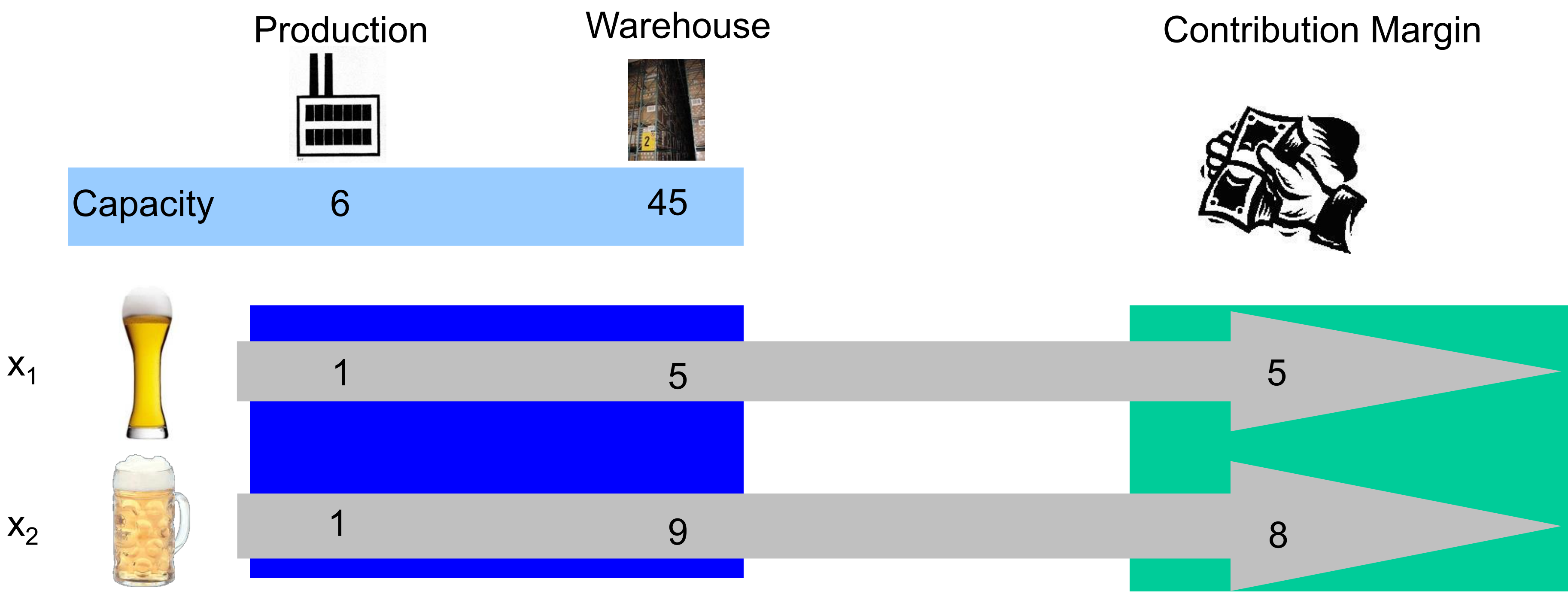
4 Integer and Mixed-Integer Programming

4.1 Examples

Integer Program

<u>Example 1</u>: Production Planning



<u>Question</u>: How many <u>integer</u> units shall be produced to maximize the contributed margin?

<u>Note</u>: The parameters of the problem are different from the ones in Chapter 3.

Integer Program

Example 1: Production Planning

Max $$z = 5 \cdot x_1 + 8 \cdot x_2 \quad (2.5)$$

subject to

$$1 \cdot x_1 + 1 \cdot x_2 \leq 6 \quad (2.6)$$

$$5 \cdot x_1 + 9 \cdot x_2 \leq 45 \quad (2.7)$$

$$x_1, x_2 \geq 0 \quad \text{and integer} \quad (2.8)$$

Definition: An integer program is a linear program where all variables are integer.

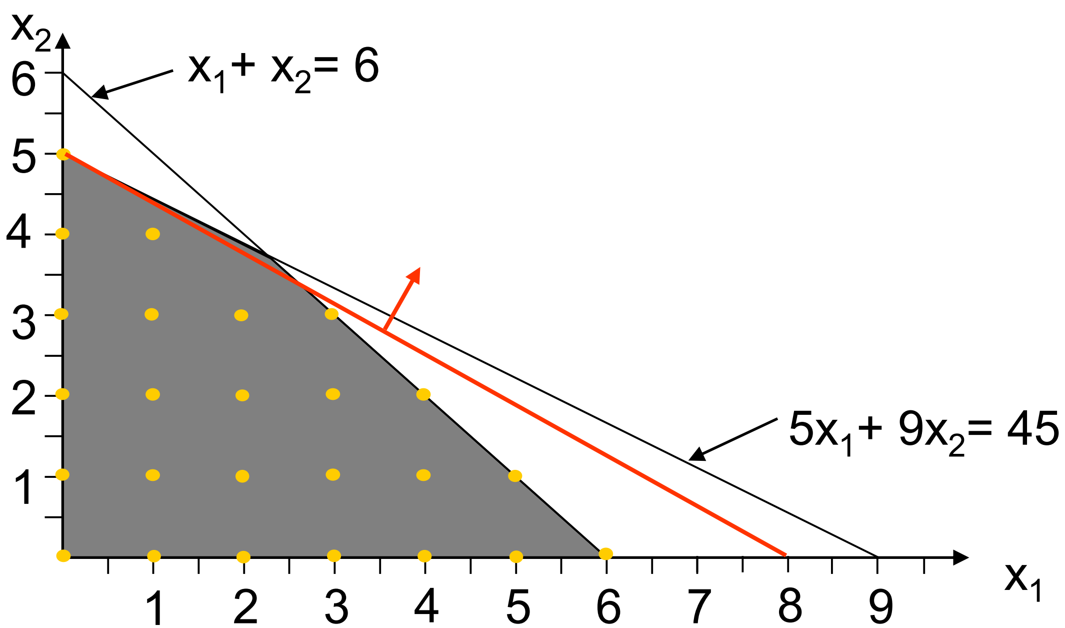
Integer Program: Solution Space

Max $z = 5 \cdot x_1 + 8 \cdot x_2$

subject to

$$1 \cdot x_1 + 1 \cdot x_2 \leq 6$$
$$5 \cdot x_1 + 9 \cdot x_2 \leq 45$$
$$x_1, x_2 \geq 0$$ <u>and integer</u> (for short we write: $x_1, x_2 \in \mathbb{Z}_{\geq}$)

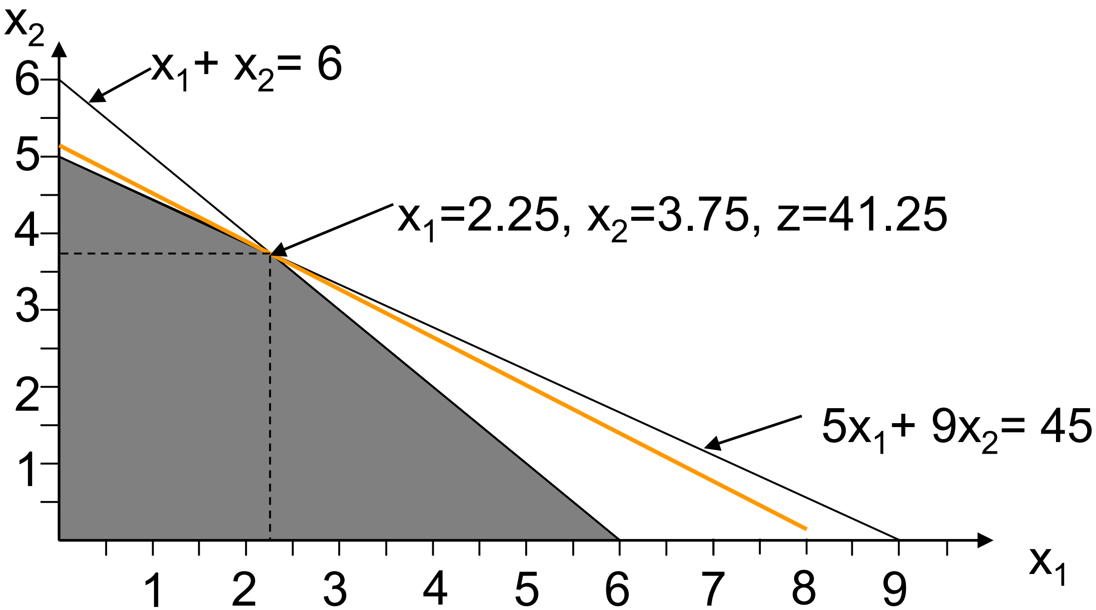
All feasible solutions:



4.2 Properties of Integer Programs

LP-Relaxation

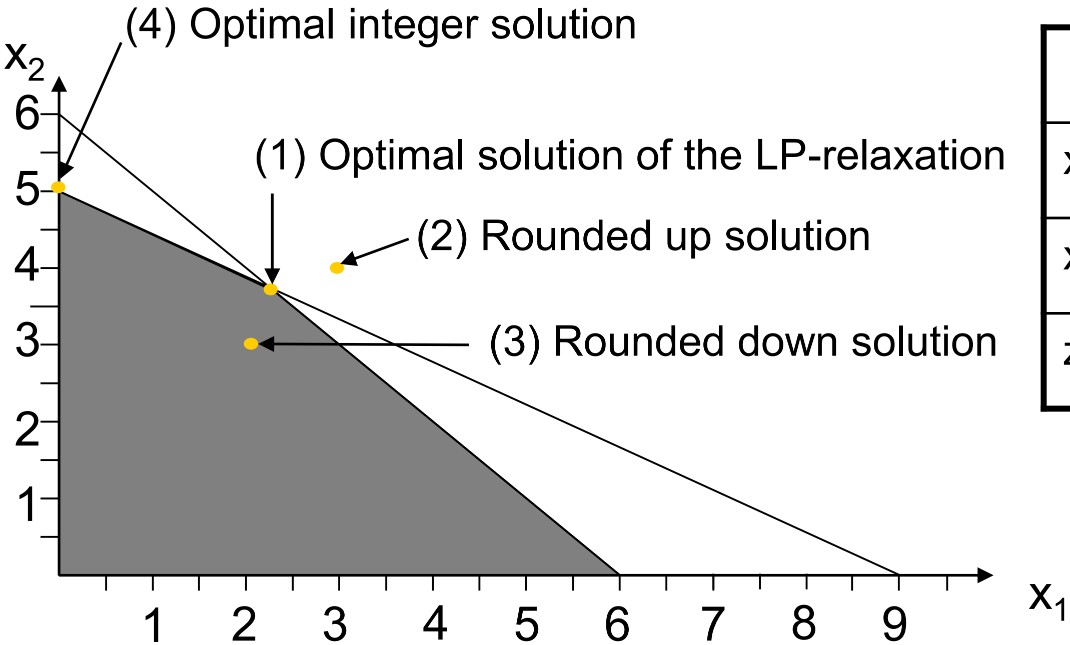
Definition 21: The LP-relaxation of an integer program is obtained by relaxing the integrality constraint of the integer variables. The LP-relaxation can be solved with the Simplex method.



Optimal Solution of the LP-Relaxation and the Integer Program

<u>Observations</u> (for our example, does not hold in general):

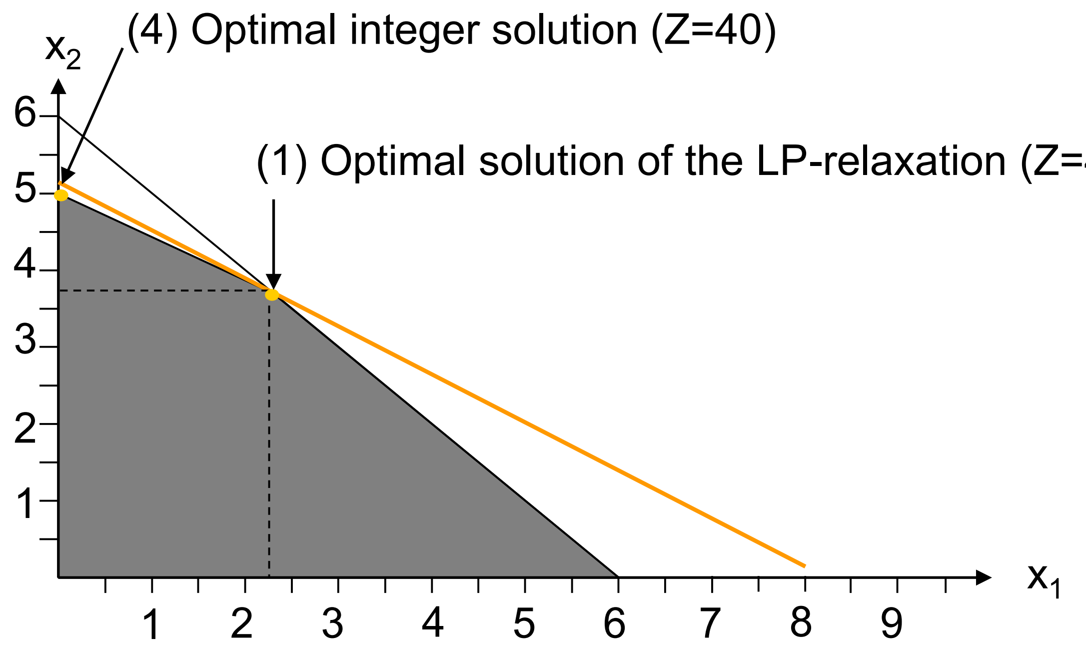
- The optimal solution of the LP-relaxation (1) is not integer.
- The optimal solution of the LP-relaxation (1) is not integer.
- Rounding up (2) does lead to an infeasible solution.
- Rounding down gives a feasible but not an optimal solution.
- The optimal solution of the integer problem can be far from the optimal solution of the LP-relaxation.



	(1)	(2)	(3)	(4)
x_1	2.25	3	2	0
x_2	3.75	4	3	5
z	41.25	-	34	40

LP-Relaxation and Integer Program

Observation: The objective function value of the optimal LP-relaxation is larger than or equal to the objective function value of the optimal integer solution.



Property 9: For a maximization problem, the optimal objective function value of the LP-relaxation $z^*(LP)$ is always greater than or equal to the optimal objective function value of the integer program $z^*(IP)$.

Maximization Problem: $z^*(IP) \leq z^*(LP)$

Minimization Problem: $z^*(LP) \leq z^*(IP)$

4.3 Branch-and-Bound

Branch-and-Bound: Example and Main Idea

We will explain the fundamentals of Branch-and-Bound (B&B) with the following problem:

P_0: Max $z = 5 \cdot x_1 + 8 \cdot x_2$

subject to

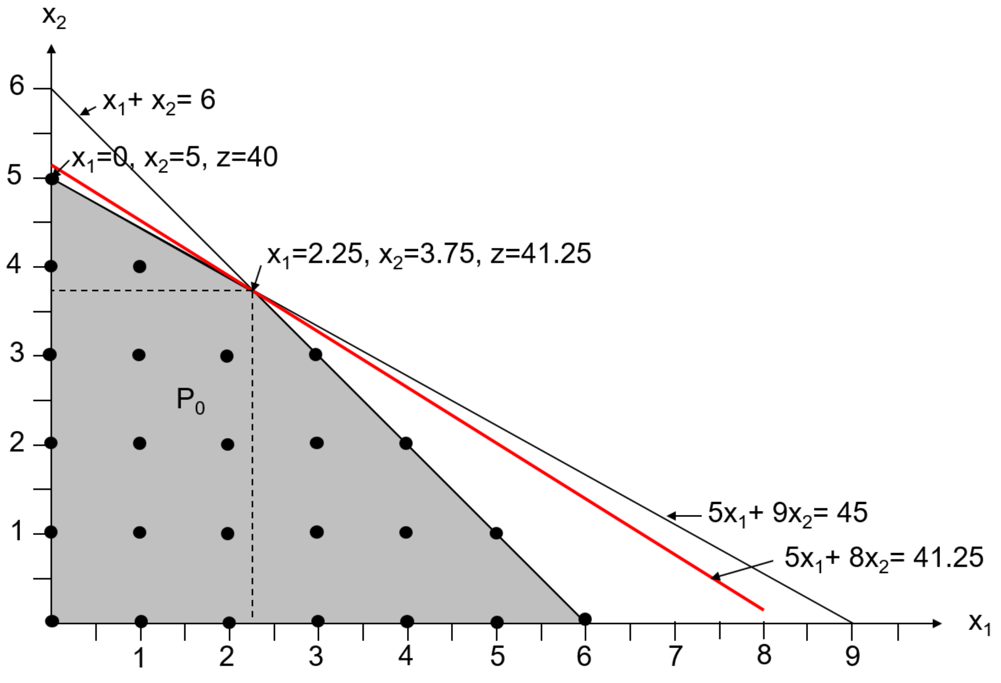
$$1 \cdot x_1 + 1 \cdot x_2 \leq 6$$

$$5 \cdot x_1 + 9 \cdot x_2 \leq 45$$

$$x_1, x_2 \in \mathbb{Z}_{\geq}$$

<u>Main idea</u>: In a systematic way, substitute the start problem by new problems where non-integer solutions are forbidden by adding constraints. New problems are solved with the dual simplex.

Upper Bound, Lower Bound, and Solution Gap



Branching

In case of a non-integer solution, we substitute the original problem.

<u>Rule for generating and numbering the sub-problems</u>:

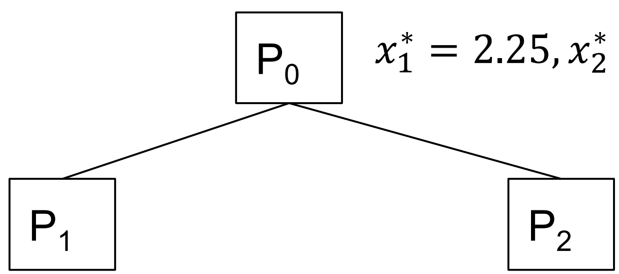
- "Left" sub-problem: Adding the ≤-constraint. Number = Number of parent problem $+1$
- "Right" sub-problem: Adding the ≥-constraint. Number = Number of parent problem $+2$

P_0: Max $z = 5 \cdot x_1 + 8 \cdot x_2$

subject to

$$1 \cdot x_1 + 1 \cdot x_2 \leq 6$$
$$5 \cdot x_1 + 9 \cdot x_2 \leq 45$$
$$x_1, x_2 \geq 0$$

Example: Branching on x_2



$x_1^* = 2.25, x_2^* = 3.75$

P_1 : Max $z = 5 \cdot x_1 + 8 \cdot x_2$

subject to

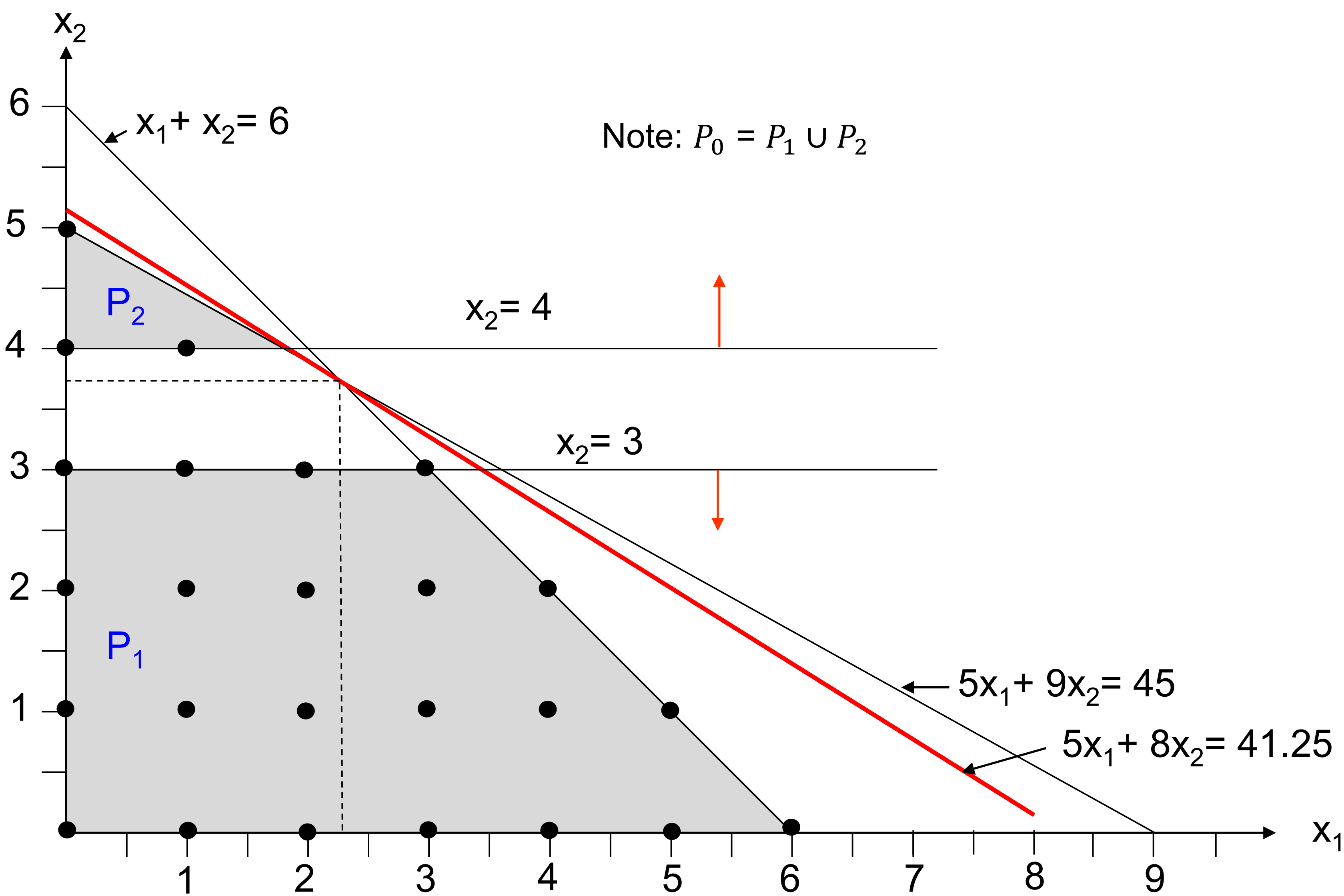
$$1 \cdot x_1 + 1 \cdot x_2 \leq 6$$
$$5 \cdot x_1 + 9 \cdot x_2 \leq 45$$
$$x_1, x_2 \geq 0$$

P_2 : Max $z = 5 \cdot x_1 + 8 \cdot x_2$

subject to

$$1 \cdot x_1 + 1 \cdot x_2 \leq 6$$
$$5 \cdot x_1 + 9 \cdot x_2 \leq 45$$
$$x_1, x_2 \geq 0$$

Branching: Graphical Representation



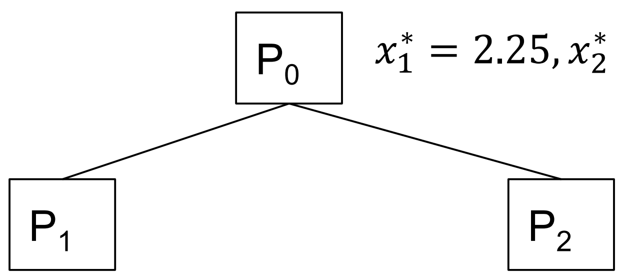
Managing Open Problems with a Candidate List

- Let K be a list of the problems, which have to be solved.

P_0: Max $z = 5 \cdot x_1 + 8 \cdot x_2$

subject to

$$1 \cdot x_1 + 1 \cdot x_2 \leq 6$$
$$5 \cdot x_1 + 9 \cdot x_2 \leq 45$$
$$x_1, x_2 \geq 0 \text{ and integer}$$



$x_1^* = 2.25, x_2^* = 3.75$

P_1: Max $z = 5 \cdot x_1 + 8 \cdot x_2$

subject to

$$1 \cdot x_1 + 1 \cdot x_2 \leq 6$$
$$5 \cdot x_1 + 9 \cdot x_2 \leq 45$$
$$x_2 \leq 3$$
$$x_1, x_2 \geq 0 \text{ and integer}$$

P_2: Max $z = 5 \cdot x_1 + 8 \cdot x_2$

subject to

$$1 \cdot x_1 + 1 \cdot x_2 \leq 6$$
$$5 \cdot x_1 + 9 \cdot x_2 \leq 45$$
$$x_2 \geq 4$$
$$x_1, x_2 \geq 0 \text{ and integer}$$

Updating Upper Bound (UB) and Lower Bound (LB)

- The UB equals the maximum value of the objective functions of the LP-relaxation of all sub-problems in the candidate list K.

- The LB is updated whenever we solve a sub-problem and find an improved integer solution.

P_0: Max $z = 5 \cdot x_1 + 8 \cdot x_2$

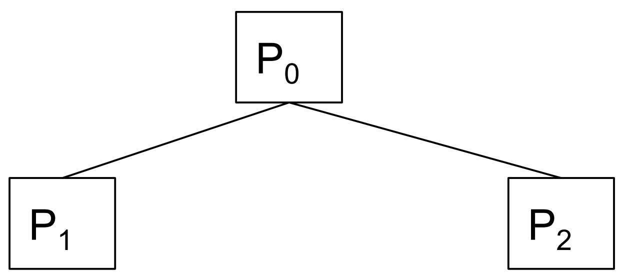
subject to

$$1 \cdot x_1 + 1 \cdot x_2 \leq 6$$
$$5 \cdot x_1 + 9 \cdot x_2 \leq 45$$
$$x_1, x_2 \geq 0$$

$x_1^* = 2.25, x_2^* = 3.75, z^* = 39$

$\overline{z} = 41.25$

$\underline{z} = 34$



P_1: Max $z = 5 \cdot x_1 + 8 \cdot x_2$

subject to

$$1 \cdot x_1 + 1 \cdot x_2 \leq 6$$
$$5 \cdot x_1 + 9 \cdot x_2 \leq 45$$
$$x_2 \leq 3$$
$$x_1, x_2 \geq 0$$

$x_1^* = 3, x_2^* = 3, z^* = 39$

P_2: Max $z = 5 \cdot x_1 + 8 \cdot x_2$

subject to

$$1 \cdot x_1 + 1 \cdot x_2 \leq 6$$
$$5 \cdot x_1 + 9 \cdot x_2 \leq 45$$
$$x_2 \geq 4$$
$$x_1, x_2 \geq 0$$

$x_1^* = 1{,}8, x_2^* = 4, z^* = 41$

Solution Gap

- For any state in the branch-and-bound algorithm the solution gap can be calcluated.

$$\Delta = \frac{|z(\text{best bound}) - z(\text{incumbent})|}{|z(\text{incumbent})|}$$

- z(incumbent) is the objective function of the best found feasible solution for the integr problem.
- z(best bound) is the objective function of the best possible solution, which still can be reached.

- If Δ= 0, we found the optimal solution.
- If Δ> 0, the best possible solution is Δ *100 % away from the so far best found solution.

- During the branch-and-bound algorithm, Δ is decreasing.

Selecting the Next Candidate Problem from the List

- Last In First Out (LIFO): We select the problem, which has been added to the list last.
- Tie Breaker is the largest problem number.

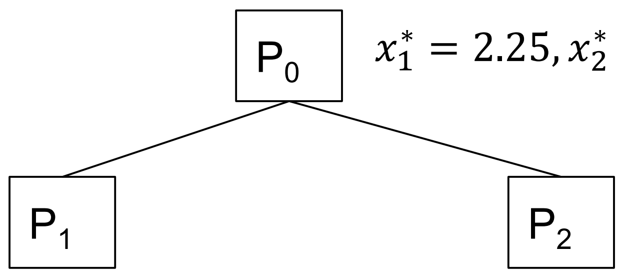
P_0: Max $z = 5 \cdot x_1 + 8 \cdot x_2$ (2.5)

subject to

$1 \cdot x_1 + 1 \cdot x_2 \leq 6$ (2.6)

$5 \cdot x_1 + 9 \cdot x_2 \leq 45$ (2.7)

$x_1, x_2 \geq 0$ and integer (2.8)



$x_1^* = 2.25, x_2^* = 3.75$ $\bar{z} = 41.25, \underline{z} = 34$

P_1: Max $z = 5 \cdot x_1 + 8 \cdot x_2$ (2.5)

subject to

$1 \cdot x_1 + 1 \cdot x_2 \leq 6$ (2.6)

$5 \cdot x_1 + 9 \cdot x_2 \leq 45$ (2.7)

$x_2 \leq 3$

$x_1, x_2 \geq 0$ and integer (2.8)

$x_1^* = 3, x_2^* = 3, z^* = 39$

P_2: Max $z = 5 \cdot x_1 + 8 \cdot x_2$ (2.5)

subject to

$1 \cdot x_1 + 1 \cdot x_2 \leq 6$ (2.6)

$5 \cdot x_1 + 9 \cdot x_2 \leq 45$ (2.7)

$x_2 \geq 4$

$x_1, x_2 \geq 0$ and integer (2.8)

$x_1^* = 1{,}8, x_2^* = 4, z^* = 41$

One Iteration: Processing one Sub-Problem

1) We select a problem from the list

2) We delete the problem from the list

3) We distinguish cases 1 - 4:

   Case 1: If the LP-solution is integer, check if LB and incumbent solution can be updated.

   Case 2: If the LP-solution is not integer and $z^*(LP) \leq \underline{z}$, no further action.

   Case 3: If the problem does not have a feasible solution, no further action.

   Case 4: If the LP-solution is not integer and $z^*(LP) > \underline{z}$, create two new sub-problems.

Example: Processing Problem P_0

P_0
$x_1 = 2.25$ $x_2 = 3.75$ $z = 41.25$

$\overline{z} = 41.25$

$\underline{z} = 34$

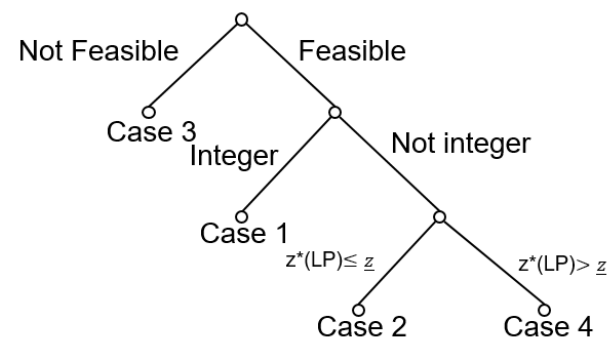
1) We select a problem from the list
2) We delete the problem from the list
3) We distinguish four cases:

   Case 1: If the LP-solution is integer, check if LB and incumbent solution can be updated.

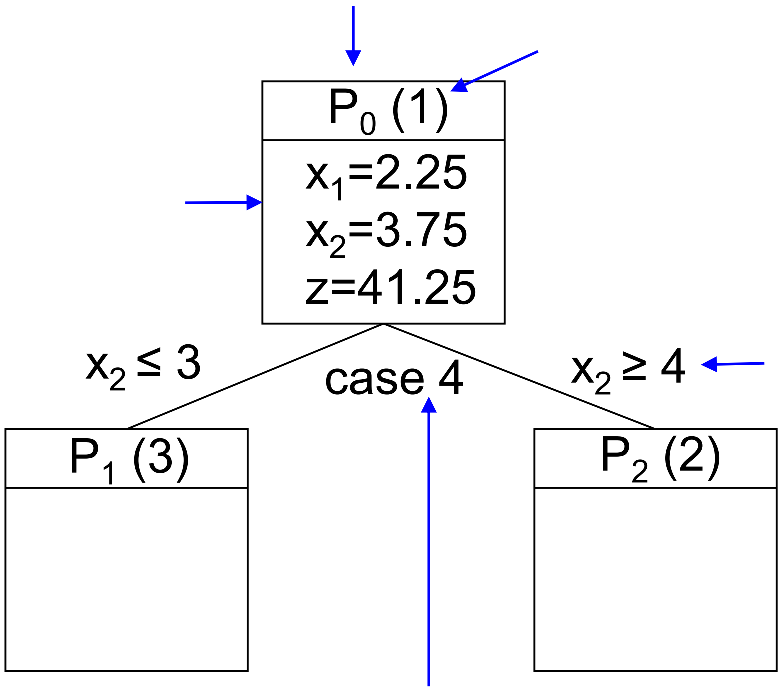
   Case 2: If the LP-solution is not integer and $z^*(LP) < \underline{z}$, no further action.

   Case 3: If the problem does not have a feasible solution, no further action.

   Case 4: If the LP-solution is not integer and $z^*(LP) > \underline{z}$, create two new sub-problems.



Representation of the Branch-and-Bound Tree



Summary: Design Decisions for Branch-and-Bound

Fractional variable used for branching:

- Here: Arbitrary choice of x_2 when both variables are fractional
- Alternative: Choose the variable, which is most fractional.

Creating and numbering sub-problems:

- Left sub-problem: Adding $\leq -$constraint $\rightarrow$ sub-problem P_{i+1}
- Right sub-problem: Adding $\geq -$constraint $\rightarrow$ sub-problem P_{i+2}

Selecting the next candidate problem from list K:

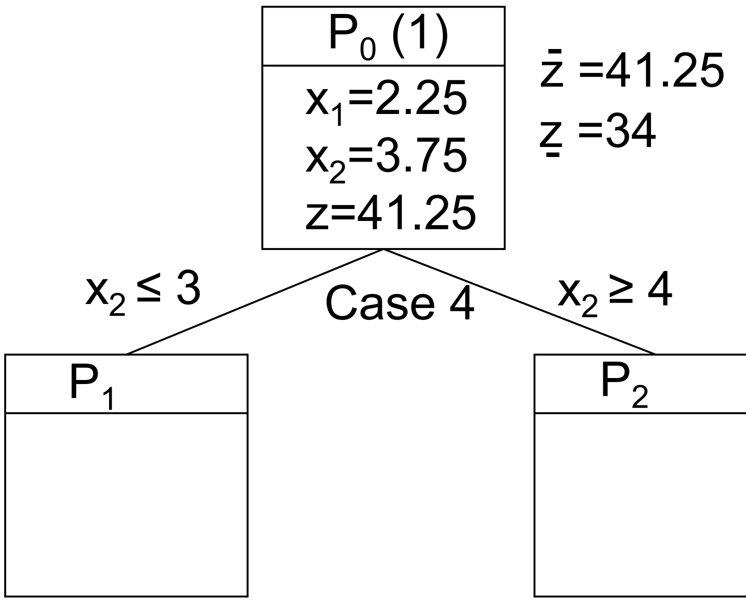
- Here: Last-in-first-out (LIFO): Select the sub-problem, which has been added to K last.
- Alternative: Maximum upper bound (MUB): Select the sub-problem with max upper bound.
- Tie-breaker: Largest problem number

Table-Format for Branch-and-Bound
1st Iteration: Problem P_0

Cases:
(1) Integer solution
(2) $z^*(LP) \le \underline{z}$
(3) No feasible solution
(4) $z^*(LP) > \underline{z}$

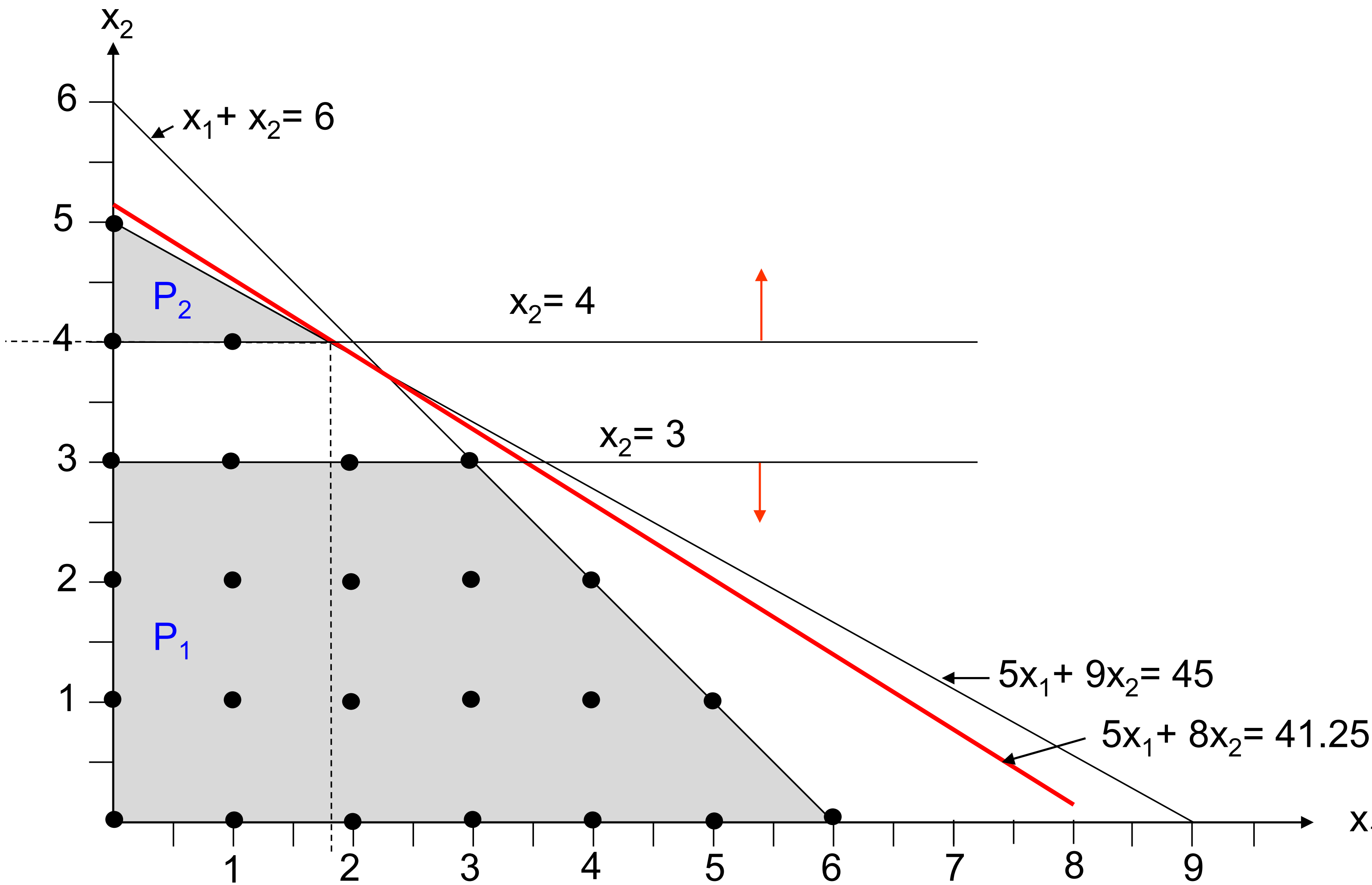
		Sol. LP-relax.						Incumbent				
K	P_{i^*}	x_1	x_2	z	$\overline{z}$	$\underline{z}$	Δ	IP sol.	Case	Branch	P_i	$z^*(LP)$

Branch-and-Bound Tree



$$K = < P_2, P_1]$$

Graphical Representation of Problems P_1 and P_2



Solving Sub-Problem P_2 with the Dual Simplex

P_0 → P_1, P_2 ($x_2 \geq 4$)

Optimal tableau for P_0:

$x_2 \geq 4$

$x_2 - x_5 = 4$

$-x_2 + x_5 = -4$

BV	Value	x1	x2	x3	x4	x_5	Row
x1	2.25	1	0	2.25	-0.25	0	(1)
x2	3.75	0	1	-1.25	0.25	0	(2)
x_5	-4	0	-1 (should be 0)	0	0	1	(3)
-z	-41.25	0	0	-1.25	-0.75	0	(4)

BV	Value	x1	x2	x3	x4	x5	
x_1	2.25	1	0	2.25	-0.25	0	(5)
x_2	3.75	0	1	-1.25	0.25	0	(6)
x_5	-0.25	0	0	Pivot elem. -1.25	0.25	0	(7) = (3) + (2)
-z	-41.25	0	0	-1.25	-0.75	0	(8)

BV	Value	x1	x2	x3	x4	x5
x_3	0.2	0	0	1	-0.2	-0.8
-z	-41					

(9) = (5) - 2.25 (11)

(10) = (6) - 1.25 (11)

(11) = (7) / -1.25

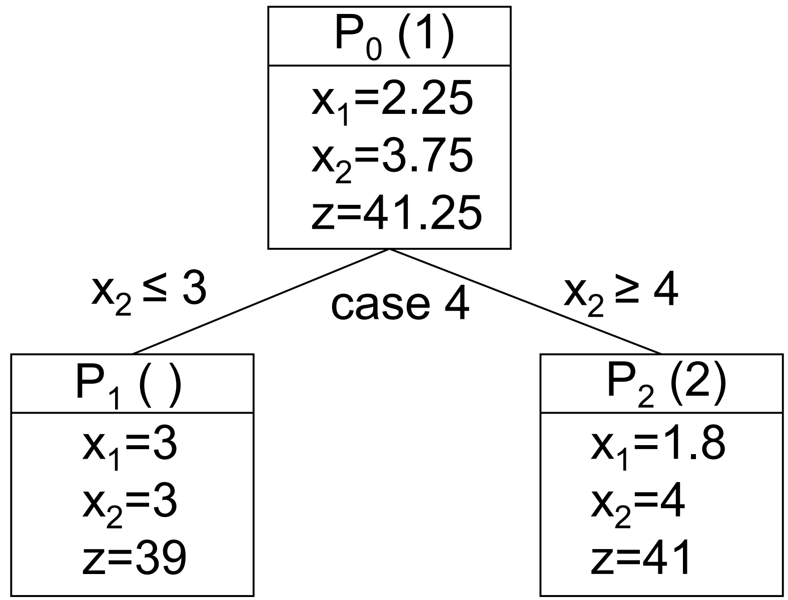
(12) = (8) - (-1.25)(11)

Table 2nd Iteration: Problem P_2

Cases:
(1) Integer solution
(2) z*(LP-relaxation) ≤ $\underline{z}$
(3) No feasible solution
(4) z*(LP-relaxation) > $\underline{z}$

K	P_{i^*}	Sol. LP-relax. x_1	x_2	z	$\overline{z}$	$\underline{z}$	Δ	Incumbent IP sol.	Case	Branch	P_i	$z^*(LP)$
$\langle P_0]$	P_0	2.25	3.75	41.25	41.25	34	21.32%	$x_1 = 2$	4)	$x_2 \leq 3$	P_1	39
								$x_2 = 3$		$x_2 \geq 4$	P_2	41
$\langle P_2, P_1]$												

B&B-Tree 2nd Iteration: Problem P_2

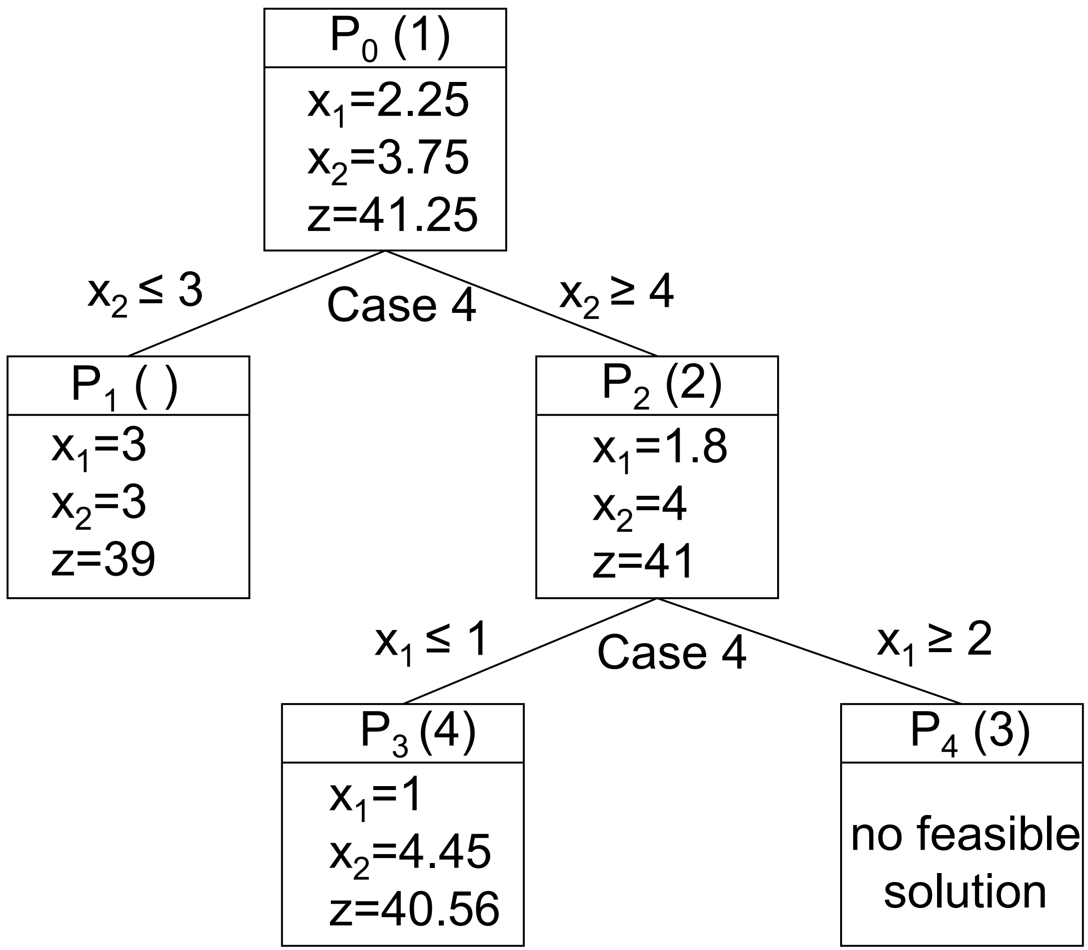


Cases:
(1) Integer solution
(2) z*(LP-relaxation) ≤ $\underline{z}$
(3) No feasible solution
(4) z*(LP-relaxation) > $\underline{z}$

Table 3rd Iteration: Problem P_4

K	P_{i^*}	Sol. LP-relax. x_1	x_2	z	$\overline{z}$	$\underline{z}$	Δ	Incumbent IP sol.	Case	Branch	P_i	$z^*(LP)$
$\langle P_0]$	P_0	2.25	3.75	41.25	41.25	34	21.32%	$x_1 = 2$	4)	$x_2 \leq 3$	P_1	39
								$x_2 = 3$		$x_2 \geq 4$	P_2	41
$\langle P_2, P_1]$	P_2	1.8	4	41	41	34	20.59%		4)	$x_1 \leq 1$	P_3	40.56
										$x_1 \geq 2$	P_4	-
$\langle P_4, P_3, P_1]$												

B&B-Tree 3rd Iteration: Problem P_4



Graphical Representation P1, P3 and P4

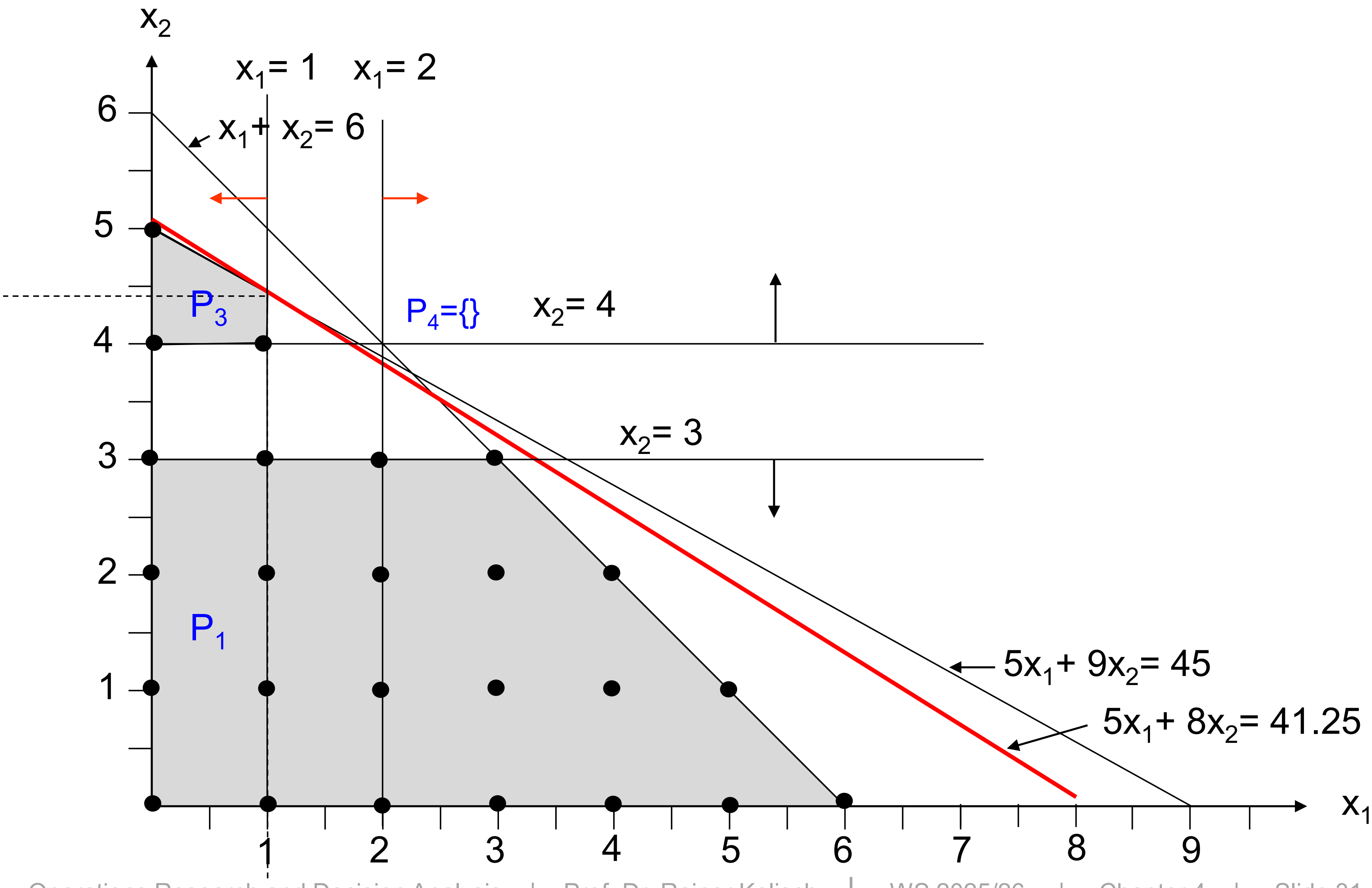


Table 4th Iteration: Problem P_3

Cases:
(1) Integer solution
(2) z*(LP-relaxation) ≤ $\underline{z}$
(3) No feasible solution
(4) z*(LP-relaxation) > $\underline{z}$

		Sol. LP-relax.						Incumbent				
K	P_{i^*}	x_1	x_2	z	$\overline{z}$	$\underline{z}$	Δ	IP sol.	Case	Branch	P_i	$z^*(LP)$
$\langle P_0]$	P_0	2.25	3.75	41.25	41.25	34	21.32%	$x_1 = 2$ $x_2 = 3$	4)	$x_2 \leq 3$ $x_2 \geq 4$	P_1 P_2	39 41
$\langle P_2, P_1]$	P_2	1.8	4	41	41	34	20.59%		4)	$x_1 \leq 1$ $x_1 \geq 2$	P_3 P_4	40.56 -
$\langle P_4, P_3, P_1]$	P_4	–	–	–	40.56	34	19.29%		3)			
$\langle P_3, P_1]$												

B&B-Tree 4th Iteration: Problem P_3

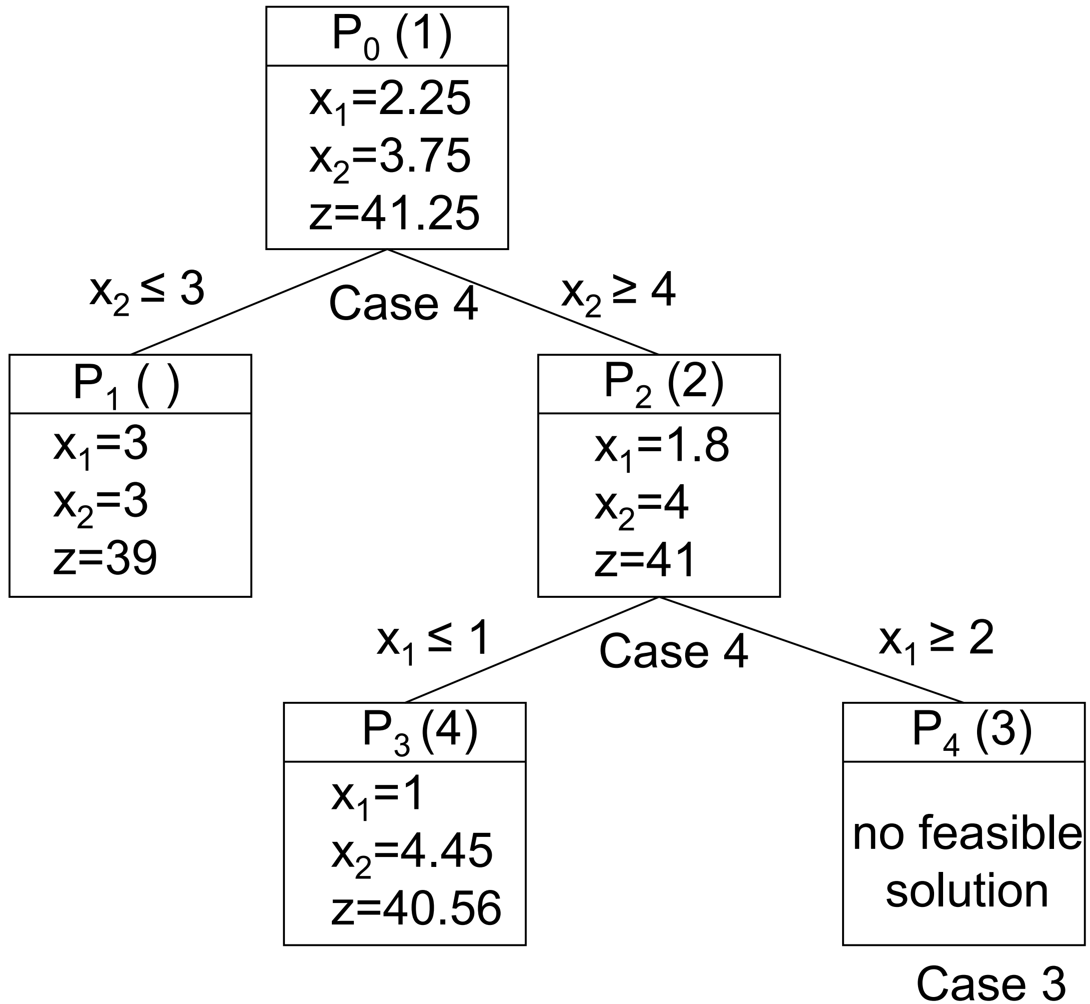
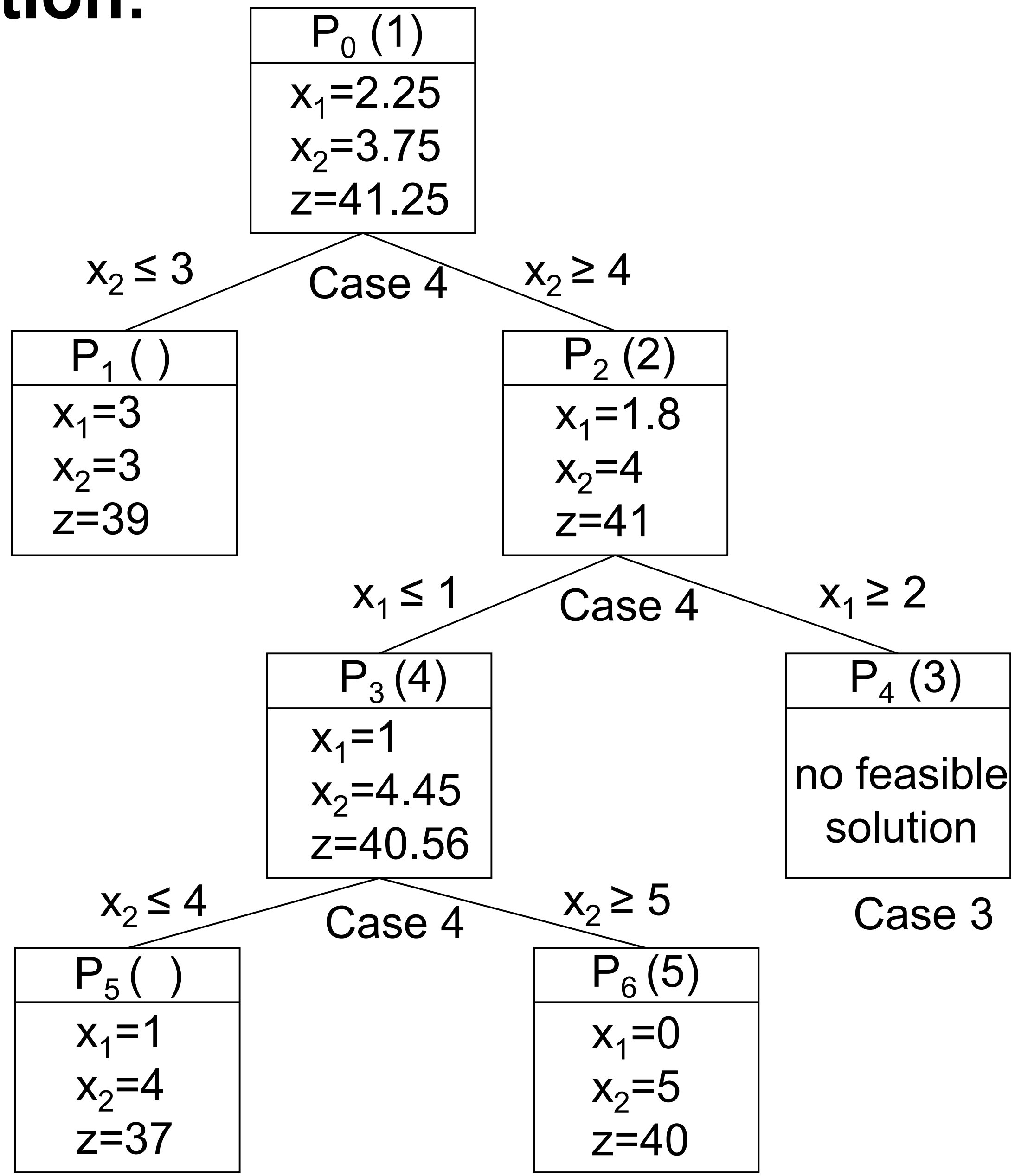


Table 5th Iteration: Problem P_6

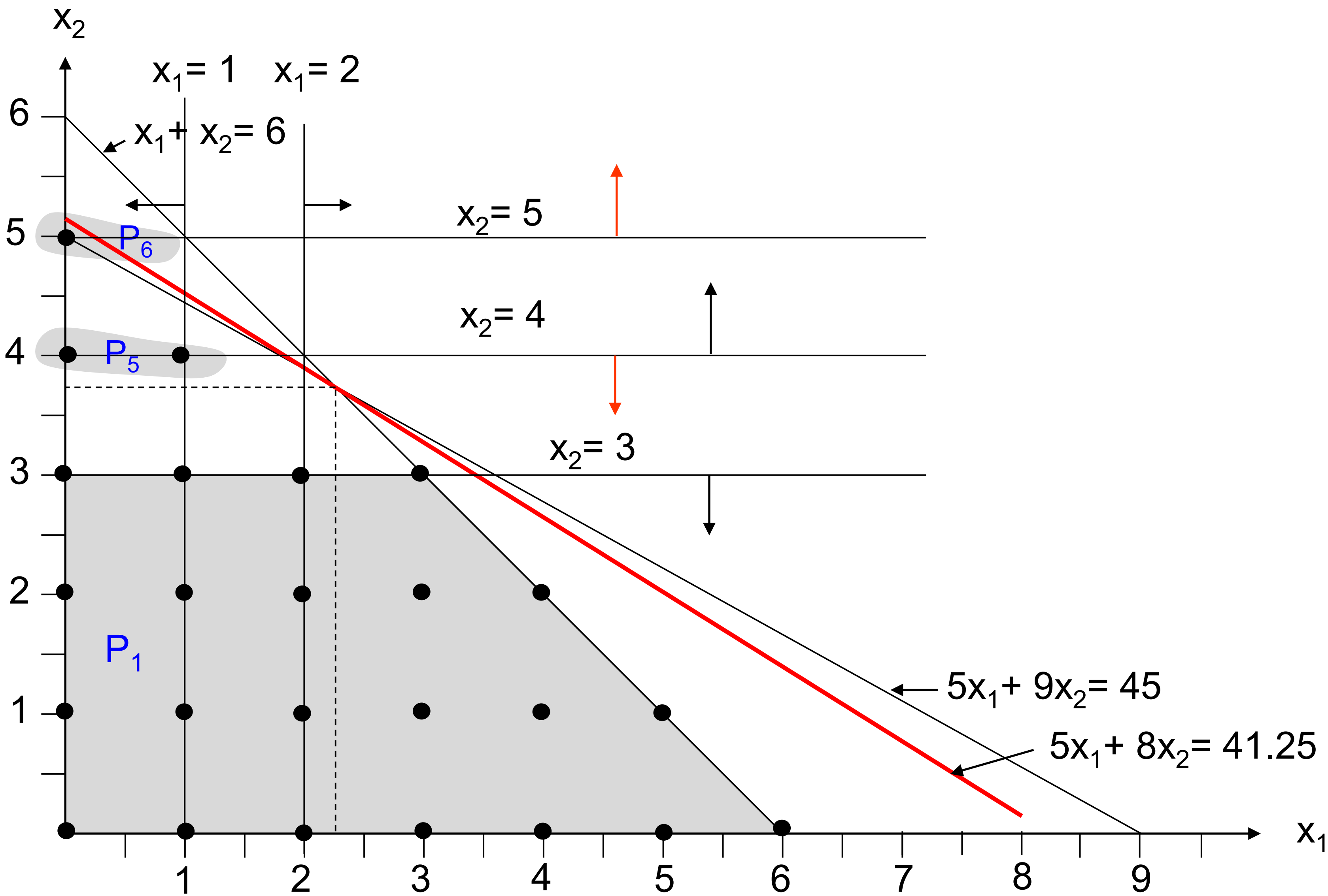
Cases:
(1) Integer solution
(2) z*(LP-relaxation) ≤ $\underline{z}$
(3) No feasible solution
(4) z*(LP-relaxation) > $\underline{z}$

K	P_{i^*}	Sol. LP-relax. x_1	x_2	z	$\overline{z}$	$\underline{z}$	Δ	Incumbent IP sol.	Case	Branch	P_i	$z^*(LP)$
$\langle P_0]$	P_0	2.25	3.75	41.25	41.25	34	21.32%	$x_1 = 2$	4)	$x_2 \leq 3$	P_1	39
								$x_2 = 3$		$x_2 \geq 4$	P_2	41
$\langle P_2, P_1]$	P_2	1.8	4	41	41	34	20.59%		4)	$x_1 \leq 1$	P_3	40.56
										$x_1 \geq 2$	P_4	-
$\langle P_4, P_3, P_1]$	P_4	–	–	–	40.56	34	19.29%		3)			
$\langle P_3, P_1]$	P_3	1	4.44	40.56	40.56	34	19.29%		4)	$x_2 \leq 4$	P_5	37
										$x_2 \geq 5$	P_6	40
$\langle P_6, P_5, P_1]$												

B&B-Tree 5th Iteration: Problem P_6



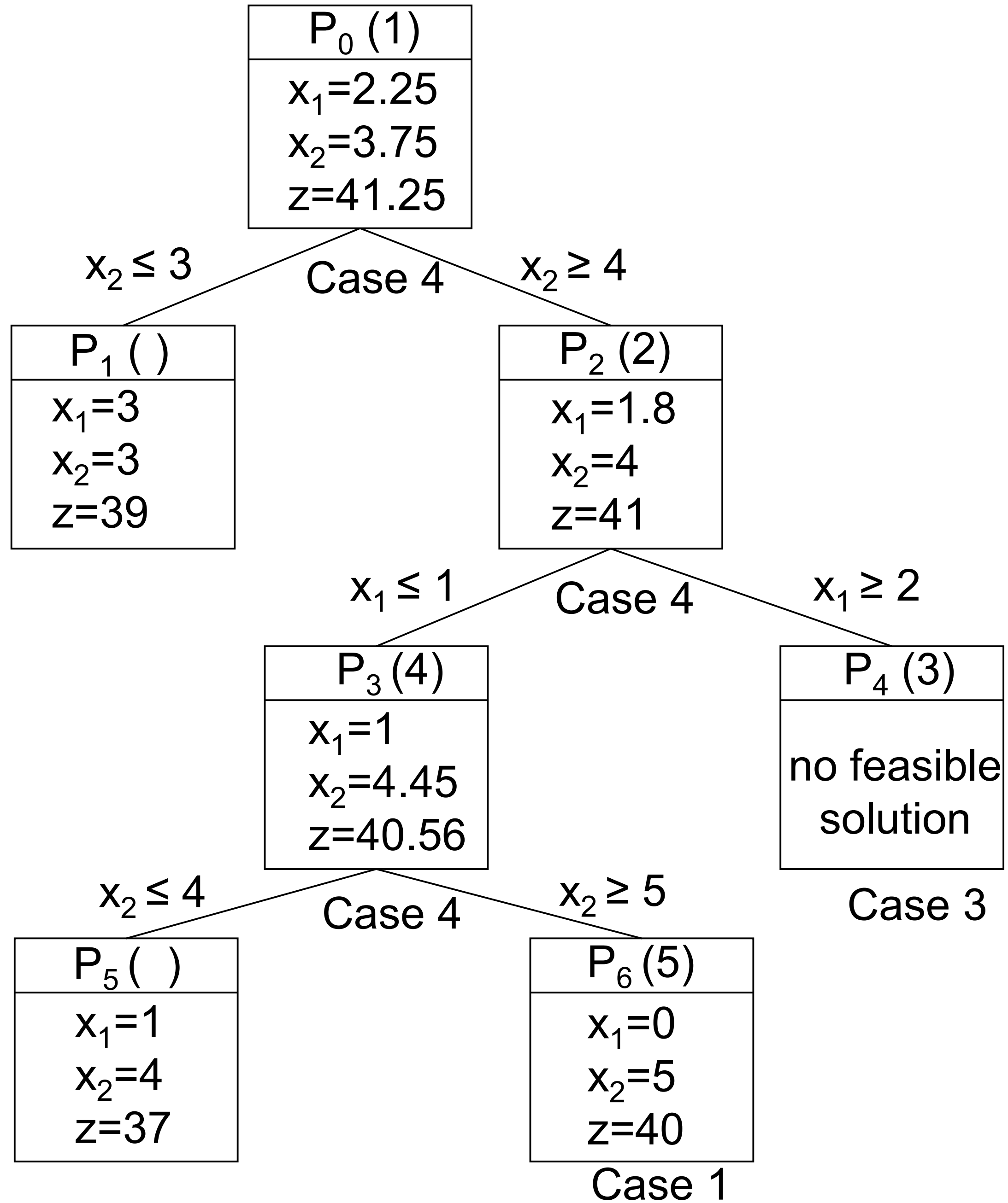
Graphical Representation P_1, P_5, and P_6



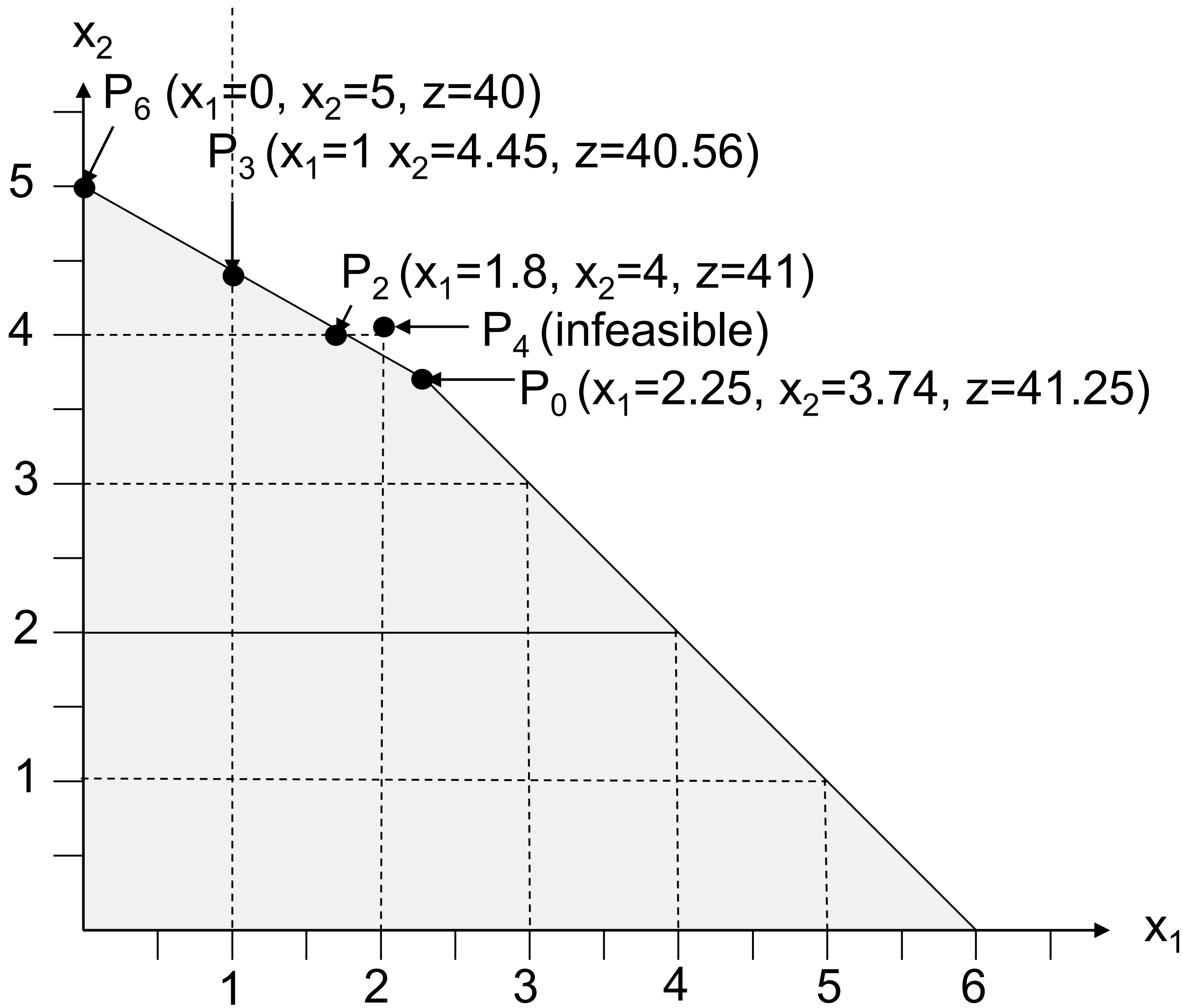
Final B&B-Table

K	P_{i^*}	Sol. LP-relax. x_1	x_2	z	$\overline{z}$	$\underline{z}$	Δ	Incumbent IP sol.	Case	Branch	P_i	$z^*(LP)$
$\langle P_0]$	P_0	2.25	3.75	41.25	41.25	34	21.32%	$x_1 = 2$, $x_2 = 3$	4)	$x_2 \leq 3$	P_1	39
										$x_2 \geq 4$	P_2	41
$\langle P_2, P_1]$	P_2	1.8	4	41	41	34	20.59%		4)	$x_1 \leq 1$	P_3	40.56
										$x_1 \geq 2$	P_4	-
$\langle P_4, P_3, P_1]$	P_4	–	–	–	40.56	34	19.29%		3)			
$\langle P_3, P_1]$	P_3	1	4.44	40.56	40.56	34	19.29%		4)	$x_2 \leq 4$	P_5	37
										$x_2 \geq 5$	P_6	40
$\langle P_6, P_5, P_1]$	P_6	0	5	40	40	40	0%	$x_1 = 0$, $x_2 = 5$	1)			
Opt. Sol.		0	5	40								

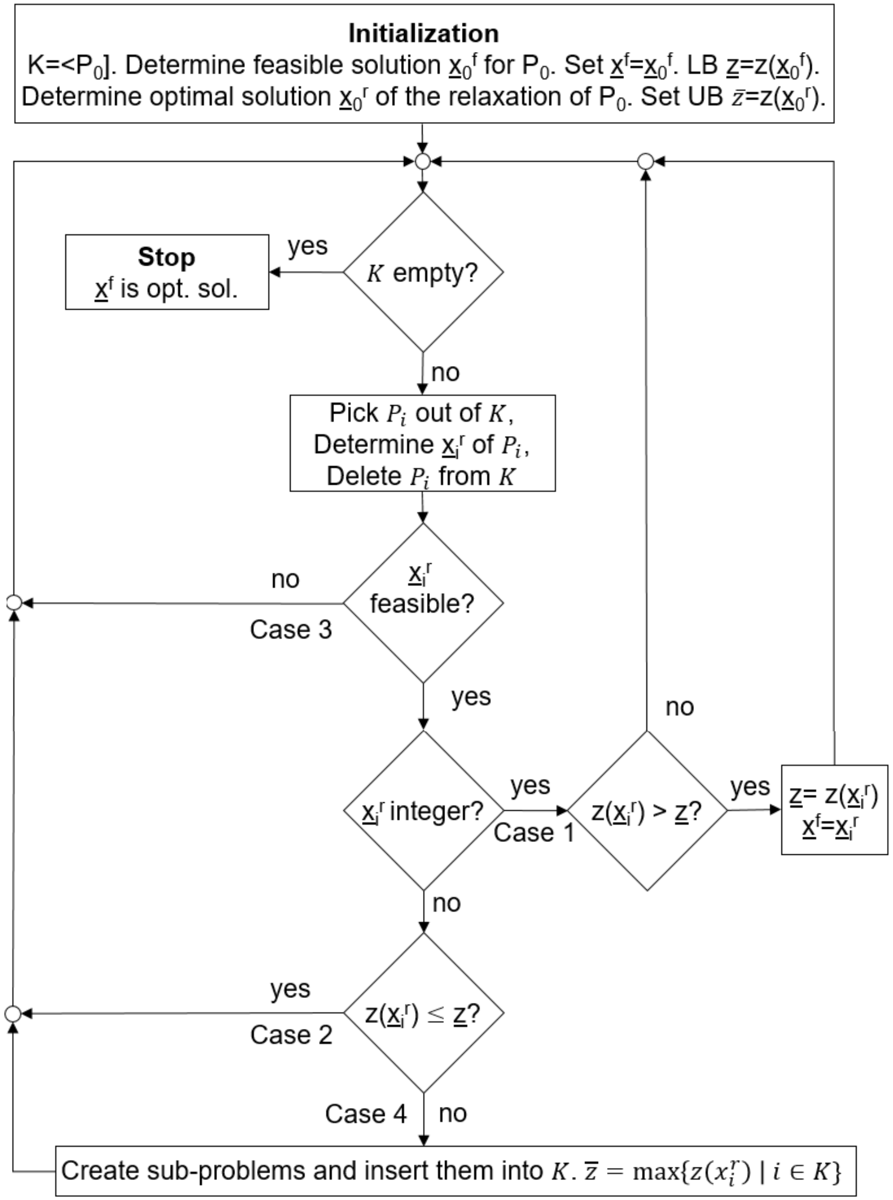
Final B&B-Tree



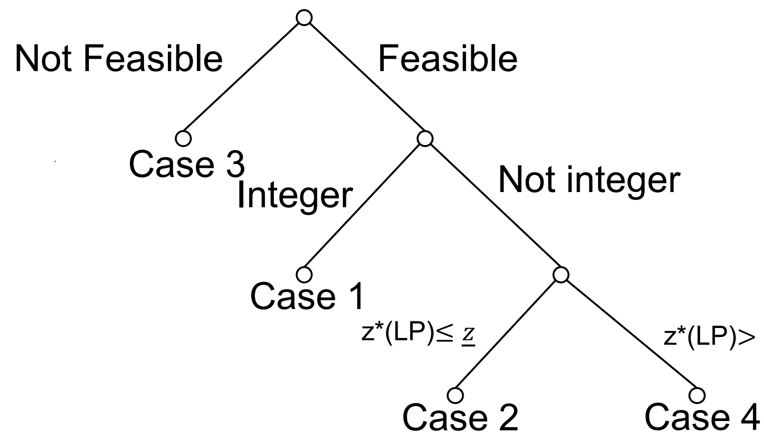
Solution Space and Visited Solutions



Flowchart of the B&B-Algorithm



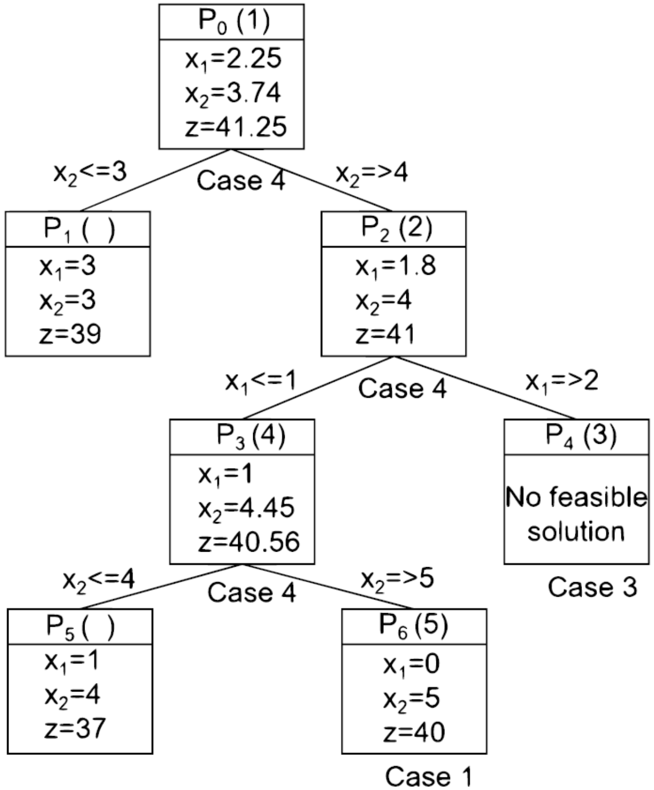
Cases:



Property 10: Non-Increasing Objective Function Value

Observe: z^*(LP-relaxation of P_2) < z^* (LP-relaxation of P_0). Generally, we have:

<u>Property 10</u>: In case of a maximization problem, the following holds for the optimal objective function value of parent-and child-problem, both for the LP-relaxation and the integer problem: z^*(child-problem) ≤ z^*(parent-problem).



Chapter 4.3.2

Truncated Branch-and-Bound

Truncated Branch-and-Bound: Idea

We stop the algorithm as soon as the solution gap Δ falls below a given threshold $\overline{\Delta}$.

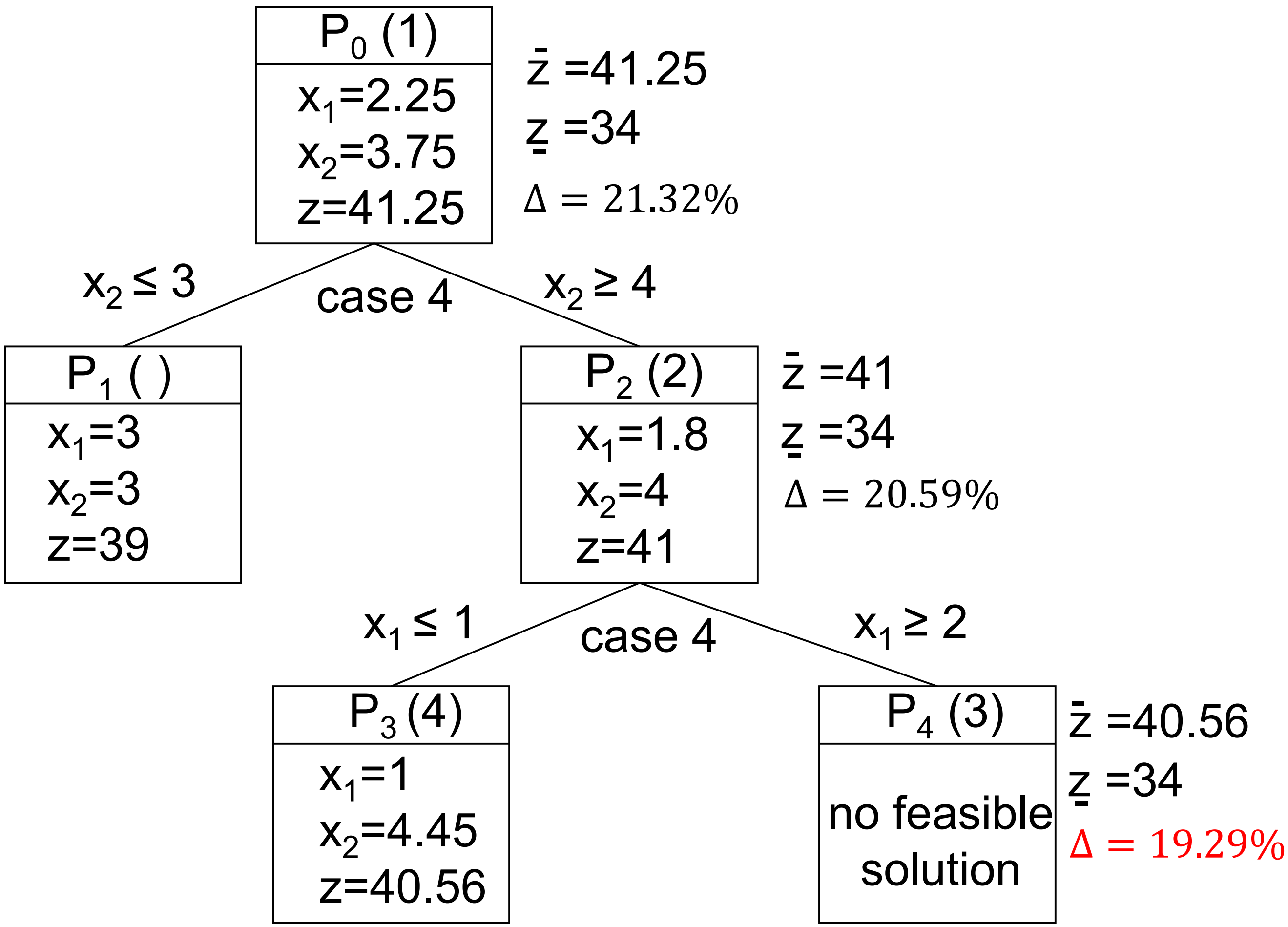
We then have a guarantue that the best possible solution is not more than $\overline{\Delta}$*100% percent away from the found solution.

Truncated Branch-and-Bound: Example

For $\overline{\Delta}$ = 20% we obtain:

K	P_{i^*}	Sol. LP-relax. x_1	x_2	z	$\overline{z}$	$\underline{z}$	Δ	Incumbent IP sol.	Case	Branch $P_i: z^*(LP)$
$\langle P_0]$	P_0	2.25	3.75	41.25	41.25	34	21.32%	$x_1 = 2$ $x_2 = 3$	4)	$x_2 \leq 3 \rightarrow P_1$: 39 $x_2 \geq 4 \rightarrow P_2$: 41
$\langle P_2, P_1]$	P_2	1.8	4	41	41	34	20.59%		4)	$x_1 \leq 1 \rightarrow P_3$: 40.56 $x_1 \geq 2 \rightarrow P_4$: -
$\langle P_4, P_3, P_1]$	P_4	–	–	–	40.56	34	19.29%		3)	

Final Truncated B&B-Tree



Chapter 4.3.3

Branch-and-Bound without Simplex Algorithm

Example Capital Budgeting Problem

A budget of 14 is available and each investment (project) can be selected or not selected.

Project	NPV	Budget demand
1	16	5
2	22	7
3	12	4
4	8	3

Decision**:** Which projects should be selected in order to maximize the total NPV?

Binary Program for the Capital Budgeting Problem (CBP)

Decision variables: $x_i \in \{0,1\}$ ← Binary variable is special case of integer

↳ $x_i = 1$ if project $i \{i=1,\dots,4\}$ is selected

Parameters:

c_i = NPV of project i

b_i = budget demand of project i

B = available budget

General

$$\text{Max} \sum_{i=1}^{4} c_i x_i$$

$$\text{s.t.} \sum_{i=1}^{4} b_i x_i \leq B$$

$$x_i \in \{0,1\} \quad i=1,\dots,4$$

Specific (prev. slide)

$$\Rightarrow \text{Max } 16x_1 + 22x_2 + 12x_3 + 8x_4$$

$$\text{s.t. } 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14$$

$$x_1 \in \{0,1\}, \dots$$

Solving the LP-Relaxation of the CBP with the Simplex Algorithm

```
Max 16x1 + 22x2 + 12x3 + 8x4
st
5x1 + 7x2 + 4x3 + 3x4 <= 14
x1 <= 1
x2 <= 1
x3 <= 1
x4 <= 1
```

Decision variables

ROW	(BASIS)	X1	X2	X3	X4	SLK 2	SLK 4	SLK 5	SLK 6	
Obj. f. 1	ART	0.000	0.000	0.000	1.000	3.000	1.000	0.000	0.000	44.000
2	X3	0.000	0.000	1.000	0.750	0.250	-1.750	0.000	0.000	0.500
3	X1	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000
4	X2	0.000	1.000	0.000	0.000	0.000	1.000	0.000	0.000	1.000
5	SLK 5	0.000	0.000	0.000	-0.750	-0.250	1.750	1.000	0.000	0.500
6	SLK 6	0.000	0.000	0.000	1.000	0.000	0.000	0.000	1.000	1.000

Calculating the Upper Bound in Table-Format

without Simplex

Project	NPV (c_j)	Budget demand (a_j)	$\frac{c_j}{a_j}$ (Value per budget demand; b)	Selection Sequence	x_j	$\sum c_j \cdot x_j$ (acc. value)	$\sum a_j \cdot x_j$ (b; acc. budget dem.)	Budget left: $B-\sum b_j x_j$
								14
1	16	5	$\frac{16}{5} = 3.2$	largest → 1	$x_1 = 1$	16	5	9
2	22	7	$\frac{22}{7} = 3.14$	2	$x_2 = 1$	38	12	2
3	12	4	$\frac{12}{4} = 3$	3	$x_3 = \frac{1}{2}$ (budget left not enough)	44	14	0
4	8	3	$\frac{8}{3} = 2.17$	smallest → 4	$x_4 = 0$	44	14	

Relaxed solution

UB=44

Calculating the Lower Bound in Table-Format

Project	NPV (c_j)	Budget demand (a_j)	$\frac{c_j}{a_j}$ b	Selection Sequence	x_j	$\sum c_j \cdot x_j$	$\sum a_j \cdot x_j$ b
1	16	5	3.2	1	$x_1 = 1$	16	5
2	22	7	3.14	2	$x_2 = 1$	38	12
3	12	4	3	3	$x_3 = 0$	"	"
4	8	3	2.67	4	$x_4 = 0$	"	"

no fractional products in lower bound; not enough budget for $x_3 = 1$ or $x_4 = 1$

LB=38

<u>Cases</u>:
(1) Integer solution
(2) z(LP-relaxation) ≤ $\underline{z}$
(3) No feasible solution
(4) z(LP-relaxation) > $\underline{z}$

Branch-and-Bound Table

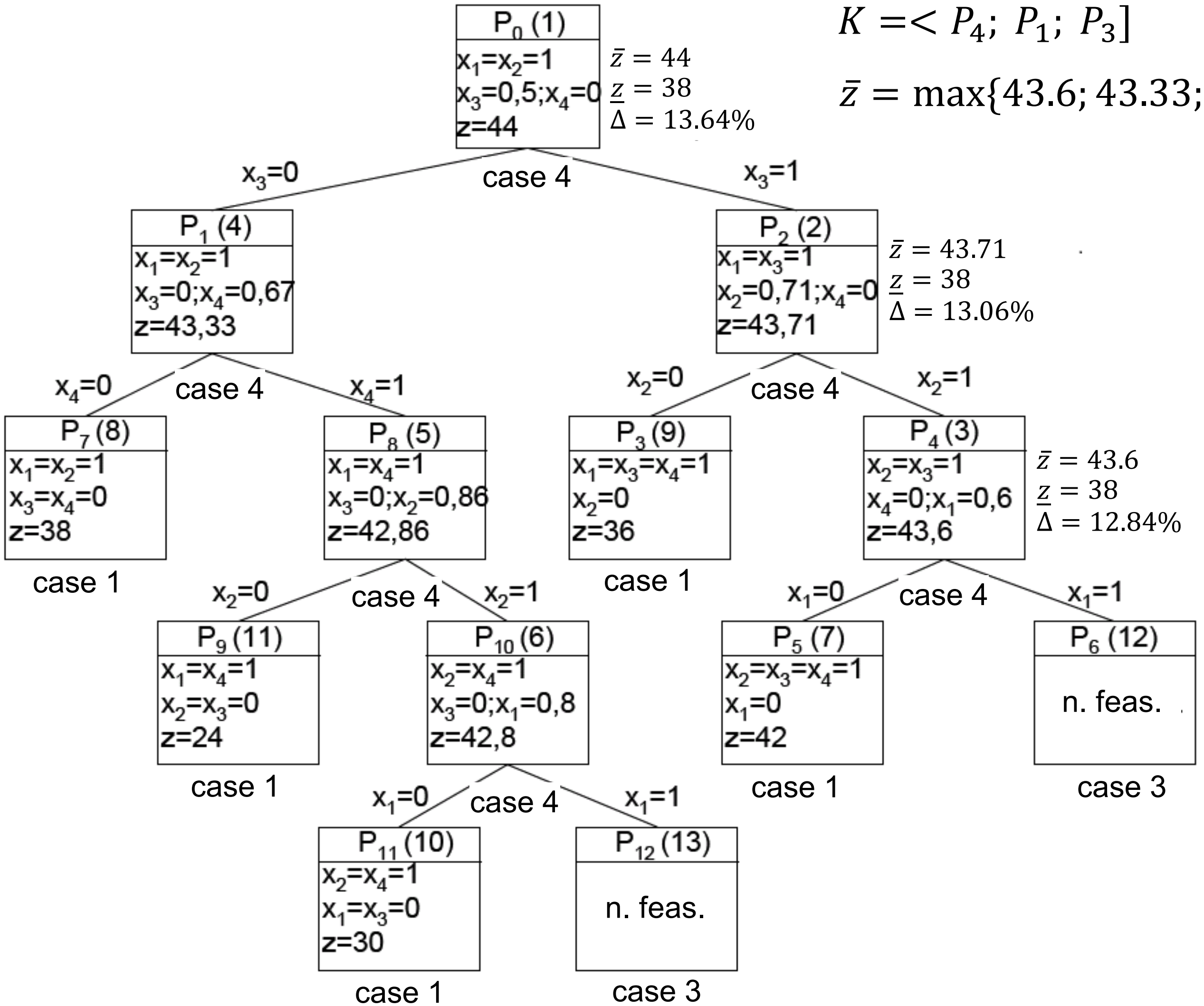
<u>Selection of the next sub-problem</u>**:** MUB with tie-breaker „smaller problem number".

		Solution of the relaxation										
K	P_{i^*}	x_1	x_2	x_3	x_4	z	$\overline{z}$	$\underline{z}$	Δ	Case	Branching	UB
<P_0]	P_0											

Updating of UB when Selecting the 3rd Subproblem

$K = < P_4;\ P_1;\ P_3]$

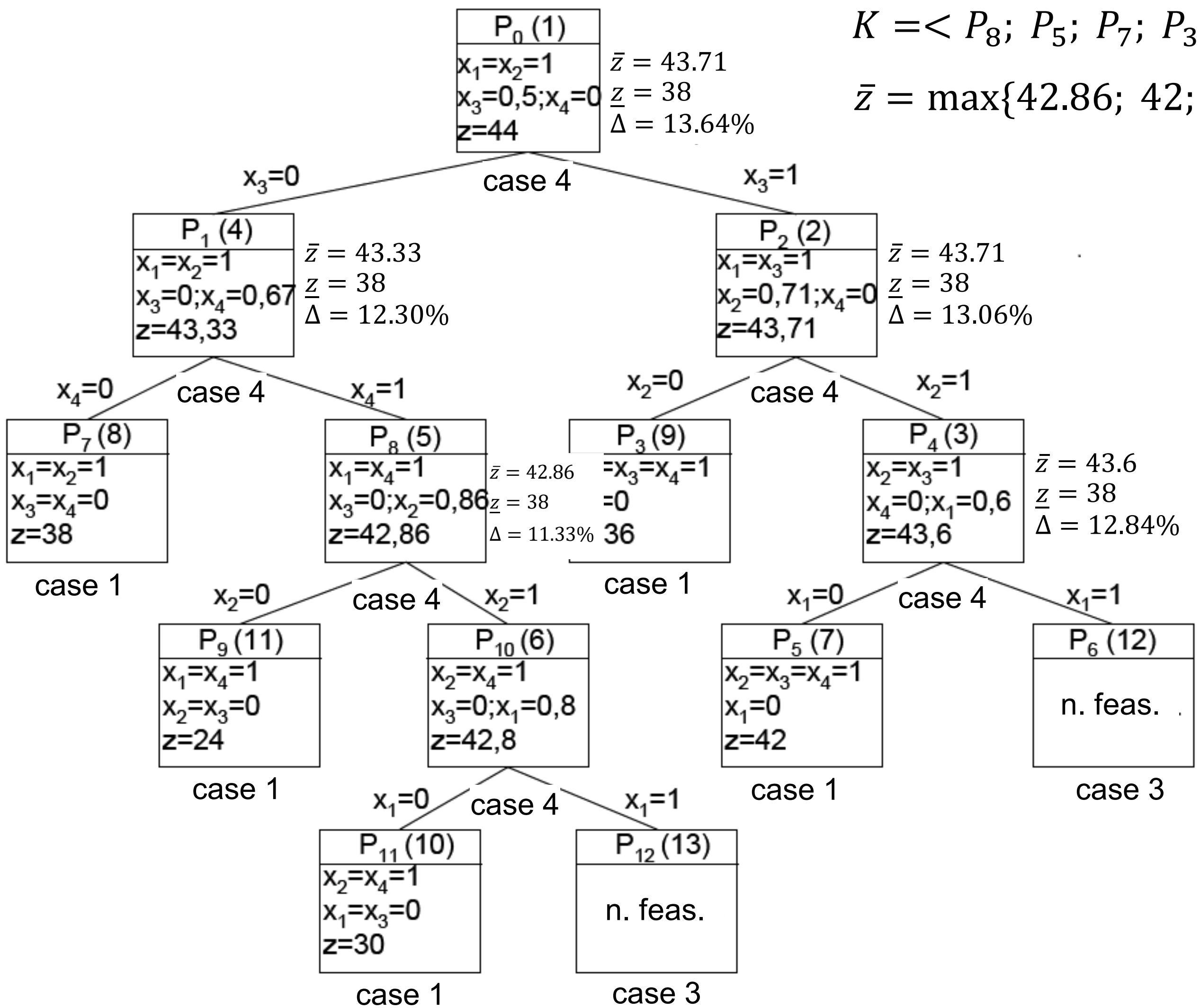
$\bar{z} = \max\{43.6; 43.33;\ 36\} = 43.6$



Updating of UB when Selecting the 5th Subproblem

$K =< P_8;\ P_5;\ P_7;\ P_3;\ P_6]$

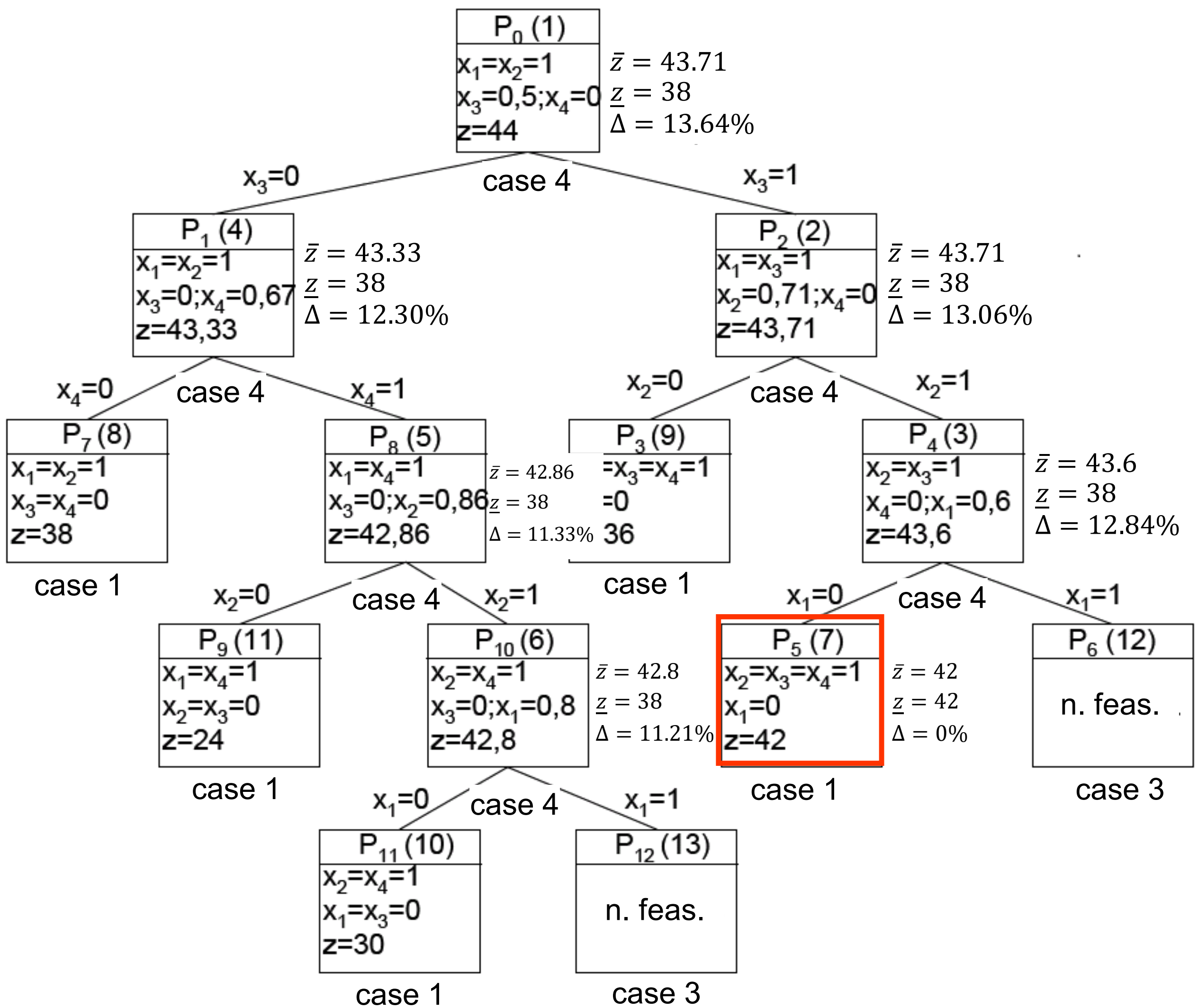
$\bar{z} = \max\{42.86;\ 42;\ 38;\ 36;\ "-"\} = 42.86$



B&B-Protocol with Updating of Upper Bounds

		Relaxation										
K	P_{i^*}	x_1	x_2	x_3	x_4	z	$\underline{z}$	$\underline{\underline{z}}$	Δ	Case	Branching	UB
<P_0]	P_0	1	1	$^1/_2$	0	44	44	38	13.64%	(4)	$x_3 = 0 \Rightarrow P_1$ $x_3 = 1 \Rightarrow P_2$	$43^1/_3$ $43^5/_7$
<P_2,P_1]	P_2	1	$^5/_7$	1	0	$43^5/_7$	43.71	38	13.06%	(4)	$x_2 = 0 \Rightarrow P_3$ $x_2 = 1 \Rightarrow P_4$	36 $43^3/_5$
<P_4,P_1,P_3]	P_4	$^3/_5$	1	1	0	$43^3/_5$	43.6	38	12.84%	(4)	$x_1 = 0 \Rightarrow P_5$ $x_1 = 1 \Rightarrow P_6$	42 n. feas.
<P_1,P_5,P_3, P_6]	P_1	1	1	0	$^2/_3$	$43^1/_3$	43.33	38	12.30%	(4)	$x_4 = 0 \Rightarrow P_7$ $x_4 = 1 \Rightarrow P_8$	38 $42^6/_7$
<P_8,P_5,P_7, P_3,P_6]	P_8	1	$^6/_7$	0	1	$42^6/_7$	42.86	38	11.33%	(4)	$x_2 = 0 \Rightarrow P_9$ $x_2 = 1 \Rightarrow P_{10}$	24 $42^4/_5$
<P_{10},P_5,P_7, P_3,P_9,P_6]	P_{10}	$^4/_5$	1	0	1	$42^4/_5$	42.8	38	11.21%	(4)	$x_1 = 0 \Rightarrow P_{11}$ $x_1 = 1 \Rightarrow P_{12}$	30 n. feas.
<P_5,P_7,P_3,P_{11}, P_9,P_6,P_{12}]	P_5	0	1	1	1	42	42	42	0%	(1)		

Branch-and-Bound Tree with Optimal Solution



Chapter 4.1

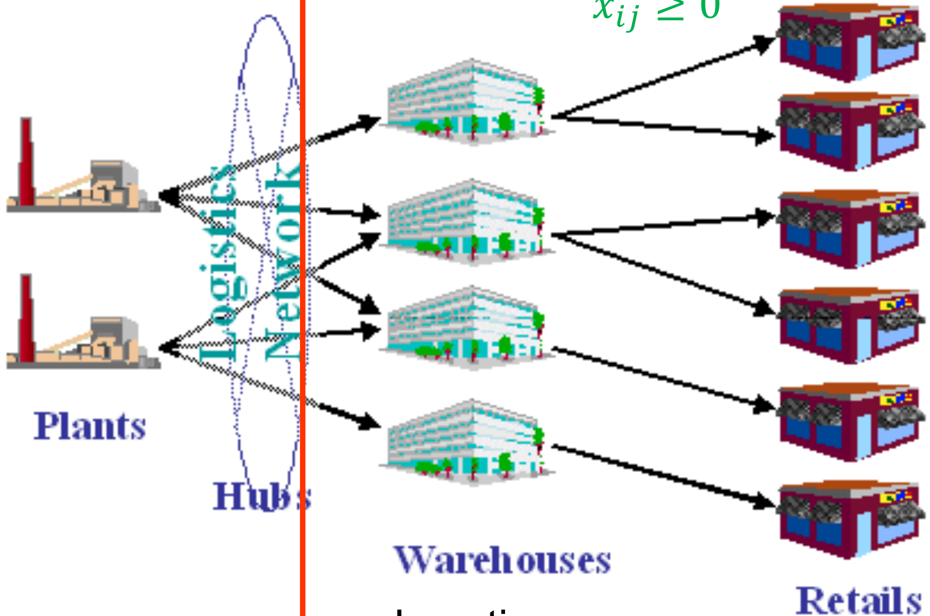
(Further) Examples

Mixed-Integer Program

<u>Example 2</u>: Warehouse Location Problem

Variable cost c_{ij} for delivering one unit from warehouse i to retailer j

$x_{ij} \geq 0$



Locations i=1,….,m with fixed cost f_i

Retailer j=1,….,n with demand d_j

$y_i \in \{1,0\}$

Source: http://www.research.ibm.com/trl/projects/optimization/wltp_e.htm

Mixed-Integer Program for the Warehouse Location Problem

Variables:

y_i = 1, if warehouse i is opened, 0 otherwise (Integer variable)

x_{ij} = Number of units transported from warehouse i to retailer j (continuous variable)

Parameter:

f_i = Fixed cost for warehouse i

d_j = Demand of retailer j

c_{ij} = Transportation cost for transporting one unit from warehouse i to retailer j

Mixed-Integer Program

$$\text{Min} \sum_{i=1}^{m} f_i \cdot y_i + \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} \cdot x_{ij} \qquad (1)$$

subject to

$$\sum_{i=1}^{m} x_{ij} = d_j \qquad (j=1,\dots,n) \qquad (2)$$

For each retailer, the Σ units shipped from warehouses to the retailer has to equal demand

$$\sum_{j=1}^{n} x_{ij} - \left(\sum_{j=1}^{n} d_j\right) \cdot y_i \leq 0 \qquad (i=1,\dots,m) \qquad (3)$$

sum of demand across retailers (total supply chain demand)

For each warehouse, the Σ units shipped from the warehouse to retailers has to be ≤ total supply chain demand

$$x_{ij} \geq 0 \qquad (i=1,\dots,m)\ (j=1,\dots,n) \qquad (4)$$

$$y_i \in \{0,1\} \qquad (i=1,\dots,m) \qquad (5)$$

$$\left\{ \begin{array}{lll} y_i \leq 1 & (i=1,\dots,m) & (5.a) \\ y_i \geq 0 & (i=1,\dots,m) & (5.b) \\ y_i \in \mathbb{Z} & (i=1,\dots,m) & (5.c) \end{array} \right.$$

Definition: A mixed-integer program is a linear program with integer and continuous variables.

Binary Program

Example 3: Capital Budgeting Problem

A budget of 14 is available and each investment (project) can be selected or not selected.

Project	NPV	Budget demand
1	16	5
2	22	7
3	12	4
4	8	3

Decision**:** Which projects should be selected in order to maximize the total NPV?

Binary Program

<u>Example 3</u>: Capital Budgeting Problem

Max $$z = 16 \cdot x_1 + 22 \cdot x_2 + 12 \cdot x_3 + 8 \cdot x_4 \quad (2.9)$$

subject to

$$5 \cdot x_1 + 7 \cdot x_2 + 4 \cdot x_3 + 3 \cdot x_4 \leq 14 \quad (2.10)$$

$$x_1, x_2, x_3, x_4 \in \{0,1\} \quad (2.11)$$

<u>Definition</u>: A binary program is a linear program where all variables are binary.

Summary: Integer and Mixed-Integer Programs

Integer Program (IP)	Mixed-Integer Program (MIP)	Binary Program (BP)
$x \in \mathbb{Z}_{\geq}$	$x \geq 0$ $y \in \mathbb{Z}_{\geq}$	$x \in \{0,1\}$
Production program with integer quantities	Warehouse location problem	Capital budgeting problem