Advanced linear programming

http://www.staff.science.uu.nl/~akker103/ALP/

Chapter 10: Integer linear programming models

Marjan van den Akker

Intro…….

Marjan van den Akker

- Master Mathematics TU/e
- PhD Mathematics of Operations Research TU/e
- Engineer Netherlands Aerospace Center (NLR)
- Lecturer/researcher Computer Science UU:
  - Research on planning algorithms, integer linear programming and simulation
  - Master courses :
    - Algorithms for decision support (COSC),
    - Advanced Linear Programming (Mastermath)
  - Coordination Software- and Gameproject

Method of working

- Lectures
- Self study material
  - Slides and your own notes
  - Book
  - Some lecture notes (under construction)
  - Additional reading material
  - Exercises (if you hand in a solution I can check, good solutions can be made available on the course website)
  - Slides and reading material published on website
    http://www.staff.science.uu.nl/~akker103/ALP/
- Written exam at the end. **ONE** retake

Topics of part 2

Large-scale LP (Ch 6)

- Column generation and Dantzig-Wolfe decomposition
- Benders decomposition

Integer Linear Programming (ILP) (Ch 10 + 11):

- Modelling
- Solving by branch-and-bound
- Cutting planes, branch-and-cut

Lagrangean relaxation (Ch 11)

Since column generation and Benders have their main applications in ILP, we will do ILP first.

This lecture

- ILP models
- Remarks on complexity LP and ILP
- Solving ILP by branch-and-bound
- Model choice matters: strength of LP-relaxation

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

$4x_1 + 8x_2 + 5x_3 + 2x_4 + 6x_5 \leq 15$

$x_1, x_2, x_3, x_4, x_5 \in \{0,1\}$

(Mixed) Integer linear programming

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

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

$x, y \geq 0$

x integral (or binary)

Extension of LP:

- Good news: more possibilities for modelling
- Bad news: larger solution times

Combinatorial optimization

- Find feasible solution with minimal cost, maximal revenue
- Number of possible solutions is finite but very, very large
- Many combinatorial optimization problem can be modeled as ILP
- ILP is NP-hard

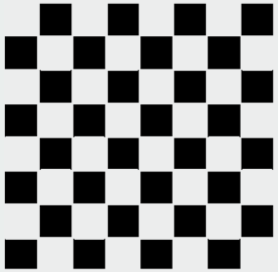
NP-hardness

- *NP*-hard !!!!
  - ***P:*** problem can be solved in **polynomial** time
  - ***NP:*** check solution for feasibility is polynomial, optimization is not provably faster than enumeration of all solutions. (non-deterministic polynomial)
- *P vs NP*

  $ 1 million **Millenium Prize** problem

  http://www.claymath.org/millennium/P_vs_NP

A long time ago Sissa ben Dahir, the Grand Vizer to the Indian king, Shirham, presented his latest creation to his ruler.



...alled chess.

The king ... he told Sissa that he could name his own re...

Sissa replied, "Majesty, give ... rupees; or give me some rice in the followin...

1 grain to place on the first square of the chessboa...
2 grains to place on the second square,
4 grains for the third square, and
8 grains for the 4th square; and
to continue in like manner...
...squares of the board."

18.446.744.073.709.551.615

Cover Netherlands and Belgium with a layer of 1 m

Universiteit Utrecht

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

(Mixed) Integer linear program

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

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

$x, y \geq 0$

x integral

(or binary)

LP-relaxation

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

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

$x, y \geq 0$

Lower bound (or upper bound in case of maximization)

Solution method for linear programming

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

$$LP \in P$$

Knapsack problem revisited

since we use it to demonstrate branch-and-bound for ILP

Knapsack volume *b*
Item *i* has profit c_i and weight a_i
$x_i = 1$ if item i is selected, 0 otherwise……

$$\max \sum_{i=1}^{n} c_i x_i$$

$$s.t.$$

$$\sum_{i=1}^{n} a_i x_i \le b$$

$$x_i \in \{0,1\} \qquad (i = 1,2,\ldots,n)$$

Knapsack problem: elements needed in branch-and-bound

LP-relaxation:

Greedy algorithm

Step 0. Order variables such that $\frac{c_1}{a_1} \geq \frac{c_2}{a_2} \geq \ldots \geq \frac{c_n}{a_n}$

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

Step 2. If $a_i \leq \bar{b}$, then $x_j \leftarrow 1$, 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 solution of LP-relaxation

Solving ILP by branch-and-bound

Let x* be the best known feasible solution

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 feasible solution x_f (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

Optional

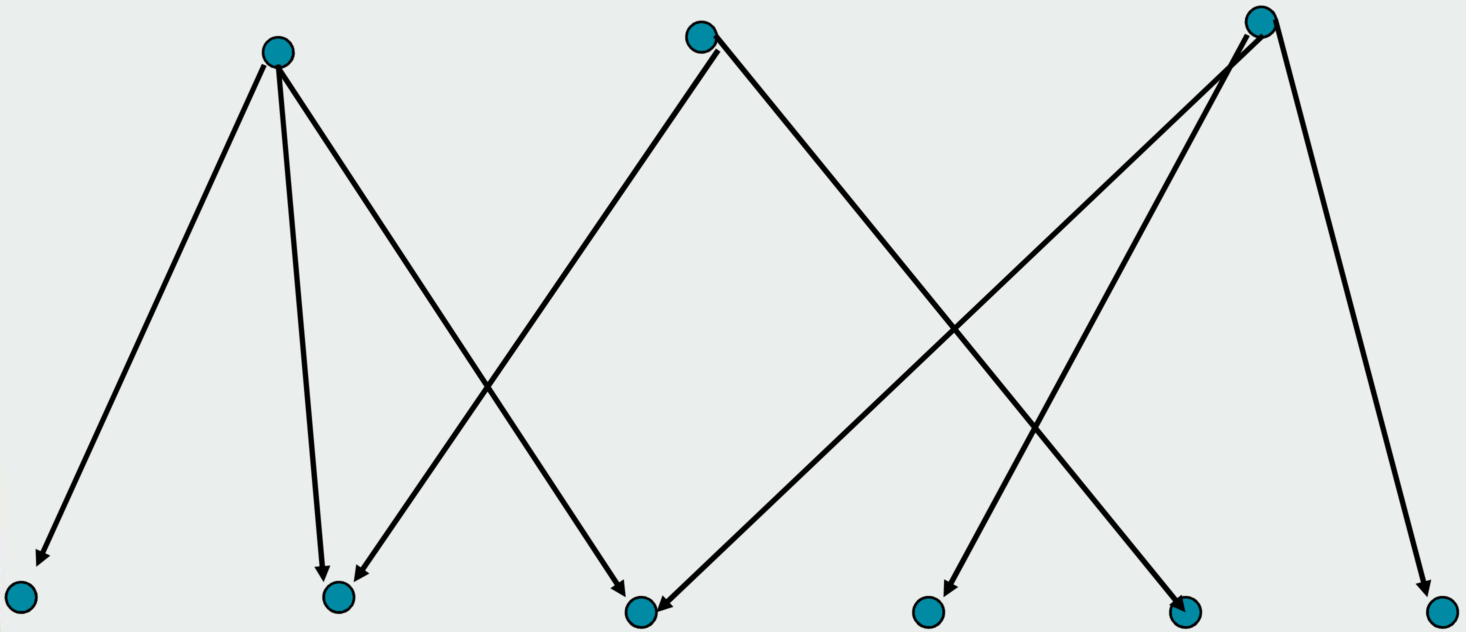
This if for maximization problem, the book uses a minimization problem.

Modeling

- Objective function
- Constraints
- Decision variables

Facility location

Possible locations: *n*



Customers: *m*

Capacitated facility location

- Data:
  - m customers, n possible locations of depot
  - c_{ij} unit cost of serving customer i by depot j
  - Customer demand: D_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 constraints***
- 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 formulations: (FL) and (AFL)
- P_F is defined as the feasible set corresponding to the LP-relaxation of F (P_F is a polyhedron)
- We show that

$$P_{FL} \subset P_{AFL}$$

- This means that (FL) gives a stronger lower bound

$$Z_{LP}\,(AFL) \leq Z_{LP}(FL) \leq Z_{IP}$$

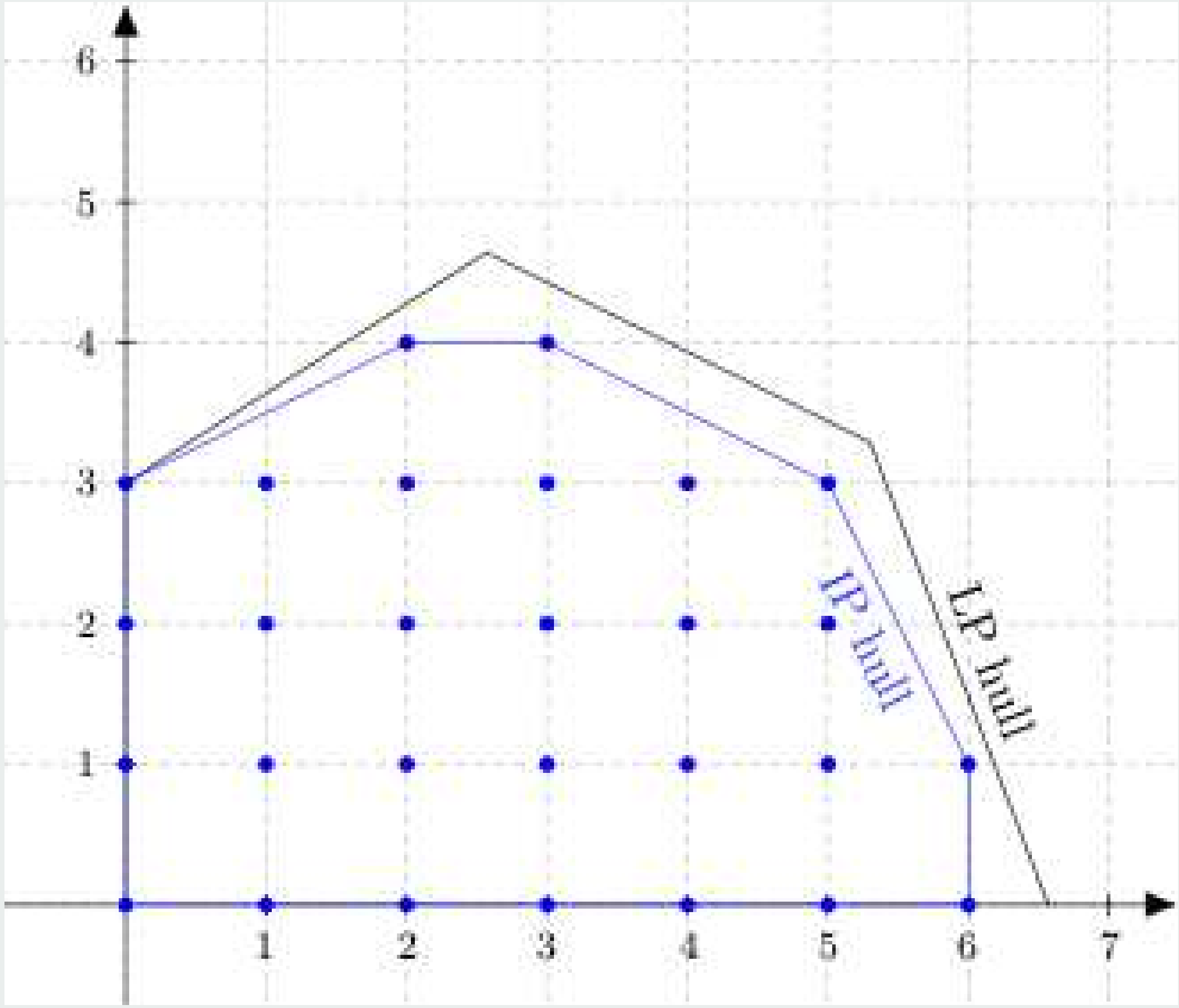
- However, (FL) has more constraints

Strength (quality) of an ILP formulation

- T set of feasible integral solutions
- For formulation F, P_F is defined as the feasible set of solutions of the LP-relaxation of F
- P_F is a polyhedron
- 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 is better
- This is likely to reduce the number of nodes in the branch-and-bound tree
- *This shows that model choices matter!*


6
5
4
3
2
1
1
2
3
4
5
6
7
IP hull
LP hull

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

- We compare formulations (Subtour) and (Cut) and show that (Subtour) is stronger.

Procurement problem

- Computer-manufacturer wants to buy 600 hard-disks
- Offers:

	Fixed cost	Minimum amount to order	Price per item	Discount Threshold	Discount price	Available number of items
A	100	50	24	250	18	500
B	75	50	28	150	20	700

- What is the optimal procurement plan?

Procurement problem

Contains important ILP modelling features:

Already seen in facility location:

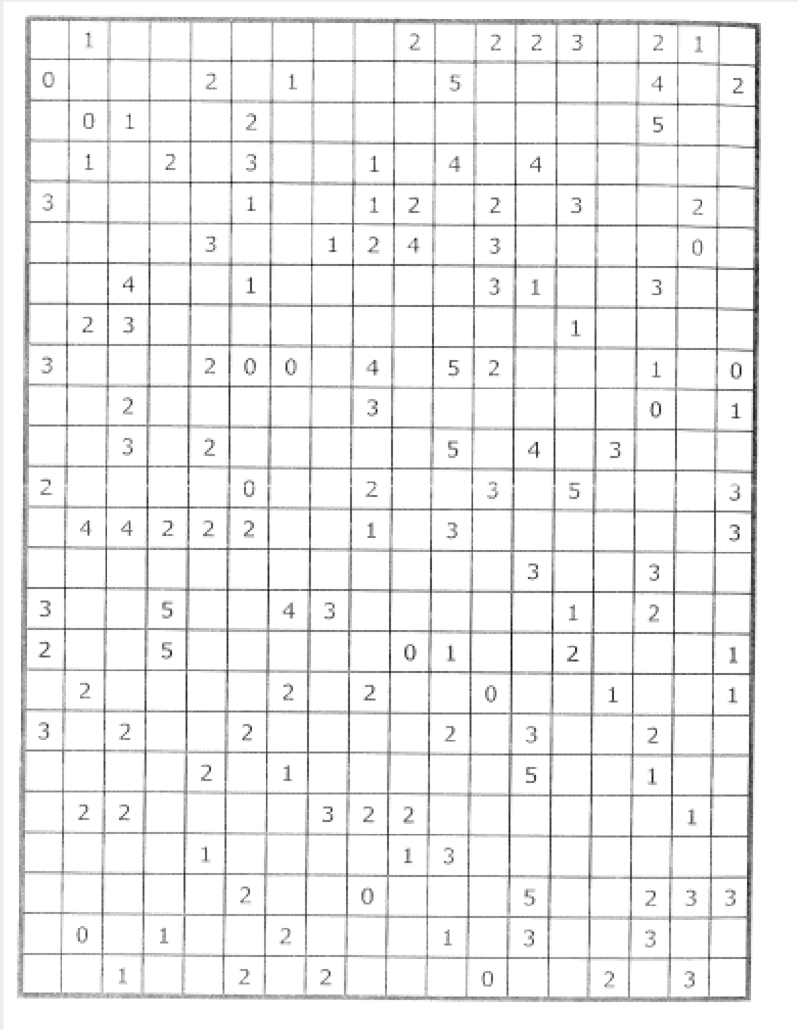
- Fixed cost
- Forcing contraints

Other features:

- Linearize piece-wise linear cost
- Choice constraints

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



	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	

Treasure island with pitfall

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

Wrap-up

- Integer linear programming (ILP) has many modelling possibilities
- ILP can be solved by branch-and-bound
- Soemtimes there are different ILP formulation for the same problem. Formulation makes a difference, e.g. because of the strength of the LP-relaxation.