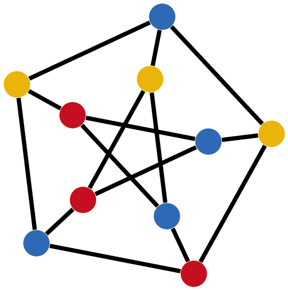
variables $x_e \in \{0, 1\}$ for $e \in E$ with interpretation $x_e = 1 \iff e \in M$

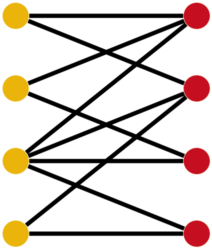
$$
\begin{aligned}
\text{maximize} \quad & \sum_{e \in E} w_e\, x_e \\
\text{subject to} \quad & \sum_{e \in \delta(v)} x_e \le 1 && \text{for all } v \in V, \\
& x_e \in \{0, 1\} && \text{for all } e \in E.
\end{aligned}
$$

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

Minimum Weighted Node Cover Problem

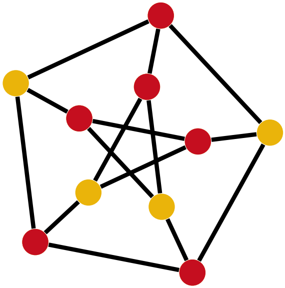
Given: undirected graph $G = (V, E)$, node weights $w_v \in \mathbb{R}_{\geq 0}$, $v \in V$

Task: find $U \subseteq V$ of minimum weight such that $U \cap e \neq \emptyset$ for all $e \in E$

Formulation as integer linear program (IP)

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

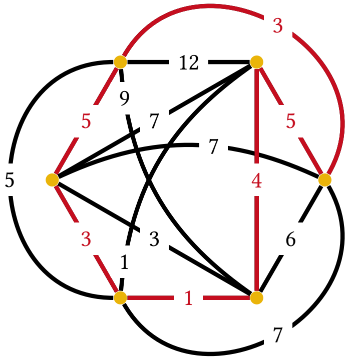
$$\begin{aligned} \min \quad & \sum_{v \in V} w_v \cdot x_v \\ \text{s.t.} \quad & x_v + x_{v'} \geq 1 && \text{for all } e = \{v, v'\} \in E, \\ & x_v \in \{0, 1\} && \text{for all } v \in V. \end{aligned}$$



Traveling Salesperson Problem (TSP)

Given: complete graph $K_n = (V, E)$ on n nodes, edge costs $c_e \in \mathbb{R}_{\geq 0}$, $e \in E$

Task: find a Hamiltonian cycle of minimum total length
(A Hamiltonian cycle is a cycle that visits every node exactly once.)



Formulation as an integer linear program? (later!)

Minimum Cost Flow Problem

Given: directed graph $D = (V, A)$

- arc capacities $u_a \in \mathbb{R}_{\geq 0}$, $a \in A$
- arc costs $c_a \in \mathbb{R}$, $a \in A$
- node balances $b_v \in \mathbb{R}$, $v \in V$

Interpretation

- single commodity (water, gas, electricity) is shipped in the network
- nodes $v \in V$ with $b_v > 0$ have demand and are called sinks
  nodes with $b_v < 0$ have supply and are called sources
- capacity u_a of arc $a \in A$ limits the amount of flow through a
- cost c_a is per-unit cost for shipping through a

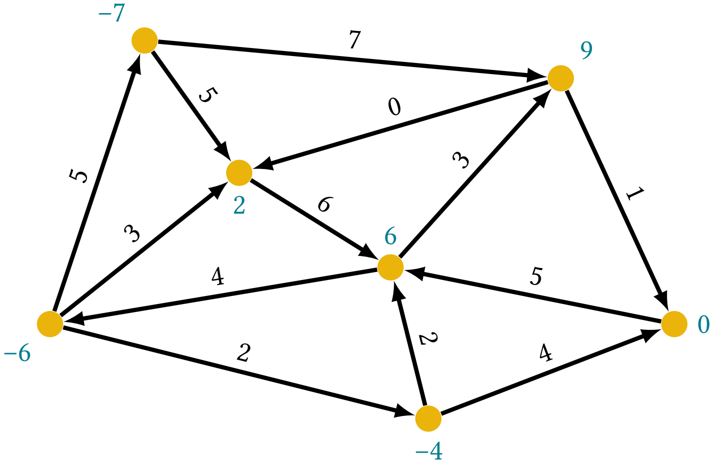
Task: find a flow $x = (x_a)_{a \in A}$, $x_a \in \mathbb{R}_{\geq 0}$, i.e.,

- $0 \leq x_a \leq u_a$ for all $a \in A$
- $\sum_{a \in \delta^-(v)} x_a - \sum_{a \in \delta^+(v)} x_a = b_v$ for all $v \in V$

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

Minimum Cost Flow Problem (Cont.)

Example: flow satisfying given supplies and demands



Minimum Cost Flow Problem (Cont.)

Formulation as a linear program (LP)

$$
\begin{aligned}
\text{minimize} \quad & \sum_{a \in A} c_a x_a & & & (1.1)\\
\text{subject to} \quad & \sum_{a \in \delta^-(v)} x_a - \sum_{a \in \delta^+(v)} x_a = b_v & & \text{for all } v \in V, & (1.2)\\
& x_a \leq u_a & & \text{for all } a \in A, & (1.3)\\
& x_a \geq 0 & & \text{for all } a \in A. & (1.4)
\end{aligned}
$$

- objective function given by (1.1).
- set of feasible solutions $X = \left\{ x \in \mathbb{R}^A \mid x \text{ satisfies (1.2), (1.3), and (1.4)} \right\}$
- objective (1.1) is linear in x and (1.2) – (1.4) are linear equations and linear inequalities, respectively $\longrightarrow$ linear program

Minimum Cost Flow with Fixed Cost

- assume arcs have fixed costs $w_a \in \mathbb{R}_{\geq 0}$, $a \in A$
- if arc $a \in A$ is used (i.e., $x_a > 0$), it must be bought at cost w_a

Formulation as mixed-integer linear program (MIP)

add variables $y_a \in \{0, 1\}$ with interpretation $y_a = 1 \iff a$ is used

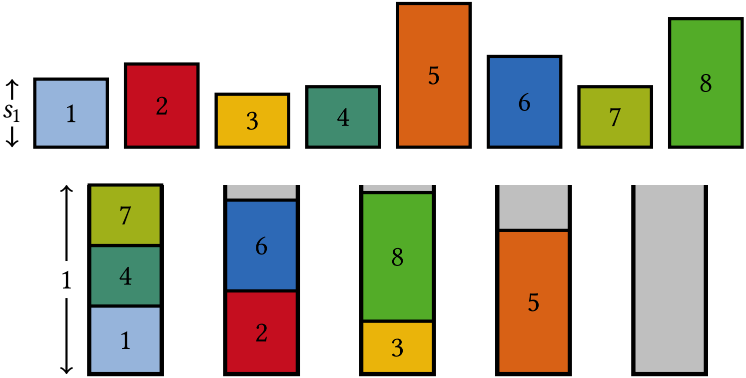
$$
\begin{aligned}
\text{minimize} \quad & \sum_{a \in A} c_a\, x_a + \sum_{a \in A} w_a\, y_a \\
\text{subject to} \quad & \sum_{a \in \delta^-(v)} x_a - \sum_{a \in \delta^+(v)} x_a = b_v && \text{for all } v \in V, \\
& x_a \leq u_a\, y_a && \text{for all } a \in A, \\
& x_a \geq 0 && \text{for all } a \in A. \\
& y(a) \in \{0, 1\} && \text{for all } a \in A.
\end{aligned}
$$

Bin Packing Problem

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

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

- $\{1, \ldots, n\} = \bigcup_{j=1}^{k} I_j$, $I_i \cap I_j = \emptyset$ for all $i \neq j$
- $\sum_{i \in I_j} s_i \leq 1$ for all j



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., minimize k such that

- $\{1, \dots, n\} = \bigcup_{j=1}^{k} I_j$, $I_i \cap I_j = \emptyset$ for all $i \neq j$
- $\sum_{i \in I_j} s_i \leq 1$ for all j

Formulation as an integer linear program (IP)

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

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

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

Knapsack Problem

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

Task: find subset $I \subseteq \{1, \dots, n\}$ 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 $i = 1, \dots, n$ with interpretation $x_i = 1 \iff i \in I$

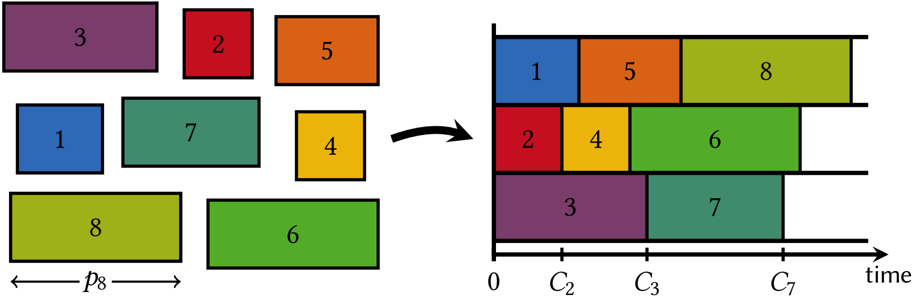
$$
\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{n} v_i \, x_i \\
\text{subject to} \quad & \sum_{i=1}^{n} w_i \, x_i \leq W \\
& x_i \in \{0, 1\} \qquad\qquad \text{for } 1 = 1, \dots, n.
\end{aligned}
$$

Parallel Machine Scheduling

Given: n jobs $j = 1, \dots, n$, 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?

1 Introduction

1.4 Outline and Literature

Rough Outline of ADM I

- 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, Minimum cost Flows, and weighted bipartite matchings
- Complexity of Linear Programming and the Ellipsoid Method
- Large-scale Linear Programming

Rough Outline of This Course

ADM I: Intro to Linear Programming & Combinatorial Optimization

ADM II: Discrete Optimization

- Maximum Weight Branchings
- Matchings
- Weighted Matchings
- T-Joins and the Postman Problem
- Matroids
- Complexity Theory and NP-hardness
- Integer Linear Programming
- Traveling Salesperson Problem

ADM3: Advanced topics

- Approximation Algorithms ?
- Algorithmic Game Theory ?
- Convex Optimization ?

Mathematical Progress vs. Faster Hardware

Mixed-Integer Linear Program (MIP)

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

$$\begin{aligned} \text{minimize} \quad & c^\top x \\ \text{subject to} \quad & Ax \geq b \\ & x_j \in \mathbb{Z} \quad \text{for certain } j \end{aligned}$$

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!

Literature on Linear and Integer Optimization

- 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

- 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.