# Applications of Linear Programming

*lecturer: András London*

University of Szeged
Institute of Informatics
Department of Computational Optimization

Lecture 4

# Introduction

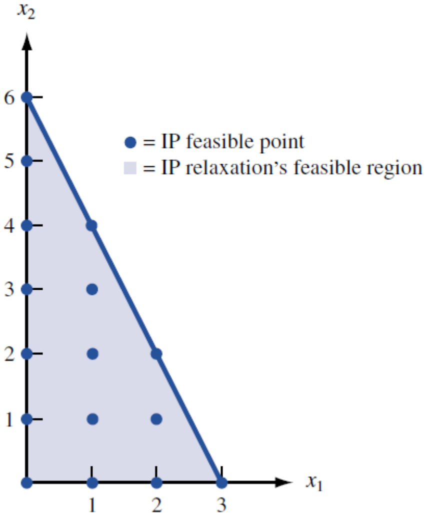
An LP in which all variables are required to be integers is called a **pure integer programming** (IP) problem.For example,

$$\begin{array}{lrcl} \max & z & = & 3x_1 + 2x_2 \\ \text{st} & x_1 + x_2 & \leq & 6 \\ & x_1, x_2 & \geq & 0 \\ & x_1, x_2 & & \text{integer} \end{array}$$

An IP in which only some of the variables are required to be integers is called a **mixed integer programming problem** (MIP). In the example above let $x_1, x_2 \geq 0$ and $x_2$ be an integer ($x_1$ is not required to be an integer).

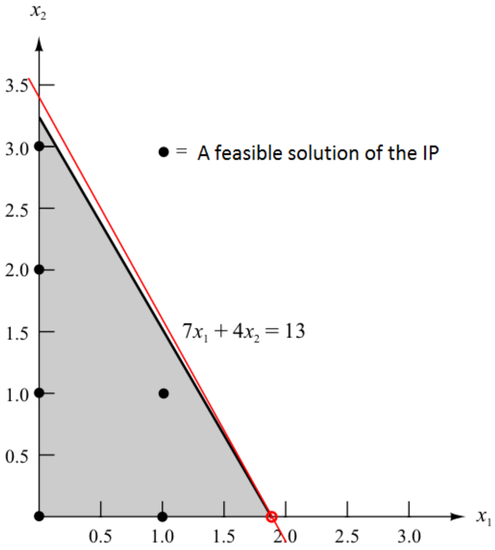
An integer programming problem in which all the variables must equal $0$ or $1$ is called a $0$–$1$ **IP**.

**LP-relaxation**: The LP obtained by **omitting all integer or** $0$–$1$ **constraints** on variables is called the LP relaxation of the IP.



**Propositions**:

- the feasible region for any IP must be contained in the feasible region of the corresponding LP relaxation
- Optimal objective function value for LP relaxation $\geq$ optimal value for IP (for max problems)
- If all corner points of the feasible region of the LP-relaxation is integer, then it has an integer optimal solution that is also optimal solution of the corresponding IP
- The optimal solution of the LP-relaxation can be arbitrarily far from the IP solution.



# The Branch-and-Bound Method for Solving IP Problems

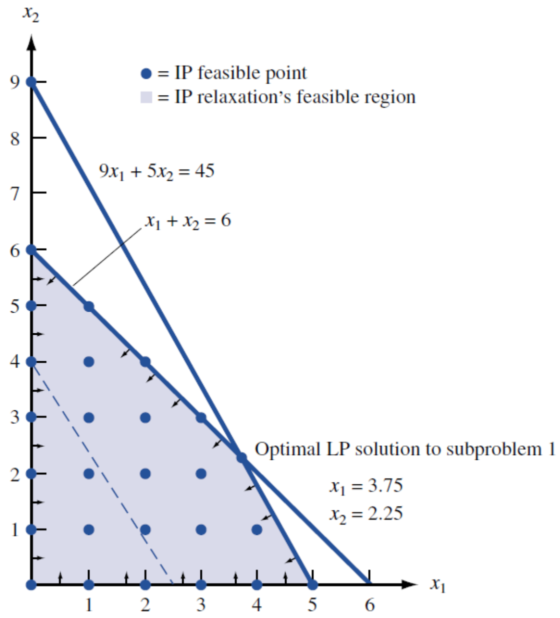
**Example**: The Telfa Corporation manufactures tables and chairs. A table requires 1 hour of labor and 9 square board feet of wood, and a chair requires 1 hour of labor and 5 square board feet of wood. Currently, 6 hours of labor and 45 square board feet of wood are available. Each table contributes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

**Solution**: Let $x_1 =$ number of tables manufactured, $x_2 =$ number of chairs manufactured. Because $x_1$ and $x_2$ must be integers, Telfa wants to solve the following IP:

$$
\begin{array}{lrcll}
\max & z & = & 8x_1 + 5x_2 & \\
\text{st} & x_1 + x_2 & \leq & 6 & \text{(Labor constraint)} \\
 & 9x_1 + 5x_2 & \leq & 45 & \text{(Wood constraint)} \\
 & x_1, x_2 & \geq & 0, & \text{integer}
\end{array}
$$

Step #1: **solving the LP-relaxation** of the IP. If all the decision variables are integer in the optimal solution, then this is the optimal solution of the IP.

Step #2 (iteration): (At least one $x_i$ is not an integer). **Partition the feasible region for the LP-relaxation** in an attempt to find out more about the location of the IP's optimal solution.
If $x_i = x_i^*$ in the LP-relaxation optimum, then add $x_i \leq \lfloor x_i^* \rfloor$ and $x_i \geq \lceil x_i^* \rceil$ to our sub-problem.

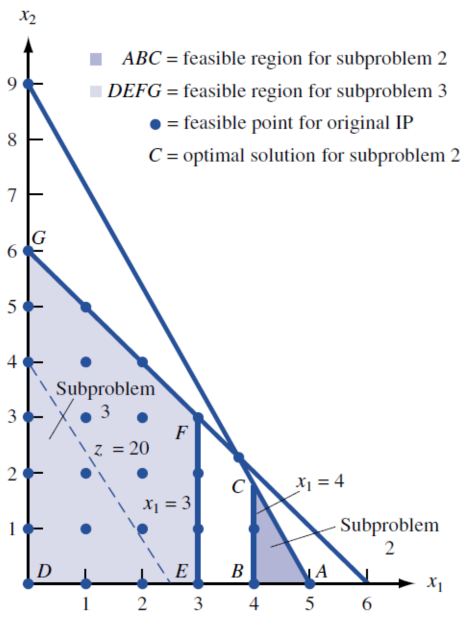


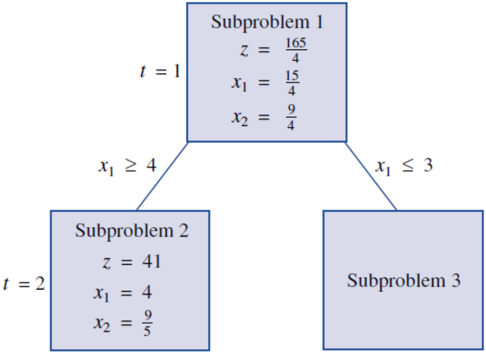
In the optimal solution of the LP-relaxation $x_1 = 3.75$ thus we obtain the following sub-problems:

**Subproblem 2** $=$ Subproblem 1 $+$ constraint $x_1 \geq 4$.

**Subproblem 3** $=$ Subproblem 1 $+$ constraint $x_1 \leq 3$.

Hence we ignore $x_1 \in ]3,4[$ non-integer solutions.



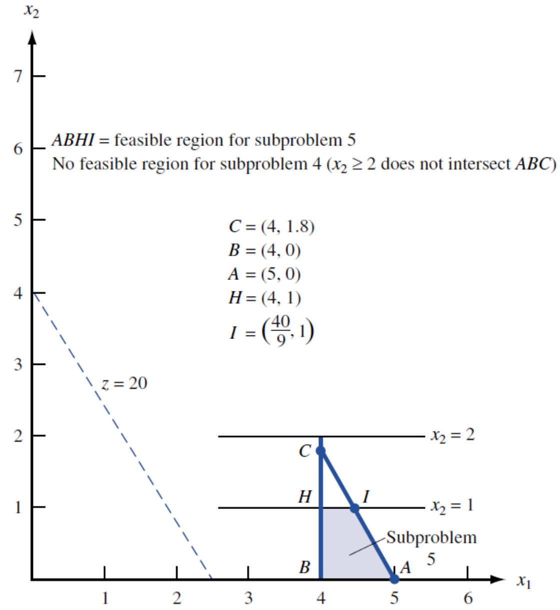


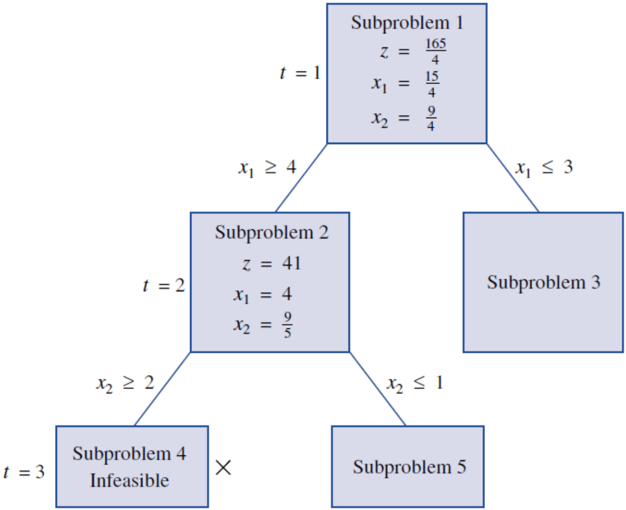

- A display of all subproblems that have been created is called a **tree**
- The root is Subproblem 1 (LP relaxation)
- The descendants are the corresponding subproblems
- The added constraint is represented on the edge
- The solutions of the LP subproblems are recorded on the nodes

The optimal solution to subproblem 2 did not yield an all-integer solution, so we choose to use subproblem 2 to create two new subproblems

**Subproblem 4**
$=$
subproblem 1 $+$
constraints $x_1 \geq 4$
and $x_2 \geq 2$
$=$
subproblem 2 $+$
constraint $x_2 \geq 2.$

**Subproblem 5**
$=$
subproblem 1 $+$
constraints $x_1 \geq 4$
and $x_2 \leq 1$
$=$
subproblem 2 $+$
constraint $x_2 \leq 1.$

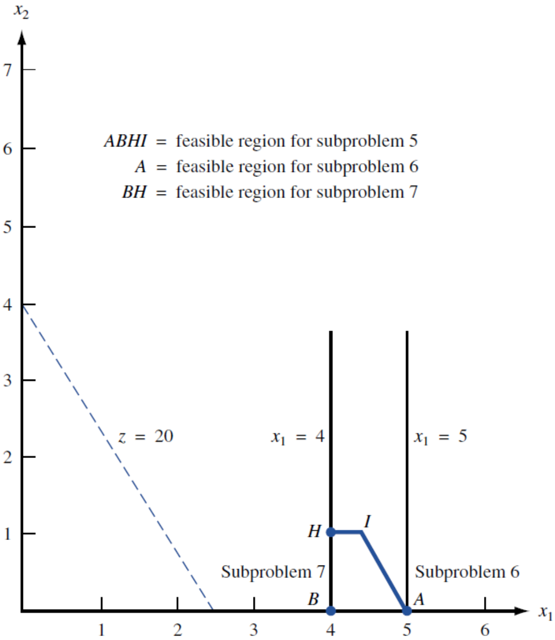


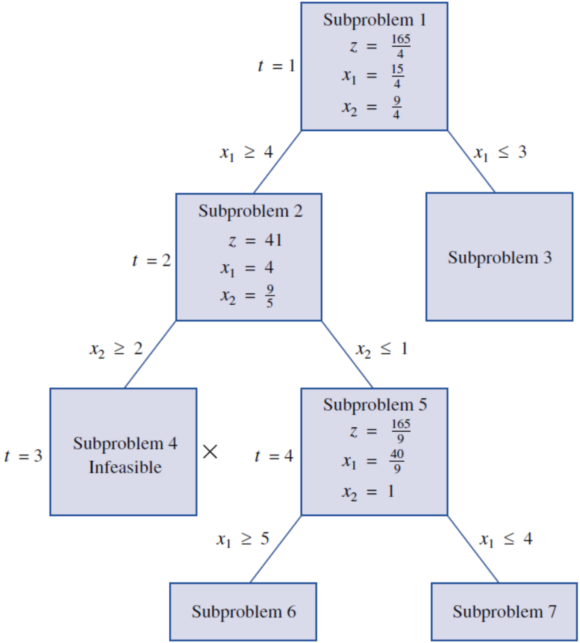



Subproblem 4 cannot yield the optimal solution to the IP. To indicate this fact, we place an $\times$ by subproblem 4 (and terminate this branch).

**Subproblem 6**
$=$
subproblem 5 $+$
constraint $x_1 \geq 5$

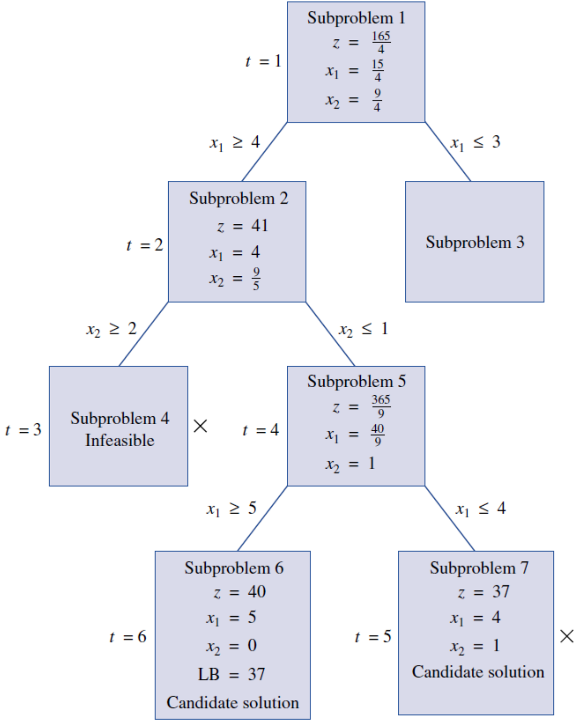
**Subproblem 7**
$=$
subproblem 5 $+$
constraint $x_1 \leq 4$



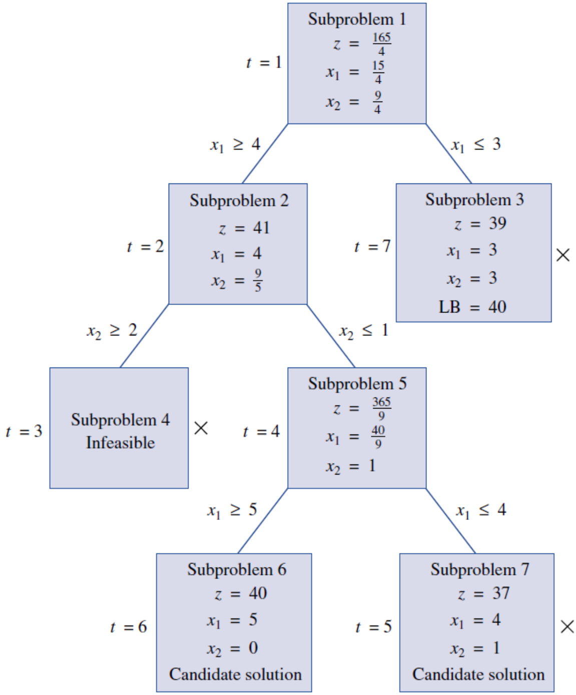



A solution obtained by solving a subproblem in which all variables have integer values is a **candidate solution**

We have a feasible solution to the original IP with $z = 37$ (subproblem 7), so we may conclude that the optimal $z$-value for the IP $\geq 37$. Thus, the $z$-value for the candidate solution is a **lower bound** on the optimal z-value for the original IP ($LB = 37$).




We find an integer solution for subproblem 6, and terminate this branch. Here $LB = 40$, thus 7 cannot yield the optimal solution of the IP (we denote this fact by placing an $\times$ by subproblem 7).




We find an integer solution for subproblem 3, and terminate this branch.

We see that there are no remaining unsolved subproblems, and that only **subproblem 6 can yield the optimal solution to the IP**. Thus, the optimal solution to the IP is for Telfa to manufacture 5 tables and 0 chairs. This solution will contribute $40 to profits.

A node of the branching tree is **„fathomed"** (closed), if

- there is no feasible solution of the corresponding subproblem
- the subproblem yields an optimal solution in which all variables have integer values
- the optimal $z$-value for the subproblem does not exceed (in a max problem) the current $LB$

A subproblem may be **eliminated** from consideration in the following situations

- The subproblem is infeasible
- the LB (representing the $z$-value of the best candidate to date) is at least as large as the z-value for the subproblem

# Solving Knapsack Problems by the Branch-and-Bound Method

A **knapsack problem** is an IP with a single constraint.

**Example**: Josie Camper is going to a 2-day trip. There are 4 items that he wants to pack, but the total weight cannot exceeds 14kg. The following table shows the weight and utility of each item.

| Item | Weight(kg) | Utility |
|---|---|---|
| 1 | 5 | 16 |
| 2 | 7 | 22 |
| 3 | 4 | 12 |
| 4 | 3 | 8 |

| Item | Weight(kg) | Utility | Relative utility | Rank |
|---|---|---|---|---|
| 1 | 5 | 16 | 3.2 | 1. |
| 2 | 7 | 22 | 3.1 | 2. |
| 3 | 4 | 12 | 3 | 3. |
| 4 | 3 | 8 | 2.7 | 4. |

**Mathematical model**:

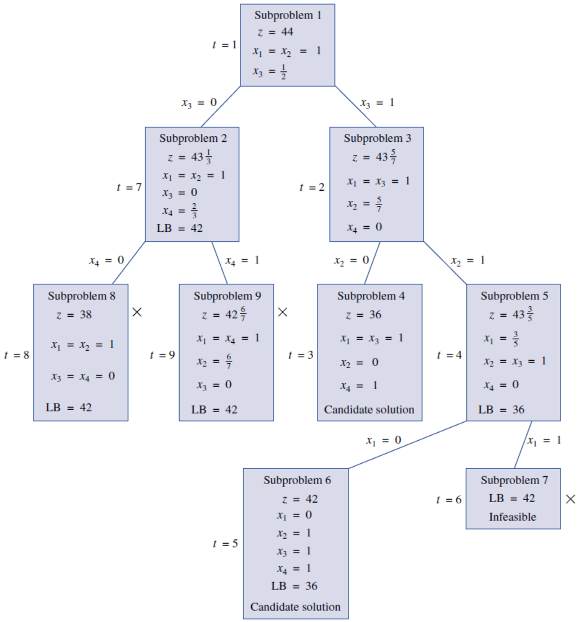
Let $x_i = 1$ if he brings item $i$ and $x_i = 0$ if not. Then the task is to solve

$$\begin{aligned} \max \quad & z = 16x_1 + 22x_2 + 12x_3 + 8x_4 \\ \text{st} \quad & 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\ & x_i \in \{0, 1\} \end{aligned}$$

The LP-relaxation can be solved easily. Put the items to the knapsack according to their relative utility (the most important first). If there is no more space for the item, put just an appropriate „part" of it.

We put the first 2 items of 12 kg total, and „half" of the 3rd item: $x_1 = 1, x_2 = 1, x_3 = 0.5, z = 16 + 22 + 0.5 * 12 = 44$.

Partitioning: $x_3 = 1$ (and we have 10kg free space) or $x_3 = 0$ (and we can use the rest of the item).




| Item | Weight | Utility |
|---|---|---|
| 1 | 5 | 16 |
| 2 | 7 | 22 |
| 3 | 4 | 12 |
| 4 | 3 | 8 |

# Remarks on the Branch and Bound technique

- In case of knapsack problems $2^n$ subproblems have to be solved in **worst case**. This means that the problem is **NP-hard** ($\sim$ cannot be solved in polynomial time).
- The situation is even worst for IP problems, $2^{Mn}$, where $M$ is the number of integers a variable can get.
- Traversing the branching tree can be LIFO (Last In First Out) that is a depth-first search or FIFO (First In First Out) that is a breadth-first search.

**Fix-charge problems**: Suppose activity $i$ incurs a fixed charge if undertaken at any positive level. Let $x_i > 0$ be the level of activity (e.g. production number).
In the model

- let $y_1 = 1$ if activity $i$ is undertaken at positive level $(x_i > 0)$ and $y_1 = 0$ if $x_i = 0$.
- constraint of the form $x_i \leq M_i y_i$ must be added to the formulation, where $M_i$ must be large enough to ensure that $x_i$ will be less than or equal to $M_i$.

**Minimum level of production**: if we produce the product $i$, then at least $L$ must be produced.
In the model

- let $y_i = 1$ if we produce at least one $i$ and $y_i = 0$ otherwise
- constraint of the form $x_i \geq 1000 y_i$ must be added.

**Either-Or constraint**: Suppose we want to ensure that at least one of the following two constraints (and possibly both) are satisfied:
$f(x_1, x_2, \ldots, x_n) \leq 0$ $g(x_1, x_2, \ldots, x_n) \leq 0$.
In the Model

- let $y = 1$ if $g \leq 0$, and $y = 0$ ha $f \leq 0$.
- add constraint $f(x_1, x_2, \ldots, x_n) \leq My$,
- add constraint $g(x_1, x_2, \ldots, x_n) \leq M(1 - y)$
- where $M \geq \max\{f, g\}$ for all $(x_1, x_2, \ldots, x_n)$

**If-Then constraint**: Suppose we want to ensure that
$f(x_1, x_2, \ldots, x_n) > 0$ implies $g(x_1, x_2, \ldots, x_n) \geq 0$.
In the model

- let $y = 1$ if $f > 0$, $y = 0$ if $f \leq 0$.
- add constraint $f(x_1, x_2, \ldots, x_n) \leq My$,
- add constraint $g(x_1, x_2, \ldots, x_n) \geq -M(1 - y)$
- where $M \geq \max\{f, g\}$ for all $(x_1, x_2, \ldots, x_n)$

# Capital Budgeting

Stockco is considering four investments. Investment 1 will yield a net present value (NPV) of \$16,000; investment 2, an NPV of \$22,000; investment 3, an NPV of \$12,000; and investment 4, an NPV of \$8,000. Each investment requires a certain cash outflow at the present time: investment 1, \$5,000; investment 2, \$7,000; investment 3, \$4,000; and investment 4, \$3,000. Currently, \$14,000 is available for investment. Formulate an IP whose solution will tell Stockco how to maximize the NPV obtained from investments 1–4.

Modify the Stockco formulation to account for each of the following requirements:

1. Stockco can invest in at most two investments.
2. f Stockco invests in investment 2, they must also invest in investment 1.
3. If Stockco invests in investment 2, they cannot invest in investment 4.

# Facility location

There are six cities (cities 1–6) in Kilroy County. The county must determine where to build fire stations. The county wants to build the minimum number of fire stations needed to ensure that at least one fire station is within 15 minutes (driving time) of each city. The times (in minutes) required to drive between the cities in Kilroy County are shown in Table 6. Formulate an IP that will tell Kilroy how many fire stations should be built and where they should be located.

| | To | | | | | |
|---|---|---|---|---|---|---|
| From | City 1 | City 2 | City 3 | City 4 | City 5 | City 6 |
| City 1 | 0 | 10 | 20 | 30 | 30 | 20 |
| City 2 | 10 | 0 | 25 | 35 | 20 | 10 |
| City 3 | 20 | 25 | 0 | 15 | 30 | 20 |
| City 4 | 30 | 35 | 15 | 0 | 15 | 25 |
| City 5 | 30 | 20 | 30 | 15 | 0 | 14 |
| City 6 | 20 | 10 | 20 | 25 | 14 | 0 |