Applied optimization

Lecture 7

Linear integer optimization

Ann-Brith Strömberg

6 februari 2002

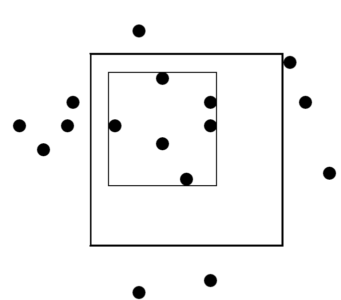
Screening of smear tests (granska cellprover)

- Prevent cancer in the womb (livmoderhalscancer)
- Regular examinations of all women above the age of 18
- Manual screening of each smear test using a microscope
- Prescreening using graphics processing $\Rightarrow \leq 50000$ points that must be manually screened
- ≈ 300 pictures/smear test (as few as possible $\Rightarrow$ more time for each picture)
- Screen the pictures in the right order (automatically by the microscope)
- Optimization?

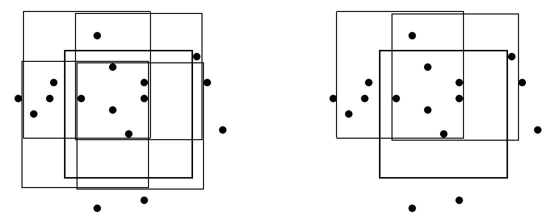
A smear test and an initial grid

- Totally 36 246 points and 392 squares (pictures)
- Can we decrease the *number* of pictures that have to be screened?

The smallest rectangle that covers all points in a square



Generation of alternative squares



The smear test and all square-candidates

- Totally 1610 square candidates
- Find the least number of squares to cover all the points

Mathematical model

The coefficient $\alpha_{kj} = \begin{cases} 1 & \text{if square } j \text{ covers point } k \\ 0 & \text{otherwise} \end{cases}$

The variable $x_j = \begin{cases} 1 & \text{if square } j \text{ is chosen} \\ 0 & \text{otherwise} \end{cases}$

Cover each point with at least one square: (SET COVERING)

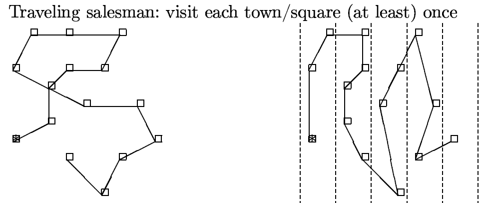
$$\begin{array}{lll} \min & \sum_j x_j & \\ \text{s.t.} & \sum_j \alpha_{kj} x_j \geq 1 & \text{for all } k \\ & x_j \in \{0,1\} & \text{for all } j \end{array}$$

Smear test with "minimum" number of squares

- 36 246 points are covered by 339 squares
- $\approx 13\%$ fewer than the original 392

In what order should the squares be screened?

- Minimize the sum of distances between successive squares
- Visit each square once
- Traveling salesman: visit each town/square (at least) once



Nearest neighbour-heuristic — Strip-heuristic

No guarantee to find an optimal solution

Screening order from the strip-heuristic

When are integer models needed?

- Products or raw materials are indivisible
- Logical constraints: "if A then B"; "A or B"
- Fixed costs
- Combinatorics (sequencing, allocation)
- On/off-decision to buy, invest, hire, generate electricity, ...

Either $0 \leq x \leq 1$ or $x \geq 7$

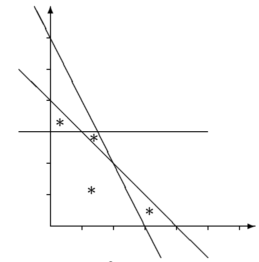
Let $M \gg 1$

$$\begin{aligned} x &\leq 1 + My \\ x &\geq 7y \\ y &\in \{0,1\} \end{aligned}$$

Variable x may only take the values 2, 45, 78 & 107

$$x = 2y_1 + 45y_2 + 78y_3 + 107y_4$$
$$y_1 + y_2 + y_3 + y_4 = 1$$
$$y_1, y_2, y_3, y_4 \in \{0,1\}$$

At least 2 of 3 constraints must be fulfilled



$*=$ feasible regions

$M \geq 2$

$$\begin{aligned} x_1 + x_2 &\leq 4 & (1) \\ 2x_1 + x_2 &\leq 6 & (2) \\ x_2 &\leq 3 & (3) \\ \text{and } x_1, x_2 &\geq 0 \end{aligned}$$

$$\begin{aligned} x_1 + x_2 &\leq 4 + M(1-y_1) & (1) \\ 2x_1 + x_2 &\leq 6 + M(1-y_2) & (2) \\ x_2 &\leq 3 + M(1-y_3) & (3) \\ y_1 + y_2 + y_3 &\geq 2 \\ y_1, y_2, y_3 &\in \{0,1\} \\ \text{and } x_1, x_2 &\geq 0 \end{aligned}$$

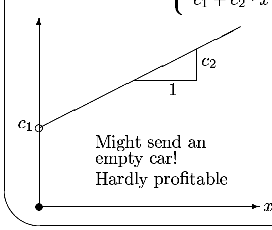
Fixed costs

$x =$ the amount of a certain product to be sent

If $x > 0$ then the initial cost c_1 (e.g. car hire) is generated

Variable cost c_2 per unit sent

Total cost: $f(x) = \begin{cases} 0 & \text{if } x = 0 \\ c_1 + c_2 \cdot x & \text{if } x > 0 \end{cases}$



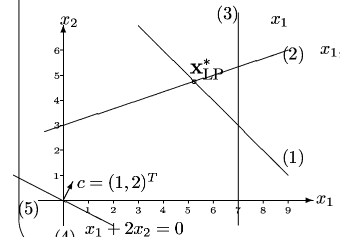
$$\begin{aligned} \text{Let } M &= \text{car capacity} \\ y &= \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \end{cases} \\ f(x,y) &= c_1 \cdot y + c_2 \cdot x \\ x &\leq M \cdot y \\ x &\geq 0, \quad y \in \{0,1\} \end{aligned}$$

Other applications of integer optimization

- Facility location (new hospitals, shopping centres, etc.)
- Scheduling (on machines, personell, projects, for schools)
- Logistics (material- and warehouse control)
- Distribution (transportation of goods, buses for disabled persons (färdtjänst))
- Production planning
- Telecommunication (network design, frequency allocation)
- VLSI-design

Linear continuous optimization model

$$\begin{aligned} \max \quad z_{\text{LP}} &= x_1 + 2x_2 \\ \text{s.t.} \quad & x_1 + x_2 \leq 10 & (1) \\ & -x_1 + 3x_2 \leq 9 & (2) \\ & x_1 \leq 7 & (3) \\ & x_1, x_2 \geq 0 & (4,5) \end{aligned}$$



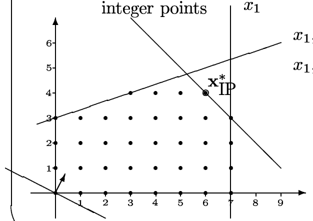
$$\mathbf{x}^*_{\text{LP}} = \begin{pmatrix} 21/4 \\ 19/4 \end{pmatrix}$$

$$z^*_{\text{LP}} = 14 + \tfrac{3}{4}$$

Linear integer optimization model

$$
\begin{array}{llrcrcll}
\max & z_{\text{IP}} = & x_1 & + & 2x_2 & & & \\
\text{s.t.} & & x_1 & + & x_2 & \leq & 10 & (1) \\
& & -x_1 & + & 3x_2 & \leq & 9 & (2) \\
& & x_1 & & & \leq & 7 & (3) \\
& & & & x_1, x_2 & \geq & 0 & (4,5) \\
& & & & x_1, x_2 & & \text{integer} &
\end{array}
$$

• = feasible integer points



$$\mathbf{x}^*_{\text{IP}} = \begin{pmatrix} 6 \\ 4 \end{pmatrix}$$

$$z^*_{\text{IP}} = 14 < z^*_{\text{LP}}$$

Combinatorial explosion

Assign n persons to carry out n jobs. # feasible solutions: $n!$

Assume that a feasible solution is evaluated in 10^{-9} seconds

n	2	5	8	10	100	1000
$n!$	2	120	$4.0 \cdot 10^4$	$3.6 \cdot 10^6$	$9.3 \cdot 10^{157}$	$4.0 \cdot 10^{2567}$
⌈time⌉	10^{-8} s	10^{-6} s	10^{-4} s	10^{-2} s	10^{142} yrs	...

Complete enumeration of all solutions is **not** an efficient algorithm!

An algorithm exists that solves this problem in time $\mathcal{O}(n^4) \propto n^4$

n	2	5	8	10	100	1000
n^4	16	625	$4.1 \cdot 10^3$	10^4	10^8	10^{12}
⌈time⌉	10^{-7} s	10^{-6} s	10^{-5} s	$10^{-5} s$	10^{-1} s	17 min

Relaxation and restriction

- **Relaxation:** remove constraints (e.g. integrality requirements) so that the remaining problem is (considerably) easier to solve

⇒ not necessarily a feasible solution

⇒ optimistic estimate of z_{IP}

- **Restriction:** restrict the search area to a subset of the feasible set: solve heuristically

⇒ feasible but not necessarily optimal solution

⇒ pessimistic estimate of z_{IP}

Example: knapsack problem

$$
\begin{array}{rrcl}
z^* = \max & z = 11x_1 + 7x_2 + 8x_3 + 5x_4 & & \\
\text{s.t.} & 3x_1 + 2x_2 + 2x_3 + 3x_4 & \leq & 6 \\
& x_1, x_2, x_3, x_4 & \in & \{0,1\}
\end{array}
$$

Relaxation

Relax the constraints $x_j \in \{0,1\}$ to $0 \leq x_j \leq 1$ for all j

Sort the quotients: $(\frac{11}{3}, \frac{7}{2}, \frac{8}{2}, \frac{5}{3}) \approx (3.67, 3.5, 4, 1.67)$

Priority order: (x_3, x_1, x_2, x_4)

Solution (not feasible): $x_3 = x_1 = 1$, $x_2 = \frac{1}{2}$, $x_4 = 0 \Rightarrow z = 22.5 \geq z^*$

Restriction

Heuristic solution (feasible): $x_3 = x_1 = 1$, $x_2 = x_4 = 0 \Rightarrow z = 19 \leq z^*$

$19 \leq z^* \leq 22$

Bounds on the optimal value

Optimistic estimate from relaxation: (Solve a simpler problem)

$$\begin{bmatrix} \max & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{bmatrix} = z_{\text{LP}} \geq z_{\text{IP}} = \begin{bmatrix} \max & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \text{ integer} \end{bmatrix}$$

Pessimistic estimate from feasible solution: (Use a heuristic)

$$z_{\text{IP}} \geq \mathbf{c}^T\bar{\mathbf{x}} \text{ for all integral } \bar{\mathbf{x}} \geq \mathbf{0} \text{ such that } \mathbf{A}\bar{\mathbf{x}} = \mathbf{b}$$

Try to find tight bounds, upper and lower:

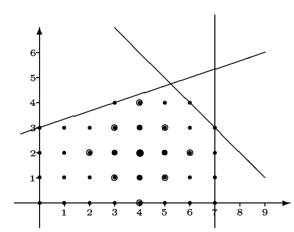
$$\mathbf{c}^T\bar{\mathbf{x}} \leq z_{\text{IP}} \leq z_{\text{LP}}$$

Heuristics

- **Building heuristic**, example:
  - TSP: Nearest neighbour/strip/...
  - SET COVERING: $\max_j(\min_i \alpha_{ij})$ for $j = k \Rightarrow$ choose column k, remove covered rows, repeat
- **Local search:** start from a feasible solution $\bar{x}$
  - NEIGHBOURHOOD OF $\bar{x}$: all feasible solutions within a certain "distance" from $\bar{x}$: $N_\delta(\bar{x}) = \{y \mid \|\bar{x} - y\| \leq \delta, Ay = b, y \geq 0\}$
  - HAMMING-DISTANCE = the number of elements that differ between two 0/1-vectors
    $N_d(\bar{x}) = \{y \in \{0,1\}^n \mid \sum_{j=1}^n |\bar{x}_j - y_j| \leq d, \ y \text{ feasible}\}$

Neighbourhood in a linear integer model

$$
\begin{array}{rcrcl}
x_1 & + & x_2 & \leq & 10 \\
-x_1 & + & 3x_2 & \leq & 9 \\
x_1 & & & \leq & 7 \\
& & x_1, x_2 & \geq & 0 \\
& & x_1, x_2 & & \text{integer}
\end{array}
$$



• = feasible integer points

● = the point $\bar{x}$

• = distance 1 from $\bar{x}$

• = distance 2 from $\bar{x}$

Local search

- Find the best solution $\bar{y} : c^T\bar{y} = \max_{y \in N_d(\bar{x})} c^T y$
  - If $c^T\bar{y} > c^T\bar{x} \Rightarrow$ move to the point $\bar{y}$, repeat from $\bar{y}$: $N_d(\bar{y})$
  - If $c^T\bar{y} \leq c^T\bar{x} \Rightarrow \bar{x}$ is a *local* maximum w.r.t. the neighbourhood, $N_d(\cdot)$, chosen

Optimizing algorithm

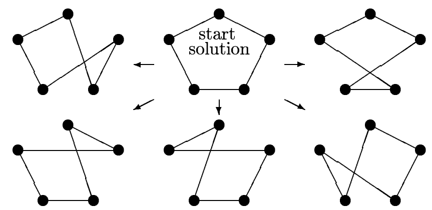
- Specialized algorithms exist for many structured problems (spanning trees, transportation models, matching, ...)
- General (raw) algorithm: Branch–and–bound
  - Relax: remove (complicating) constr. ⇒ optimistic estimate
  - Branch: Partition the feasible set ($x_2 = 0$ or 1)
  - Tree search: deep/breadth/best-first
  - Cut nodes: feasible solution/no solution ∃/cannot be optimal

Local search heuristics

"Given a feasible solution, find a better one in the neighbourhood"

. cf. Steepest descent

Ex: Tavelling salesman "2-change neighbourhood"



Choose the best solution in the neighbourhood and repeat

Improving the screening order using local search