MVE165/MMG631
Linear and integer optimization with applications
Lecture 5
Discrete optimization models and applications; complexity

Ann-Brith Strömberg

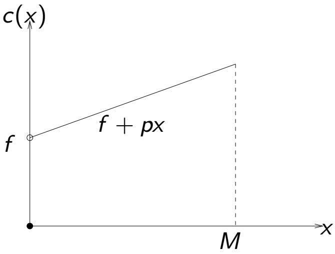
2014–04–01

Modelling with integer variables (Ch. 13.1)

- **Variables**
  - *Linear programming* (LP) uses continuous variables: $x_{ij} \geq 0$
  - *Integer linear programming* (ILP) use also *integer*, *binary*, and *discrete* variables
  - If both continuous and integer variables are used in a program, it is called a *mixed integer (linear) program (MILP)*
- **Constraints**
  - In an ILP (or MILP) it is possible to model linear constraints, but also logical relations as, e.g. if–then and either–or
  - This is done by introducing additional binary variables and additional constraints

Mixed integer modelling—fixed charges

- Send a truck $\Rightarrow$ Start–up cost $f > 0$
- Load bread loafs $\Rightarrow$ cost $p > 0$ per loaf
- $x = \#$ bread loafs to transport from bakery to store

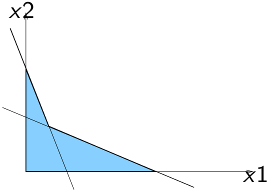


- Cost function $c(x) = \begin{cases} 0 & \text{if} \quad x = 0 \\ f + px & \text{if} \quad 0 < x \leq M \end{cases}$
- The function $c : \mathbb{R}_+ \mapsto \mathbb{R}_+$ is *nonlinear* and *discontinuos*

Integer linear programming modelling—fixed charges

- Let $y = \#$ trucks to send (here y equals 0 or 1)
- Replace $c(x)$ by $fy + px$
- Constraints: $0 \leq x \leq My$ and $y \in \{0, 1\}$
- New model: $\left[\begin{array}{lrcl} & \min fy + px & & \\ \text{s.t.} & x - My & \leq & 0 \\ & x & \geq & 0 \\ & y & \in & \{0,1\} \end{array} \right]$
  - $y = 0 \quad \Rightarrow \quad x = 0 \quad \Rightarrow \quad fy + px = 0$
  - $y = 1 \quad \Rightarrow \quad x \leq M \quad \Rightarrow \quad fy + px = f + px$
  - $x > 0 \quad \Rightarrow \quad y = 1 \quad \Rightarrow \quad fy + px = f + px$
  - $x = 0 \quad \not\Rightarrow \quad y = 0$ But: Minimization will push y to zero!

Discrete alternatives



- Suppose:
  either $x_1 + 2x_2 \leq 4$ **or** $5x_1 + 3x_2 \leq 10$,
  and $x_1, x_2 \geq 0$ must hold
- **Not** a convex set
- Let $M \gg 1$ and define $y \in \{0, 1\}$

$\Rightarrow$ New constraint set:

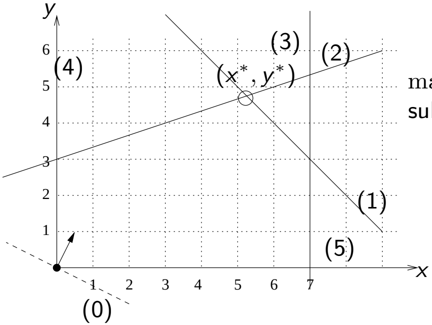
$$\left[\begin{array}{rrcl} x_1 + 2x_2 & -My & \leq & 4 \\ 5x_1 + 3x_2 & -M(1-y) & \leq & 10 \\ & y & \in & \{0,1\} \\ x_1, x_2 & & \geq & 0 \end{array}\right]$$

- $y = \begin{cases} 0 & \Rightarrow \quad x_1 + 2x_2 \leq 4 \quad \text{must hold} \\ 1 & \Rightarrow \quad 5x_1 + 3x_2 \leq 10 \quad \text{must hold} \end{cases}$

Exercises: Homework

1. Suppose that you are interested in choosing from a set of investments $\{1, \ldots, 7\}$ using $0-1$ variables. Model the following constraints.
   1.1 You cannot invest in all of them
   1.2 You must choose at least one of them
   1.3 Investment 1 cannot be chosen if investment 3 is chosen
   1.4 Investment 4 can be chosen only if investment 2 is also chosen
   1.5 You must choose either both investment 1 and 5 or neither
   1.6 You must choose either at least one of the investments 1, 2 and 3 or at least two investments from 2, 4, 5 and 6
2. Formulate the following as mixed integer progams
   2.1 $u = \min\{x_1, x_2\}$, assuming that $0 \leq x_j \leq C$ for $j = 1, 2$
   2.2 $v = |x_1 - x_2|$ with $0 \leq x_j \leq C$ for $j = 1, 2$
   2.3 The set $X \setminus \{x^*\}$ where $X = \{x \in Z^n | Ax \leq b\}$ and $x^* \in X$

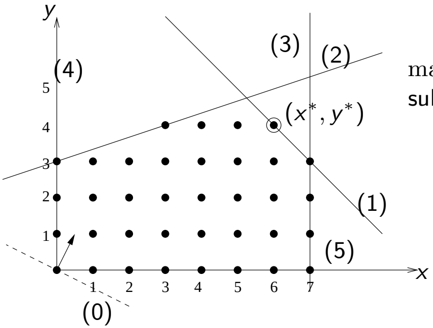
Linear programming: A small example



$$
\begin{array}{llllll}
\text{maximize} & x & + & 2y & & & (0) \\
\text{subject to} & x & + & y & \leq & 10 & (1) \\
 & -x & + & 3y & \leq & 9 & (2) \\
 & x & & & \leq & 7 & (3) \\
 & & & x, y & \geq & 0 & (4,5)
\end{array}
$$

- Optimal solution: $(x^*, y^*) = (5\frac{1}{4}, 4\frac{3}{4})$
- Optimal objective value: $14\frac{3}{4}$

Integer linear programming: A small example

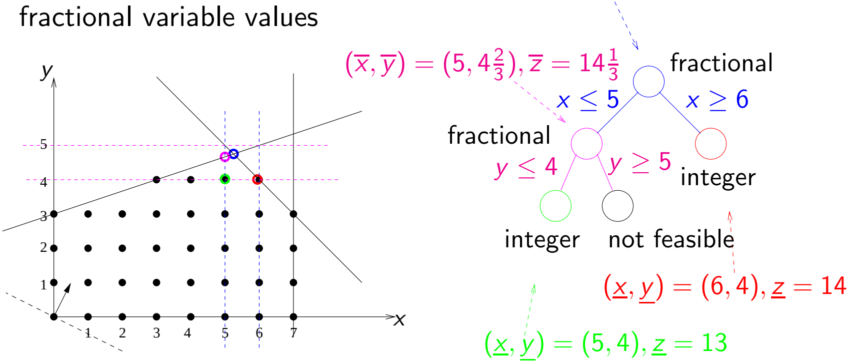


$$
\begin{array}{llcrcrll}
\text{maximize} & x & + & 2y & & & & (0) \\
\text{subject to} & x & + & y & \leq & 10 & & (1) \\
 & -x & + & 3y & \leq & 9 & & (2) \\
 & x & & & \leq & 7 & & (3) \\
 & & & x, y & \geq & 0 & & (4,5) \\
 & & & x, y & & \text{integer} & &
\end{array}
$$

- What if the variables are forced to be integral?
- Optimal solution: $(x^*, y^*) = (6, 4)$
- Optimal objective value: $14 < 14\frac{3}{4}$
- The optimal value decreases (possibly constant) when the variables are restricted to have integral values

ILP: Solution by the branch–and–bound algorithm (e.g., Cplex, XpressMP, or GLPK) (Ch. 15.1–15.2)

- Relax integrality requirements $\Rightarrow$ linear, continuous problem $\Rightarrow$ $(\overline{x}, \overline{y}) = (5\frac{1}{4}, 4\frac{3}{4}), \overline{z} = 14\frac{3}{4}$
- Search tree: branch over fractional variable values



The knapsack problem—budget constraints (Ch. 13.2)

- Select an optimal collection of objects or investments or projects ...
  - c_j = benefit of choosing object j, $j = 1, \dots, n$
- Limits on the budget
  - a_j = cost of object j, $j = 1, \dots, n$
  - b = total budget
- Variables: $x_j = \begin{cases} 1, & \text{if object } j \text{ is chosen,} \\ 0, & \text{otherwise.} \end{cases} \quad j = 1, \dots, n$
- Objective function: $\boxed{\max \sum_{j=1}^{n} c_j x_j}$
- Budget constraint: $\boxed{\sum_{j=1}^{n} a_j x_j \leq b}$
- Binary variables: $\boxed{x_j \in \{0, 1\},\ j = 1, \dots, n}$

Computational complexity (Ch. 2.6)

- A small knapsack instance

$$\begin{array}{lrcl} z_1^* = \max & 213x_1 + 1928x_2 + 11111x_3 + 2345x_4 + 9123x_5 & & \\ \text{subject to} & 12223x_1 + 12224x_2 + 36674x_3 + 61119x_4 + 85569x_5 & \leq & 89\,643\,482 \\ & x_1, \dots, x_5 & \geq & 0, \text{integer} \end{array}$$

- Optimal solution $\mathbf{x}^* = (0, 1, 2444, 0, 0)$, $z_1^* = 27\ 157\ 212$
- Cplex finds this solution in 0.015 seconds
- The equality version

$$\begin{array}{lrcl} z_2^* = \max & 213x_1 + 1928x_2 + 11111x_3 + 2345x_4 + 9123x_5 & & \\ \text{subject to} & 12223x_1 + 12224x_2 + 36674x_3 + 61119x_4 + 85569x_5 & = & 89\,643\,482 \\ & x_1, \dots, x_5 & \geq & 0, \text{integer} \end{array}$$

- Optimal solution $\mathbf{x}^* = (7334, 0, 0, 0, 0)$, $z_2^* = 1\ 562\ 142$
- Cplex computations interrupted after 1700 sec. ($\approx \frac{1}{2}$ hour)
  - No integer solution found
  - Best upper bound found: 25 821 000
  - 55 863 802 branch–and–bound nodes visited
  - Only *one* feasible solution exists!

Computational complexity

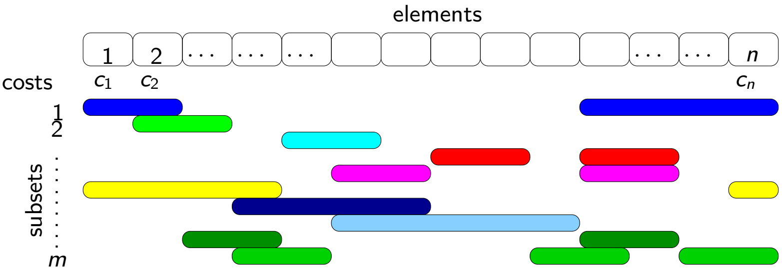
- Mathematical insight yields successful algorithms
- Example: Assignment problem: Assign n persons to n jobs.
- $\#$ feasible solutions: $n!$ $\Rightarrow$ Combinatorial explosion
- An algorithm $\exists$ that solves this problem in time $\mathcal{O}(n^4) \propto n^4$
- Binary knapsack: $\mathcal{O}(2^n)$
- Complete enumeration of all solutions is *not* efficient

n	2	5	8	10	100	1000
$n!$	2	120	40000	3600000	$9.3 \cdot 10^{157}$	$4.0 \cdot 10^{2567}$
2^n	4	32	256	1024	$1.3 \cdot 10^{30}$	$1.1 \cdot 10^{301}$
n^4	16	625	4100	10000	100000000	$1.0 \cdot 10^{12}$
($n \log n$	0.6	3.5	7.2	10	200	3000)

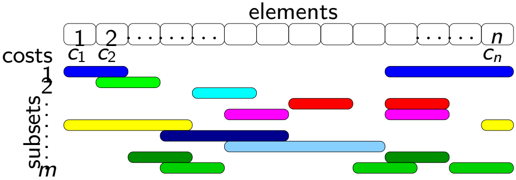
- (Continuous knapsack (sorting of $\frac{c_j}{a_j}$): $\mathcal{O}(n \log n)$)

The set covering problem (Ch. 13.8)

- A number (n) of items and a cost for each item
- A number (m) of subsets of the n items
- Find a selection of the items such that each subset contains at least one selected item and such that the total cost for the selected items is minimized



The set covering problem (Ch. 13.8)



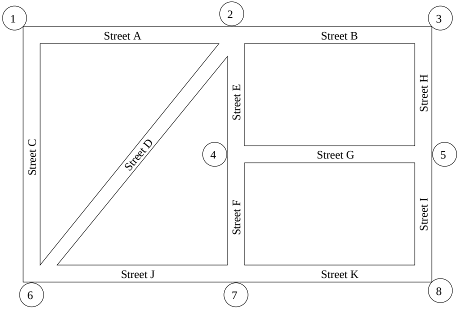
- Mathematical formulation:

$$\begin{array}{lrcl} \min & \mathbf{c}^{\mathrm{T}}\mathbf{x} & & \\ \text{subject to} & \mathbf{A}\mathbf{x} & \geq & \mathbf{1} \\ & \mathbf{x} & & \text{binary} \end{array}$$

- $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{1} = (1, \dots, 1)^{\mathrm{T}} \in \mathbb{R}^m$ are constant vectors
- $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a matrix with entries $a_{ij} \in \{0, 1\}$
- $\mathbf{x} \in \mathbb{R}^n$ is the vector of variables
- Related models: *set partitioning* ($\mathbf{Ax} = \mathbf{1}$), *set packing* ($\mathbf{Ax} \leq \mathbf{1}$)

Example: Installing security telephones

- The road administration wants to install emergency telephones such that each street has access to at least one phone
- It is logical to place the phones at street crossings
- Each crossing has an installation cost: $\mathbf{c} = (2, 2, 3, 4, 3, 2, 2, 1)$
- Find the cheapest selection of crossings to provide all streets with phones

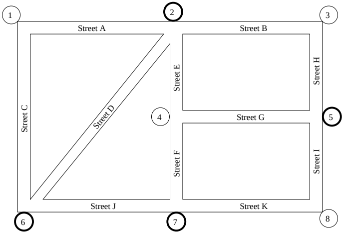


- Define variables and constraints

Installing security telephones: Mathematical model

- Binary variables for each crossing: $x_j = 1$ if a phone is installed at j, $x_j = 0$ otherwise.
- For each street, introduce a constraint saying that a phone should be placed at—at least—one of its crossings:

  A: $x_1 + x_2 \geq 1$, B: $x_2 + x_3 \geq 1$,
  C: $x_1 + x_6 \geq 1$, D: $x_2 + x_6 \geq 1$,
  E: $x_2 + x_4 \geq 1$, F: $x_4 + x_7 \geq 1$,
  G: $x_4 + x_5 \geq 1$, H: $x_3 + x_5 \geq 1$,
  I: $x_5 + x_8 \geq 1$, J: $x_6 + x_7 \geq 1$,
  K: $x_7 + x_8 \geq 1$



- Objective function:
  min $2x_1 + 2x_2 + 3x_3 + 4x_4 + 3x_5 + 2x_6 + 2x_7 + x_8$
- An optimal solution: $x_2 = x_5 = x_6 = x_7 = 1$, $x_1 = x_3 = x_4 = x_8 = 0$. Objective value: 9.

- Given three telephone companies A, B and, C which charge a fixed start-up price of 16, 25 and, 18, respectively
- For each minute of call-time A, B, and, C charge 0.25, 0.21 and, 0.22
- We want to phone 200 minutes. Which company should we choose?
- $x_i =$ number of minutes called by $i \in \{A, B, C\}$
- Binary variables $y_i = 1$ if $x_i > 0$, $y_i = 0$ otherwise (pay start-up price only if calls are made with company i)
- Mathematical model

$$\begin{array}{llcll} \min & 0.25x_1 + 0.21x_2 + 0.22x_3 & + & 16y_1 + 25y_2 + 18y_3 & \\ \text{subject to} & x_1 + x_2 + x_3 & = & 200 & \\ & 0 \leq x_i & \leq & 200y_i, & i = 1, 2, 3 \\ & y_i & \in & \{0, 1\}, & i = 1, 2, 3 \end{array}$$

More modelling examples (2) (Ch. 13.9)

- We wish to process three jobs on one machine
- Each job j has a processing time p_j, a due date d_j, and a penalty cost c_j if the due date is missed
- How should the jobs be scheduled to minimize the total penalty cost?

Job	Processing time (days)	Due date (days)	Late penalty \$/day
1	5	25	19
2	20	22	12
3	15	35	34

MODEL ON THE BOARD!

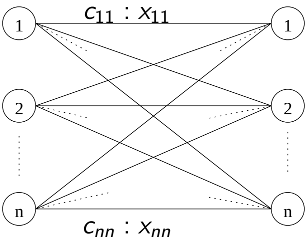
Assign each task to one resource, and each resource to one task

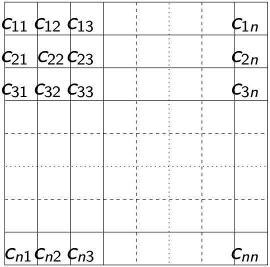
- Linear cost c_{ij} for assigning task i to resource j, $i, j \in \{1, \dots, n\}$
- Variables: $x_{ij} = \begin{cases} 1, & \text{if task } i \text{ is assigned to resource } j \\ 0, & \text{otherwise} \end{cases}$

$$
\begin{array}{lrcll}
\min & \displaystyle\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} & & & \\
\text{subject to} & \displaystyle\sum_{j=1}^{n} x_{ij} & = & 1, & i = 1, \dots, n \\
& \displaystyle\sum_{i=1}^{n} x_{ij} & = & 1, & j = 1, \dots, n \\
& x_{ij} & \geq & 0, & i, j = 1, \dots, n
\end{array}
$$

The assignment model

- Choose *one* element from each row and each column

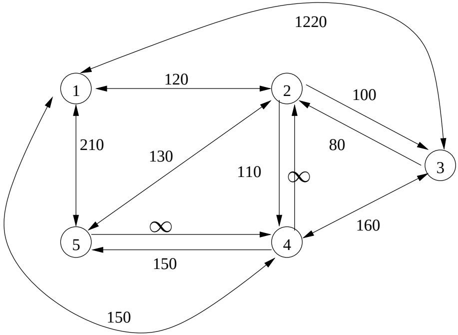




- This integer linear model has integral extreme points, since it can be formulated as a network flow problem
- Therefore, it can be efficiently solved using specialized (network) linear programming techniques
- Even more efficient special purpose (primal–dual–graph-based) algorithms exist

The travelling salesperson problem (TSP) (Ch. 13.10)

- Given n cities and connections between all cities (distances on each connection)
- Find shortest tour that passes through all the cities



- A problem that is very easy to describe and understand but very difficult to solve (combinatorial explosion)
- ∃ different versions of TSP: Euclidean, metric, symmetric, ...

An ILP formulation of the TSP problem

- Let the distance from city i to city j be d_{ij}
- Introduce binary variables x_{ij} for each connection
- Let $V = \{1, \ldots, n\}$ denote the set of nodes (cities)

$$
\begin{array}{llcll}
\min & \displaystyle\sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij}, & & & \\
\text{s.t.} & \displaystyle\sum_{j \in V} x_{ij} & = & 1, & i \in V, \quad (1) \\
& \displaystyle\sum_{i \in V} x_{ij} & = & 1, & j \in V, \quad (2) \\
& \displaystyle\sum_{i \in U, j \in V \setminus U} x_{ij} & \geq & 1, & \forall U \subset V : 2 \leq |U| \leq |V| - 2, \quad (3) \\
& x_{ij} & & \text{binary} & i, j \in V
\end{array}
$$

- Cf. the assignment problem $\boxed{\text{DRAW GRAPH * 2 !}}$
- Enter and leave each city exactly once $\Leftrightarrow$ (1) and (2) $\boxed{\text{DRAW!}}$
- Constraints (3): *subtour elimination* $\boxed{\text{DRAW!}}$
- Alternative formulation of (3): $\boxed{\text{DRAW!}}$
  $\sum_{(i,j) \in U} x_{ij} \leq |U| - 1, \quad \forall U \subset V : 2 \leq |U| \leq |V| - 2$