MVE165/MMG631
Linear and integer optimization with applications
Lecture 7
Discrete optimization models and applications; complexity

Ann-Brith Strömberg

2018–04–17

Recall the diet problem

- Sets
  - $\mathcal{J} = \{1, \dots, n\}$ — kinds of food
  - $\mathcal{I} = \{1, \dots, m\}$ — kinds of nutrients
- Variables
  - x_j, $j \in \mathcal{J}$ — purchased amount of food j per day
- Parameters
  - c_j, $j \in \mathcal{J}$ — cost of food j
  - a_j, $j \in \mathcal{J}$ — available amount of food j
  - p_{ij}, $i \in \mathcal{I}$, $j \in \mathcal{J}$ — content of nutrient i in food j
  - q_i — lower limit on the amount of nutrient i per day
  - Q_i — upper limit on the amount of nutrient i per day

The diet problem

The linear optimization model

$$
\begin{array}{llll}
\underset{x}{\text{minimize}} & & \displaystyle\sum_{j=1}^{n} c_j x_j, & \\
\text{subject to} & q_i \leq & \displaystyle\sum_{j=1}^{n} p_{ij} x_j \leq Q_i, & i = 1, \ldots, m, \\
& 0 \leq & x_j \leq a_j, & j = 1, \ldots, n.
\end{array}
$$

- What if we are allowed to buy at most k different kinds of food, where $k \leq n$?
- Define new variables: $y_j = \begin{cases} 1 & \text{if food } j \text{ is in the diet} \\ 0 & \text{otherwise} \end{cases}$
- Model the following relations:

$$
\begin{array}{rcl}
y_j = 0 & \Longrightarrow & x_j = 0 \\
y_j = 1 & \Longrightarrow & x_j \geq 0
\end{array}
$$

The cardinality constrained diet problem

- Add a *cardinality constraint*: $\sum_{j=1}^{n} y_j \leq k$
- Modify the availability constraints: $0 \leq x_j \leq a_j y_j$

An integer (binary) linear optimization model

$$
\begin{array}{llll}
\underset{x,y}{\text{minimize}} & & \displaystyle\sum_{j=1}^{n} c_j x_j, & \\
\text{subject to} & q_i \leq & \displaystyle\sum_{j=1}^{n} p_{ij} x_j \leq Q_i, & i = 1, \ldots, m, \\
& & \displaystyle\sum_{j=1}^{n} y_j \leq k, & \\
& 0 \leq & x_j \leq a_j y_j, & j = 1, \ldots, n, \\
& & y_j \in \{0, 1\}, & j = 1, \ldots, n.
\end{array}
$$

The cardinality constrained diet problem—an instance

- Buy at most k types of food
- *Totally* $n{=}20$ *types of food*: SourMilk, Milk, Potato, Carrot, HaricotVerts, GreenBeans, Spinache, Tomato, Cabbage, Banana, Queenberries, OrangeJuice, Chicken, Salmon, Cod, Rice, Pasta, Egg, Apple, Ham
- *Constraints on* $m{=}13$ *nutrients*: Energy, Carbohydrates, Fat, Protein, Fibres, SaturFat, SingleUnsaturFat, MultiUnsaturFat, VitaminD, VitaminC, Folate, Iron, Salt

Optimal solutions for $k \in \{20, 10\}$

k	20	10
Apple	3	3
Banana	2	2
Carrot	2.3	3
Chicken	0.4	– –
Egg	2	2
HaricotVerts	0.1	– –
Milk	3	3
Pasta	2	2
Potato	2.3	2.4
Rice	1	1
Salmon	0.5	0.8
SourMilk	2	2

For $k \leq 9$ *no feasible solution exists*

Variables

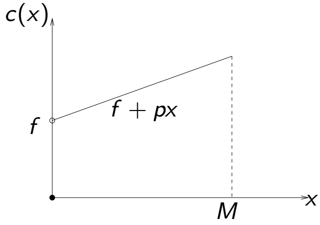
- *Linear programming* (LP) uses continuous variables: $x_{ij} \geq 0$
- *Integer linear programming* (ILP) uses *integer* variables: $x_{ij} \in \mathbb{Z}$
- *Binary linear programming* (BLP) uses *binary* variables: $x_{ij} \in \mathbb{B}$
- If *both* continuous and integer/binary variables are used in a program, it is called a *mixed integer/binary linear program* (MILP)/(MBLP)

Constraints

- An ILP (or MILP) possesses linear constraints and integer requirements on the variables
- Also logical relations, e.g., *if–then* and *either–or*, can be modelled
- This is done by introducing additional (binary) variables and additional constraints

MILP modelling—fixed charges

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



The cost function $c : \mathbb{R}_+ \mapsto \mathbb{R}_+$ is *nonlinear* and *discontinuos*

$$c(x) := \begin{cases} 0 & \text{if} \quad x = 0 \\ f + px & \text{if} \quad 0 < x \leq M \end{cases}$$

MILP modelling—fixed charges

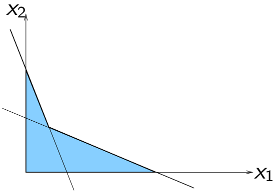
- Let $y = \#$ trucks to send (here, y equals 0 or 1)
- Replace $c(x)$ by $fy + px$
- Constraints: $0 \leq x \leq My$ and $y \in \{0, 1\}$

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

- Suspose:
  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 $y \in \{0, 1\}$

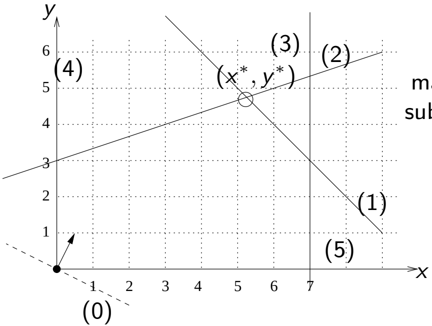
$\Rightarrow$ New set of constraints:
$$\left[\begin{array}{rcl} x_1 + 2x_2 & -My & \leq 4 \\ 5x_1 + 3x_2 & -M(1-y) & \leq 10 \\ & y & \in \{0,1\} \\ x_1, x_2 & & \geq 0 \end{array}\right]$$

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

Exercises: Homework

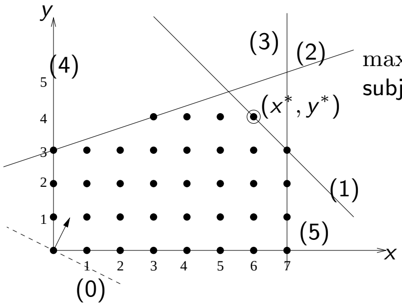
1. Suppose that you are interested in choosing from a set of investments $\{1, \ldots, 7\}$ using 0/1 variables. Model the following constraints:
   1. You cannot invest in all of them
   2. You must choose at least one of them
   3. Investment 1 cannot be chosen if investment 3 is chosen
   4. Investment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

Linear programming: A small example



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

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

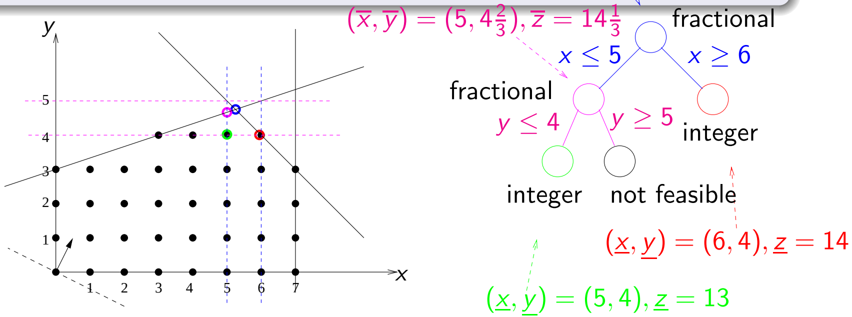


$$
\begin{array}{lrcrcrr}
\text{maximize} & x & + & 2y & & & (0) \\
\text{subject to} & x & + & y & \leq & 10 & (1) \\
& -x & + & 3y & \leq & 9 & (2) \\
& x & & & \leq & 7 & (3) \\
& & & x, y & \geq & 0 & (4,5) \\
& & & x, y & \text{integer} & &
\end{array}
$$

- What if the variables must take integer values?
- 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 possess only 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



For n binary variables: $\leq 2^n$ branches in the search tree

The knapsack problem—budget constraint (Ch. 13.2)

- Select an optimal collection of objects or investments or projects or ...
  - 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—the knapsack problem (Ch 2.6)

A small knapsack instance

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

- 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

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

- 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!

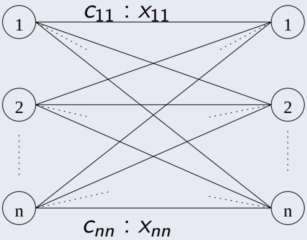
Assign each task to one resource, and each resource to one task

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

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

The assignment model

Choose *one* element from each row and each column



c_{11}	c_{12}	c_{13}					c_{1n}
c_{21}	c_{22}	c_{23}					c_{2n}
c_{31}	c_{32}	c_{33}					c_{3n}
c_{n1}	c_{n2}	c_{n3}					c_{nn}

- This integer linear model has *integral extreme points*, since it can be formulated as a network flow problem (Lect. 11–12)
- Therefore, it can be efficiently solved using specialized (network) linear programming techniques
- Even more efficient special purpose (primal–dual–graph-based) algorithms exist

Computational complexity

- Mathematical insight yields efficient algorithms
- E.g., the *assignment problem*
  - $\#$ feasible solutions: $n!$ $\Longrightarrow$ Combinatorial explosion
  - An algorithm $\exists$ that solves this problem in time $\mathcal{O}(n^4) \propto n^4$

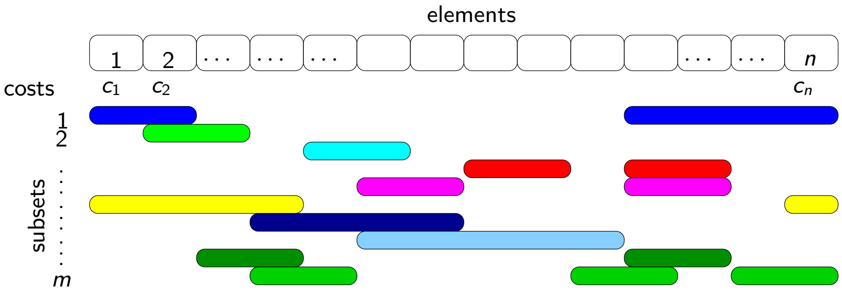
Complete enumeration of all solutions is *not* efficient

n	2	5	8	10	100	1000
$n!$	2	120	40 000	3 600 000	$9.3 \cdot 10^{157}$	$4.0 \cdot 10^{2567}$
2^n	4	32	256	1 024	$1.3 \cdot 10^{30}$	$1.1 \cdot 10^{301}$
n^4	16	625	4 100	10 000	$1.0 \cdot 10^{8}$	$1.0 \cdot 10^{12}$
$n \log n$	0.6	3.5	7.2	10	200	3 000

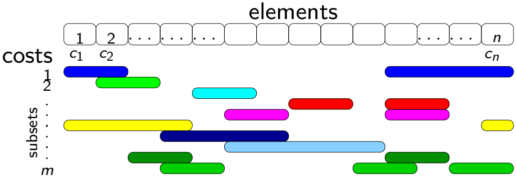
- Binary knapsack: $\mathcal{O}(2^n)$
- Continuous knapsack (sorting of $\frac{c_j}{a_j}$): $\mathcal{O}(n \log n)$

Set covering problem—exponential complexity (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

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

- $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{1} = (1, \ldots, 1)^\top \in \mathbb{R}^m$ 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* ($\mathbf{A}\mathbf{x} = \mathbf{1}$), *set packing* ($\mathbf{A}\mathbf{x} \leq \mathbf{1}$)