Example: 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:

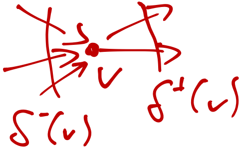
- nodes $v \in V$ with $b(v) > 0$ ($b(v) < 0$) have *supply* (*demand*) and are called *sources* (*sinks*)
- the capacity $u(a)$ of arc $a \in A$ limits the amount of flow that can be sent through arc a.

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

$$0 \leq x(a) \leq u(a) \qquad \text{for all } a \in A,$$

$$\sum_{a \in \delta^+(v)} x(a) - \sum_{a \in \delta^-(v)} x(a) = b(v) \qquad \text{for all } v \in V,$$

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



Example: Minimum Cost Flow Problem (cont.)

Formulation as a linear program (LP):

$$\text{minimize} \quad \sum_{a \in A} c(a) \cdot x(a) \tag{1.1}$$

$$\text{subject to} \quad \sum_{a \in \delta^+(v)} x(a) - \sum_{a \in \delta^-(v)} x(a) = b(v) \quad \text{for all } v \in V, \tag{1.2}$$

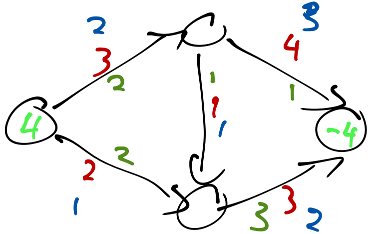
$$x(a) \leq u(a) \quad \text{for all } a \in A, \tag{1.3}$$

$$x(a) \geq 0 \quad \text{for all } a \in A. \tag{1.4}$$

Flow conservation

capacity

x(a)



cost(x) = 16

u(a) ... capacity of a

c(a) ... cost of shipping 1 unit over a

Example: Minimum Cost Flow Problem (cont.)

Formulation as a linear program (LP):

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

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

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

Example: Minimum Cost Flow Problem (cont.)

Formulation as a linear program (LP):

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

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

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

- 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

Example (cont.): Adding Fixed Cost

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

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

Example (cont.): Adding Fixed Cost

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

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.

Example (cont.): Adding Fixed Cost

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

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):

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

Example (cont.): Adding Fixed Cost

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

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):

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

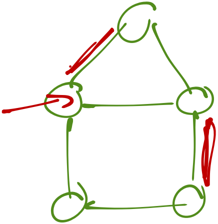
MIP: Linear program where some variables may only take integer values.

Example: Maximum Weighted Matching Problem

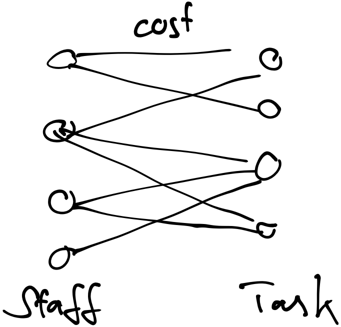
Given: undirected graph $G = (V, E)$, weight function $w : E \to \mathbb{R}$.

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.)



Application:
Task assignment



Example: 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 $x_e = 1$ if and only if $e \in M$.

Example: 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 $x_e = 1$ if and only if $e \in M$.

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

set of incident edges of v

Example: 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 $x_e = 1$ if and only if $e \in M$.

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

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

Example: Traveling Salesperson Problem (TSP)

Given: complete graph K_n on n nodes, weight function $w : E(K_n) \to \mathbb{R}$.

Task: find a Hamiltonian circuit with minimum total weight.

(A Hamiltonian circuit visits every node exactly once.)

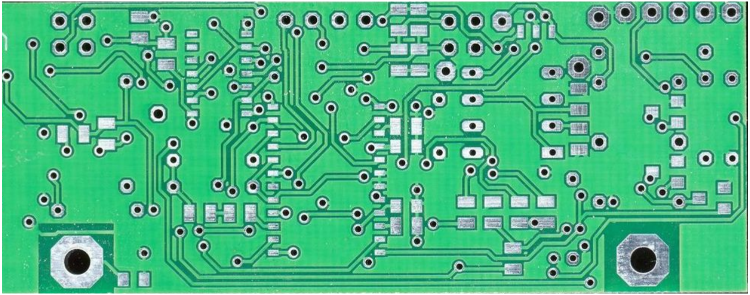
Example: Traveling Salesperson Problem (TSP)

Given: complete graph K_n on n nodes, weight function $w : E(K_n) \to \mathbb{R}$.

Task: find a Hamiltonian circuit with minimum total weight.

(A Hamiltonian circuit visits every node exactly once.)

Application: Drilling holes in printed circuit boards.



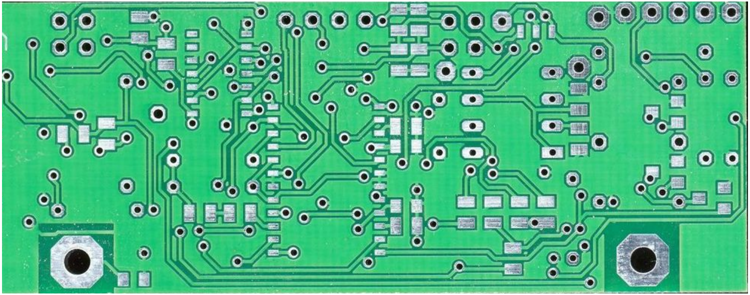
Example: Traveling Salesperson Problem (TSP)

Given: complete graph K_n on n nodes, weight function $w : E(K_n) \to \mathbb{R}$.

Task: find a Hamiltonian circuit with minimum total weight.

(A Hamiltonian circuit visits every node exactly once.)

Application: Drilling holes in printed circuit boards.



Formulation as an integer linear program? (maybe later!)

Example: Weighted Vertex 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.

Example: Weighted Vertex 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.

Formulation as an integer linear program (IP):

Variables: $x_v \in \{0, 1\}$ for $v \in V$ with $x_v = 1$ if and only if $v \in U$.

Example: Weighted Vertex 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.

Formulation as an integer linear program (IP):

Variables: $x_v \in \{0, 1\}$ for $v \in V$ with $x_v = 1$ if and only if $v \in U$.

$$\begin{aligned} \text{minimize} \quad & \sum_{v \in V} w(v) \cdot x_v \\ \text{subject to} \quad & x_v + x_{v'} \geq 1 && \text{for all } e = \{v, v'\} \in E, \\ & x_v \in \{0, 1\} && \text{for all } v \in V. \end{aligned}$$

Markowitz' Portfolio Optimisation Problem

Given: n different securities (stocks, bonds, etc.) with random returns, target return R, for each security $i \in [n]$:

- expected return μ_i, variance σ_i

For each pair of securities i, j:

- covariance ρ_{ij},

Markowitz' Portfolio Optimisation Problem

Given: n different securities (stocks, bonds, etc.) with random returns, target return R, for each security $i \in [n]$:

- expected return μ_i, variance σ_i

For each pair of securities i, j:

- covariance ρ_{ij},

Task: Find a portfolio $x_1, \ldots, x_n$ that minimises *"risk"* (aka *variance*) and has expected return $\geq R$.

Formulation as a quadratic programme (QP):

Markowitz' Portfolio Optimisation Problem

Given: n different securities (stocks, bonds, etc.) with random returns, target return R, for each security $i \in [n]$:

- expected return μ_i, variance σ_i

For each pair of securities i, j:

- covariance ρ_{ij},

Task: Find a portfolio $x_1, \ldots, x_n$ that minimises *"risk"* (aka *variance*) and has expected return $\geq R$.

Formulation as a quadratic programme (QP):

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i,j} \rho_{ij}\sigma_i\sigma_j x_i x_j \\
\text{subject to} \quad & \sum_i x_i = 1 \\
& \sum_i \mu_i x_i \geq R \\
& x_i \geq 0, \qquad \text{for all } i.
\end{aligned}
$$

Typical Questions

For a given optimization problem:

- How to find an optimal solution?

Typical Questions

For a given optimization problem:

- How to find an optimal solution?
- How to find a feasible solution?

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?

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?

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?

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?
- Does there exist an *efficient algorithm* with “small” worst-case running time?

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?
- Does there exist an *efficient algorithm* with "small" worst-case running time?
- How to formulate the problem as a (mixed integer) linear program?

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?
- Does there exist 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?

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. Lovàsz, A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*. Springer, 1988.
- J. Matousek, 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.

Chapter 2:
Linear Programming Basics

(Bertsimas & Tsitsiklis, Chapter 1)

Example of a Linear Program

$$
\begin{array}{lrcrcrcrl}
\text{minimize} & 2x_1 & - & x_2 & + & 4x_3 & & & \\
\text{subject to} & x_1 & + & x_2 & & & + & x_4 & \leq 2 \\
 & & & 3x_2 & - & x_3 & & & = 5 \\
 & & & & & x_3 & + & x_4 & \geq 3 \\
 & x_1 & & & & & & & \geq 0 \\
 & & & & & x_3 & & & \leq 0
\end{array}
$$

objective function

constraints

Example of a Linear Program

$$\begin{array}{lrcrcrcrl}
\text{minimize} & 2x_1 & - & x_2 & + & 4x_3 & & & \\
\text{subject to} & x_1 & + & x_2 & & & + & x_4 & \leq 2 \\
 & & & 3x_2 & - & x_3 & & & = 5 \\
 & & & & & x_3 & + & x_4 & \geq 3 \\
 & x_1 & & & & & & & \geq 0 \\
 & & & & & x_3 & & & \leq 0
\end{array}$$

Remarks.

- objective function is linear in vector of variables $x = (x_1, x_2, x_3, x_4)^T$
- constraints are linear inequalities and linear equations
- last two constraints are special
  (non-negativity and non-positivity constraint, respectively)

General Linear Program

$$
\begin{aligned}
\text{minimize} \quad & c^T \cdot x = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \\
\text{subject to} \quad & a_i{}^T \cdot x \geq b_i && \text{for } i \in M_1, && (2.1) \\
& a_i{}^T \cdot x = b_i && \text{for } i \in M_2, && (2.2) \\
& a_i{}^T \cdot x \leq b_i && \text{for } i \in M_3, && (2.3) \\
& x_j \geq 0 && \text{for } j \in N_1, && (2.4) \\
& x_j \leq 0 && \text{for } j \in N_2, && (2.5)
\end{aligned}
$$

General Linear Program

$$
\begin{aligned}
\text{minimize} \quad & c^T \cdot x & & \\
\text{subject to} \quad & {a_i}^T \cdot x \geq b_i & & \text{for } i \in M_1, & (2.1)\\
& {a_i}^T \cdot x = b_i & & \text{for } i \in M_2, & (2.2)\\
& {a_i}^T \cdot x \leq b_i & & \text{for } i \in M_3, & (2.3)\\
& x_j \geq 0 & & \text{for } j \in N_1, & (2.4)\\
& x_j \leq 0 & & \text{for } j \in N_2, & (2.5)
\end{aligned}
$$

with $c \in \mathbb{R}^n$, $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$ for $i \in M_1 \dot\cup M_2 \dot\cup M_3$ (finite index sets), and $N_1, N_2 \subseteq \{1, \ldots, n\}$ given.

General Linear Program

$$
\begin{aligned}
\text{minimize} \quad & c^T \cdot x & & \\
\text{subject to} \quad & {a_i}^T \cdot x \geq b_i & \text{for } i \in M_1, & \quad (2.1)\\
& {a_i}^T \cdot x = b_i & \text{for } i \in M_2, & \quad (2.2)\\
& {a_i}^T \cdot x \leq b_i & \text{for } i \in M_3, & \quad (2.3)\\
& x_j \geq 0 & \text{for } j \in N_1, & \quad (2.4)\\
& x_j \leq 0 & \text{for } j \in N_2, & \quad (2.5)
\end{aligned}
$$

with $c \in \mathbb{R}^n$, $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$ for $i \in M_1 \,\dot\cup\, M_2 \,\dot\cup\, M_3$ (finite index sets), and $N_1, N_2 \subseteq \{1, \ldots, n\}$ given.

- $x \in \mathbb{R}^n$ satisfying constraints (2.1) – (2.5) is a feasible solution.

General Linear Program

$$
\begin{aligned}
\text{minimize} \quad & c^T \cdot x \\
\text{subject to} \quad & a_i{}^T \cdot x \geq b_i && \text{for } i \in M_1, && (2.1)\\
& a_i{}^T \cdot x = b_i && \text{for } i \in M_2, && (2.2)\\
& a_i{}^T \cdot x \leq b_i && \text{for } i \in M_3, && (2.3)\\
& x_j \geq 0 && \text{for } j \in N_1, && (2.4)\\
& x_j \leq 0 && \text{for } j \in N_2, && (2.5)
\end{aligned}
$$

with $c \in \mathbb{R}^n$, $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$ for $i \in M_1 \dot\cup M_2 \dot\cup M_3$ (finite index sets), and $N_1, N_2 \subseteq \{1, \ldots, n\}$ given.

- $x \in \mathbb{R}^n$ satisfying constraints (2.1) – (2.5) is a feasible solution.
- feasible solution x^* is optimal solution if

$$c^T \cdot x^* \leq c^T \cdot x \qquad \text{for all feasible solutions } x.$$

General Linear Program

$$
\begin{aligned}
\text{minimize} \quad & c^T \cdot x & & \\
\text{subject to} \quad & a_i{}^T \cdot x \geq b_i & \text{for } i \in M_1, & \quad (2.1) \\
& a_i{}^T \cdot x = b_i & \text{for } i \in M_2, & \quad (2.2) \\
& a_i{}^T \cdot x \leq b_i & \text{for } i \in M_3, & \quad (2.3) \\
& x_j \geq 0 & \text{for } j \in N_1, & \quad (2.4) \\
& x_j \leq 0 & \text{for } j \in N_2, & \quad (2.5)
\end{aligned}
$$

with $c \in \mathbb{R}^n$, $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$ for $i \in M_1 \dot\cup M_2 \dot\cup M_3$ (finite index sets), and $N_1, N_2 \subseteq \{1, \ldots, n\}$ given.

- $x \in \mathbb{R}^n$ satisfying constraints (2.1) – (2.5) is a feasible solution.
- feasible solution x^* is optimal solution if

$$c^T \cdot x^* \leq c^T \cdot x \qquad \text{for all feasible solutions } x.$$

- linear program is unbounded if, for all $k \in \mathbb{R}$, there is a feasible solution $x \in \mathbb{R}^n$ with $c^T \cdot x \leq k$.