Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

Algorithms for Decision Support

Integer linear programming models

People with reduced mobility (PRM) require assistance when travelling through the airport

http://www.schiphol.nl/Travellers/AtSchiphol/InformationForPassengersWithReducedMobility.htm

PRMs: Possible techniques for decision support

1. Management considers to change the locations of the lounges and to extend the number of lounges
2. Management want to find good strategies for assigning PRMs to employees
    1. Define a few different options
    2. Imitate the process of supervising PRMs in the computer
    3. Run each option and compute the performance
    4. Compare the performance to the currrent situation and select the best option

- **Discrete-event simulation**

PRMs: Possible techniques for decision support (2)

1. Management wants to find the best locations of the lounges
2. Management wants to determine the assignment of PRM’s to employees such the waiting time for the PRM’s outside the lounges is minimal.

- **Combinatorial optimization**

Combinatorial optimization

- Define variables x representing a schedule
- Express waiting time $w(x)$ in terms of these variables
- Formulate constraints in terms of these variables: $x \in C$
- C contains a finite but very large number of element
- Optimize:

$$\min w(x) \text{ subject to } x \in C$$

Integer linear programming

- Mainstream optimization method in scientific research
- Extremely important optimization algorithm in practice
- Used by Tennet to analyse the system adequacy (leveringszekerheid) of the Dutch electricity network
- Used by U-OV to determine the sequence of trips for each of their buses
- Record for solving Travelling Salesman Problem

Contents

- We forget stochastics for a while
- We move to discrete (or combinatorial) optimization
- Integer linear programming is well-known modelling and solution technique
- First, linear programming
- Modelling integer linear programming problems
- Solving integer linear programming problems by branch-and-bound.
- Branch-and-cut

Example: Ajax

- Three types of computers: Alpha, Beta, and Gamma.
- Net profit: $350,- per Alpha, $470,- per Beta, and $610,- per Gamma.
- Every computer can be sold at the given profit.
- Testing: Alpha and Beta computers on the A-line, Gamma computers on the C-line.
- Testing takes 1 hour per computer.
- Capacity A-line: 120 hours; capacity C-line: 80 hours.
- Required labor: 10 hours per Alpha, 15 hours per Beta, and 20 hours per Gamma.
- Total amount of labor available: 2000 hours.

Example: Ajax

Decision variables: MA number of alpha's produced, etc

$$\max Z = 350\,MA + 470\,MB + 610\,MC$$

Objective function

$$\begin{aligned}
\text{subject to}\quad & MA + MB && \leq 120 \quad (A-\text{line}) \\
& MC && \leq 48 \quad (C-\text{line}) \\
& 10MA + 15MB + 20MC && \leq 2000 \quad (\text{labor}) \\
& MA, MB, MC \geq 0 &&
\end{aligned}$$

Constraints

Linear programming

Min $c^T x$

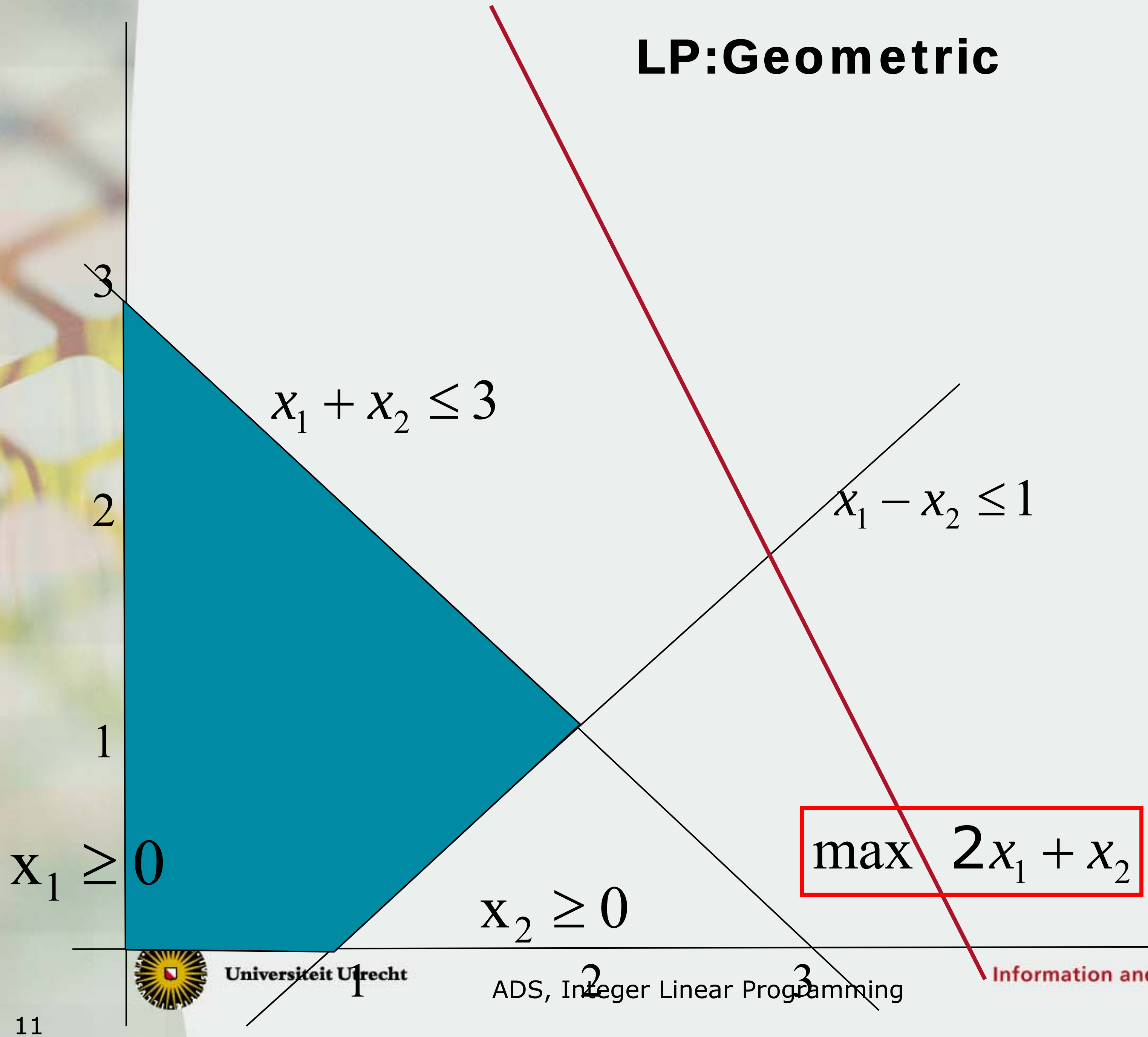
s.t. $Ax \leq b$

$x \geq 0$

With

$x \in \mathbb{R}^n$, $c \in \mathbb{Q}^n$, $\mathrm{A} \in \mathbb{Q}^{m \times n}$, and $b \in \mathbb{Q}^m$

LP:Geometric



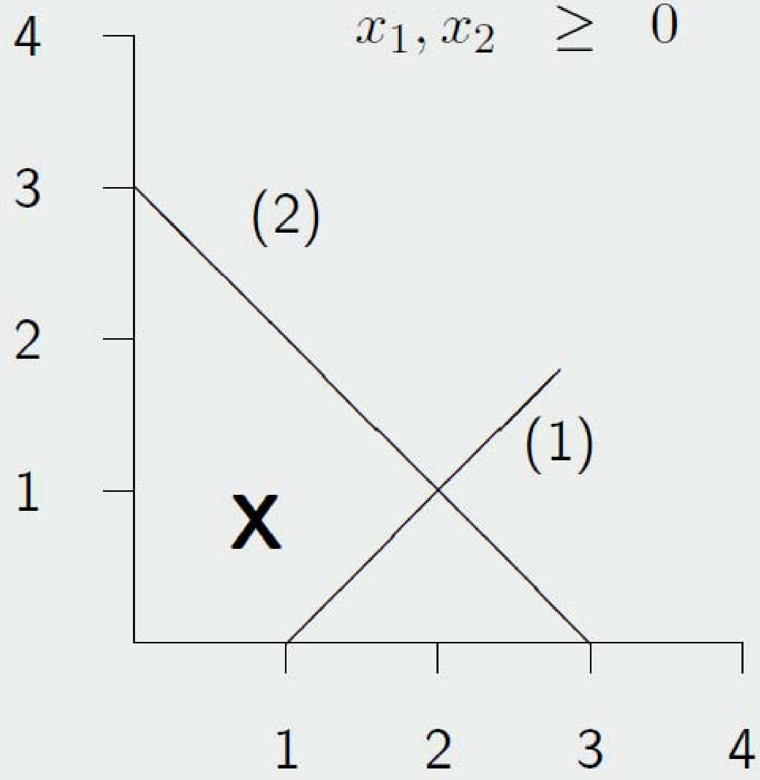
The simplex method

- Example dictionaries

Feasible region

Constraints describing the feasible region $\mathbf{X}$:

$$\begin{aligned} x_1 - x_2 &\leq 1 \\ x_1 + x_2 &\leq 3 \\ x_1, x_2 &\geq 0 \end{aligned}$$



$$\begin{aligned} (1):\quad x_1 - x_2 &= 1 \\ (2):\quad x_1 + x_2 &= 3 \end{aligned}$$

Solving the LP

Use the objective of maximizing $z = 2x_1 + x_2$.

- Introduce slack variables $x_3, x_4 \geq 0$
- New description feasible region:

$$\begin{aligned} x_1 - x_2 + x_3 &= 1 \\ x_1 + x_2 + x_4 &= 3 \\ x_1, x_2, x_3, x_4 &\geq 0 \end{aligned}$$

- The borders correspond to the equations $x_1 = 0, x_2 = 0, x_3 = 0, x_4 = 0$.
- Start with $(x_1, x_2, x_3, x_4) = (0, 0, 1, 3)$ (the origin in the two-dimensional case).

Knowledge

In case of a bounded optimum, there is always an extreme point (vertex of the feasible region) in which the optimum is attained. In an extreme point the number of variables with a positive value is at most equal to the number of equations.

Improving current solution (1)

Problem: maximize z subject to the constraints

$$\begin{aligned} z - 2x_1 - x_2 &= 0 \\ x_1 - x_2 + x_3 &= 1 \\ x_1 + x_2 + x_4 &= 3 \\ x_1, x_2, x_3, x_4 &\geq 0 \end{aligned}$$

Put all variables with value 0 at the right-hand-side. Denote only the equations.

$$\begin{aligned} z &= 2x_1 + x_2 \\ x_3 &= 1 - x_1 + x_2 \\ x_4 &= 3 - x_1 - x_2 \end{aligned}$$

What to do next?

Improving current solution (2)

- Increase the value of *one* variable with current value equal to 0 (x_1 or x_2).
- Increase a variable such that the value of z increases; if this is not possible, then you have found an optimum solution.
- All other variables at the right-hand-side remain equal to 0; adjust the left-hand-side variables according to the equations.

$$\begin{aligned} z &= 2x_1 + x_2 \\ x_3 &= 1 - x_1 + x_2 \\ x_4 &= 3 - x_1 - x_2 \end{aligned}$$

Improving current solution (3)

- The objective equation $z = 2x_1 + x_2$ indicates that increasing x_1 improves the objective value with 2 per unit; increasing x_2 leads to a gain of 1 per unit.
- Choose (greedy) to increase x_1 (walk along the border.)
- Increase x_1 maximally until one of the other variables becomes 0 (you hit at an extreme point).

$$\begin{aligned} z &= 2x_1 + x_2 \\ x_3 &= 1 - x_1 + x_2 \\ x_4 &= 3 - x_1 - x_2 \end{aligned}$$

New point: $x_3 = 0$ and $x_1 = 1$.

Improving current solution (4)

- Reformulate the problem, such that z, x_1, x_4 get expressed in x_2 and x_3 (zero-value variables).
- Use $x_3 = 1 - x_1 + x_2$, that is, $x_1 = 1 - x_3 + x_2$; use this to rearrange the other equations.

$$\begin{aligned} z &= 2 + 3x_2 - 2x_3 \\ x_1 &= 1 + x_2 - x_3 \\ x_4 &= 2 - 2x_2 + x_3 \end{aligned}$$

Improving current solution (5)

$$\begin{aligned} z &= 2 + 3x_2 - 2x_3 \\ x_1 &= 1 + x_2 - x_3 \\ x_4 &= 2 - 2x_2 + x_3 \end{aligned}$$

- Increasing x_2 yields 3 per unit; increasing x_3 costs 2 per unit.
- Increase x_2 maximally $\Longrightarrow$ x_4 drops to 0.
- Adjust again the equations, use $x_4 = 2 - 2x_2 + x_3 \Leftrightarrow x_2 = 1 + 0.5x_3 - 0.5x_4$
- Express z, x_1, x_2 in x_3 and x_4.

$$\begin{aligned} z &= 5 - 0.5x_3 - 1.5x_4 \\ x_1 &= 2 - 0.5x_3 - 0.5x_4 \\ x_2 &= 1 + 0.5x_3 - 0.5x_4 \end{aligned}$$

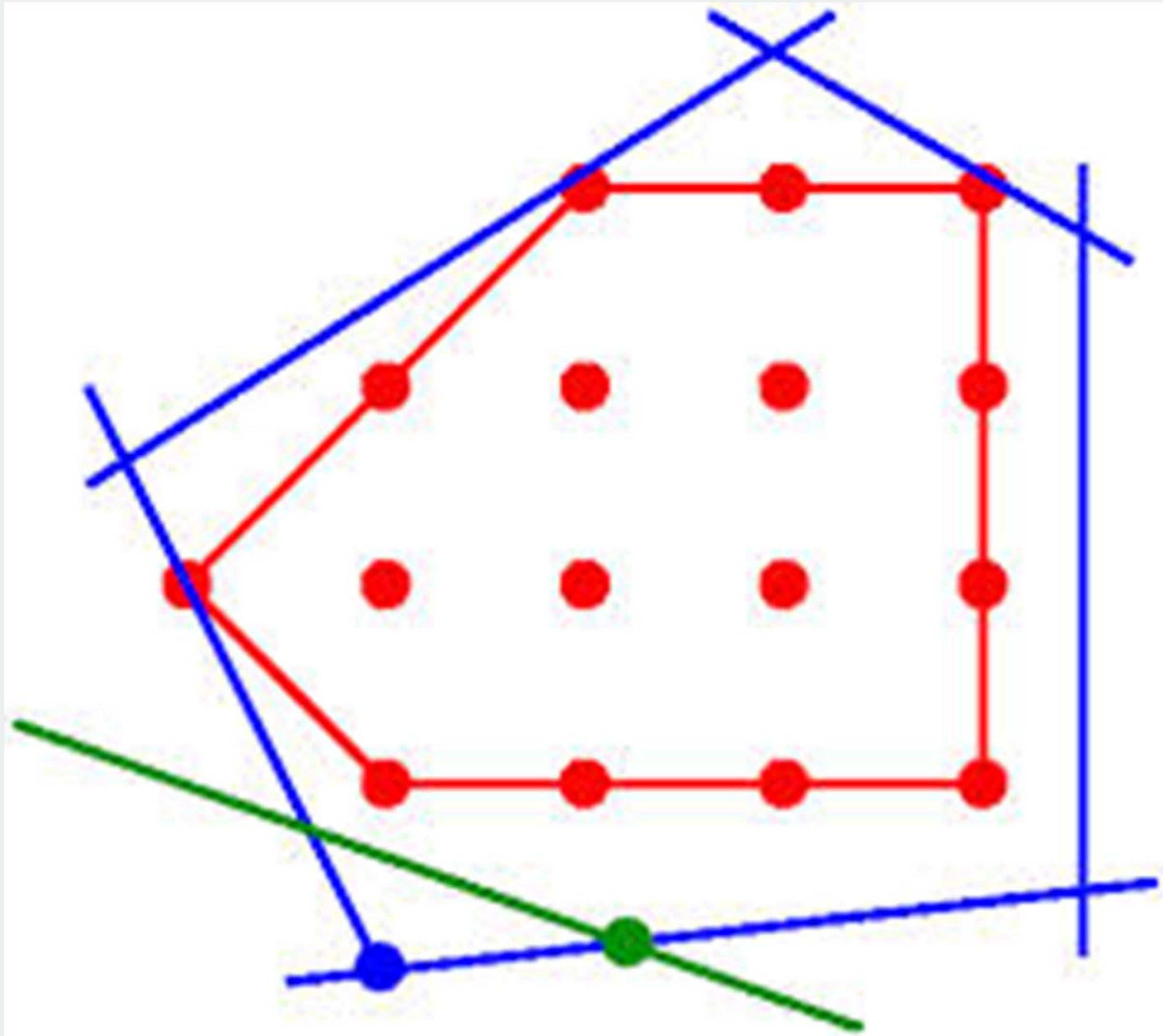
Optimal solution

$(x_1, x_2, x_3, x_4) = (2, 1, 0, 0)$ with value 5.

Solution options

A linear programming problem can

- be infeasible
  - Example:
    $\max\{6x_1 + 4x_2 \mid x_1 + x_2 \leq 3, 2x_1 + 2x_2 \geq 8, x_1, x_2 \geq 0\}$
- be unbounded
  - Example:
    $\max\{6x_1 + 4x_2 \mid x_1 + x_2 \geq 3, 2x_1 + 2x_2 \geq 8, x_1, x_2 \geq 0\}$
    for any $\lambda \geq 2$ we have that $x_1 = \lambda, x_2 = \lambda$ is feasible
- have a bounded optimum



Agenda Tue 8-10

- Last week:
  - Example of solving LP by Simplex method
  - This is to help you understand MIP
  - Execution of Simplex method will not be asked at exam
- Today
  - Introduction MIP
  - Solving MIP by branch-and-bound
  - Modelling

Linear programming

Min $c^T x$

s.t. $Ax \leq b$

$x \geq 0$

With

$x \in \mathbb{R}^n$, $c \in \mathbb{Q}^n$, $\mathrm{A} \in \mathbb{Q}^{m \times n}$, and $b \in \mathbb{Q}^m$

Solution method for linear programming

- Simplex method
  - Slower than polynomial
  - Practical
- Ellipsoid method
  - Polynomial (Khachian, 1979)
  - Not practical
- Interior points methods
  - Polynomial (Karmakar, 1984)
  - Outperforms Simplex for very large instances

Seems not too hard to implement. But, for larger problems you run into numerical problems. Use a standard solver (Gurobi, CPLEX, GLPK)

$$LP \in P$$

Knapsack problem

Knapsack with volume 15
What should you take with you to maximize utility?

Item	1:paper	2:book	3:bread	4:smart-phone	5:water
Utility	8	12	7	15	12
Volume	4	8	5	2	6

Knapsack problem (2)

x_1 = 1 if item 1 is selected, 0 otherwise, x_2, ...

max $z = 8x_1 + 12x_2 + 7x_3 + 15x_4 + 12x_5$

subject to

$$4x_1 + 8x_2 + 5x_3 + 2x_4 + 6x_5 \leq 15$$
$$x_1, x_2, x_3, x_4, x_5 \in \{0,1\}$$

Knapsack problem (3)

- n items, knapsack volume b
- Item j has
  - Utility (revenue) c_j
  - Volume (weight) a_j
- MIP formulation
  - Decision variables: $x_j = 1$ if item j is selected and $x_j = 0$ otherwise

$$\max \sum_{j=1}^{n} c_j x_j$$

s.t.

$$\sum_{j=1}^{n} a_j x_j \leq b$$

$$x_j \in \{0,1\} \qquad (j = 1, \ldots, n)$$

(Mixed) Integer linear Programming (MIP)

Min $c^T x + d^T y$

$s.t.\ Ax + By \leq b$

$x, y \geq 0$

x integral (or binary)

Extension of LP:

Success factor!

- Good news: more possibilities for modelling
  - Many problems from transportation, logistics, rostering, health care planning etc
- Bad news: larger solution times

Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

(Mixed) Integer linear program

Min $c^T x$

$s.t.\ Ax + By \leq b$

$x, y \geq 0$

x integral

(or binary)

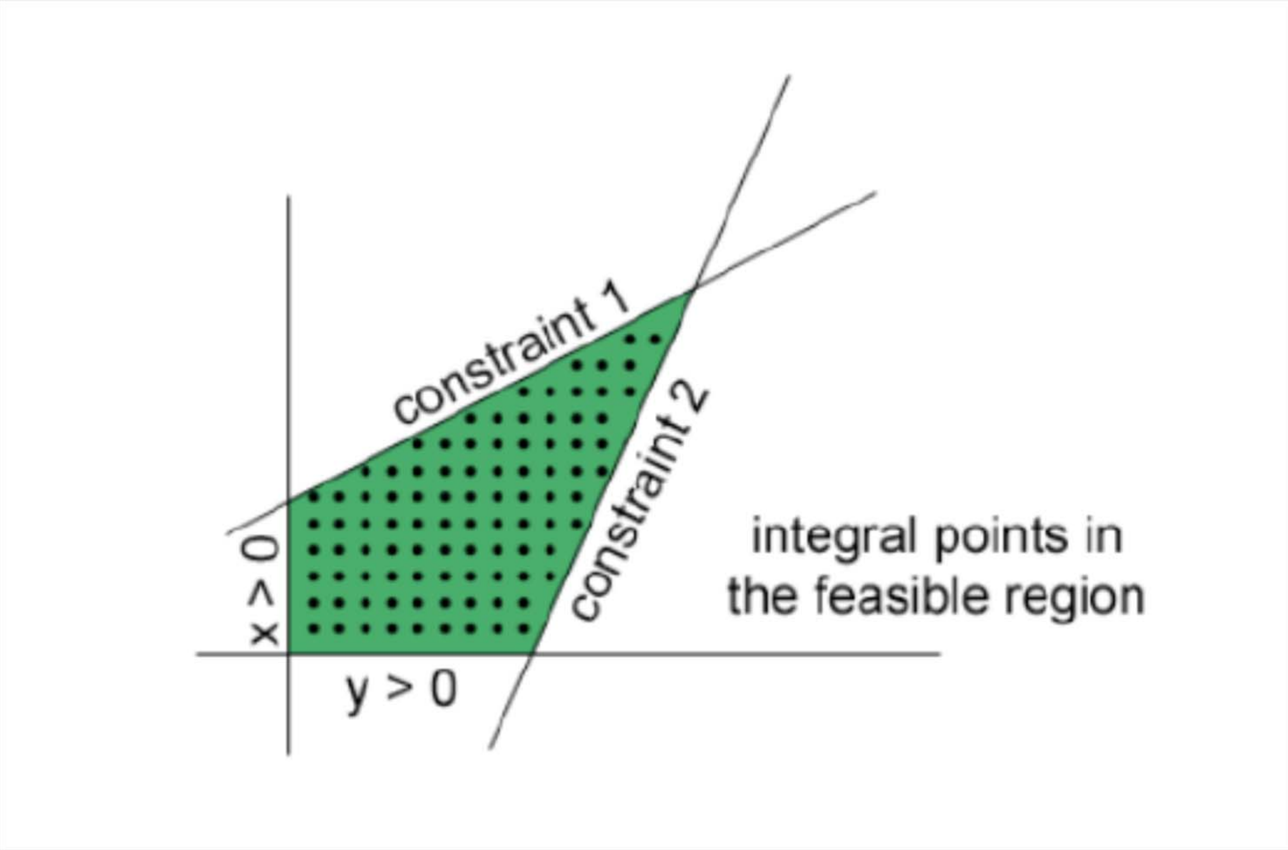
LP-relaxation

Min $c^T x$

$s.t.\ Ax + By \leq b$

$x, y \geq 0$

Lower bound (or upper bound in case of maximization)


constraint 1
constraint 2
x > 0
y > 0
integral points in
the feasible region

Knapsack problem

- n items, knapsack volume b
- Item j has
  - Utility (revenue) c_j
  - Volume (weight) a_j
- MIP formulation
  - Decision variables: $x_j = 1$ if item j is selected and $x_j = 0$ otherwise

$$\max \sum_{j=1}^{n} c_j x_j$$

s.t.

$$\sum_{j=1}^{n} a_j x_j \leq b$$

$$x_j \in \{0, 1\} \qquad (j = 1, \dots, n)$$

Knapsack problem: LP-relaxation

- LP-relaxation: Greedy algorithm

Step 0. Order variables such that $\frac{\mathrm{c}_1}{\mathrm{a}_1} \geq \frac{\mathrm{c}_2}{\mathrm{a}_2} \geq \ldots \geq \frac{\mathrm{c}_\mathrm{n}}{\mathrm{a}_\mathrm{n}}$

Step 1. $x_i \leftarrow 0\ \forall_i$; restcapacity $\bar{b} = b; i = 1$

Step 2. If $a_i \leq \bar{b}$, then $x_i \leftarrow 1$, else $x_j \leftarrow \frac{\bar{b}}{a_i}$. Set $\bar{b} \leftarrow \bar{b} - a_i x_i; j \leftarrow j + 1$

Step 3. If $\bar{b} > 0$, go to Step 2.

- Feasible solution: rounding down

How to solve an Integer Linear Program

- First solve the LP-relaxation
  - You may have learned how to do this by the Simplex method in the bachelor course Optimization
  - In this course, we got an idea how to do this from previous lecture
  - There is excellent software for this (even Excel can solve a (not too large) LP)
- If LP-relaxation has integral solution: finished ☺ ☺
- Otherwise, proceed by branch-and-bound

Solving MIP by branch-and-bound: informally (for maximization)

Split into sub problems, e.g. by fixing a variable to 0 or 1. This is *branching*, you get nodes of a tree.

Select a node to evaluate. You can compute:

- upper bound by solving LP-relaxation
- a feasible solution (e.g. by rounding heuristic)

4 options:

1. Infeasible: do not search further from this node
2. LP-relaxation upper bound less than or equal to the best known feasible solution: hopeless node, do not search further here *(bounding)*
3. LP-relaxation has integral solution: this node is completely solved. ☺
4. LP-relaxation upper bound larger than heuristic solution. Search further by splitting (*branching)*

Solving MIP by branch-and-bound

Let x^* be the best known feasible solution and let $v(x^*)$ be its objective value.

1. Select an active sub problem F_i (unevaluated node)
2. Solve LP-relaxation of F_i. If F_i is infeasible: delete node and go to 1
3. Consider upper bound $Z_{LP}(F_i)$ from LP-relaxation and compute feasible solution x_f (e.g. by rounding)
   1. If $Z_{LP}(F_i) \leq v(x)^*$ delete node
   2. If $v(x_f) > v(x^*)$: update x^*
   3. If solution x_{LP} to LP-relaxation is integral, then node finished, otherwise split node into two new active sub problems
4. Go to step 1

Optional

This if for maximization problem.

Modeling

- Decision variables
- Objective function
- Constraints

Assignment problem

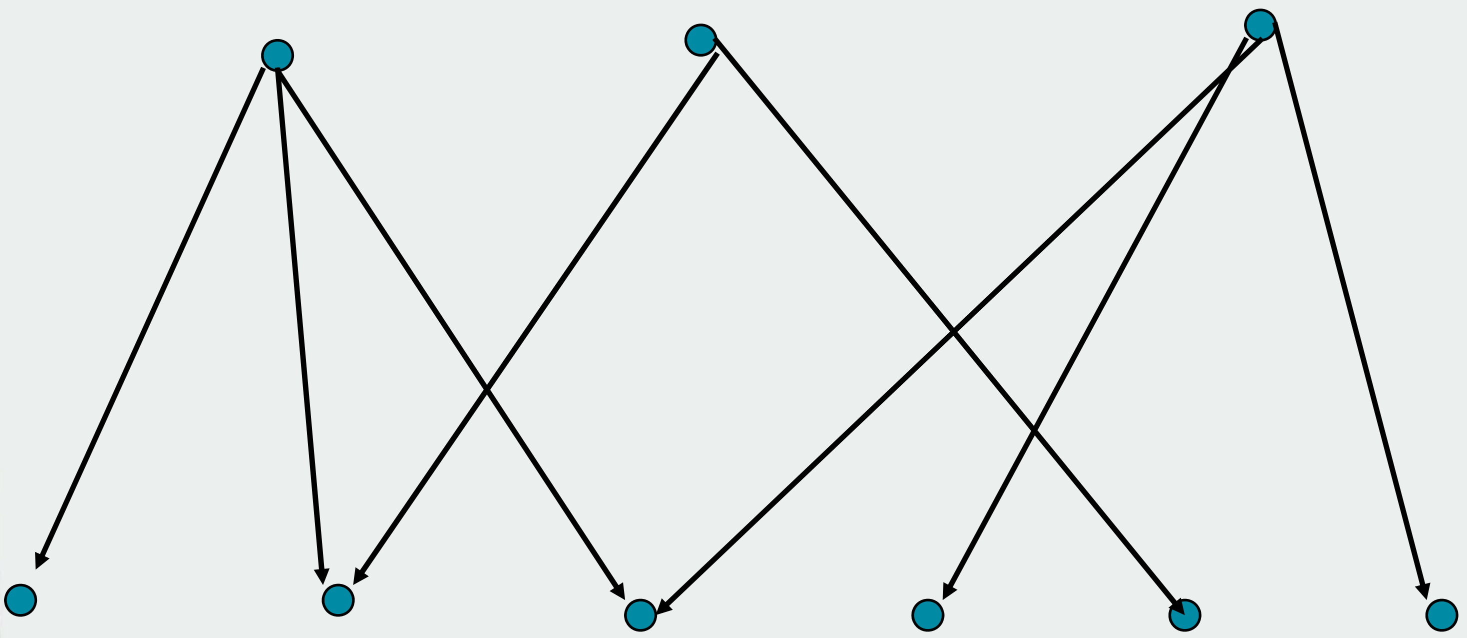
- n persons, n jobs.
- Each person can do at most one job
- Each job has to be executed
- C_{ij} cost if person i performs job j
- We want to minimize cost

Maximum independent set

- Given a graph G=(V,E)
- V: nodes
- E: edges
- An independent set I is a set of nodes, such that every edge has at most one nodes in I, i.e., no pair of nodes in I is connected.
- What is the maximum number of nodes in an independent set?

Facility location

Possible locations: n



Customers: m

Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

Capacitated facility location

- Data:
  - m customers,
  - Customer demand: D_i
  - n possible locations of depots (facilities)
  - c_{ij} cost per unit product transported from depot j to customer i
  - Capacity depot: C_j
  - Fixed cost for opening depot DC: F_j
- Which depots are opened and which customer is served by which depot?

Capacitated facility location:

- Our example shows modelling possibilities with binary variables
- Our model uses binary variables for ***fixed cost***
- Our model uses binary variables ***forcing constraints:***
  - depot can only be used when it is open.

Uncapacitated facility location

- Data:
  - m customers, n possible locations of depot
  - Each customer is assigned to one depot
  - d_{ij} cost of serving customer i by depot j
  - Fixed cost for opening depot DC: F_j
- Which depots are opened and which customer is served by which depot?

Uncapacitated facility location

- Two models *FL* and *AFL*
- Let the polyhedron P_{FL} (P_{AFL}) be the set of feasible solutions for LP-relaxation of *FL (AFL)*
- *FL* and *AFL* have the same optimal value Z_{IP} for the integer formulation
- But

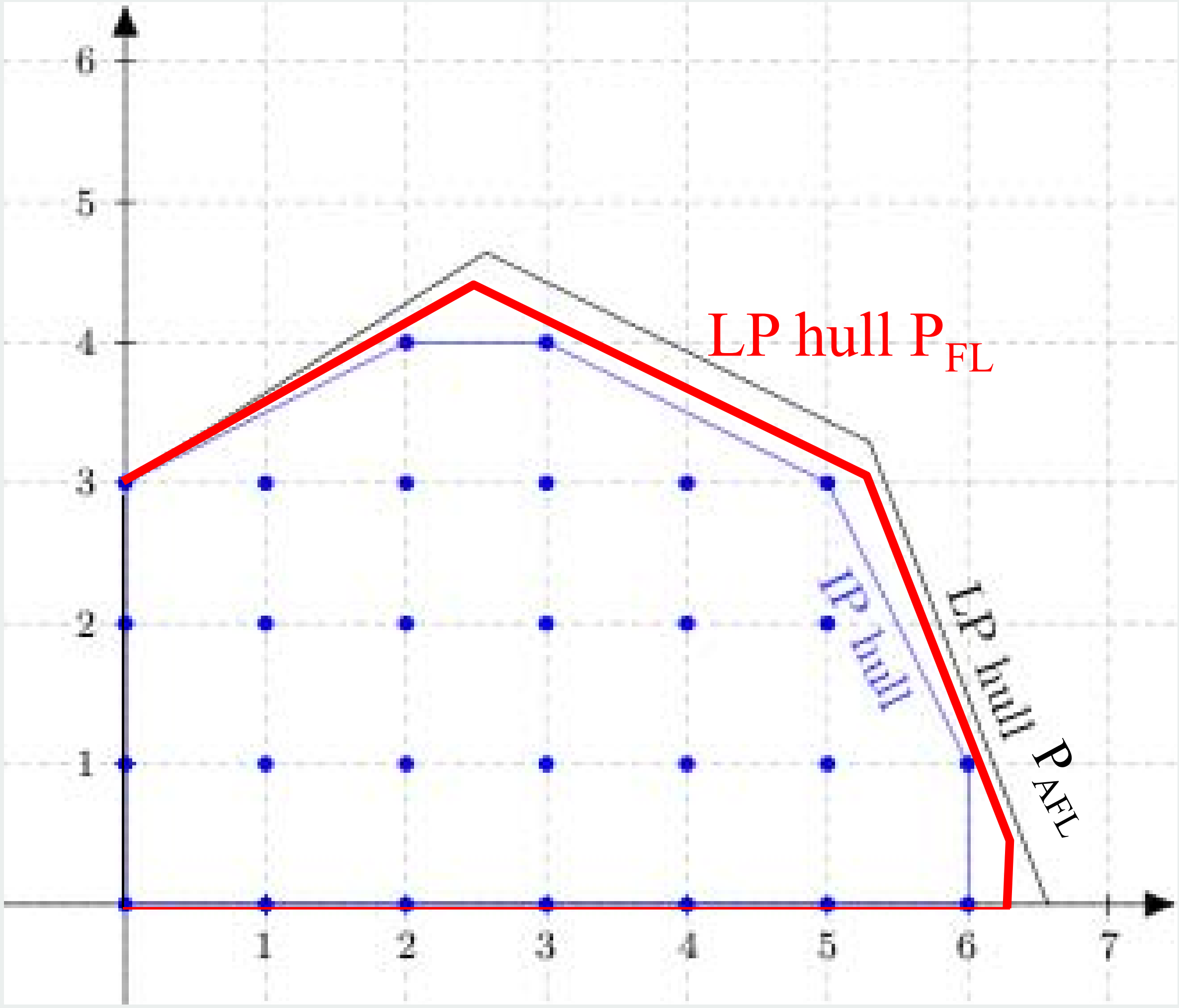
$$P_{FL} \subseteq P_{AFL}$$

- Now, the set of optimal values satisfy:

$$Z_{AFL} \leq Z_{FL} \leq Z_{IP}$$

- The LP-relaxation of *FL* gives a stronger bound
- A stronger bound reduces the number of nodes in the branch-and-bound algorithm

Just to show you the idea



Treasure island

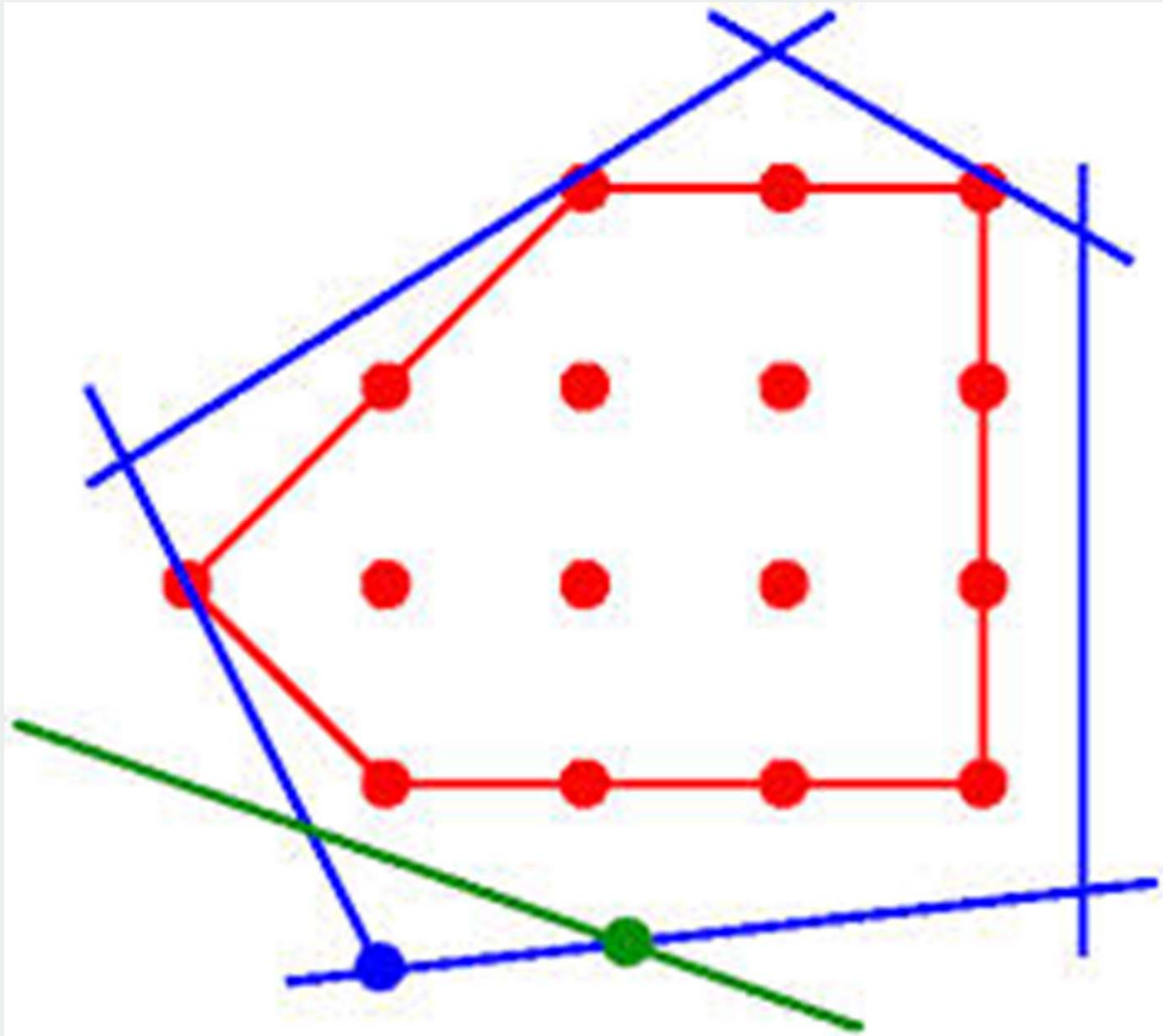
- Diamonds are buried on an island
- Numbers give number of diamonds in neighboring positions (include diagonal)
- At most one diamond per position
- No diamond at position with number

- We now have a feasibility problem

	1								2		2	2	3		2	1	
0				2		1				5					4		2
	0	1			2										5		
	1		2		3			1		4		4					
3					1			1	2		2		3			2	
				3			1	2	4		3					0	
		4			1						3	1			3		
	2	3											1				
3				2	0	0		4		5	2				1		0
		2						3							0		1
		3		2						5		4		3			
2					0			2			3		5				3
	4	4	2	2	2			1		3							3
												3			3		
3			5			4	3						1		2		
2			5						0	1			2				1
	2					2		2			0			1			1
3		2			2					2		3			2		
				2		1						5			1		
	2	2					3	2	2							1	
				1					1	3							
					2			0				5			2	3	3
	0		1			2				1		3			3		
		1			2		2				0			2		3	

Challenge: Treasure island with pitfall

- Like treasure island but exactly one given number is incorrect.



Agenda Thu 10-10

- Last lectures:
  - Example of solving LP by Simplex method
  - This is to help you understand MIP
  - Execution of Simplex method will not be asked at exam
  - Introduction MIP
  - Solving MIP by branch-and-bound
  - Modelling
- Today
  - More modelling: minimum spanning tree, Unit Commitment
  - Cutting planes
  - Branch-and-cut
  - MIP solvers

(Mixed) Integer linear program

Min $c^T x$

$s.t.\ Ax + By \leq b$

$x, y \geq 0$

x integral

(or binary)

LP-relaxation

Min $c^T x$

$s.t.\ Ax + By \leq b$

$x, y \geq 0$

Lower bound (or upper bound in case of maximization)

Minimum spanning tree

- G=(N,E)
- N set of *n* nodes
- E set of *m* edges
- C_e cost of edge *e*
- Tree is a subgraph without cycles
- Spanning tree is a tree containing all nodes
- Find a spanning tree with minimum cost

Minimum spanning tree

- Two models: Sub and Cut
- For ***LP*** formulation *F*, P_F is defined as the feasible set of solutions of F

- ***Theorem***: $P_{sub} \subseteq P_{cut}$ *and* P_{cut} *contains fractional solutions*

Modelling choice matters: Strength (quality) of an MIP formulation

- *T* set of feasible integral solutions
- For ***LP*** formulation *F*, P_F is defined as the feasible set of solutions of F
- Ideal situation: P_F is the convex hull of *T*
- Formulation *A* is stronger than formulation *B* if

$$P_A \subset P_B$$

- Hence, the bound from model *A* is stronger
- Most likely, solving the MIP by branch-and-bound will be faster

Modelling choice matters !!!

Modelling choice matters:

If you have a MIP formulation and it solves very slow

- This may not be the final answer
- You might try an alternative formulation

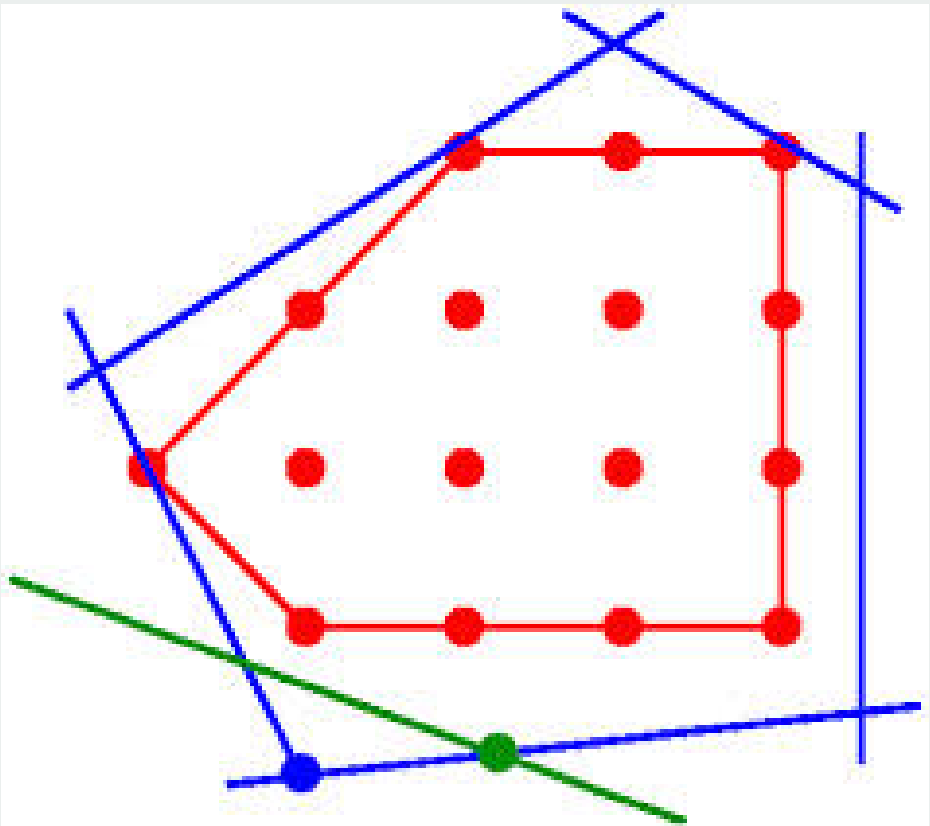
Example Energy Management: Unit Commitment

- See separate pdf-file

Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

Cutting plane algorithm



Universiteit Utrecht

[Faculty of **Science**
Information and Computing Sciences]

Cutting plane algorithm

1. Solve the LP-relaxation. Let x^* be an optimal solution.
2. If x^* is integral, stop x^* optimal solution to the integer linear programming problem.
3. If not, add a *valid inequality* that is not satisfied by x^*

and go to Step 1. To find such an inequality you have to solve separation problem.

Valid inequality: linear constraint that is satisfied by all integral solutions.

Valid inequalities

- General
  - Gomory cuts
- Problem specific

Solving LP-relaxation

- The Simplex method gives two types of variables:
  - $x_i \in B$: basic variables, one for each constraint, can be non-zero (left-side in Dictionary)
  - $x_i \in N$: non-basic variables, are zero (right-side in Dictionary)
  - during the algorithm B changes step by step
- When running the Simplex method you find the following type of equations (are rows of the dictionary)
  - for each $x_i \in B$:

$$x_i = \bar{a}_{io} - \sum_{j \in N} \bar{a}_{ij} x_j$$

$$\Leftrightarrow$$

$$x_i + \sum_{j \in N} \bar{a}_{ij} x_j = \bar{a}_{io}$$

Example

$$\max 2x_1 + x_2 \text{ subject to}$$
$$x_1 - x_2 \le 1$$
$$2\,x_1 + 2x_2 \le 7$$
$$x_1, x_2 \ge 0$$

$$\Leftrightarrow$$

$$\max 2x_1 + x_2 \text{ subject to}$$
$$x_1 - x_2 + x_3 = 1$$
$$2\,x_1 + 2x_2 + x_4 = 7$$
$$x_1, x_2, x_3, x_4 \ge 0$$

- Final dictionary:

$$z = 5\,\frac{3}{4} - \frac{1}{2}x_3 - \frac{3}{4}x_4$$
$$x_1 = 2\frac{1}{4} - \frac{1}{2}x_3 - \frac{1}{4}x_4$$
$$x_2 = \frac{5}{4} + \frac{1}{2}x_3 - \frac{1}{4}x_4$$

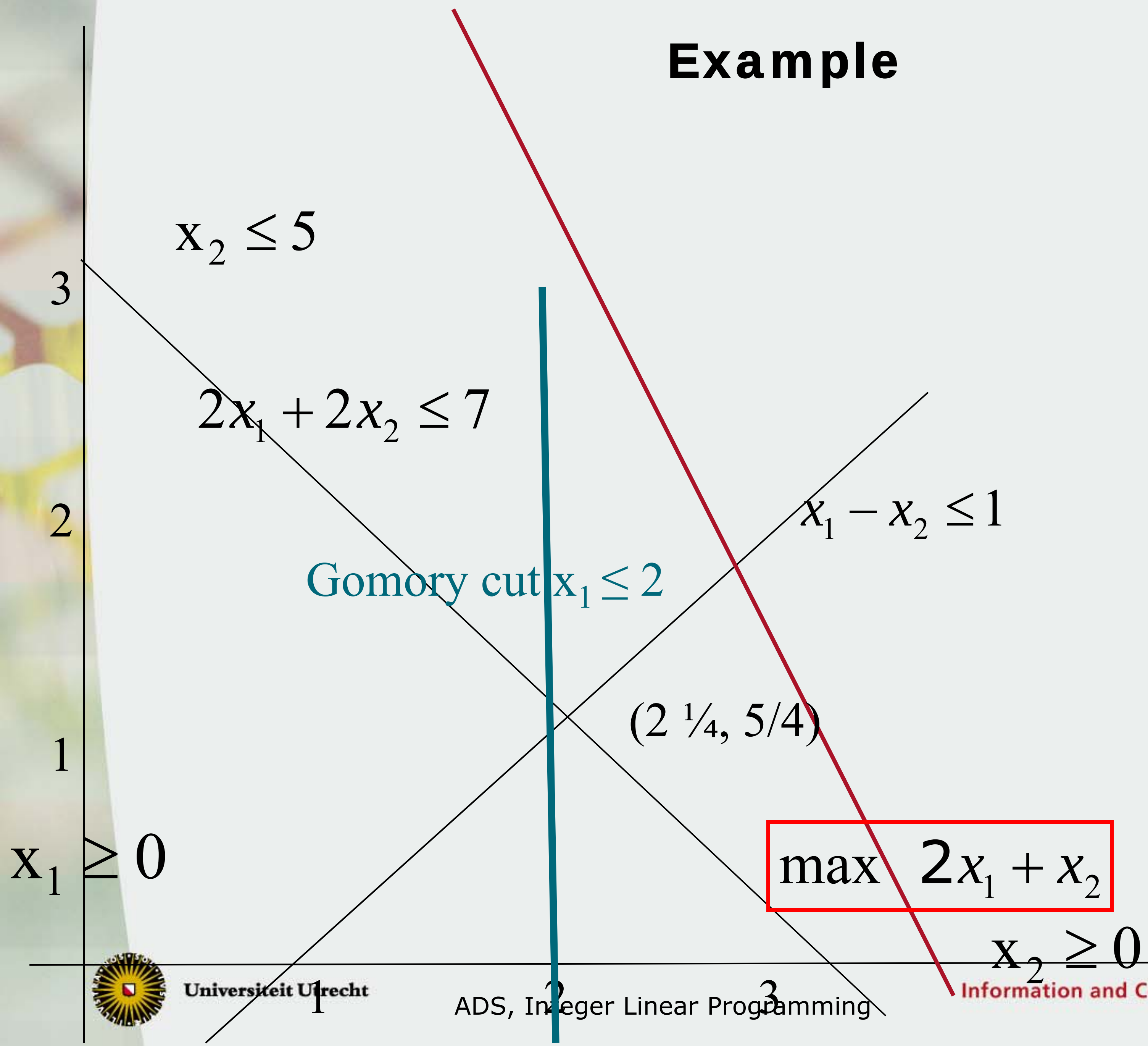
Example

- $x_2 - \frac{1}{2}x_3 + \frac{1}{4}x_4 = \frac{5}{4}$
- $x_2 - \ x_3 \leq \frac{5}{4}$
- For integral solutions
  - $x_2 - \ x_3 \leq 1$, hence $x_2 - (1 - x_1 + x_2) \leq 1 \Leftrightarrow \mathrm{x}_1 \leq 2$

Universiteit Utrecht

[Faculty of **Science**
Information and Computing Sciences]

Example



$x_2 \leq 5$
3
$2x_1 + 2x_2 \leq 7$
2
$x_1 - x_2 \leq 1$
Gomory cut $x_1 \leq 2$
$(2\ 1/4, 5/4)$
1
$x_1 \geq 0$
$\max \quad 2x_1 + x_2$
$x_2 \geq 0$
1
2
3

Gomory cuts

- If solution is fractional, then at least one of the $\overline{a}_{i0}$ is fractional
- Take row from final dictionary corresponding to fractional basic variable

$$x_i + \sum_{j \in N} \overline{a}_{ij} x_j = \overline{a}_{io} \text{ with } \overline{a}_{io} \text{ fractional}$$

- Gomory cut

$$x_i + \sum_{j \in N} \lfloor \overline{a}_{ij} \rfloor x_j \leq \lfloor \overline{a}_{io} \rfloor$$

Problem-specific valid inequalities

- Usually classes of inequalities
- Class of inequalities: set of inequalities of a specific form

- Weighted independent set
- Cover inequalities for knapsack problems

- *Finding valid inequalities is a combinatorial challenge*

Branch-and-cut

- Combination of cutting planes and MIP solving by branch-and-bound

- Applied in well-known MIP-solvers:
  - CPLEX
  - GUROBI
  - GNU Linear Programming Kit (**GLPK**)
  - COIN-OR

- World-record exact TSP solving
  - CONCORDE: http://www.math.uwaterloo.ca/tsp/concorde/index.html

Solving MIP by branch-and-bound or branch-and-cut

Let x* be the best known feasible solution

Search strategy

1. Select an active sub problem F_i (unevaluated node)
2. If F_i is infeasible: delete node
3. Compute upper bound $Z_{LP}(F_i)$ by solving LP-relaxation and adding *cutting planes*. Find feasible solution x_f (e.g. by rounding)

   If $Z_{LP}(F_i) \leq$ value x* delete node (bounding)

   If x_f is better than x*: update x*

   If solution x_{LP} to LP-relaxation is integral,

   then If x_{LP} is better than x*: update x* and node finished,

   otherwise split node into two new subproblems (branching)
4. Go to step 1

Primal heuristic

How many cuts?
Which classes?

Branching strategy

Optional

This if for maximization problem, the book uses a minimization problem.

Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

Branch-and-cut: choices

- Search strategy:
  - x=1 before x=0, other the other way around
  - Depth first
  - Breadth first
  - Best bound: for maximization go to the node with the largest LP-bound

- Cut generation:
  - As many cuts as possible
  - All cuts in root node, nothing in the remainder of the tree
  - All cuts in root node, only a subset from the classes in the remainder of the tree

Branch-and-cut: choices (2)

- Primal heuristic, usually based on rounding LP solution

- Branching strategy:
  - Branch on fractional variables closest to ½
  - Branch on fractional variables closest to 1
  - Branch on important variable (ratio in knapsack)
  - SOS-branching:
    - Special Ordered Sets (SOS): a set of variables, at most one of which can take a non-zero value
    - Let S be a Special Ordered Set.
    - Nodes $\sum_{j\in S} x_j = 1$ and $\sum_{j\in S} x_j = 0$

Branch-and-cut is a framework algorithm!!

Last century, tailoring to your own problem was necessary in most cases and a huge amount of research has been undertaken in doing this:

- Pre-processing
- Classes of valid inequalities
- How many cuts in each node?
- Search strategy
- Branching strategy
- Primal heuristics
- Reduced cost fixing

Well-known MIP solvers like CPLEX and Gurobi successfully (and secretly) apply a lot of the above techniques.

- **Tailoring of branch-and-cut sometimes successful, especially valid inequalities in case of a weak LP-relaxations**

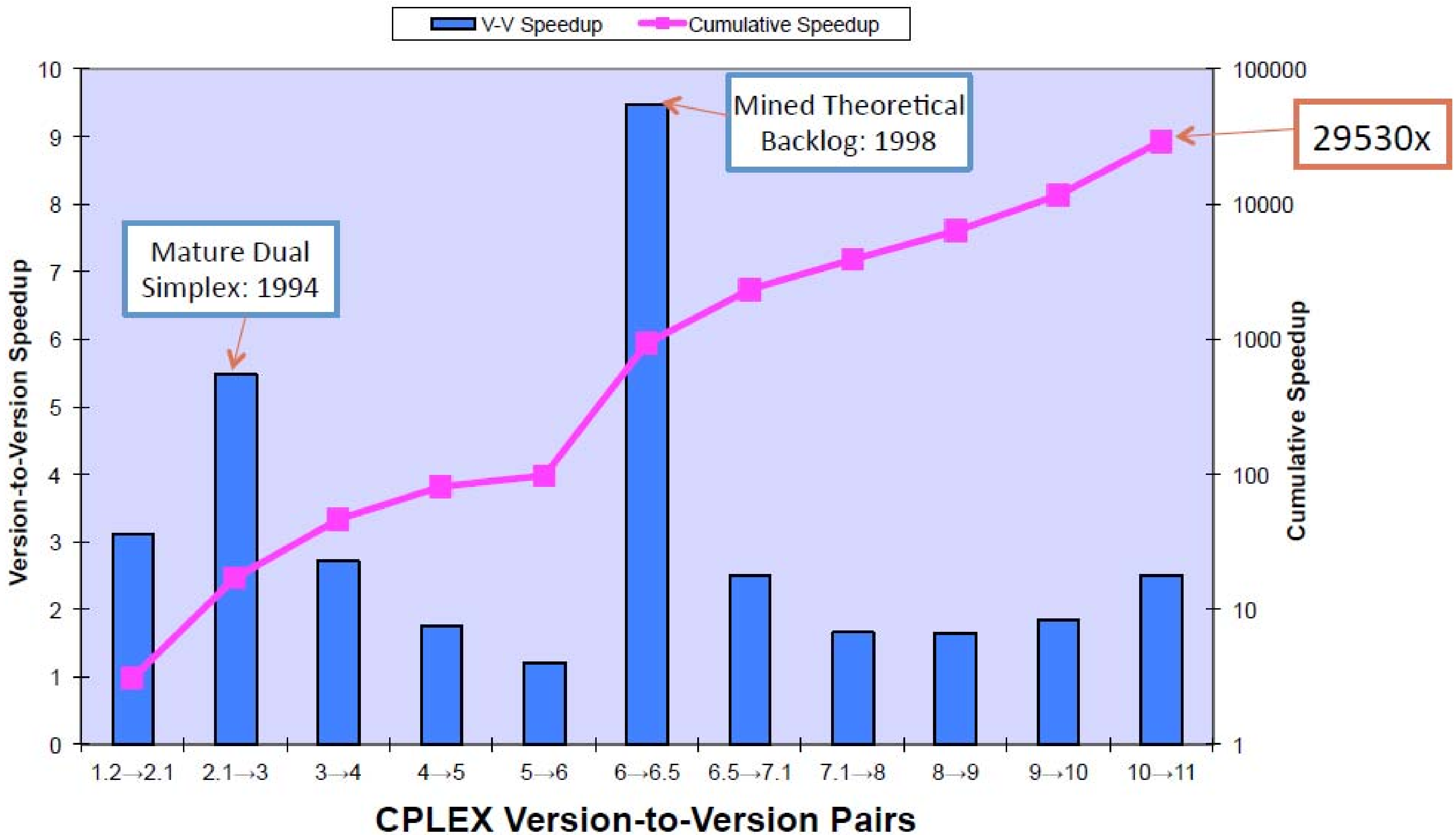
MIP solvers

- CPLEX
- GUROBI
- GNU Linear Programming Kit (**GLPK**)
- COIN-OR

- GLPK and COIN-OR are open source,
- CPLEX and GUROBI are commercial but free for academic purposes
- GUROBI recommended

from http://www.lnmb.nl/conferences/2015/programlnmbconference/LNMB-NGB_Bixby.pdf

Speedups 1991-2008



10

Still we are talking about combinatorial optimization

- We have many, many, many variables
- Still large computation times
- Decomposition approaches often help
- Algorithmic challenges: you need clever algorithms that every now and then use LP-solvers as a component
- **More in**
  - **Scheduling and timetabling**

After the lectures on MIP

- You have an idea of the simplex method
- You can model a combinatorial optimization problem as MIP: exercises (see course website) are strongly recommended
- You know how to solve MIP branch-and-bound
- You understand the strength of an LP-relaxation
- You know some valid inequalites (cutting planes)
- You understand branch-and-cut
- Additional challenge?: slides on polyhedral combinatorics are available on the website