Lecture 6: Mixed integer linear programming

11/15/17

Overview

- **Mixed integer linear programming**
  - **General**
- **Formulating**
  - **Translation of logical constructs**
  - **Quality of formulations**
- **Solving**
  - **Branch & bound**
  - **Cutting planes**

General

- **MILP is an LP with integrality constraints on some decision variables**
  - **usually, non-negative integers**
  - **break-up with convex optimization**

- **more general and more common in real-life**
- **much harder to solve (NP-hard)**
- **still subject of research**

- **there are nonlinear generalizations**

Examples

- **Pure integer linear program**

$$\max z = x_1 - 3x_2$$
$$x_1 - x_2 \leq 10$$
$$x_1, x_2 \geq 0,\ x_1, x_2 \in Z$$

- **Mixed integer linear program**

$$\max z = x_1 - 3x_2$$
$$x_1 - x_2 \leq 10$$
$$x_1, x_2 \geq 0,\ x_1 \in Z$$

Examples

- **0-1 integer linear program**
  - **Any pure IP can be reformulated as 0-1 IP**

$$\max z = x_1 - 3x_2$$
$$x_1 - x_2 \leq 10$$
$$x_1, x_2 \geq 0,\ x_1, x_2 \in \{0,1\}$$

- **Knapsack problem**
  - **IP with one constraint**

$$\max z = x_1 + 3x_2$$
$$x_1 + x_2 \leq 10$$
$$x_1, x_2 \geq 0,\ x_1, x_2 \in Z$$

Formulation

- Reformulating logical and other combinatorial concepts to mathematical programs
  - At times, no-intuitive

- binary variables are crucial
  - get comfortable with them

Transformations I – cardinality and implications for variables

Stock broker selects a portfolio of 4 investments. He can invest in at most two investments. If he invests in investment 2, he also must invest in 1. also, if he invests in 2, he can't invest in 4. Write down the constraints.

- **$x_{1,}x_{2,}x_{3,}x_4 \in \{0,1\}$: 1 if invested in option i, 0 otherwise**

1. *at most (or at least) elements from a set*

$$\mathbf{x_1+x_2+x_3+x_4 \leq 2}$$

2. *Implication $(x_2=1) \Rightarrow (x_1=1)$*

$$\mathbf{x_2 \leq x_1}$$

3. *$(x_2=1) \Rightarrow (x_4=0)$; at most 1 out of 2*

$$\mathbf{x_2+x_4 \leq 1}$$

Transformations II – fixed charges

Clothes manufacturer is capable of manufacturing shirts and pants on rented equipment. Pants can be sold for 12$ and shirts for 4$. There is 160m of cloth available for both, where pants take up 4m/piece and shirt 1m/piece. For each type of clothes, fixed charges of 180$ and 160$ have to be paid if they are produced. Find the optimal plan.

$$\max 4x_1 + 12x_2 - 160y_1 - 180y_2$$

$$1x_1 + 4x_2 \leq 160$$

$$x_1, x_2 \in Z \text{ : produced shirts and pants}$$

- *Fixed charges*
  - *introduce new binary variables and appropriate constraints*

$$y_1, y_2 \in \{0,1\} \text{ : 1 if producing any of i, 0 otherwise}$$

$$x_1 \leq M_1 y_1$$

$$x_2 \leq M_2 y_2$$

- **M_i sufficiently large in order not to constrain solution for $y_i=1$**

Transformations III – restricted range of values

- Write down the following constraint in IP form

$$x \in S = \{1,5,7,11,23\}$$

- *Restricted range of values*
  - *Introduce |S| new binary variables*

$$y_i \in \{0,1\},\ i=1,\dots,\ |S|$$

$$x = 1y_1 + 5y_2 + 7y_3 + 11y_4 + 23y_5$$

$$y_1 + y_2 + y_3 + y_4 + y_5 = 1$$

Transformations IV – set covering

- Two sets:
  - Set-to-be-covered S_1
  - Covering set S_2

Each member in S_1 must be covered by an acceptable member of S_2. Number of used elements from S_2 has to be minimized in some sense.

Example

There are 4 cities in a county. Minimal number of fire stations has to be built in order for each city to have at least one fire station within 15min. The distances are given in the following table:

	To			
From	City 1	City 2	City 3	City 4
City 1	0	10	20	30
City 2	10	0	25	35
City 3	20	25	0	15
City 4	30	35	15	0

Transformations IV – set covering

From	To City 1	City 2	City 3	City 4
City 1	0	10	20	30
City 2	10	0	25	35
City 3	20	25	0	15
City 4	30	35	15	0

- In this case, $S_1=S_2$
- $\mathbf{x_i \in \{0,1\}, i=1,\ldots, |S_2|}$
  - **1 if station built in i, 0 otherwise**

$$\min x_1+x_2+x_3+x_4$$

$x_1+x_2 \geq 1$; for city 1, it has to be built in 1 or 2, likewise for city 2

$x_3+x_4 \geq 1$; for city 3, it has to be built in 3 or 4, likewise for city 4

Transformations V – disjunctive constraints

- At least 1 out of 2 constraints has to be satisfied

Example

Out of two constraints of the form:

$$f(\mathbf{x_1,x_2,\ldots,x_n}) \le \mathbf{0}$$
$$g(\mathbf{x_1,x_2,\ldots,x_n}) \le \mathbf{0}$$

At least one has to be satisfied.

- $\mathbf{y \in \{0,1\}}$ + appropriate constraints

$$f(\mathbf{x_1,x_2,\ldots,x_n}) \le \mathbf{My}$$
$$g(\mathbf{x_1,x_2,\ldots,x_n}) \le \mathbf{M(1-y)}$$

- M large enough not to constrain the rest of the problem

Transformations VI – implication for constraints

$$f(x_1,x_2,\ldots,x_n) > 0 \Rightarrow g(x_1,x_2,\ldots,x_n) \geq 0$$

$y \in \{0,1\}$ + appropriate constraints

$$-g(x_1,x_2,\ldots,x_n) \leq My$$

$$f(x_1,x_2,\ldots,x_n) \leq M(1-y)$$

- M large enough not to constrain the rest of the problem
- For y=0, f-constr. satisfied, g-constr. satisfied
- For y=1, f-constr. unsatisfied and g-constr. whatever

Solving mixed integer linear programs – “lucky” example

Four jobs A, B, C, D have to be assigned to four machines. Costs of performing each job on each of machines are given in the table:

	1	2	3	4
A	9	2	1	5
B	4	5	6	7
C	2	1	3	6
D	5	3	9	4

a) Set the problem as a ILP (don't solve)

b) Solve the problem using branch&bound

Solving mixed integer linear programs – "lucky" example

	1	2	3	4
A	9	2	1	5
B	4	5	6	7
C	2	1	3	6
D	5	3	9	4

Let c_{ij} mark costs of doing i-th job on j-th machine.

$$x_{ij} = \begin{cases} 1, \text{if } i \leftrightarrow j \text{ assignment is active} \\ 0, \text{otherwise} \end{cases}$$

$$\min z = \sum_i \sum_j c_{ij} \cdot x_{ij}$$ cost of the assignment

$$\sum_j x_{ij} = 1, i = 1, \dots, 4$$ Each job is done on one machine

$$\sum_i x_{ij} = 1, j = 1, \dots, 4$$ Each machine does one job

$\forall i \forall j \; x_{ij} \geq 0$

Matrix of constraint coefficients A is ***unimodular*** *and vector of constraint RHS is integral. LP relaxation solves the ILP!*

Solving mixed integer linear programs

- Explicit enumeration of all feasible points?
  - Inefficient, prohibitive

Branch-and-bound method

- Efficient **implicit** enumeration

- **LP relaxation of MILP**
  - LP obtained by omitting all integrality constraints
  - If solution of LP relaxation satisfies MILP integrality constraints, it is optimal for MILP as well
  - Upper bound for maximization MILP

- Form a search tree on integral variables
  - Use LP relaxations for implicit enumeration
    - Potentially skipping huge parts of search-space

Solving MILPs – B&B (maximization)

1. Set of branched nodes N= { LP relaxation of starting MILP *O*}
2. LB=-∞ (or some known lower bound)
3. Repeat until N is empty
   a) According to ***some*** rule pickout a node (program) **n** from N
   b) Solve LP relaxation of n; n' and get solution UB_n
      a) If n' is infeasible: **continue**
      b) If $UB_n \leq LB$: **continue** **BOUND**
      c) If solution UB_n of n' is integral according to O
         a) if $UB_n > LB$: $LB = UB_n$
      d) else pick ***some*** would-be integral variable **v** with fractional value **f**
         a) create two subprograms based on n: **BRANCH**
            a) n_1 with additional constraint $v \geq \lceil f \rceil$
            b) n_2 with additional constraint $v \leq \lfloor f \rfloor$
            c) $N := N \cup \{n_1, n_2\}$
4. Output solution LB

Solving MILPs - Example

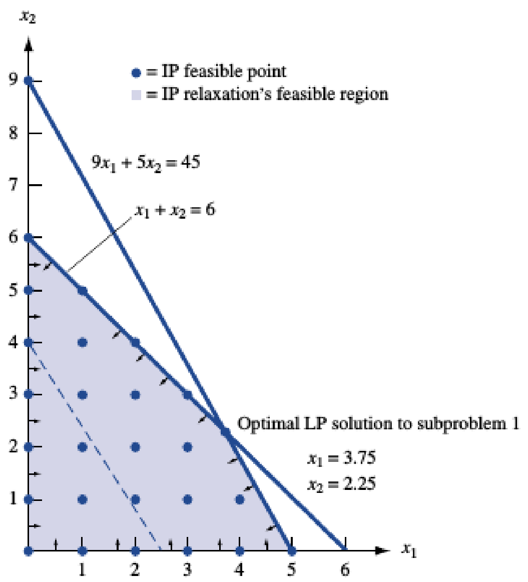
The Telfa Corporation manufactures tables and chairs. A table requires 1 hour of labor and 9 square board feet of wood, and a chair requires 1 hour of labor and 5 square board feet of wood. Currently, 6 hours of labor and 45 square board feet of wood are available. Each table contributes \$8 to profit, and each chair contributes \$5 to profit. Formulate and solve an IP to maximize Telfa's profit.

$$\max z = 8x_1 + 5x_2$$

$$x_1 + x_2 \leq 6 \quad \text{(Labor constraint)}$$

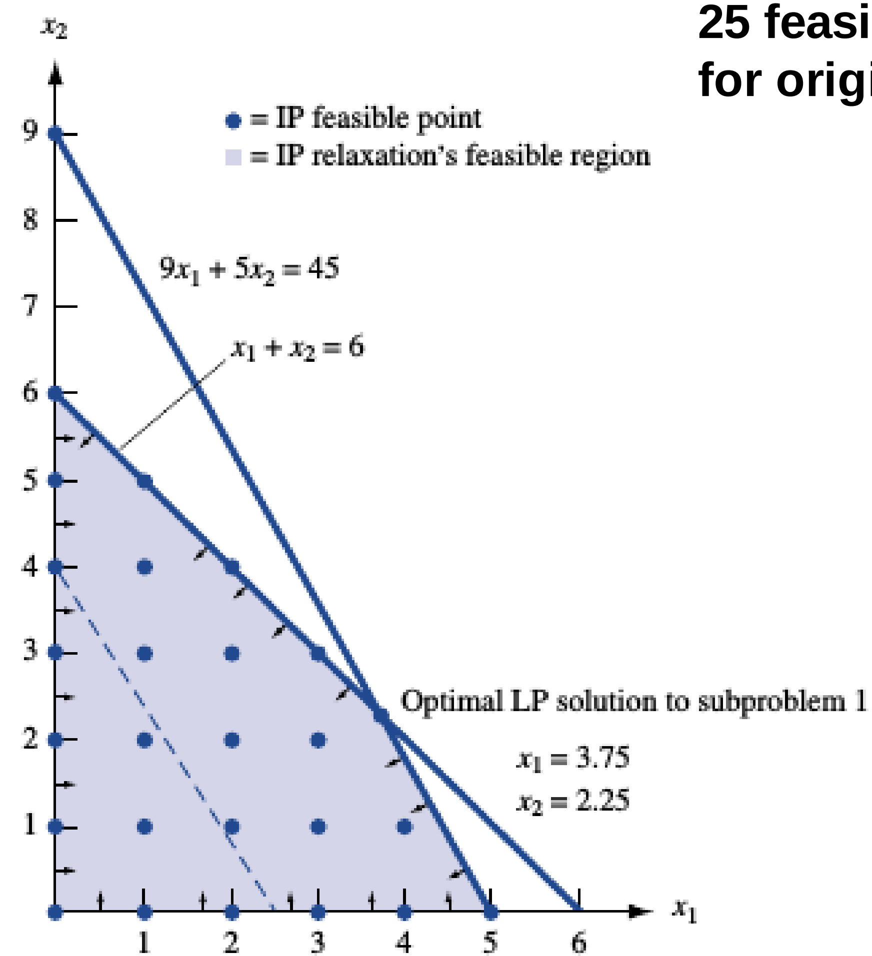
$$9x_1 + 5x_2 \leq 45 \quad \text{(Wood constraint)}$$

$$x_1, x_2 \geq 0;\ x_1, x_2 \text{ integer}$$



Solving MILPs – Solving subproblem LP1

$$\max z = 8x_1 + 5x_2$$
$$x_1 + x_2 \leq 6$$
$$9x_1 + 5x_2 \leq 45$$
$$x_1, x_2 \geq 0$$



25 feasible points for original MILP!

Let's branch on x_1! (arbitrary)

Create **two** subproblems with added constraints:

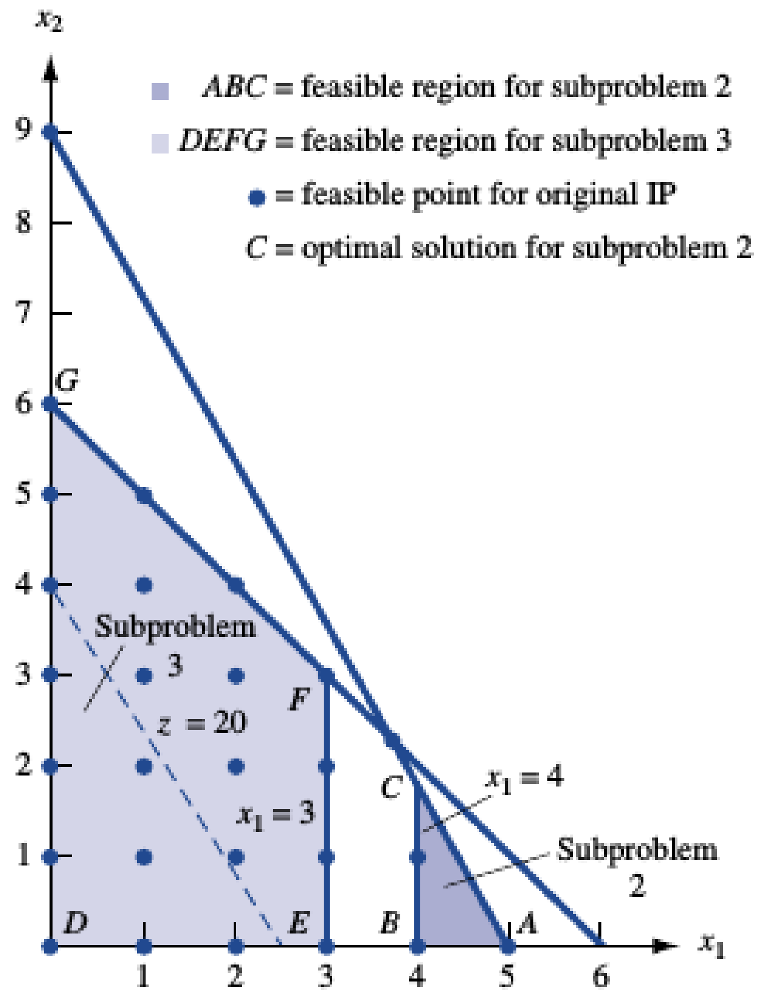
- $x_1 \geq 4$ (LP 2)
- $x_1 \leq 3$ (LP 3)

Solving MILPs – Branching LP1

Let's branch on x_1!

Create **two** subproblems with added constraints:

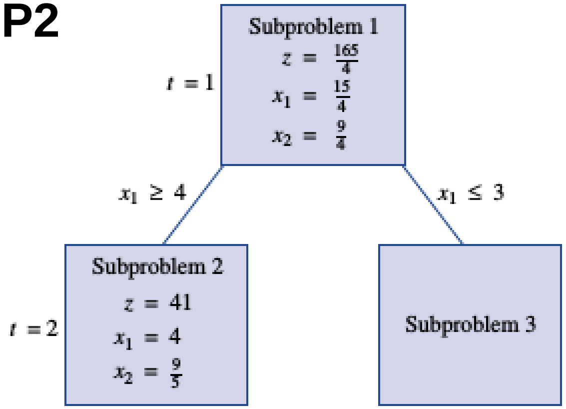
- $x_1 \geq 4$ (LP 2)
- $x_1 \leq 3$ (LP 3)



LP2:= LP1 + $x_1 \geq 4$
LP3:= LP1 + $x_1 \leq 3$

Solving MILPs – Solving and branching on…

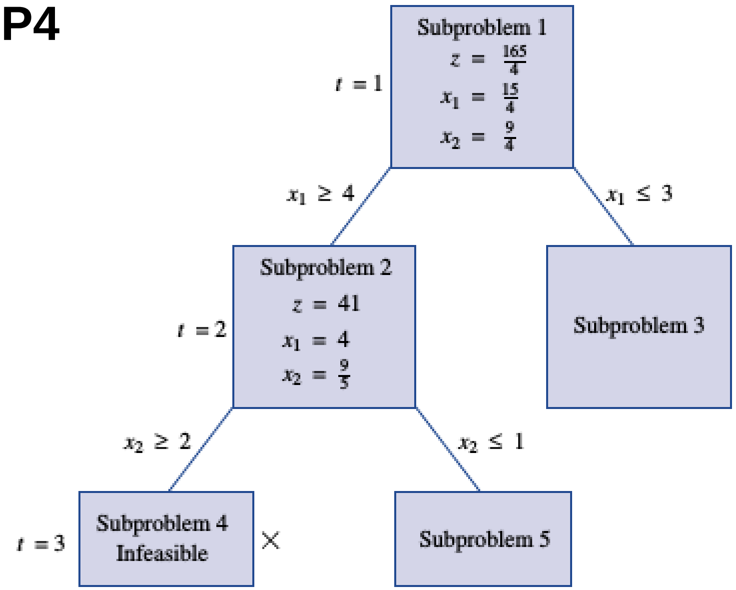
LP2

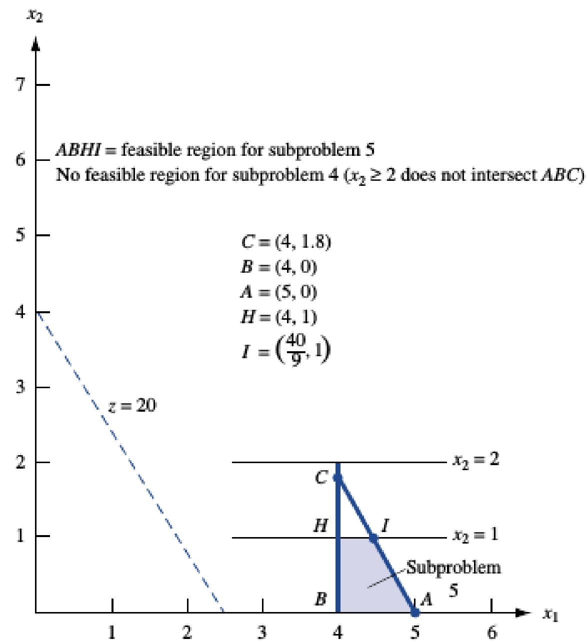


x_2 is fractional (but needs to be integral), let's branch on it!

LP4:= LP2 + $x_2 \geq 2$
LP5:= LP2 + $x_2 \leq 1$

LP4





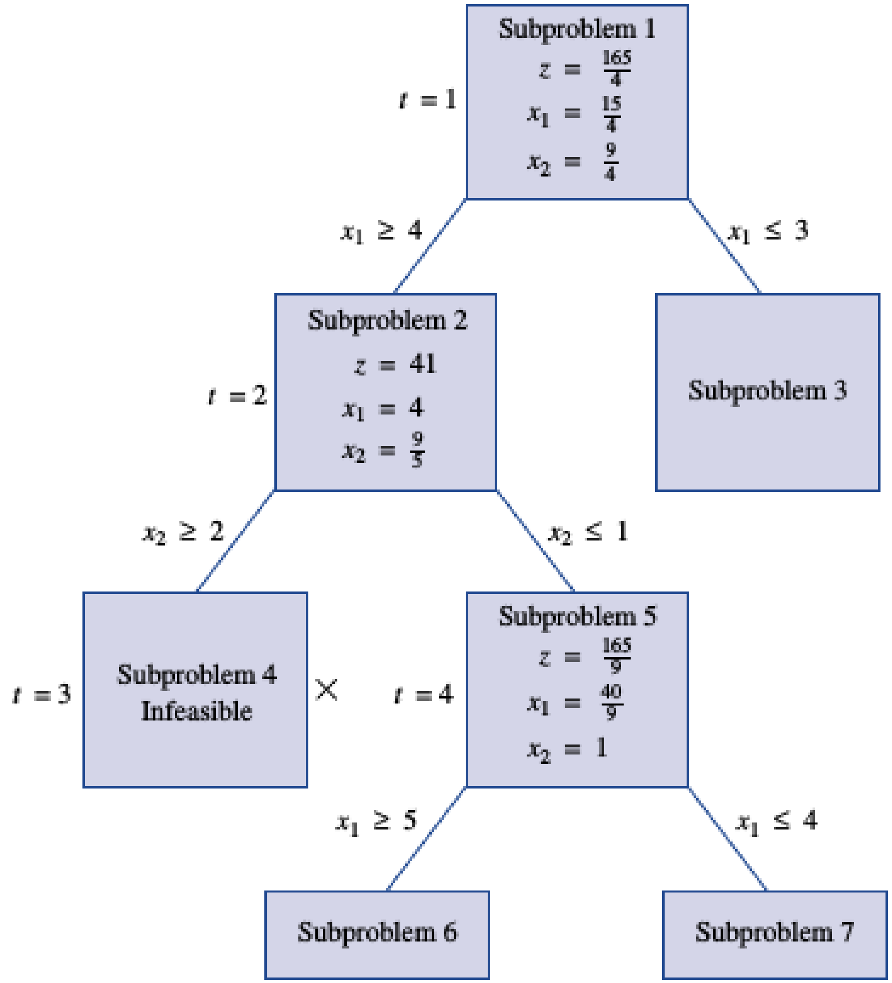
Solving MILPs – Solving and branching on…

LP5

Optimal solution
$z=365/9$
$x_1=40/9$
$x_2=1$

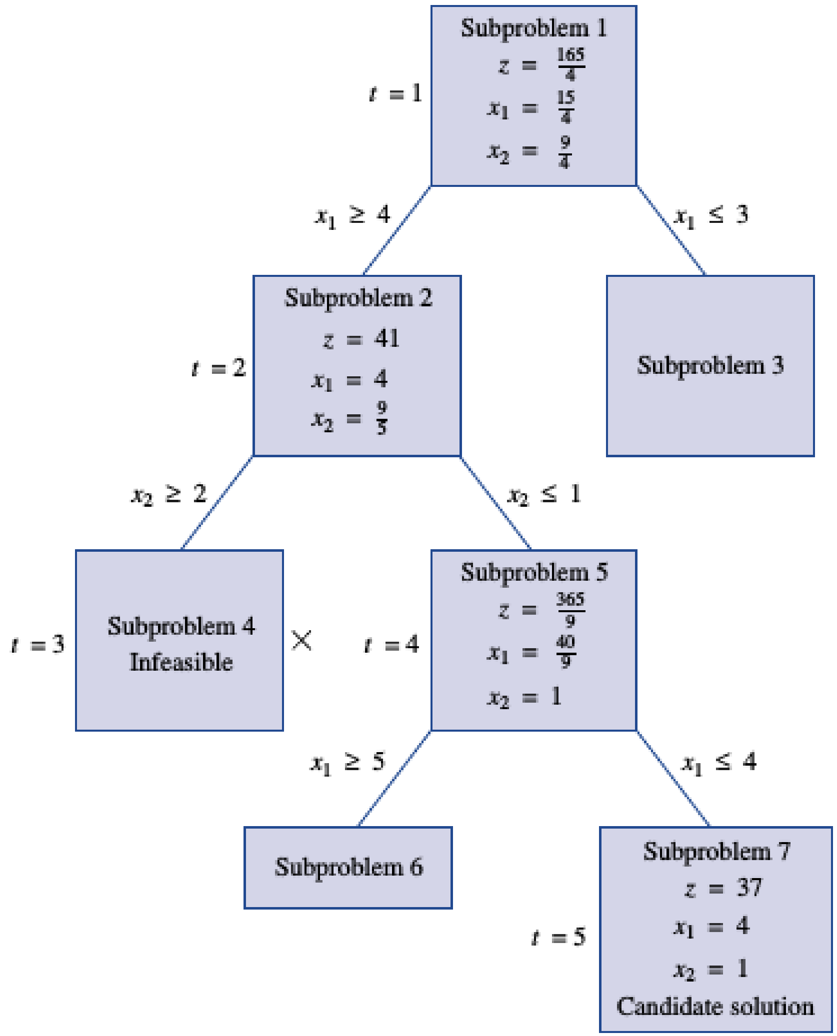
x_1 is fractional (but needs to be integral), let's branch on it!

LP6:= LP5 + $x_1 \geq 5$
LP7:= LP5 + $x_1 \leq 4$



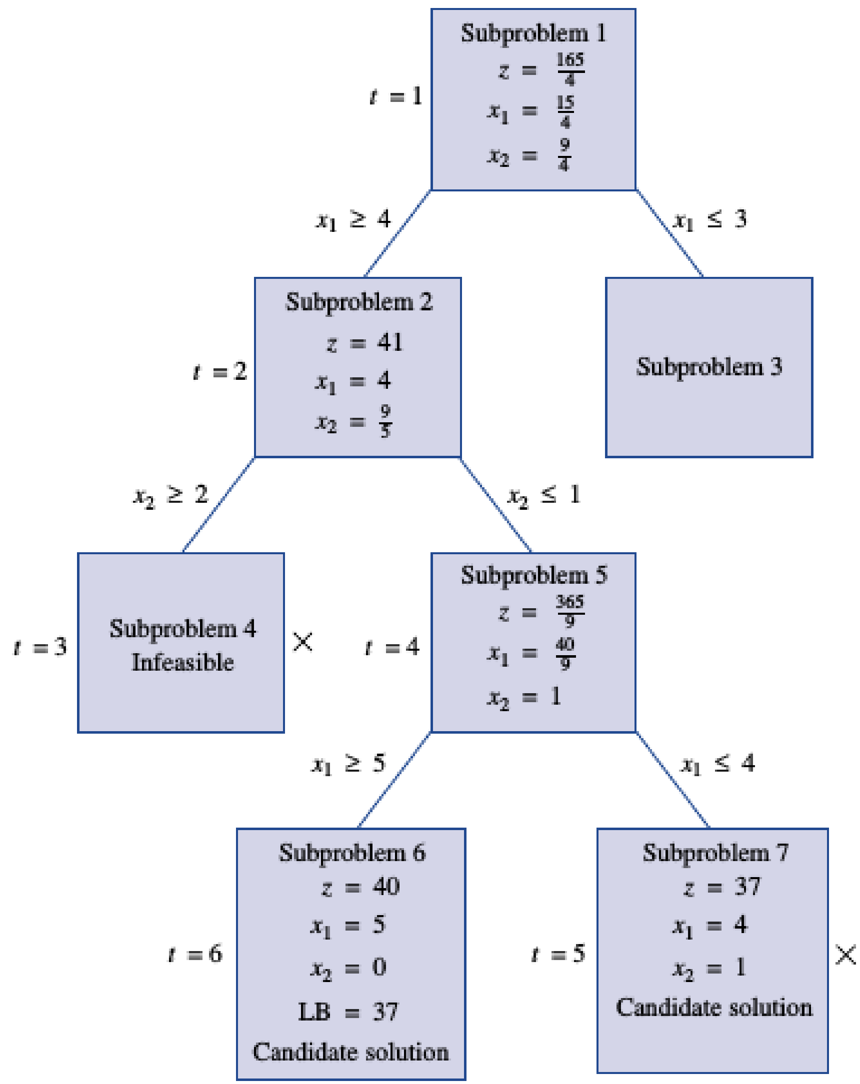
Solving MILPs – Solving LP7

LP7



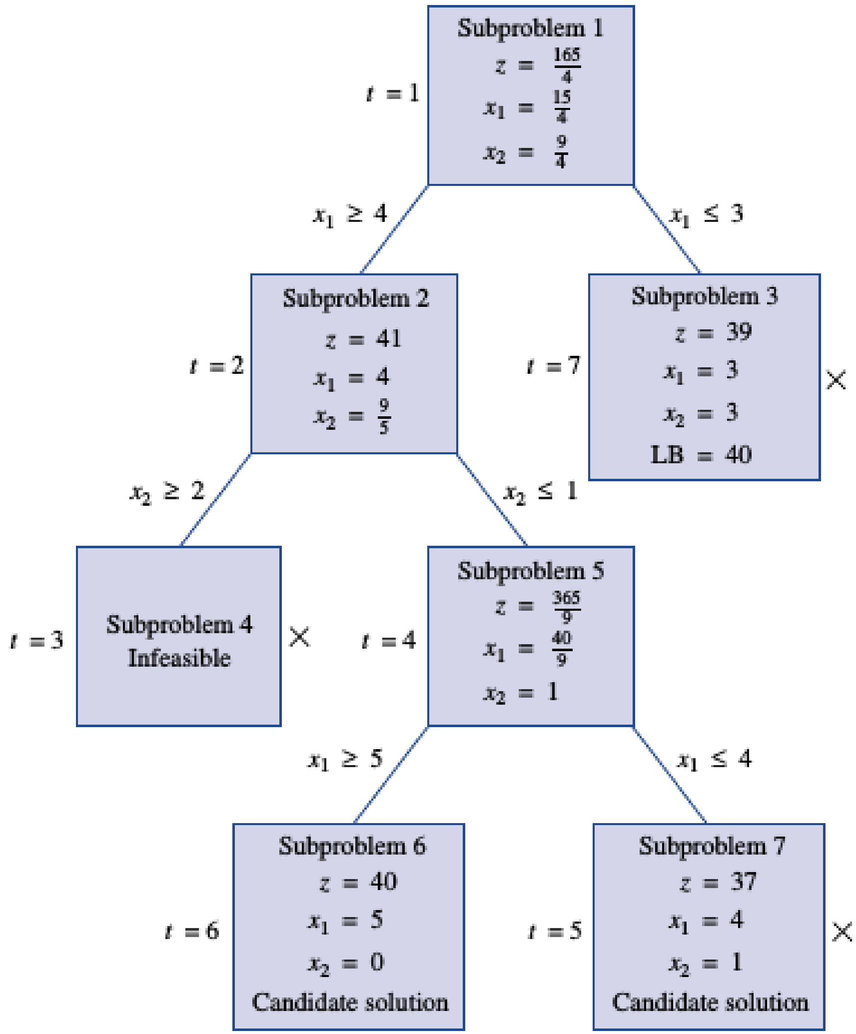
Solving MILPs – Solving LP6

LP6



Solving MILPs – Solving LP3

LP6



Solved!

25 feasible points for original MILP. But, we have explicitly checked only 7 nodes!

Solving MILPs – technicalities

- How to pick subproblem to solve?
  - Two common strategies
    - Backtracking (depth-first search)
    - Jumptracking
      - branch on node from N with the best solution of LP relaxation
- How to pick variable to branch on?
  - On variable with the greatest economic importance
- Solving so many LP relaxations!?
  - **Very** efficient using dual simplex on added constraints

Solving MILPs – technicalities

- When to finish?
  - To the bare end, at all computational costs!?
  - Often, after achieving the pre-chosen gap to the optimal z-value of problem’s LP relaxation
- Are all formulation the same (for all practical purposes)?
  - No, some are easier to solve than others

Solving ILPs : 0-1 ILPs

- Easier to solve than other pure IPs

- Implicit **enumeration**
  - Variant of B&B for 0-1 IPs, with the following differences:

  - Two sets of binary variables: *fixed* and *free*
    - all are free at the beginning
  - At picked node n, do **fast** completion of free variables to achieve greedy optimum UB_n(for max), not necessarily feasible
  - Branching on some free variable v
    - Two branches: for values 0 and 1
    - Below the branch v becomes fixed

Solving mixed integer linear programs – “lucky” example

Four jobs A, B, C, D have to be assigned to four machines. Costs of performing each job on each of machines are given in the table:

	1	2	3	4
A	9	2	1	5
B	4	5	6	7
C	2	1	3	6
D	5	3	9	4

b) Solve the problem using branch&bound

0-1 ILP: we use implicit enumeration

Cutting planes

- Generate additional constraints to optimally solved LP relaxation of the problem in order to “peel off” non-integrality
- Gomory cuts for general ILP

- Different problem-specific cuts
- Dual simplex deals with added constraints for efficient solving