Exact Optimization Methods based on Integer Linear Programming

Simulação e Otimização

Mestrado em Engenharia Informática
Mestrado em Robótica e Sistemas Inteligentes

Amaro de Sousa, Nuno Lau

DETI-UA, 2021/2022

Mathematical programming

- In an ***optimization problem***, the aim is to maximize (or minimize) a given quantity designated as the ***objective*** that depends on a finite number of variables.
- The variables might be independent or might be related between them through one or more ***constraints***.

- A ***mathematical programming problem*** is an optimization problem such that the objective and the constraints are defined by mathematical functions and functional relations.
- A ***mathematical programming model*** describes a mathematical programming problem.

Mathematical programming model

For a given set of n variables $X=\{x_1, x_2, \ldots, x_n\}$, the standard way of defining a Mathematical Programming Model is:

Minimize (or Maximize)

$$f(X)$$

Subject to:

$$g_i(X) \le k_i \qquad , i = 1, 2, \ldots, m$$
$$(=)$$
$$(\ge)$$

---

where:

– m is the number of constraints

– $f(X)$ and all $g_i(X)$ are mathematical functions of the variables

– k_i are real parameters

(Mixed Integer) Linear Programming model

- A ***Linear Programming*** *(LP) model* is a mathematical programming model where all variables $X=\{x_1, x_2, ..., x_n\}$ are non-negative reals and $f(X)$ and $g_i(X)$ are linear functions:
  - functions in the form $a_1 x_1 + a_2 x_2 + ... + a_n x_n$ where all a_i are real parameters
- An ***Integer Linear Programming*** *(ILP) model* is an LP model where all variables $X=\{x_1, x_2, ..., x_n\}$ are non-negative integers.
- A ***Mixed Integer Linear Programming*** *(MILP) model* is an LP model where some of the variables $X=\{x_1, x_2, ..., x_n\}$ are non-negative integers and others are non-negative reals.

(Mixed Integer) Linear Programming model

Minimize (or Maximize)

$$c_1 x_1 + c_2 x_2 + \ldots + c_n x_n$$

The aim is to assign the values to all variables $x_1 \ldots x_n$ that optimize the objective function

Subject to:

$$a_{11} x_1 + a_{12} x_2 + \ldots + a_{1n} x_n \leq k_1$$

$$a_{21} x_1 + a_{22} x_2 + \ldots + a_{2n} x_n \leq k_2$$

$$\ldots$$

$$a_{m1} x_1 + a_{m2} x_2 + \ldots + a_{mn} x_n \leq k_\mathrm{m}$$

All constraints must be met by the values assigned to all variables $x_1 \ldots x_n$

- Constraints with '$\geq$' can be formulated with '$\leq$' as:

$$g_i(X) \geq k_i \quad \rightarrow \quad -g_i(X) \leq -k_i$$

- Constraints with '$=$' can be formulated with '$\leq$' as:

$$g_i(X) = k_i \quad \rightarrow \quad g_i(X) \leq k_i \quad \text{and} \quad -g_i(X) \leq -k_i$$

Illustrative example

- Consider a logistic operator that has been requested to deliver the following items from its head quarters to a particular destination:

Item i:	1	2	3	4	5	6
Revenue (r_i):	2.3	4.5	1.5	5.4	2.9	3.2
Size (s_i):	30	75	20	80	35	40

- The company has 2 available vans for this delivery:

  - van 1 has a capacity of 100
  - van 2 has a capacity of 60

- Since 30+75+20+80+35+40 (=280) > 100+60 (=160), it is not possible to deliver all items with the 2 vans.
- So, the problem is to choose the items to be carried on each van aiming to maximize the total revenue.
- Solving steps:

1st – define and implement an ILP model of the optimization problem

2nd – solve the ILP model (using an available solver)

Illustrative example

Item i:	1	2	3	4	5	6
Revenue (r_i):	2.3	4.5	1.5	5.4	2.9	3.2
Size (s_i):	30	75	20	80	35	40

VARIABLES DEFINING THE PROBLEM:

means that the value of x1 can be only 0 or 1

`x1` – Binary variable that, if is 1 in the solution, indicates that item 1 is delivered
`x2` – Binary variable that, if is 1 in the solution, indicates that item 2 is delivered
…
`x6` – Binary variable that, if is 1 in the solution, indicates that item 6 is delivered

`y1_1` – Binary variable that, if is 1 in the solution, indicates that item 1 is carried by van 1
`y1_2` – Binary variable that, if is 1 in the solution, indicates that item 1 is carried by van 2
…
`y6_1` – Binary variable that, if is 1 in the solution, indicates that item 6 is carried by van 1
`y6_2` – Binary variable that, if is 1 in the solution, indicates that item 6 is carried by van 2

Illustrative example

Item i:	1	2	3	4	5	6
Revenue (r_i):	2.3	4.5	1.5	5.4	2.9	3.2
Size (s_i):	30	75	20	80	35	40

INTEGER LINEAR PROGRAMMING (ILP) MODEL (LP format):

The objective function is the total revenue of the delivered items

```
max + 2.3 x1 + 4.5 x2 + 1.5 x3 + 5.4 x4 + 2.9 x5 + 3.2 x6
subject to
+ 30 y1_1 + 75 y2_1 + 20 y3_1 + 80 y4_1 + 35 y5_1 + 40 y6_1 <= 100
+ 30 y1_2 + 75 y2_2 + 20 y3_2 + 80 y4_2 + 35 y5_2 + 40 y6_2 <= 60
+ y1_1 + y1_2 - x1 = 0
+ y2_1 + y2_2 - x2 = 0
+ y3_1 + y3_2 - x3 = 0
+ y4_1 + y4_2 - x4 = 0
+ y5_1 + y5_2 - x5 = 0
+ y6_1 + y6_2 - x6 = 0
binary
x1 x2 x3 x4 x5 x6
y1_1 y1_2 y2_1 y2_2 y3_1 y3_2 y4_1 y4_2 y5_1 y5_2 y6_1 y6_2
end
```

The total size of the items carried on each van must be within the van capacity

If an item is carried in one van, then, the item is delivered
+
An item cannot be carried by both vans

List of binary variables

Illustrative example: mathematical notation

Parameters:

n – number of items — r_i – revenue of delivering item i, with $i = 1,\ldots,n$

s_i – size of item i, with $i = 1,\ldots,n$

v – number of vans — c_j – capacity of van j, with $j = 1,\ldots,v$

Variables:

x_i – binary variable that is 1 if item i is delivered, $i = 1,\ldots,n$

y_{ij} – binary variable that is 1 if item i is carried on van j, $i = 1,\ldots,n$ and $j = 1,\ldots,v$

ILP model: Maximize $\sum_{i=1}^{n} r_i x_i$

Subject to:

$\sum_{i=1}^{n} s_i y_{ij} \leq c_j \quad , j = 1 \ldots v$

$\sum_{j=1}^{v} y_{ij} = x_i \quad , i = 1 \ldots n$

$x_i \in \{0,1\} \quad , i = 1 \ldots n$

$y_{ij} \in \{0,1\} \quad , i = 1 \ldots n \ , j = 1, \ldots v$

Illustrative example: LP file with a MATLAB script

```
r= [2.3 4.5 1.5 5.4 2.9 3.2];
s= [30 75 20 80 35 40];
c= [100 60];
n= length(r);
v= length(c);
fid = fopen('example.lpt','wt');
```

Maximize $\sum_{i=1}^{n} r_i x_i$

```
fprintf(fid,'max ');
for i=1:n
    fprintf(fid,'+ %f x%d ',r(i),i);
end
fprintf(fid,'\nsubject to\n');
```

$\sum_{i=1}^{n} s_i y_{ij} \leq c_j \quad ,j = 1 \dots v$

```
for j=1:v
    for i=1:n
        fprintf(fid,'+ %f y%d_%d ',s(i),i,j);
    end
    fprintf(fid,'<= %f\n',c(j));
end
```

$\sum_{j=1}^{v} y_{ij} = x_i \quad ,i = 1 \dots n$

```
for i=1:n
    for j=1:v
        fprintf(fid,'+ y%d_%d ',i,j);
    end
    fprintf(fid,'- x%d = 0\n',i);
end
fprintf(fid,'binary\n');
```

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

```
for i=1:n
    fprintf(fid,'x%d ',i);
end
```

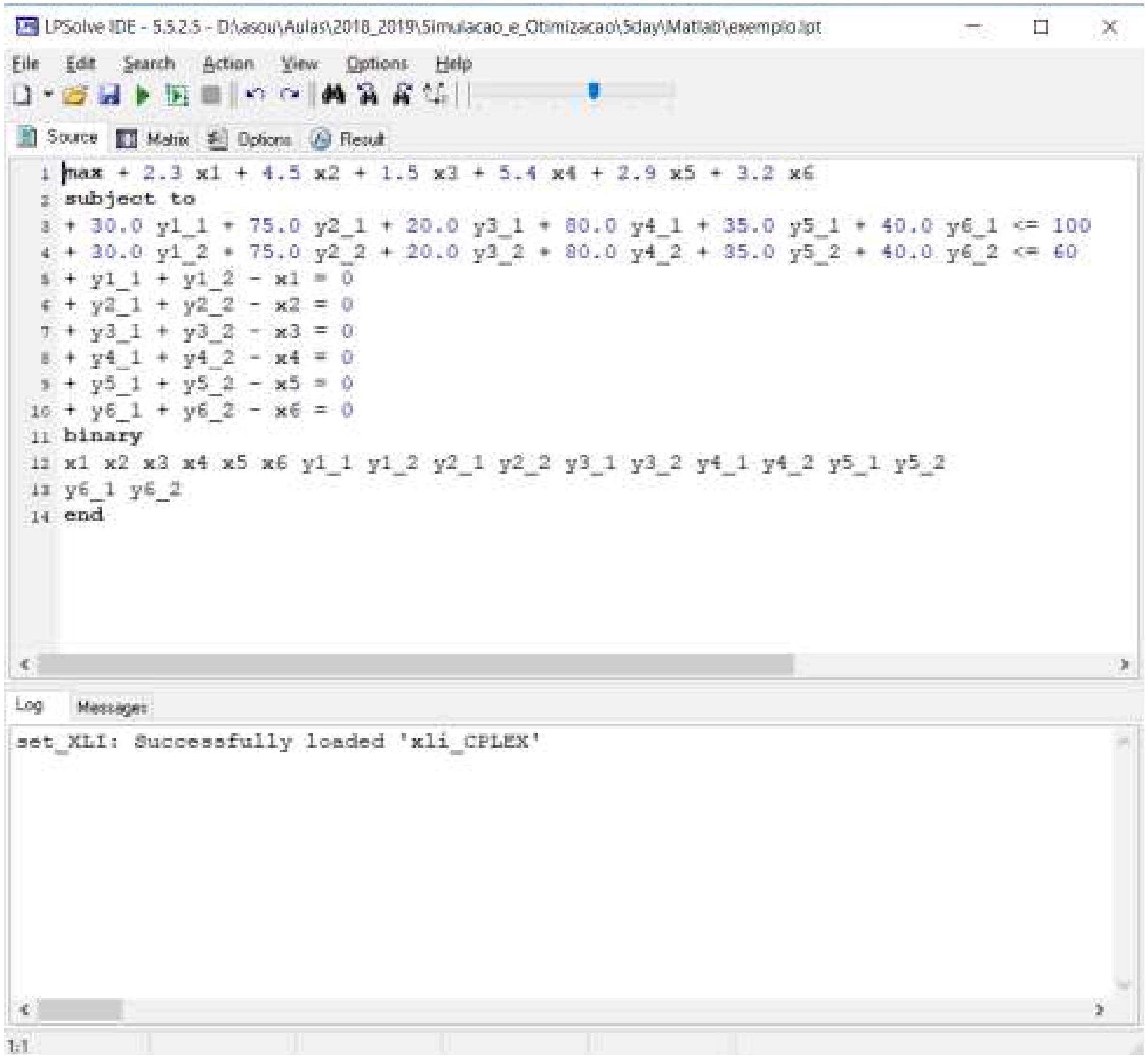
$y_{ij} \in \{0,1\}\, ,i = 1 \dots n \;\; ,j = 1, \dots v$

```
for i=1:n
    for j=1:v
        fprintf(fid,'y%d_%d ',i,j);
    end
end
fprintf(fid,'\nend');
fclose(fid);
```

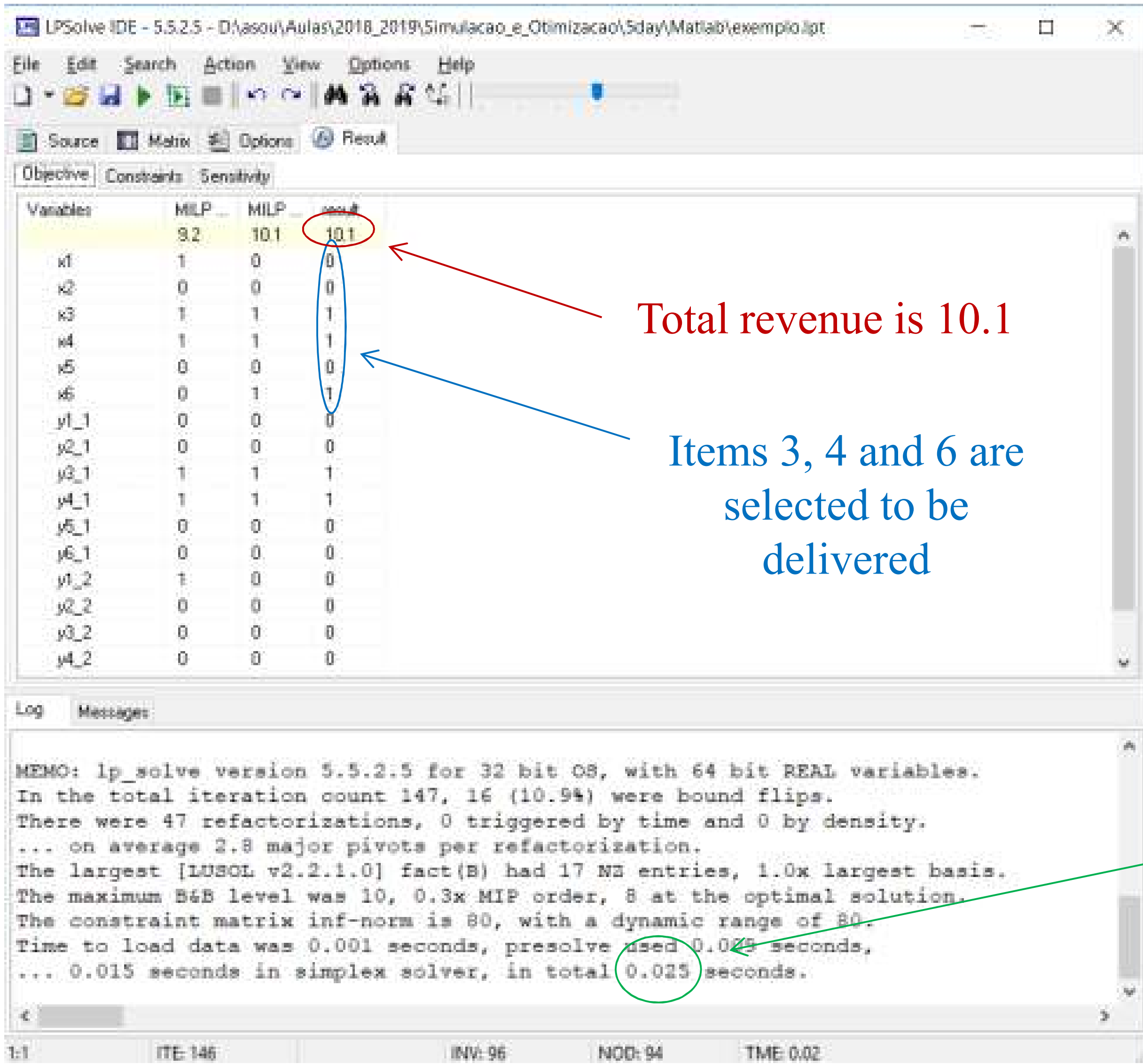
ILP solvers

- There are various commercial software packages providing algorithms to solve LP and ILP problems such as: CPLEX, Gurobi, XPRESS, etc...
- The ‘lpsolve’ IDE is a free software.
- Download ‘lpsolve’: http://sourceforge.net/projects/lpsolve/
- Help: http://lpsolve.sourceforge.net/5.5/

Illustrative example: using ‘lpsolve’ (1)



Illustrative example: using ‘lpsolve’ (2)

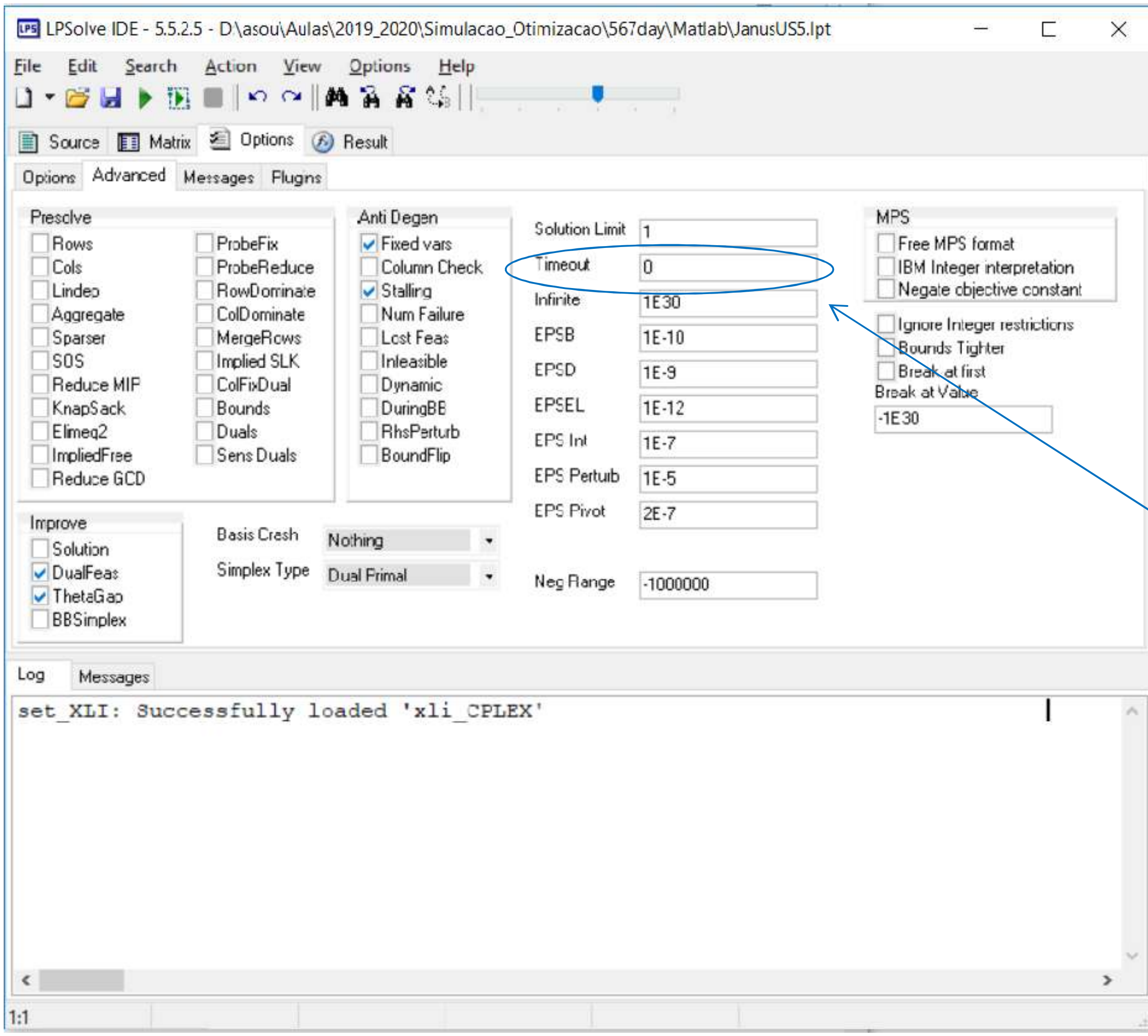


Total revenue is 10.1

Items 3, 4 and 6 are selected to be delivered

Problem solved in 0.025 seconds

Illustrative example: using ‘lpsolve’ (3)



By default, ‘lpsolve’ runs until it finds an optimal solution

In hard problems, the running time can be too long to reach an optimal solution

You can set a Timeout (in seconds)

If Timeout is reached, ‘lpsolve’ provides the best solution found at that time.

Eight assignment – step 1

- Download and install ‘lpsolve’.
- Use the provided MATLAB script to generate a file named *example.lpt* with the ILP description of illustrative example.
- Use ‘lpsolve’ to solve the illustrative example.
- Register the optimal solution and the time taken by ‘lpsolve’ to obtain it in your computer.

Eight assignment – step 2

- Change the MATLAB script to generate a file of the problem in LP format for the items and vans described below.
- Use ‘lpsolve’ to solve this new problem.
- Register the optimal solution and the time taken by ‘lpsolve’ to obtain it in your computer. Compare this runtime with the runtime while solving the previous optimization problem.

Items:

i	1	2	3	4	5	6	7	8	9	10	11	12	13
r_i	2.3	4.5	1.5	5.4	2.9	3.2	5.9	2.2	5.4	1.4	2.3	2.1	2.7
s_i	30	75	20	80	35	40	85	15	70	20	25	15	40

Vans:

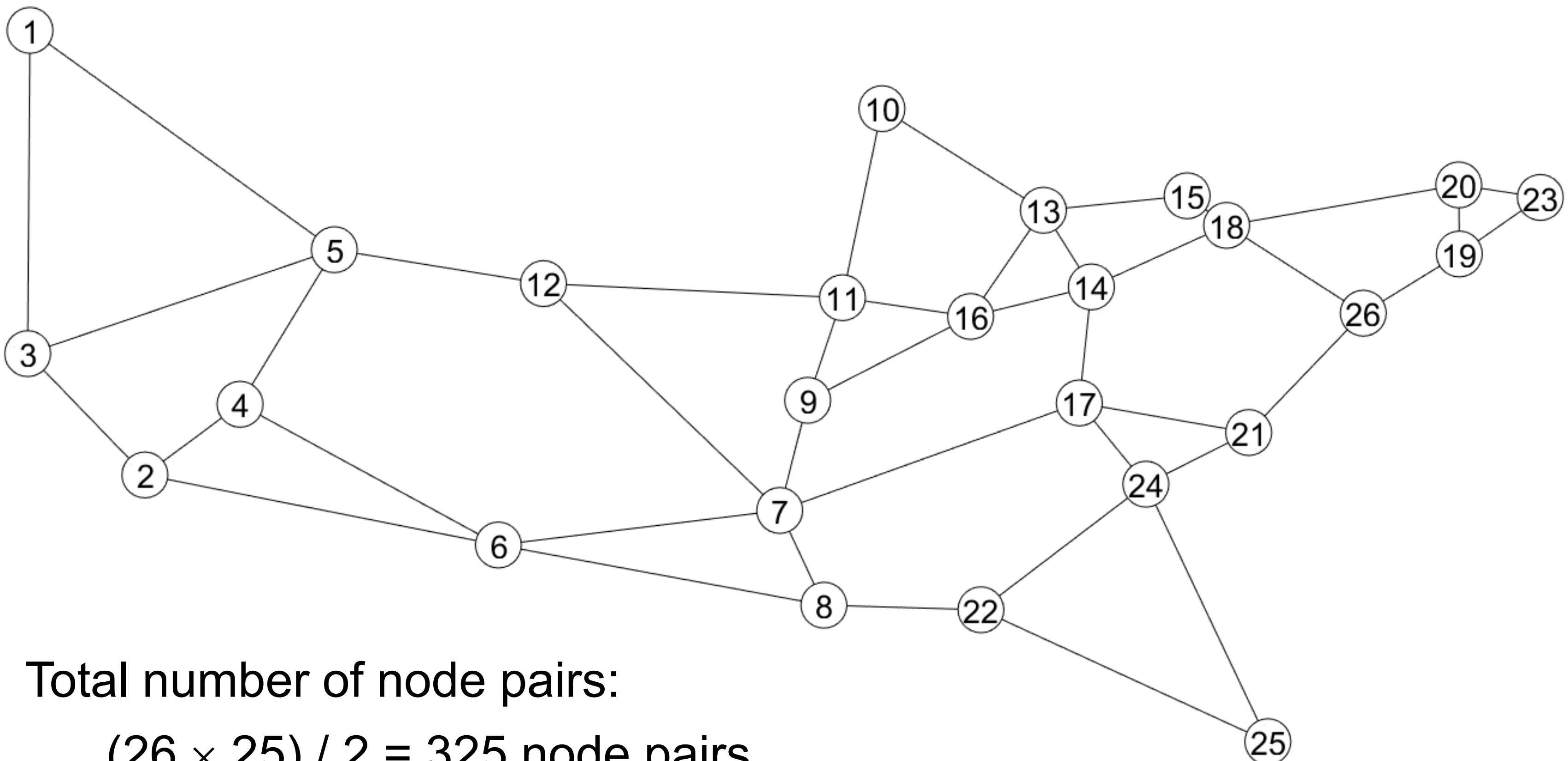
j	1	2	3	4
c_j	100	60	60	60

Detection of Critical Elements of a Network

- Consider a network modelled by a graph $G = (N, E)$ and a given positive integer c.
- The aim is to select c network elements that maximally degrade the network connectivity when all selected elements are eliminated.
- Network connectivity degradation can be defined in different ways (depending on the problem context):
  - Minimization of the number of connected node pairs (i.e., number of node pairs that can communicate)
  - Maximization of the number of connected components of the network
  - Minimization of the maximum number of nodes among all connected components of the network
- Network elements can be links (Critical Link Detection problem) or nodes (Critical Node Detection problem)

Detection of Critical Elements of a Network

Consider the JanusUS network with 26 nodes and 42 links.

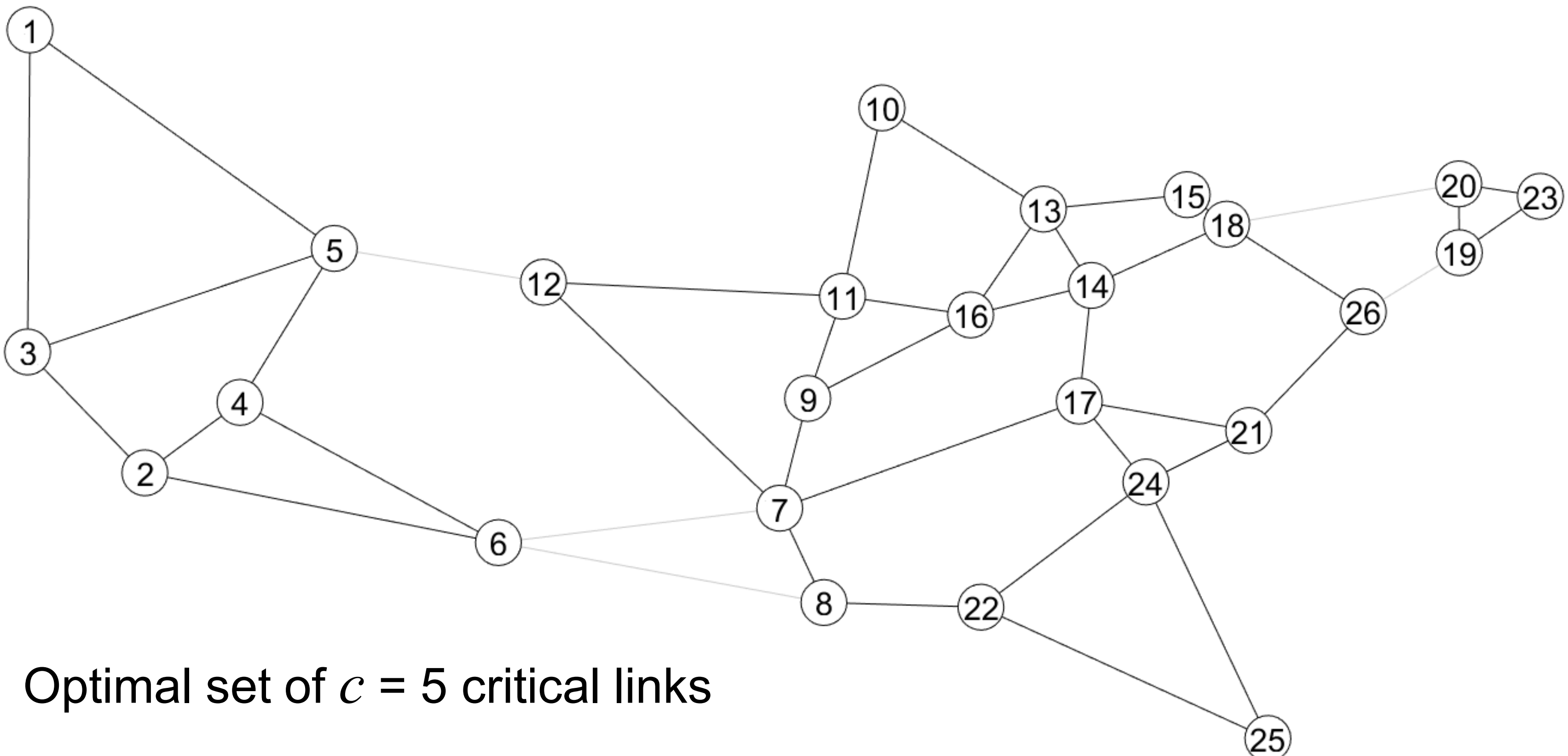


Total number of node pairs:

$(26 \times 25) / 2 = 325$ node pairs

Detection of Critical Links of a Network

Connectivity degradation: minimizing the number of connected node pairs



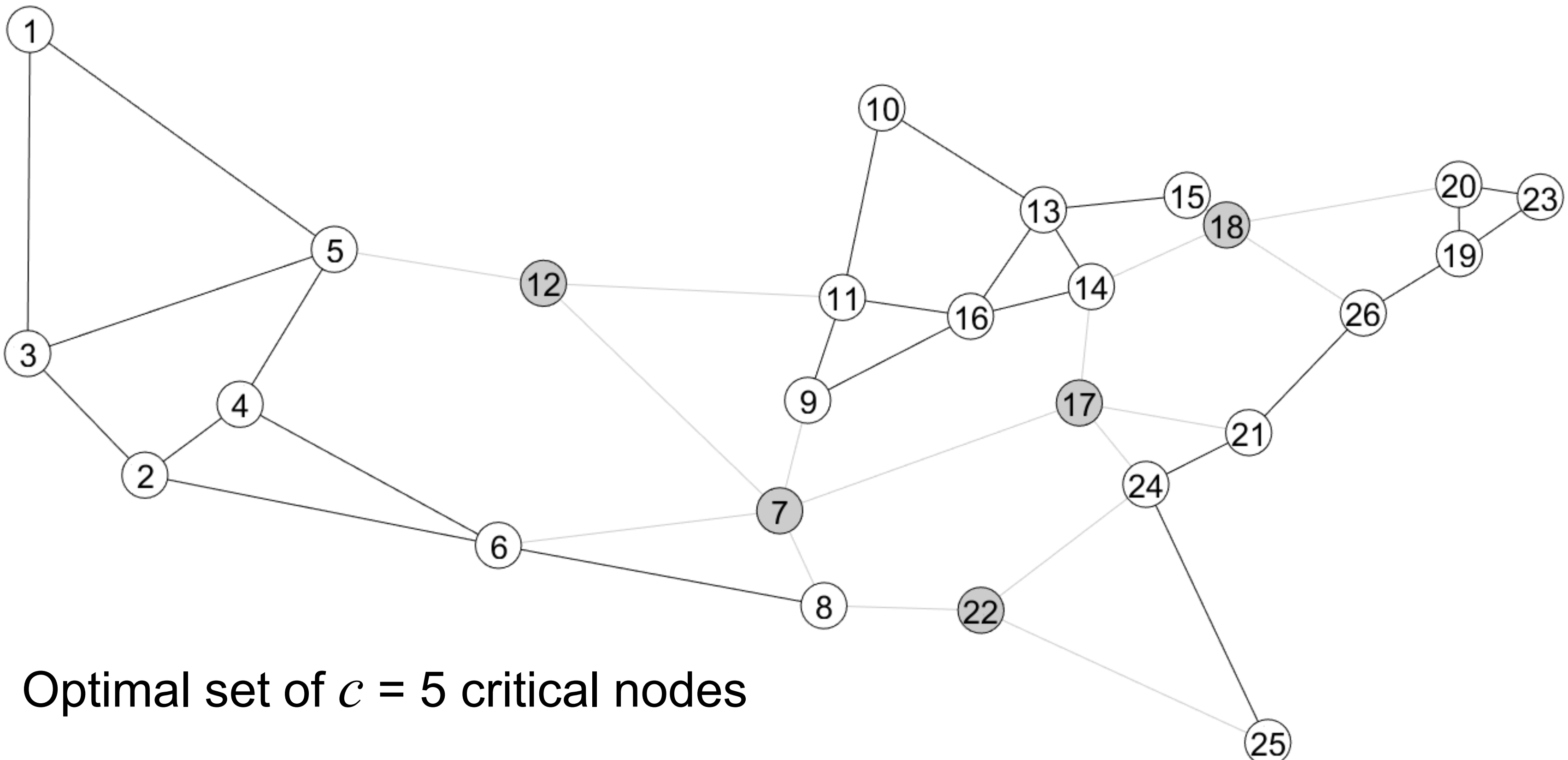
Optimal set of c = 5 critical links

No. of connected node pairs:

$(6 \times 5) / 2 + (17 \times 16) / 2 + (3 \times 2) / 2 = 154$ node pairs

Detection of Critical <u>Nodes</u> of a Network

Connectivity degradation: minimizing the number of connected node pairs



Optimal set of c = 5 critical nodes

No. of connected node pairs:

$(7 \times 6) / 2 + (7 \times 6) / 2 + (7 \times 6) / 2 = 63$ node pairs

PROBLEM 1: Critical Link Detection (CLD)

- Consider:
  - a network modelled by a graph $G = (N, E)$ with $n = |N|$ nodes;
  - an existing link between nodes $i \in N$ and $j \in N$ is represented by $(i, j) \in E$ with $i < j$;
  - set $V(i)$ is the set of neighboring nodes of node i in graph G.

- For a given positive integer c, the aim is to select c network links, named <u>critical links</u>, that minimize the number of connected node pairs when all critical links are eliminated.

- Variables:
  v_{ij} – binary variable that is equal to 1 if link $(i, j) \in E$ is selected as a critical link
  u_{ij} – binary variable that is equal to 1 if nodes i and j , with $i < j$, can communicate when all critical links are eliminated

- Variable notation:
  $u_{\{ij\}}$ – represents variable u_{ij} if $i < j$ or variable u_{ji} if $j < i$

PROBLEM 1:
Critical Link Detection (CLD)

- MILP model:

Minimize $\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} u_{ij}$

No. of connected node pairs (i.e., node pairs that can communicate)

Subject to:

$\sum_{(i,j)\in E} v_{ij} = c$

No. of critical links must be *c*

$u_{ij} + v_{ij} \geq 1 \qquad , (i,j) \in E$

If (*i*,*j*) is not a critical link, nodes *i* and *j* are connected

$u_{ij} \geq u_{\{ik\}} + u_{\{kj\}} - 1 \qquad , i = 1 \dots (n-1),\ j = (i+1) \dots n, k \in V(i)\backslash\{j\}$

If *i* is connected with its neighbour *k* and *k* is connected with *j*, node *i* is connected with node *j*

$v_{ij} \in \{0,1\} \qquad , (i,j) \in E$

$u_{ij} \in \mathbb{R}_0^+ \qquad , i = 1 \dots (n-1), j = (i+1) \dots n$

MILP model

Variables u_{ij} do not need to be binary

PROBLEM 2:
Critical Node Detection (CND)

- Consider:
  - a network modelled by a graph $G = (N, E)$ with $n = |N|$ nodes;
  - an existing link between nodes $i \in N$ and $j \in N$ is represented by $(i, j) \in E$ with $i < j$;
  - set $V(i)$ is the set of neighboring nodes of node i in graph G.

- For a given positive integer c, the aim is to select c network nodes, named <u>critical nodes</u>, that minimize the number of connected node pairs when all critical nodes are eliminated.

- Variables:
  v_i – binary variable that is equal to 1 if node $i \in N$ is selected as a critical node
  u_{ij} – binary variable that is equal to 1 if nodes i and j , with $i < j$, are connected when all critical nodes are eliminated

- Variable notation:
  $u_{\{ij\}}$ – represents variable u_{ij} if $i < j$ or variable u_{ji} if $j < i$

PROBLEM 2:
Critical Node Detection (CND)

- MILP model:

Minimize $\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} u_{ij}$

No. of connected node pairs

Subject to:

No. of critical nodes must be *c*

$$\sum_{i=1}^{n} v_i = c$$

If *i* is not a critical node and *j* is also not a critical node, nodes *i* and *j* are connected

$$u_{ij} + v_i + v_j \geq 1 \qquad ,(i,j) \in E$$

If *i* is connected with its neighbour *k* and *k* is connected with *j*, node *i* is connected with node *j*

$$u_{ij} \geq u_{\{ik\}} + u_{\{kj\}} - 1 + v_k \qquad ,(i,j) \notin E, k \in V(i)$$

not necessary to define the model but improves the resolution runtime

$$v_i \in \{0,1\} \qquad ,i = 1 \ldots n$$

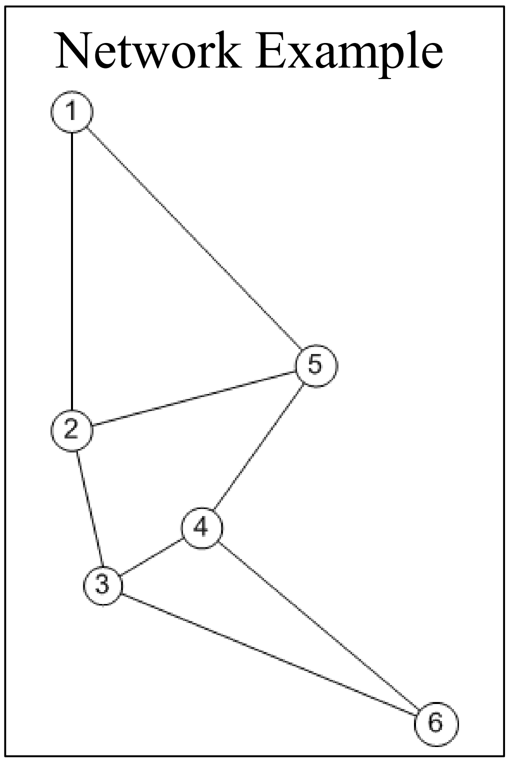
$$u_{ij} \in \mathbb{R}_0^+ \qquad ,i = 1 \ldots (n-1), j = (i+1) \ldots n$$

MILP model

Variables u_{ij} do not need to be binary

A Network Modelled by a Graph

- Consider a network modelled by an <u>undirected</u> graph $G = (N, E)$
  - N is the set of $|N|$ network nodes
  - E is the set of $|E|$ network edges (links)



$$N = \{1,2,3,4,5,6\}$$
$$|N| = 6$$

$$E = \{(1,2), (1,5), (2,3), (2,5) \\ (3,4), (3,6), (4,5), (4,6)\}$$
$$|E| = 8$$

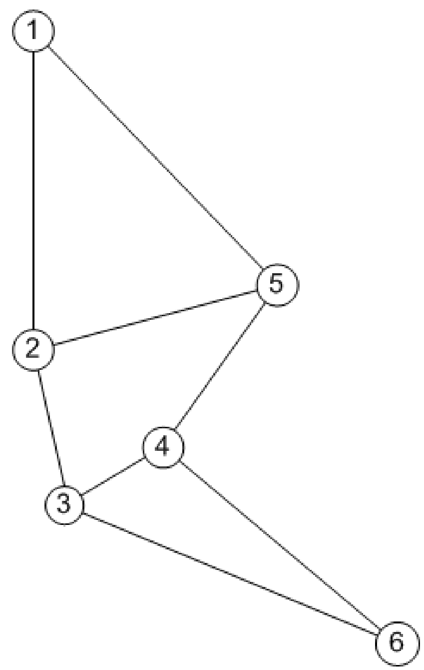
a link in E between nodes $i \in N$ and $j \in N$ is represented by (i,j), with $i < j$

A Network Modelled by a Graph

- In MATLAB, two useful ways of encoding the graph are:
  - by a list of edges (`E_list`)
  - by a matrix (`E_matrix`)

```
E_list= [1 2
         1 5
         2 3
         2 5
         3 4
         3 6
         4 5
         4 6];
```

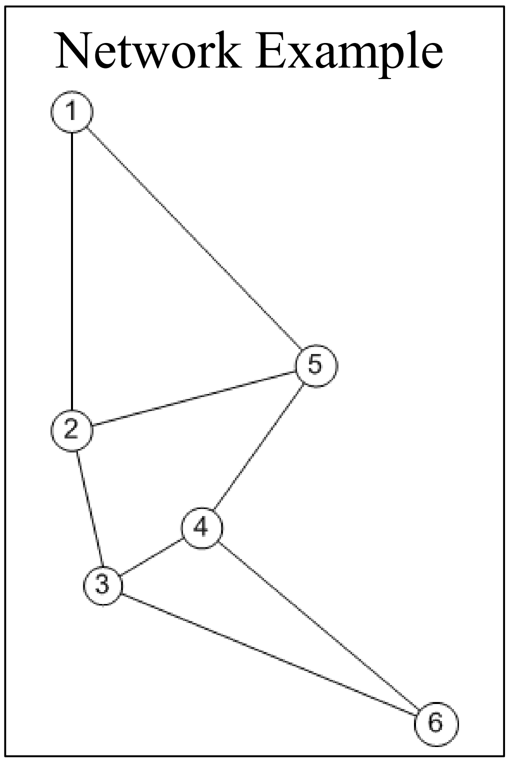
Network Example




```
E_matrix= [0 1 0 0 1 0
           1 0 1 0 1 0
           0 1 0 1 0 1
           0 0 1 0 1 1
           1 1 0 1 0 0
           0 0 1 1 0 0];
```

A Network Modelled by a Graph

- Consider $V(i)$ as the set of neighbouring nodes of node i in graph G.



In the Network Example:

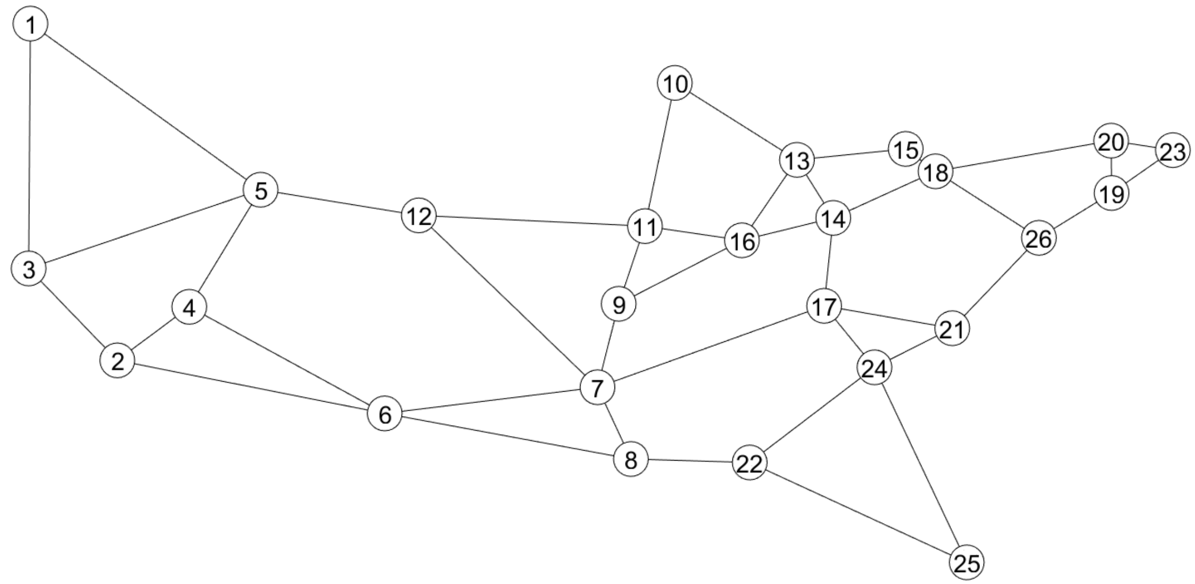
$$V(1)=\{2,5\} \qquad V(4)=\{3,5,6\}$$
$$V(2)=\{1,3,5\} \qquad V(5)=\{1,2,4\}$$
$$V(3)=\{2,4,6\} \qquad V(6)=\{3,4\}$$

In MATLAB:

```
E_matrix= [0  1  0  0  1  0
            1  0  1  0  1  0
            0  1  0  1  0  1
            0  0  1  0  1  1
            1  1  0  1  0  0
            0  0  1  1  0  0];

for k=find(E_matrix(i,:)>0)
      ...
end
```

JanusUS network



Input files of JanusUS network:

`Links_JanusUS.txt` – a matrix of 42 rows and 2 columns with the node pairs of each link, with $i < j$

`L_JanusUS.txt` – a square matrix of 26x26 with the link length l_{ij} value for existing links (i, j) or 0 otherwise

Loading input files in MATLAB:

```
Links= load('Links_JanusUS.txt');
L= load('L_JanusUS.txt');
nNodes= size(L,1);
nLinks= size(Links,1);
```

MATLAB supporting codes

```
Links= load('Links_JanusUS.txt');
L= load('L_JanusUS.txt');
nNodes= size(L,1);
nLinks= size(Links,1);
```

enumerating all links $(i,j) \in E$

```
for i= 1:nNodes-1
    for j= i+1:nNodes
        if L(i,j)>0
            ...
        end
    end
end
```

alternative way of enumerating all links $(i,j) \in E$

```
for k= 1:nLinks
    i= Links(k,1);
    j= Links(k,2);
    ...
    end
end
```

MATLAB supporting codes

```
Links= load('Links_JanusUS.txt');
L= load('L_JanusUS.txt');
nNodes= size(L,1);
nLinks= size(Links,1);
```

enumerating all node pairs
$i = 1 \dots (n-1)$,
$j = (i+1) \dots n$

```
for i= 1:nNodes-1
    for j= i+1:nNodes
        ...
    end
end
```

enumerating all nodes
i, j and k such that
$(i,j) \notin E, k \in V(i)$

```
for i= 1:nNodes-1
    for j= i+1:nNodes
        if L(i,j)==0
            for k=find(L(i,:)>0)
                ...
            end
        end
    end
end
```

Ninth assignment

- Consider the graph $G = (N, E)$ of JanusUS such that each link $(i, j) \in E$ has an associated length l_{ij} (input files: `Links_JanusUS.txt` and `L_JanusUS.txt`).
- Consider the CND problem of selecting a set of c critical nodes that minimize number of connected node pairs.
- Develop a MATLAB script to generate a MILP description of the optimization problem in LP format.
- Solve the problem with 'lpsolve' for c = 5 critical nodes (check that the obtained optimal solution is the solution in slide 20).

Graphical interpretation of a LP model

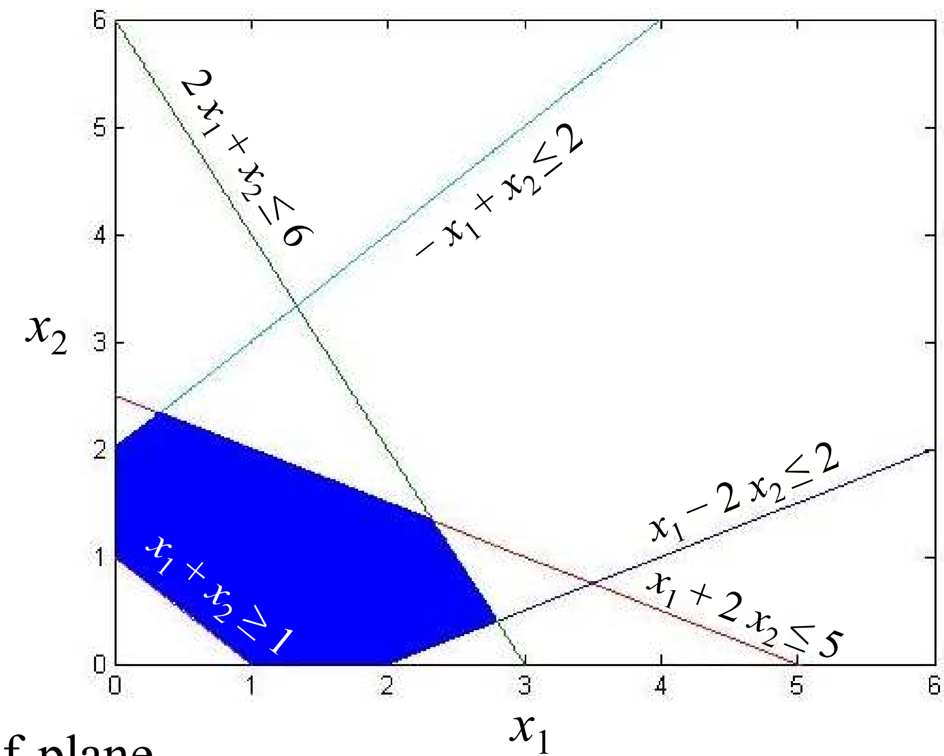
- Consider the following LP (Linear Programming) model with 2 non-negative real variables x_1 and x_2 :

Maximize

$$f(x_1,x_2) = 4\,x_1 + 5\,x_2$$

Subject to:

$$\begin{aligned} &x_1 - 2\,x_2 \le 2 \\ &2\,x_1 + x_2 \le 6 \\ &x_1 + 2\,x_2 \le 5 \\ &-x_1 + x_2 \le 2 \\ &x_1 + x_2 \ge 1 \\ &x_1, x_2 \ge 0 \end{aligned}$$



- Each constraint defines an half-plane
- The feasible solutions are all points that belong to all half-planes (blue region)

Graphical interpretation of a LP model

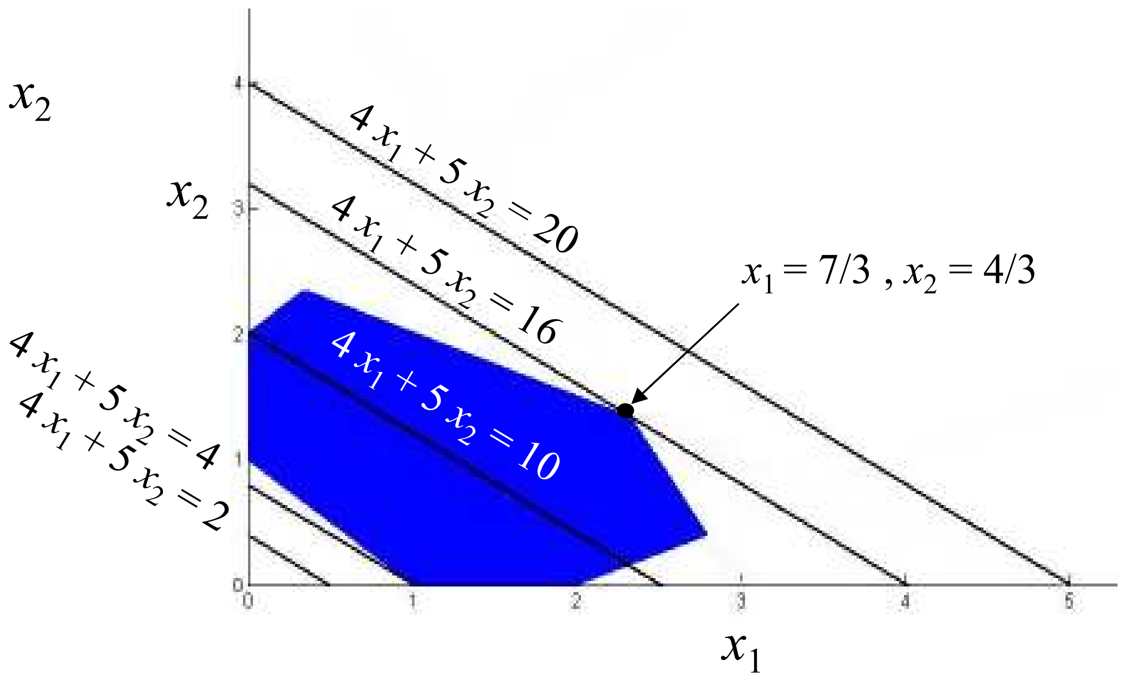
- Consider the following LP (Linear Programming) model with 2 non-negative real variables x_1 and x_2 :

Maximize

$$f(x_1,x_2) = 4\,x_1 + 5\,x_2$$

Subject to:

$$x_1 - 2\,x_2 \leq 2$$
$$2\,x_1 + x_2 \leq 6$$
$$x_1 + 2\,x_2 \leq 5$$
$$-\,x_1 + x_2 \leq 2$$
$$x_1 + x_2 \geq 1$$
$$x_1, x_2 \geq 0$$



- The aim is to find the values of x_1 and x_2 in the blue region that maximize $t = 4\,x_1 + 5\,x_2$
- The optimal solution is $x_1 = 7/3$ and $x_2 = 4/3$ whose value is $t = 16$

Graphical interpretation of a LP model

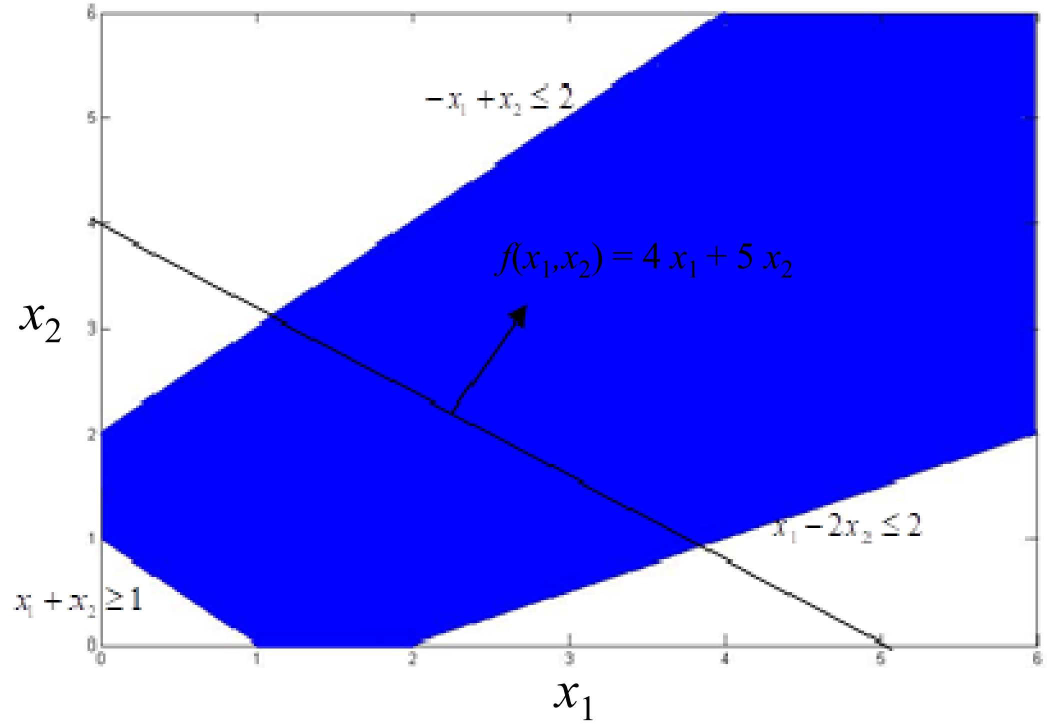
- Consider the previous LP model without 2 constraints:

Maximize

$$f(x_1,x_2) = 4\,x_1 + 5\,x_2$$

Subject to:

$$x_1 - 2\,x_2 \leq 2$$
$$\cancel{2\,x_1 + x_2 \leq 6}$$
$$\cancel{x_1 + 2\,x_2 \leq 5}$$
$$-\,x_1 + x_2 \leq 2$$
$$x_1 + x_2 \geq 1$$
$$x_1,\,x_2 \geq 0$$



- In the previous case, the problem had a close feasible region
- In this case, the problem has an open feasible region
- Due to the optimization direction, in this case there is no optimal solution: <u>the problem is unbounded</u>

Graphical interpretation of a LP model

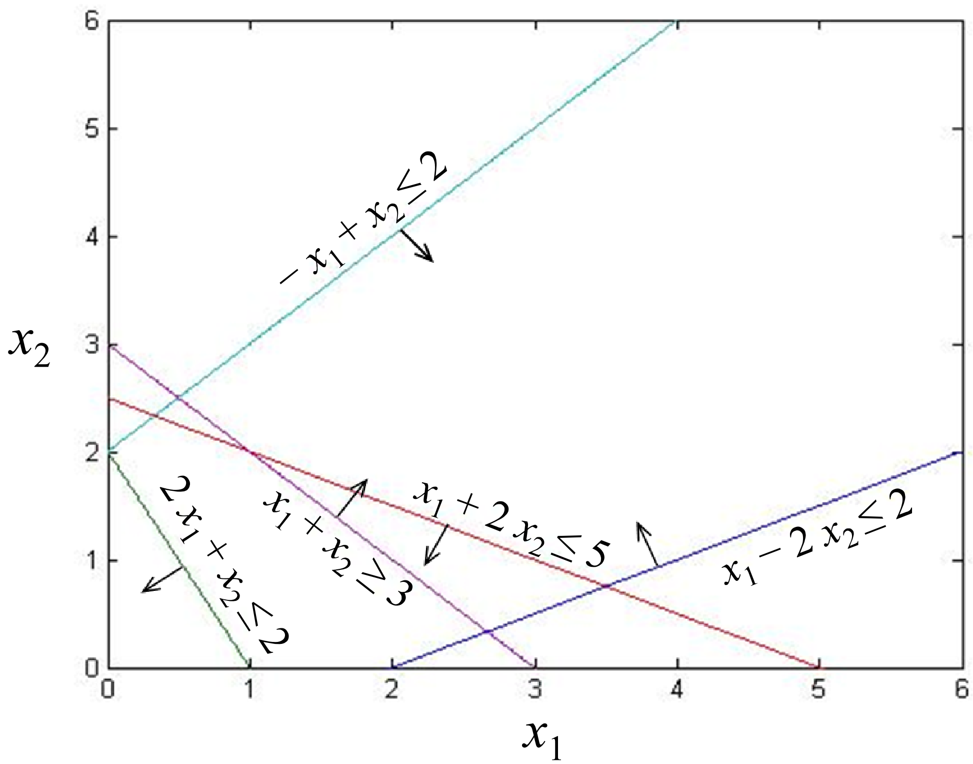
- Consider the first LP model with 2 different independent terms:

Maximize

$$f(x_1,x_2) = 4\,x_1 + 5\,x_2$$

Subject to:

$$x_1 - 2\,x_2 \leq 2$$
$$2\,x_1 + x_2 \leq \cancel{6}\ 2$$
$$x_1 + 2\,x_2 \leq 5$$
$$-\,x_1 + x_2 \leq 2$$
$$x_1 + x_2 \geq \cancel{1}\ 3$$
$$x_1, x_2 \geq 0$$



- In this case, the intersection of all half-planes is an empty region
- There is no pair of values x_1 and x_2 that can fulfil all constraints: <u>the problem is infeasible</u>

Solving a LP problem

- Start by considering new non-negative variables so that all constraints become equalities:

Maximize

$$4\,x_1 + 5\,x_2$$

Subject to:

$$
\begin{array}{ll}
x_1 - 2\,x_2 \le 2 & x_1 - 2\,x_2 + s_1 = 2 \\
2\,x_1 + x_2 \le 6 & 2\,x_1 + x_2 + s_2 = 6 \\
x_1 + 2\,x_2 \le 5 & x_1 + 2\,x_2 + s_3 = 5 \\
-\,x_1 + x_2 \le 2 & -\,x_1 + x_2 + s_4 = 2 \\
x_1 + x_2 \ge 1 & x_1 + x_2 - s_5 = 1 \\
x_1, x_2 \ge 0 & x_1, x_2, s_1, s_2, s_3, s_4, s_5 \ge 0
\end{array}
$$

- In this case, we get 5 equalities (equations) with 7 variables, which has an infinite number of solutions.
- If we assign values to 2 variables, we get a set of 5 equations with 5 variables, which has one single solution.

Solving a LP problem

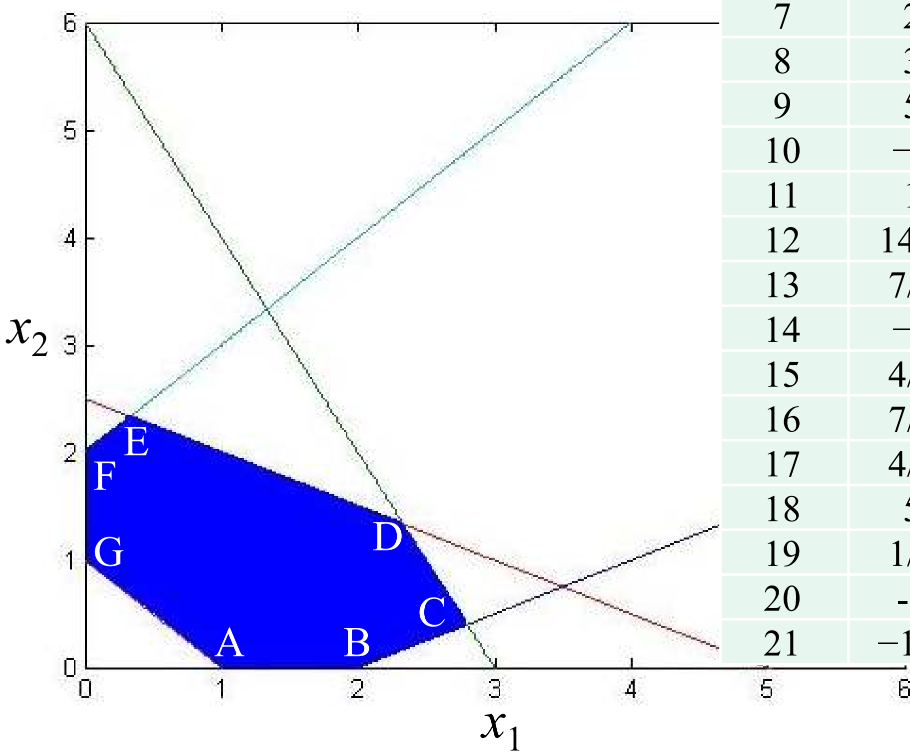
- In general, we get m equations with n variables, with $m < n$.
- Assume that all equations are linearly independent (otherwise, some original constraints are redundant).
- If we assign values to $n - m$ variables, we get a set of m equations with m variables, which has one single solution.

- A ***basic solution*** of an LP problem is the solution of the equations when $n - m$ variables are set to zero.
- A ***basic feasible solution*** of an LP problem is a basic solution such that are variables are non-negative.

- A basic feasible solution is a vertex of the feasible region of the LP problem.
- In a LP problem, one of its optimal solutions is a vertex.
- The aim is to compute the basic feasible solution with the best objective function value.

Solving a LP problem

$$x_1 - 2x_2 + s_1 = 2$$
$$2x_1 + x_2 + s_2 = 6$$
$$x_1 + 2x_2 + s_3 = 5$$
$$-x_1 + x_2 + s_4 = 2$$
$$x_1 + x_2 - s_5 = 1$$
$$x_1, x_2, s_1, s_2, s_3, s_4, s_5 \geq 0$$



All ***basic solutions***:

No.	x_1	x_2	s_1	s_2	s_3	s_4	s_5	vertex
1	0	0	2	6	5	2	−1	
2	0	−1	0	7	7	3	−2	
3	0	6	14	0	−7	−4	5	
4	0	5/2	7	7/2	0	−1/2	3/2	
5	0	2	6	4	1	0	1	F
6	0	1	4	5	3	1	0	G
7	2	0	0	2	3	4	1	B
8	3	0	−1	0	2	5	2	
9	5	0	−3	−4	0	7	4	
10	−2	0	4	10	7	0	−3	
11	1	0	1	4	4	3	0	A
12	14/5	2/5	0	0	7/5	22/5	11/5	C
13	7/2	¾	0	−7/4	0	19/4	−13/4	
14	−6	−4	0	22	19	0	11	
15	4/3	−1/3	0	11/3	13/3	11/6	0	
16	7/3	4/3	7/3	0	0	3	8/3	D
17	4/3	10/3	22/3	0	−3	0	−11/3	
18	5	−4	−11	0	8	11	0	
19	1/3	7/3	19/3	3	0	0	5/3	E
20	-3	4	13	8	0	−5	0	
21	−1/2	3/2	11/2	11/2	5/2	0	0	

Solving a LP problem

- ***Simplex method*** is an efficient algorithm to compute an optimal solution of a LP problem.
- It works as follows:
    1. Compute an initial vertex (i.e., a basic feasible solution)
    2. Find the best neighbor vertex (i.e., the neighbor vertex with the best objective value)
    3. If the neighbor vertex is better than the current vertex, the neighbor vertex becomes the current vertex and returns to step 2; otherwise, the current vertex is one optimal solution of the LP problem
- Simplex method is a hill climbing method.
- It is an exact method since in LP problems a local optimal solution is also a global optimal solution.

Graphical interpretation of a ILP model

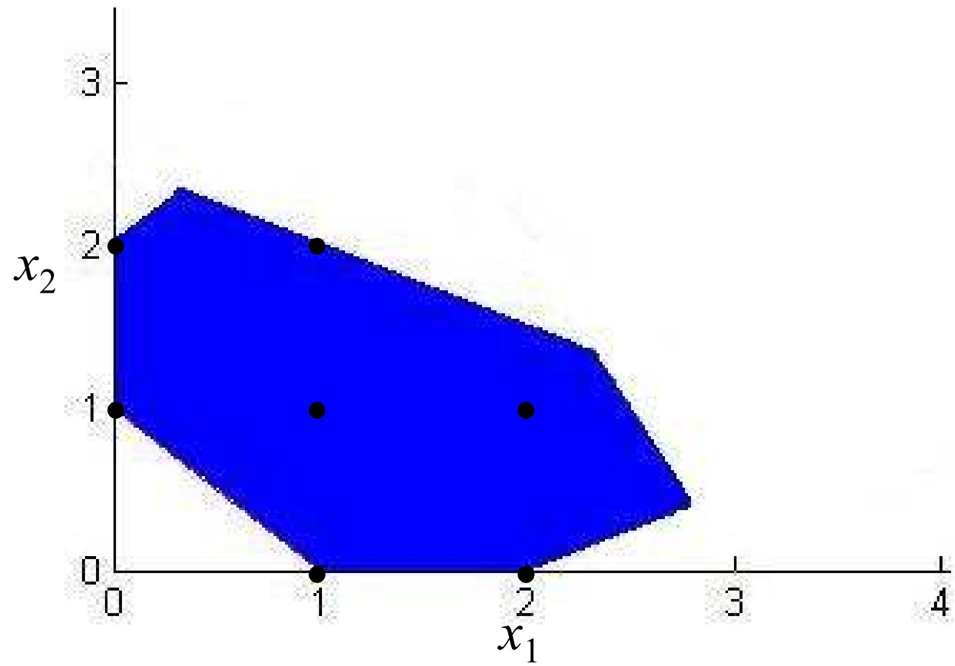
- Consider the following ILP (Integer Linear Programming) model with 2 non-negative integer variables x_1 and x_2 :

Maximize

$$f(x_1,x_2) = 4\,x_1 + 5\,x_2$$

Subject to:

$$x_1 - 2\,x_2 \le 2$$
$$2\,x_1 + x_2 \le 6$$
$$x_1 + 2\,x_2 \le 5$$
$$-\,x_1 + x_2 \le 2$$
$$x_1 + x_2 \ge 1$$
$$x_1, x_2 \in \{0,1,2,\ldots\}$$



- The *linear programming* (LP) *relaxation* of an ILP model is the LP model where the integer variables are relaxed by being considered real
- The feasible solutions of an ILP model are points (given by the integer values of the variables) inside the feasible region of its LP relaxation

Solving a ILP problem

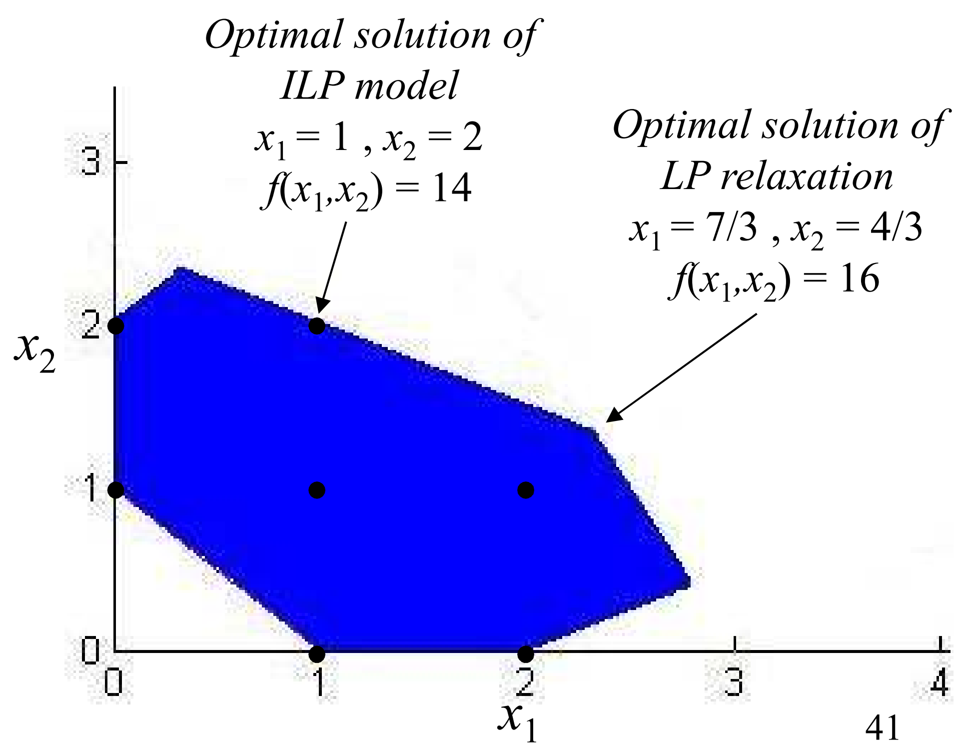
- Solving a ILP problem is much harder than solving a LP problem.
- The solution value of the LP relaxation is a mathematical bound of the solution value of a ILP model:
  - A lower bound if the objective is to minimize
  - An upper bound if the objective is to maximize, as in the example below

Maximize

$$f(x_1,x_2) = 4\,x_1 + 5\,x_2$$

Subject to:

$$x_1 - 2\,x_2 \leq 2$$
$$2\,x_1 + x_2 \leq 6$$
$$x_1 + 2\,x_2 \leq 5$$
$$-\,x_1 + x_2 \leq 2$$
$$x_1 + x_2 \geq 1$$
$$x_1, x_2 \in \{0,1,2,\ldots\}$$



Graphical interpretation of a ILP model

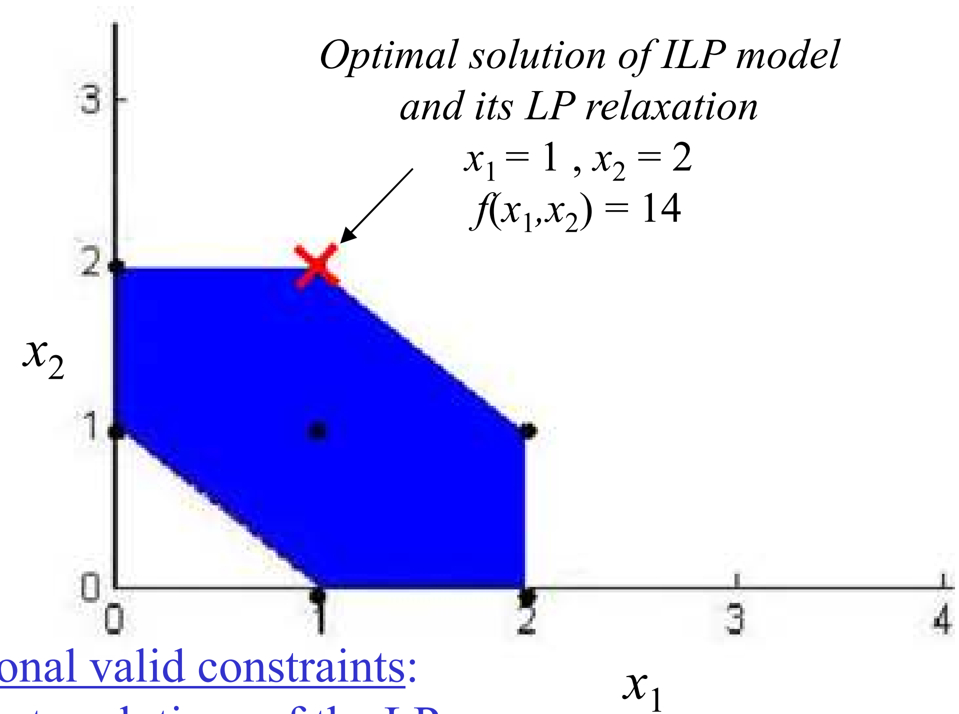
- Consider the previous ILP model with 3 additional constraints:

Maximize

$$f(x_1,x_2) = 4\,x_1 + 5\,x_2$$

Subject to:

$$x_1 - 2\,x_2 \leq 2$$
$$2\,x_1 + x_2 \leq 6$$
$$x_1 + 2\,x_2 \leq 5$$
$$-\,x_1 + x_2 \leq 2$$
$$x_1 + x_2 \geq 1$$
$$x_1 \leq 2$$
$$x_2 \leq 2$$
$$x_1 + x_2 \leq 3$$
$$x_1, x_2 \in \{0,1,2,\ldots\}$$



additional valid constraints: eliminate solutions of the LP relaxation but keep all integer solutions

- When the vertices are integer solutions, the optimal solution of the LP relaxation is the optimal integer solution
- Since in this case it is a close feasible region, this is true for any objective

Solving a ILP problem

Branch-and-bound (B&B) method is an iterative procedure to compute an optimal solution of a ILP problem.

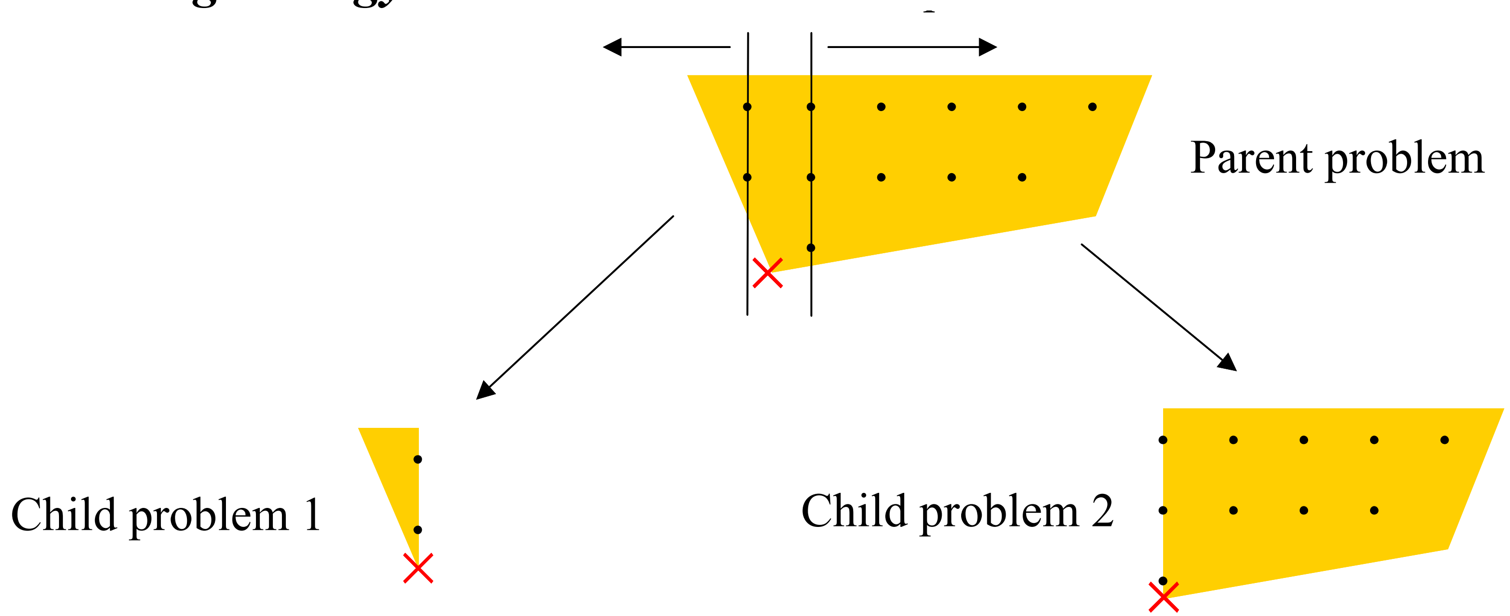
- It uses two strategies:
  - to branch one variable of a parent problem to create two child problems
  - to determine a bound by solving the LP relaxation of each created problem

Branching strategy:

- Consider one integer variable x_n whose optimal value b in the LP relaxation of the parent problem is not integer:
  - one child problem is created by adding to the parent problem the constraint $x_n \leq \lfloor b \rfloor$, where $\lfloor b \rfloor$ means the largest integer smaller than b
  - another child problem is created by adding to the parent problem the constraint $x_n \geq \lceil b \rceil$, where$\lceil b \rceil$ means the smallest integer bigger than b
- For example, if b = 2.3, then, the constraints are $x_n \leq 2$ and $x_n \geq 3$.
- Since x_n is integer, the optimal solution must be in one of the two child problems.

Solving a ILP problem

Branching strategy illustration:



- The feasible region that is eliminated in the LP relaxation of the parent problem has no integer solutions. So, the optimal solution must belong to the feasible region of one of the child problems.
- The LP relaxation value of a child problem is closer to the optimal solution. So, the worst value among both child problems improves, on average, the bound towards the optimal solution value.

B&B illustration

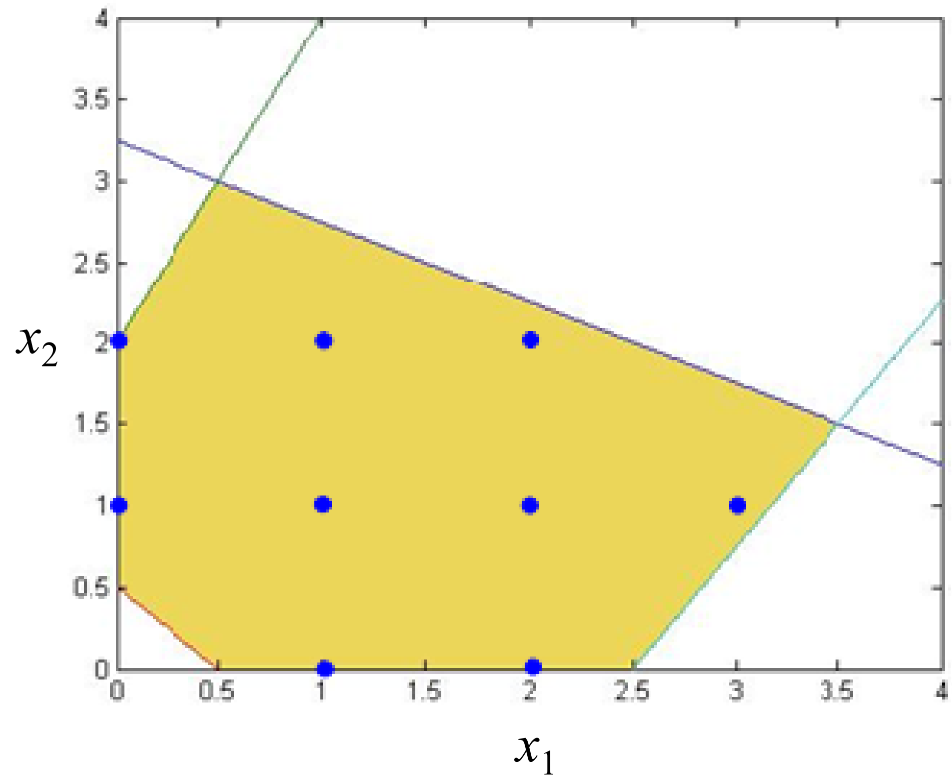
Consider the following ILP model with 2 integer variables x_1 and x_2 :

Minimize

$$f(x_1,x_2) = -3\,x_1 - 4\,x_2$$

Subject to:

$$2x_1 + 4x_2 \leq 13$$
$$-2x_1 + x_2 \leq 2$$
$$2x_1 + 2x_2 \geq 1$$
$$6x_1 - 4x_2 \leq 15$$
$$x_1, x_2 \in \{0,1,2,\ldots\}$$



B&B illustration

Minimize

$$f(x_1,x_2) = -3x_1 - 4x_2$$

Subject to:

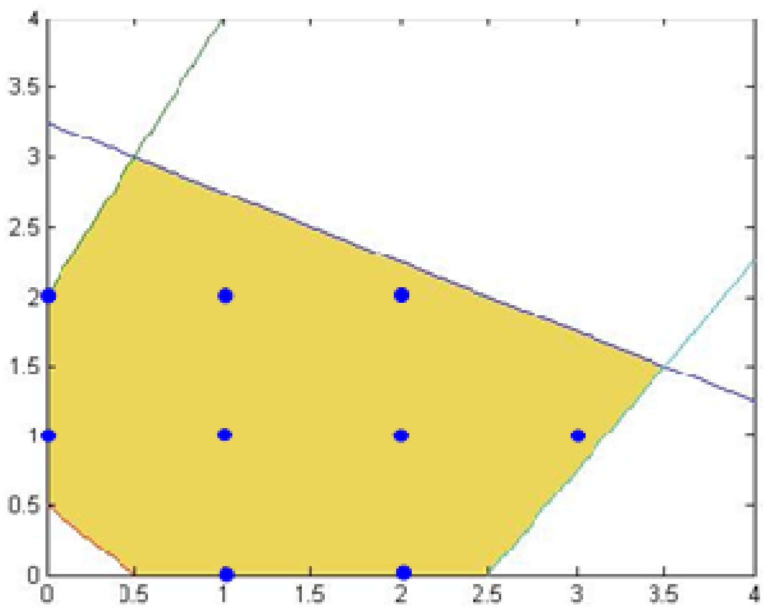
$$2x_1 + 4x_2 \le 13$$

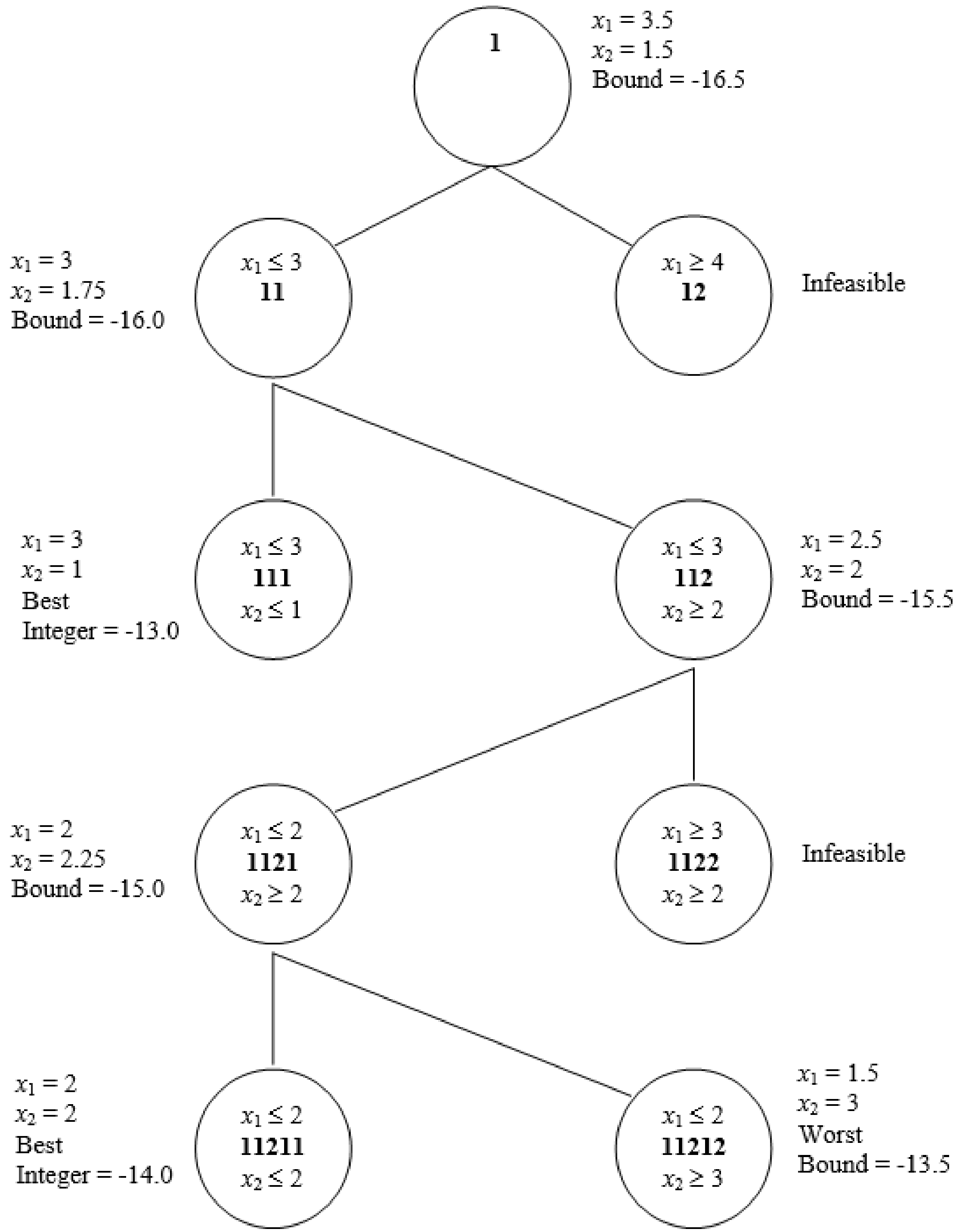
$$-2x_1 + x_2 \le 2$$

$$2x_1 + 2x_2 \ge 1$$

$$6x_1 - 4x_2 \le 15$$

$$x_1, x_2 \in \{0,1,2,\ldots\}$$





Solving a ILP problem

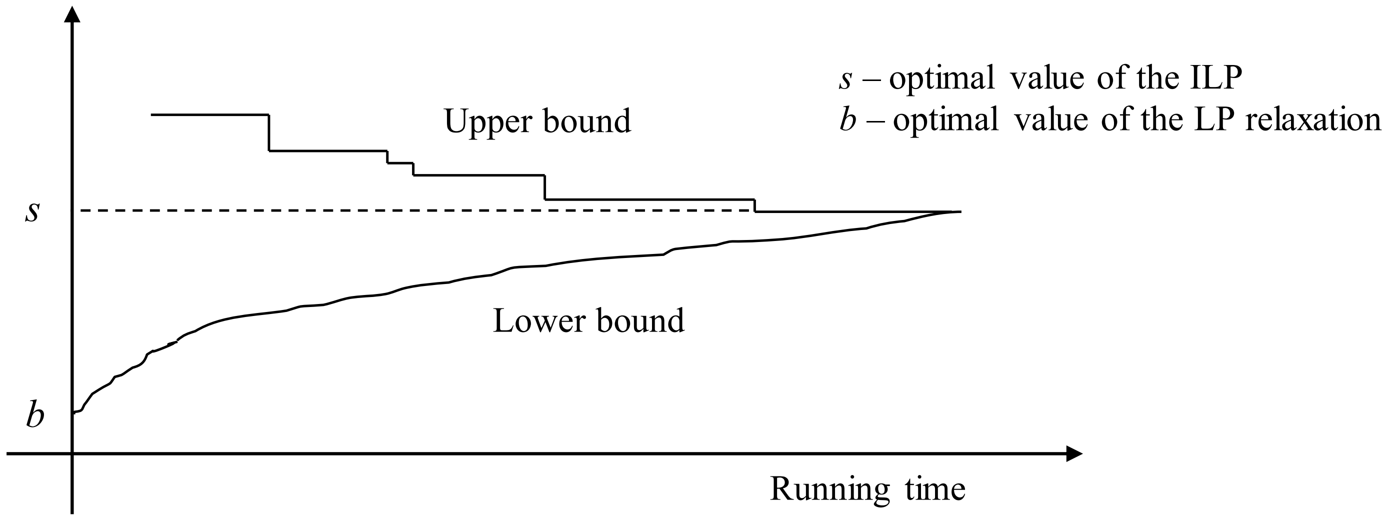
Branch-and-bound (B&B) algorithm:

1. Initialize the set of problems S with the ILP model.
2. **While** S is not empty **do**
    3. Select (and remove) one problem from S
    4. Solve the LP relaxation of the selected problem
    5. **If** the solution is integer, **then** save it if it is the best integer solution found so far
    6. **If** the solution is not integer and has a better value than the best solution found so far, **then** select a variable to branch the selected problem and add to S both child problems
7. **EndWhile**

- B&B reaches the optimal integer solution by solving LP relaxations and branching to force the vertices of the child problems to be integer
- The problem selection strategy (Step 3) and the variable selection strategy (Step 6) influence the efficiency of B&B

Solving a ILP problem

B&B evolution *(assuming a minimization problem)*



- At the beginning, the LP relaxation value *b* of the ILP model gives an initial lower bound.
- Branching generates child problems that improve the lower bound.

  At each iteration, the lowest LP relaxation value of all child problems still not branched is the current lower bound.
- When a better integer solution is found, its value is a better upper bound.
- B&B terminates when the lower bound reaches the upper bound.

CPLEX output while solving an ILP

At the beginning:

```
        Nodes                                              Cuts/
   Node  Left     Objective   IInf  Best Integer      Best Node    ItCnt

      0     0    1616.7782     14                     1616.7782      159
    100    98    6687.0969     17                     1627.4873     2846
    200   188    7844.4932     27                     1641.8359     6110
    300   252    2266.4667     17                     1739.3385     9700
*   363   198    5060.4000      0      5060.4000      1739.3385    10773
    400   233    3310.3566     26      5060.4000      1753.8190    12073
    500   333    2763.4553     21      5060.4000      1844.7267    14954
    600   431    3889.0949     15      5060.4000      1872.8646    17092
    700   528    3936.8238     27      5060.4000      1900.2539    19652
    800   628    3854.4460     20      5060.4000      1904.7431    22192
    900   722    3014.9230     17      5060.4000      1916.7131    25053
   1000   822    4244.3373     29      5060.4000      1938.4084    28172
Elapsed b&b time =   3.35 sec. (tree size =  0.42 MB)
   1100   920    3559.3986     11      5060.4000      1948.9554    30403
   1200  1015    2786.7921     29      5060.4000      1969.3917    32222
   1300  1113    4383.9618     23      5060.4000      1972.2794    35072
   1400  1206    2403.6301     14      5060.4000      1981.7726    38086
   1500  1303    3461.0657     29      5060.4000      1991.0393    41135
   1600  1397    2729.6676     31      5060.4000      2001.3802    43505
   1700  1494    3534.1418     25      5060.4000      2016.3489    46298
   1800  1590    2696.6417     20      5060.4000      2023.6108    48681
```

CPLEX output while solving an ILP

In the middle:

```
   9800  4318      2499.0212      8      2635.2000      2477.4380    266849
   9900  4372      2516.4219     17      2635.2000      2477.6865    269365
  10000  4430         cutoff             2635.2000      2478.3701    271858
Elapsed b&b time =  25.98 sec. (tree size =  2.48 MB)
  10100  4481      2576.9732     16      2635.2000      2478.6094    273582
  10200  4543      2539.8511     18      2635.2000      2478.9666    276070
  10300  4587         cutoff             2635.2000      2479.5206    278309
  10400  4648      2583.2774     14      2635.2000      2479.6259    280693
* 10412  3288      2590.5000      0      2590.5000      2479.6259    281003
* 10476  3127      2585.2000      0      2585.2000      2479.7663    282693
  10500  3131      2499.2180     29      2585.2000      2479.8316    283164
  10600  3148      2566.8968     19      2585.2000      2480.5380    285544
  10700  3178      2583.5000     21      2585.2000      2481.0380    287916
  10800  3214      2537.3264     16      2585.2000      2481.0380    290302
  10900  3260      2558.3365     11      2585.2000      2481.0380    292525
  11000  3316      2481.0380      8      2585.2000      2481.0380    295042
Elapsed b&b time =  29.06 sec. (tree size =  1.85 MB)
  11100  3368      2528.8357     11      2585.2000      2481.0569    297527
  11200  3409     infeasible             2585.2000      2481.4536    299832
  11300  3437      2546.4500     22      2585.2000      2481.8382    302106
  11400  3459         cutoff             2585.2000      2482.3623    304049
  11500  3474      2483.3663      7      2585.2000      2482.7171    305506
  11600  3518      2530.2363     24      2585.2000      2483.0864    307987
  11700  3554      2512.1779     14      2585.2000      2483.2259    310220
  11800  3601      2507.6632      1      2585.2000      2483.2259    312790
```

CPLEX output while solving an ILP

At the end:

```
  54000   747     2524.9094    13      2527.5000     2524.9094   850794
Elapsed b&b time =  79.36 sec. (tree size =  0.61 MB)
  54100   723     2527.0279    18      2527.5000     2525.0370   851438
  54200   678        cutoff            2527.5000     2525.2254   852057
  54300   618     2525.3176     6      2527.5000     2525.3176   852395
  54400   596     2525.4309    11      2527.5000     2525.4309   853023
  54500   575        cutoff            2527.5000     2525.5552   854095
  54600   528        cutoff            2527.5000     2525.6413   854361
  54700   475    infeasible            2527.5000     2525.7959   854982
  54800   428     2525.9293    19      2527.5000     2525.9293   855518
  54900   391    infeasible            2527.5000     2526.0934   856405
  55000   343    infeasible            2527.5000     2526.2915   857084
Elapsed b&b time =  80.90 sec. (tree size =  0.35 MB)
  55100   300        cutoff            2527.5000     2526.4496   857686
  55200   262    infeasible            2527.5000     2526.5999   858490
  55300   243     2526.7380     7      2527.5000     2526.7380   859079
  55400   205        cutoff            2527.5000     2526.7839   859920
  55500   152        cutoff            2527.5000     2526.9734   860603
  55600   101     2527.1528    19      2527.5000     2527.1528   861123

Integer optimal solution (0.0001/0):  Objective =   2.5275000000e+003
Current MIP best bound =   2.5272501068e+003 (gap = 0.249893)
Solution time =   81.97 sec.  Iterations = 861697  Nodes = 55678 (73)
```

Exercise 1

Consider the optimization problem defined by the following Linear Programming Model with two variables x_1 and x_2:

Maximize

$$x_1 + x_2$$

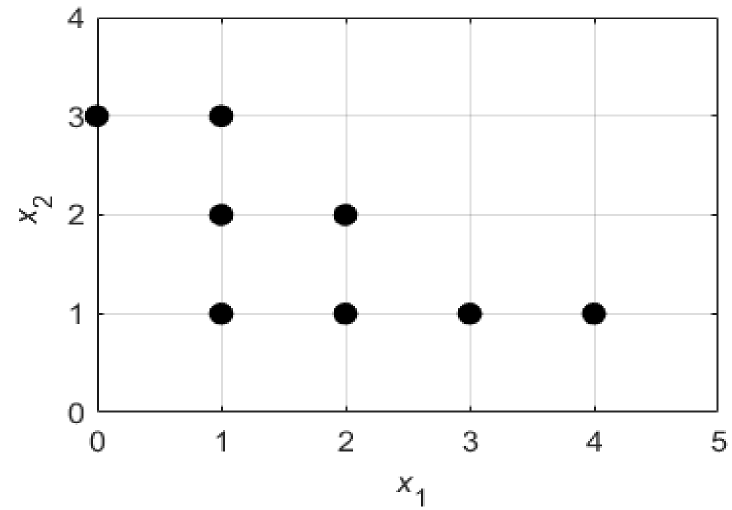
Subject to:

$$x_1 + 2\,x_2 \geq 2$$
$$-2\,x_1 + x_2 \leq 2$$
$$5\,x_1 + 3\,x_2 \leq 15$$
$$x_1, x_2 \geq 0$$

1. Draw the space of all feasible solutions of this problem.
2. Determine the optimal value z and one optimal solution of this problem.

Exercise 2

Consider an Integer Linear Programming model of two variables x_1 and x_2 and with the solution space represented in the figure.



1. Specify a set of constraints such that the optimal solution of the linear relaxation of the problem is integer for any objective function.
2. Considering that the objective function of the problem is the maximization of $f(x_1, x_2) = 2\,x_1 + 3\,x_2$, determine all optimal solutions of the problem and their objective value.
3. Is there any objective function such that its optimal solution is $(x_1, x_2) =$ (2,2) in this solution space? Justify you answer.