Optimization based on Integer Linear Programming

Desempenho e Dimensionamento de Redes

Prof. Amaro de Sousa (asou@ua.pt)

DETI-UA, 2017/2018

Mathematical programming model

- In an *optimization problem*, the aim is to maximize (or minimize) a given quantity designated by 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:

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

$$(=)$$

$$(\geq)$$

where:

- m is the number of constraints
- $f(X)$ and all $g_i(X)$ are functions of the variables
- k_i are real constants

(Mixed Integer) Linear Programming model

- A *Linear Programming (LP) model* is a mathematical programming model where all variables $X=\{x_1, x_2, \ldots, x_n\}$ are non-negative reals and $f(X)$ and $g_i(X)$ are linear functions:
  - functions in the form $a_1x_1 + a_2x_2 + \ldots + a_nx_n$ where all a_i are real constants.
- An *Integer Linear Programming (ILP) model* is an LP model where all variables $X=\{x_1, x_2, \ldots, x_n\}$ are non-negative integers.
- A *Mixed Integer Linear Programming (MILP) model* is an LP model where some variables $X=\{x_1, x_2, \ldots, x_n\}$ are non-negative integers and others are non-negative reals.

Illustrative example

Consider a transportation company that has been requested to deliver the following items 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	70	20	80	35	40

The company has 2 vans for item delivery:

- the first van has a capacity of 100
- the second van has a capacity of 60.

Since it is not possible to deliver all items with the 2 vans, the aim is to choose the items to be carried on each van to maximize the revenue.

Solving steps:

1st - define the 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	70	20	80	35	40

VARIABLES DEFINING THE PROBLEM:

`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 first van
`y1,2` – Binary variable that, if is 1 in the solution, indicates that item 1 is carried by second van
…
`y6,1` – Binary variable that, if is 1 in the solution, indicates that item 6 is carried by first van
`y6,2` – Binary variable that, if is 1 in the solution, indicates that item 6 is carried by second van

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	70	20	80	35	40

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

The objective function is the total revenue of the delivered items

```
Maximize
 + 2.3 x1 + 4.5 x2 + 1.5 x3 + 5.4 x4 + 2.9 x5 + 3.2 x6
Subject To
 + 30 y1,1 + 70 y2,1 + 20 y3,1 + 80 y4,1 + 35 y5,1 + 40 y6,1 <= 100
 + 30 y1,2 + 70 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

List of binary variables

Illustrative example – using CPLEX (1)

Starting CPLEX:

```
Welcome to IBM(R) ILOG(R) CPLEX(R) Interactive Optimizer 12.6.1.0
  with Simplex, Mixed Integer & Barrier Optimizers
5725-A06 5725-A29 5724-Y48 5724-Y49 5724-Y54 5724-Y55 5655-Y21
Copyright IBM Corp. 1988, 2014.  All Rights Reserved.

Type 'help' for a list of available commands.
Type 'help' followed by a command name for more
information on commands.

CPLEX>
```

Reading file ‘exemplo.lp’ on CPLEX:

```
CPLEX> read exemplo.lp
Problem 'exemplo.lp' read.
Read time = 0.01 sec. (0.00 ticks)
CPLEX>
```

Illustrative example – using CPLEX (2)

Solving the problem on CPLEX:

```
CPLEX> optimize

        Nodes                                               Cuts/
   Node  Left     Objective  IInf  Best Integer    Best Bound    ItCnt     Gap

*     0+    0                          1.5000        19.8000               ---
*     0+    0                         11.2000        19.8000            76.79%
      0     0        11.8286     1    11.2000        11.8286        4     5.61%
*     0+    0                         11.5000        11.8286             2.86%
      0     0         cutoff          11.5000                       4     0.00%
Elapsed time = 0.14 sec. (1.12 ticks, tree = 0.00 MB, solutions = 3)

Root node processing (before b&c):
  Real time             =    0.14 sec. (1.12 ticks)
Parallel b&c, 4 threads:
  Real time             =    0.00 sec. (0.00 ticks)
  Sync time (average)   =    0.00 sec.
  Wait time (average)   =    0.00 sec.
                          ------------
Total (root+branch&cut) =    0.14 sec. (1.12 ticks)

Solution pool: 4 solutions saved.

MIP - Integer optimal solution:  Objective = 1.1500000000e+001
Solution time =    0.16 sec.  Iterations = 4  Nodes = 0
Deterministic time = 1.12 ticks  (7.17 ticks/sec)
```

Optimal solution value

Illustrative example – using CPLEX (3)

Displaying the values of the optimal solution:

Items 1, 2, 3 and 6 are selected to be delivered

Items 1 and 2 are carried by first van

Items 3 and 6 are carried by second van

```
CPLEX> display solution variables -
Incumbent solution
Variable Name           Solution Value
x1                            1.000000
x2                            1.000000
x3                            1.000000
x6                            1.000000
y1,1                          1.000000
y2,1                          1.000000
y3,2                          1.000000
y6,2                          1.000000
All other variables in the range 1-18 are 0.
CPLEX>
```

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	70	20	80	35	40

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:

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

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

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

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

Illustrative example – generating LP file with MATLAB

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

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

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

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

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

```
r= [2.3 4.5 1.5 5.4 2.9 3.2];
s= [30 70 20 80 35 40];
c= [100 60];
n= length(r);
v= length(c);
fid = fopen('exemplo.lp','wt');
fprintf(fid,'Maximize\n');
for i=1:n
    fprintf(fid,' + %f x%d',r(i),i);
end
fprintf(fid,'\nSubject To\n');
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
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');
for i=1:n
    fprintf(fid,' x%d\n',i);
    for j=1:v
        fprintf(fid,' y%d,%d\n',i,j);
    end
end
fprintf(fid,'End\n');
fclose(fid);
```

Illustrative example - using Gurobi on Internet (1)

- Prepare an ASCII file with the problem defined in LP format and compress it with Zip:

  for example: exemplo.zip

- Go to https://neos-server.org/neos/solvers/index.html
- Select Mixed Integer Linear Programming tools
- Select Gurobi [LP Input]

Mixed Integer Linear Programming –

- Cbc [AMPL Input][GAMS Input][MPS Input]
- CPLEX [AMPL Input][GAMS Input][LP Input][MPS Input]
- feaspump [AMPL Input][CPLEX Input][MPS Input]
- FICO-Xpress [AMPL Input][GAMS Input][MOSEL Input][MPS Input]
- Gurobi [AMPL Input][GAMS Input][LP Input][MPS Input]
- MINTO [AMPL Input]
- MOSEK [AMPL Input][GAMS Input][LP Input][MPS Input]
- proxy [CPLEX Input][MPS Input]
- qsopt_ex [AMPL Input][LP Input][MPS Input]
- scip [AMPL Input][CPLEX Input][GAMS Input][MPS Input][OSIL Input][ZIMPL Input]
- SYMPHONY [MPS Input]

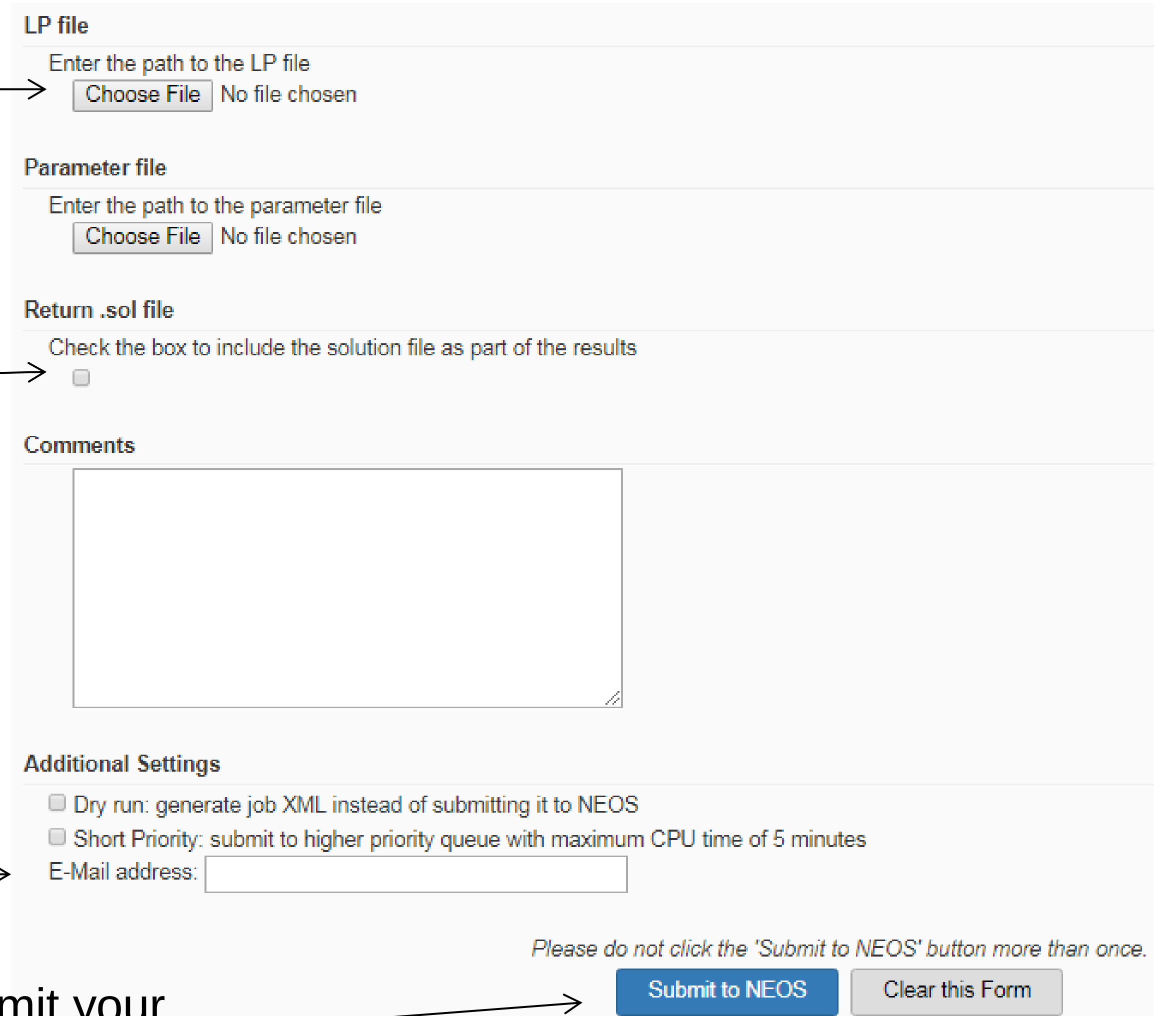
Illustrative example - using Gurobi on Internet (2)

1 . Upload exemplo.zip

2. Check the box

3. Insert a valid email address

4. Submit your ILP problem


LP file

Enter the path to the LP file

Choose File No file chosen

Parameter file

Enter the path to the parameter file

Choose File No file chosen

Return .sol file

Check the box to include the solution file as part of the results

Comments

Additional Settings

Dry run: generate job XML instead of submitting it to NEOS

Short Priority: submit to higher priority queue with maximum CPU time of 5 minutes

E-Mail address:

Please do not click the 'Submit to NEOS' button more than once.

Submit to NEOS Clear this Form

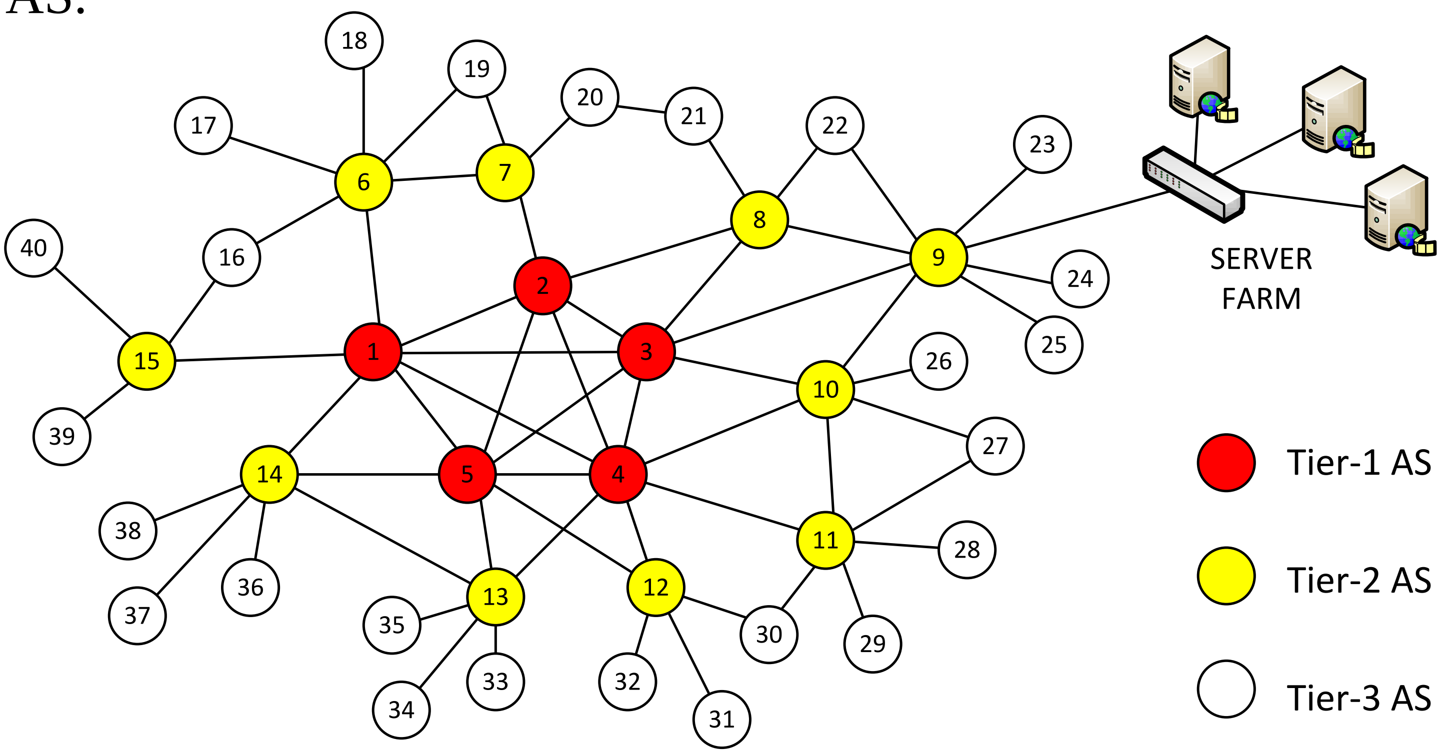
Illustrative example - using Gurobi on Internet (3)

- After the problem is solved, the solution is displayed (and also sent to the email address):

```
Optimal solution found (tolerance 1.00e-04)
Best objective 1.150000000000e+01, best bound 1.150000000000e+01, gap 0.0000%
Optimal objective: 11.5
********** Begin .sol file *************
# Objective value = 11.5
x1 1
x2 1
x3 1
x4 0
x5 0
x6 1
y1,1 1
y2,1 1
y3,1 0
y4,1 0
y5,1 0
y6,1 0
y1,2 0
y2,2 0
y3,2 1
y4,2 0
y5,2 0
y6,2 1
********** End .sol file **************
```

Solving the server farm location problem with ILP

- We have a set of Autonomous Systems (ASs) and we aim to select a subset of ASs to connect one server farm on each selected AS.
- Only Tier-2 of Tier-3 ASs provide the Internet access service.
- The solution must guarantee that there is a path between each Tier-2 and Tier-3 AS and at least one server farm with no more than one intermediate AS.



Server farm location problem: Notation and Variables

NOTATION:

n – number of Tier-2 and Tier-3 ASs where server farms can be connected to;

c_i – OPEX cost of connecting a server farm to AS i, with $1 \leq i \leq n$;

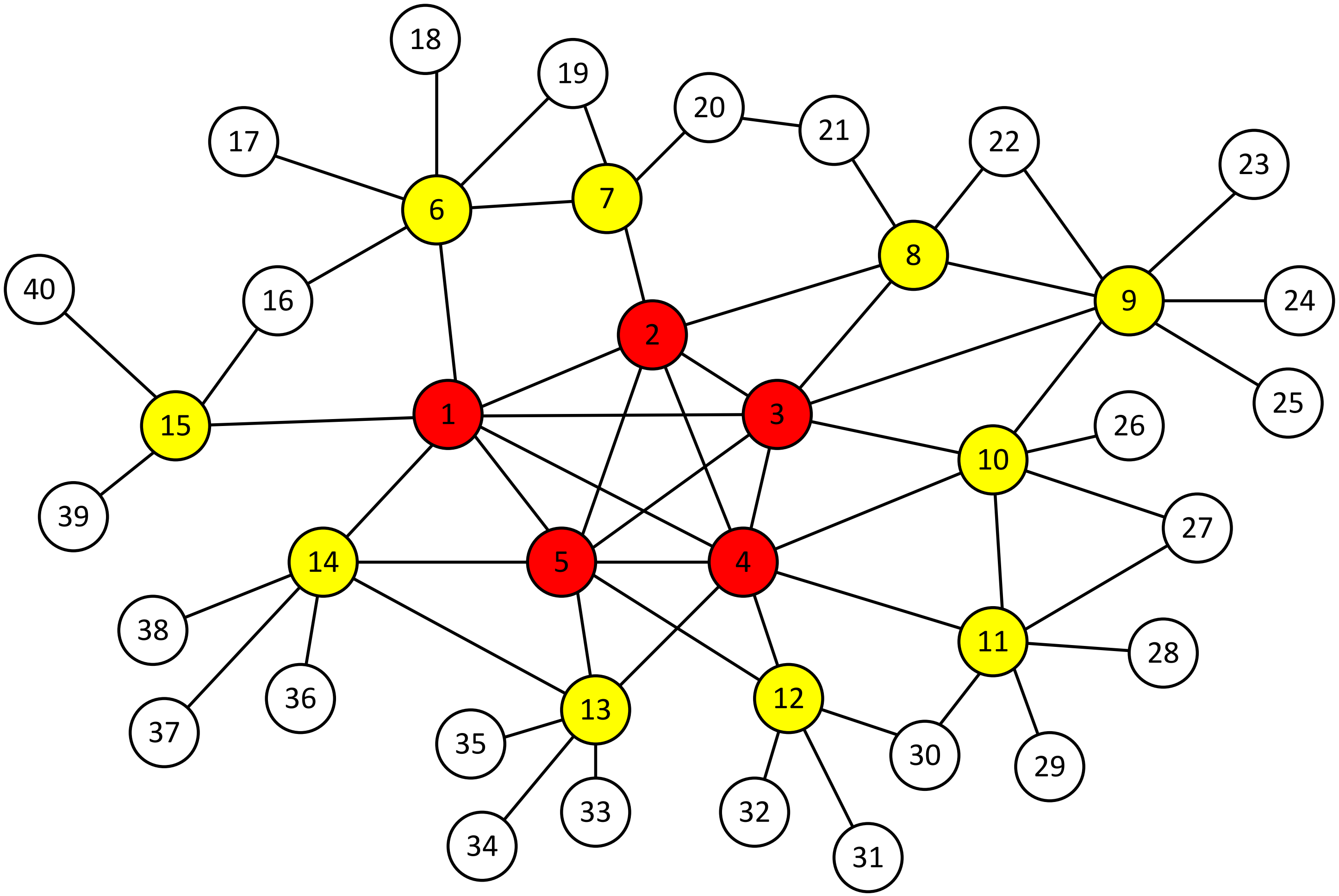
$I(j)$ – set of Tier-2 and Tier-3 ASs such that there is a shortest path between AS j and each AS $i \in I(j)$ with at most one intermediate AS.

VARIABLES:

x_i – binary variable, with $1 \leq i \leq n$, that when is equal to 1 means that AS i must be connected to one server farm;

y_{ji} – binary variable, with $1 \leq j \leq n$ and $i \in I(j)$, that when is equal to 1 means that AS j is associated with AS i.

Server farm location problem: examples of sets $I(j)$



Set $I(j)$ for $j = 6$ is: $\{6,7,14,15,16,17,18,19,20\}$

for $j = 16$ is: $\{6,7,15,16,17,18,19,39,40\}$

Server farm location problem: ILP Model

Minimize $\sum_{i=1}^{n} c_i x_i$ (1)

Subject to:

$\sum_{i \in I(j)} y_{ji} = 1 \quad , j = 1 \dots n$ (2)

$y_{ji} \leq x_i \quad , j = 1 \dots n, i \in I(j)$ (3)

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

$y_{ji} \in \{0,1\} \quad , j = 1 \dots n, i \in I(j)$ (5)

- The objective (1) is the minimization of the OPEX costs of the selected server farms.
- Constraints (2) guarantee that each AS j is associated with one AS $i \in I(j)$ while constraints (3) guarantee that an associated AS $i \in I(j)$ must have one server farm connected. So, constraints (2–3) guarantee that each AS j has always one server farm whose shortest path has at most one intermediate AS.
- Constraints (4–5) define all variables as binary variables.