# Lecture 11: Integer programming

Michael Patriksson

24 February 2004





## 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
- Pre-screening using graphics processing $\Rightarrow \leq 50000$ points that must be manually screened
- $\approx 300$ pictures/smear test (as few as possible $\Rightarrow$ more time for each picture)
- Optimization?
- Screen the pictures in the right order (automatically by the microscope)—not in this lecture

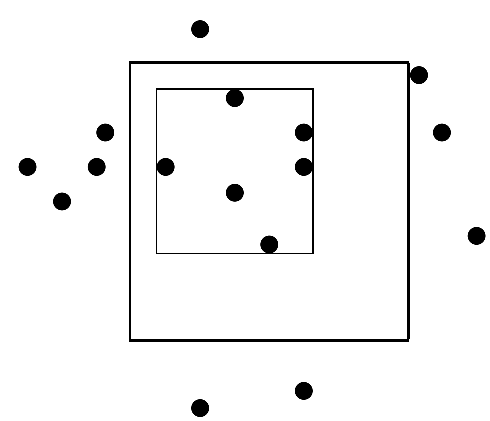


## A smear test and an initial grid

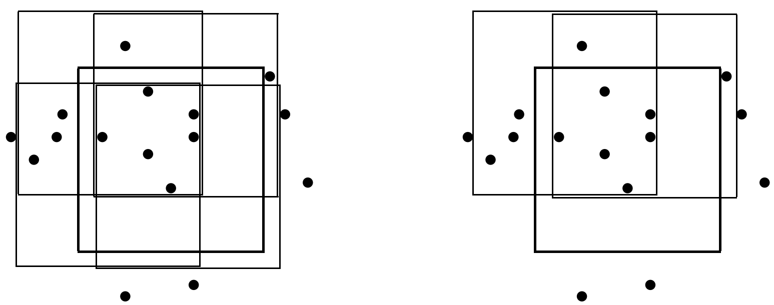
- Totally 36 246 points and 392 squares (pictures)
- Can we decrease the *number* of pictures that have to be screened?



## 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)

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



## Smear test with “minimum” number of squares

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

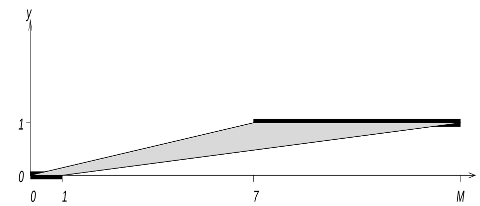
8

## 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, ...

9

## Either $0 \le x \le 1$ or $x \ge 7$




Let $M \gg 1$ : $\quad x \le 1 + My,\ x \ge 7y,\ y \in \{0,1\}$

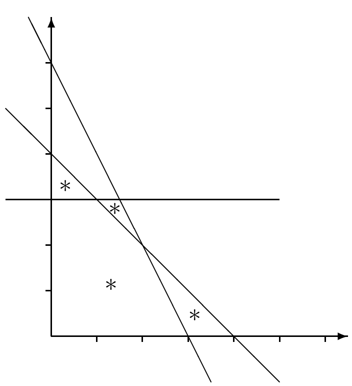
## Variable $x$ may only take the values 2, 45, 78 & 107

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

10

## At least 2 of 3 constraints must be fulfilled

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




$*$ = feasible regions

$M \ge 2$

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

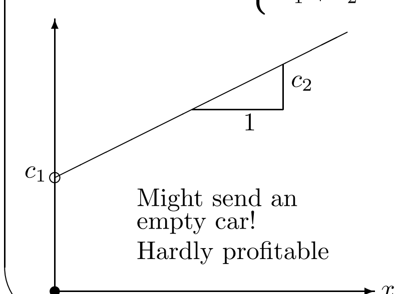
11

## 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: $f(x) = \begin{cases} 0 & \text{if } x = 0 \\ c_1 + c_2 \cdot x & \text{if } x > 0 \end{cases}$ effect / wanted!




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

effect / wanted!

## Other applications of integer optimization

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



## The 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 |
|---|---|---|---|---|---|
| $n!$ | 2 | 120 | $4.0 \cdot 10^4$ | $3.6 \cdot 10^6$ | $9.3 \cdot 10^{157}$ |
| ⌈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!

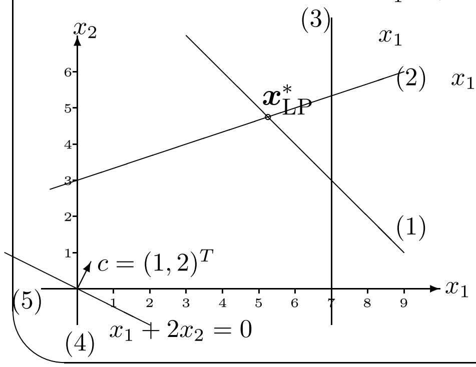
An algorithm exists that solves this problem in time $\mathcal{O}(n^4) \propto n^4$

| $n$ | 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 |



## Linear continuous optimization model

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



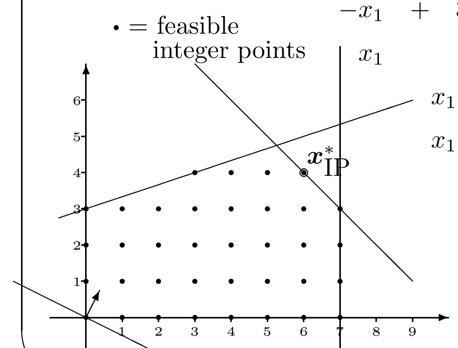

$$x^*_{\text{LP}} = \begin{pmatrix} 21/4 \\ 19/4 \end{pmatrix}$$

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



## Linear integer optimization model

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

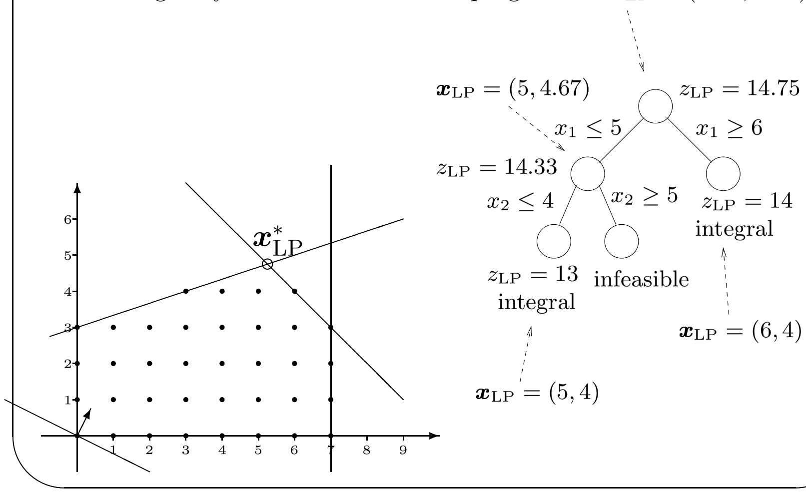



$$x^*_{\text{IP}} = \begin{pmatrix} 6 \\ 4 \end{pmatrix}$$

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

## The branch–and–bound-algorithm

Relax integrality constraints ⇒ linear program ⇒ $\boldsymbol{x}_{\rm LP} = (5.25, 4.75)$






## The complexity of integer optimization: An example

- The Mexico LP has (in the version which is handed out) 113 variables and 84 linear constraints. Solution by a *slow* (333 MHz Unix) computer: 0.01 s.
- We create an integer programming (IP) variant: add a fixed cost for using a railway link for the raw material transport. 78 binary (0/1) variables.
- Cplex uses Branch & Bound (B & B), in which to a continuous relaxation is added integer requirements on some of the integer values that received a fractional value in the LP solution.



- Solution times:

| | | |
|---|---|---|
| Fixed cost | 100 $\Longrightarrow$ | 20 s. |
| | | 18,000 B & B nodes |
| | | 60,000 simplex iterations |
| | 300 $\Longrightarrow$ | 3 min. |
| | | 208,000 B & B nodes |
| | | 650,000 simplex iterations |

- There are $2^{78} \approx 0.3 \cdot 10^{24}$ possible combinations. B & B is good at *implicitly* enumerating them all.



- The higher the fixed cost, the more difficult the problem. Why?
- Continuous relaxation worse and worse approximation.

20

## The Philips example—TSP solved heuristically

- Let $c_{ij}$ denote the distance from city $i$ to city $j$, with $i < j$, and $i, j \in \mathcal{N} = \{1, 2, \dots, n\}$, and
- $x_{ij} = \begin{cases} 1, & \text{if link } (i,j) \text{ is part of the TSP tour,} \\ 0, & \text{otherwise.} \end{cases}$

21

- The Traveling Salesman Problem (TSP):

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} \sum_{j=1:j\neq i}^{n} c_{ij} x_{ij} \\ \text{subject to} \quad & \sum_{i\in\mathcal{S}} \sum_{j\in\mathcal{S}} x_{ij} \leq |\mathcal{S}| - 1, \quad \mathcal{S} \subset \mathcal{N}, \quad (1) \\ & \sum_{i=1}^{n} \sum_{j=1:j\neq i}^{n} x_{ij} = n, \quad (2) \\ & \sum_{i=1}^{n} x_{ij} = 2, \quad j \in \mathcal{N}, \quad (3) \\ & x_{ij} \in \{0,1\}, \quad i, j \in \mathcal{N}. \end{aligned}$$

22

## Interpretations

- Constraint (1) implies that there can be no *sub-tours*, that is, a tour where fewer than $n$ cities are visited (that is, if $\mathcal{S} \subset \mathcal{N}$ then there can be at most $|\mathcal{S}| - 1$ links between nodes in the set $\mathcal{S}$, where $|\mathcal{S}|$ is the cardinality–number of members of–the set $\mathcal{S}$);
- Constraint (2) implies that in total $n$ cities must be visited;
- Constraint (3) implies that each city is connected to two others, such that we make sure to arrive from one city and leave for the next.

23

## Lagrangian relaxation

- TSP is NP-hard—no known polynomial algorithms exist
- Lagrangian relax (3) for all nodes except starting node
- Remaining problem: 1-MST—find the minimum spanning tree in the graph without the starting node and its connecting links; then, add the two cheapest links to connect the starting node

- Objective function of the Lagrangian problem:

$$q(\boldsymbol{\lambda}) = \underset{\boldsymbol{x}}{\text{minimum}} \sum_{i=1}^{n} \sum_{j=1:j\neq i}^{n} c_{ij}x_{ij} + \sum_{j=2}^{n} \lambda_j \Big(2 - \sum_{i=1:i\neq j}^{n} x_{ij}\Big)$$

$$= 2\sum_{j=1}^{n} \lambda_j + \underset{\boldsymbol{x}}{\text{minimum}} \sum_{i=1}^{n} \sum_{j=1:j\neq i}^{n} (c_{ij} - \lambda_i - \lambda_j)x_{ij}.$$

- A high (low) value of the multiplier $\lambda_j$ makes node $j$ attractive (unattractive) in the 1-MST problem, and will therefore lead to more (less) links being attached to it.
- Subgradient method for updating the multipliers.



- Updating step:

$$\lambda_j := \lambda_j + \alpha \left(2 - \sum_{i=1:i\neq j}^{n} x_{ij}\right), \qquad j = 2, \dots, n,$$

where $\alpha > 0$ is a step length.

- Update means:

Current degree at node $j$ :

$$\begin{cases} > 2 \Longrightarrow \lambda_j \downarrow \text{(link cost } \uparrow) \\ = 2 \Longrightarrow \lambda_j \leftrightarrow \text{(link cost constant)} \\ < 2 \Longrightarrow \lambda_j \uparrow \text{(link cost } \downarrow) \end{cases}$$

Link cost shifted upwards (downwards) if too many (too few) links connected to node $j$ in the 1-MST.



## Feasibility heuristic

- Adjusts Lagrangian solution $\boldsymbol{x}$ such that it becomes feasible.
- Often a good thing to do when approaching the dual optimal solution—$\boldsymbol{x}$ often then only mildly infeasible.
- Identify path in 1-MST with many links; form a subgraph with the remaining nodes which is a path; connect the two.
- Result: A Hamiltonian cycle (TSP tour).
- We then have both an upper bound (feasible point) and a lower bound ($q$) on the optimal value—a quality measure!



## The Philips example

- Fixed number of subgradient methods.
- Feasibility heuristic used every $K$ iterations ($K > 1$), starting at a late subgradient iteration.
- Typical example: Optimal path length in the order of 2 meters; upper and lower bounds produced concluded that the relative error in the production plan is *less than 7 %*.
- Also: increase in production by some 70 %.