Course on mathematical modelling: AMPL and CPLEX

teacher: Giacomo Lanza

Dipartimento di Informatica, Università di Pisa

a.a. 2019-2020

Set Covering Problem (Chapter 4 p.32-33)

The Set Covering location problem is to locate the minimum number of facilities required to "cover" all clients, by considering the following input data:

- I = set of clients (or *demand nodes*)
- J = set of candidate *facility locations*
- d_{ij} = distance between $i \in I$ and $j \in J$, $\forall i \in I, j \in J$
- D_C = coverage distance
- $N_i = \{j \in J : d_{ij} \leq D_C\}$, i.e. the set of all candidate facility locations that can cover i, $\forall i \in I$

Set Covering Problem (Chapter 4 p.32-33)

$$x_j = \begin{cases} 1 & \text{if we locate at site } j \\ 0 & \text{otherwise} \end{cases}, \ \forall j \in J.$$

$$\min \sum_{j \in J} c_j \, x_j$$

$$\sum_{j \in N_i} x_j \geq 1, \ \forall i \in I \quad \textit{covering constraints} \qquad \text{(SCLP)}$$

$$x_j \in \{0, 1\}, \ \forall j \in J$$

Set Covering Problem (Chapter 4 p.32-33)

$$x_j = \begin{cases} 1 & \text{if we locate at site } j \\ 0 & \text{otherwise} \end{cases}, \ \forall j \in J.$$

$$\min \sum_{j \in J} c_j \, x_j$$

$$\sum_{j \in N_i} x_j \geq 1, \ \forall i \in I \quad \textit{covering constraints} \qquad \text{(SCLP)}$$

$$x_j \in \{0, 1\}, \ \forall j \in J$$

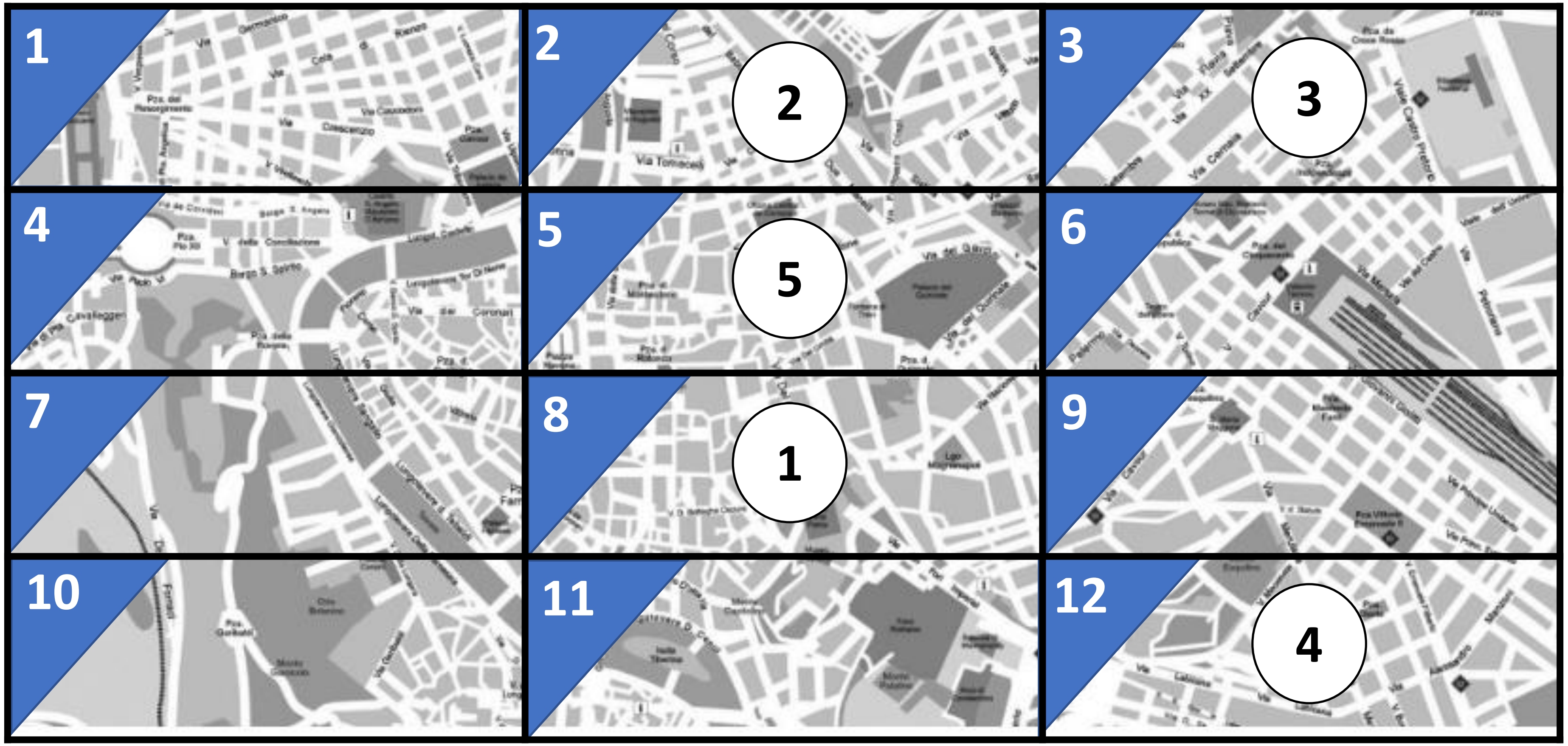
An example of SCLP

The director of a new bank needs to decide where to open a number of ATM so as to cover entirely the center of the city in which the bank will open. The director and the team split the area of the city center as a grid, defining 12 sub-areas. 5 of the 12 sub-areas are also chosen as locations where an ATM could be opened. Each ATM may serve the sub-area in which it is located and all the sub-areas adjacent to it. There is also a cost of construction of an ATM, which depends on the sub-area. The director wants to open the suitable number of ATM in order to cover all the sub-areas by minimizing the cost of construction.

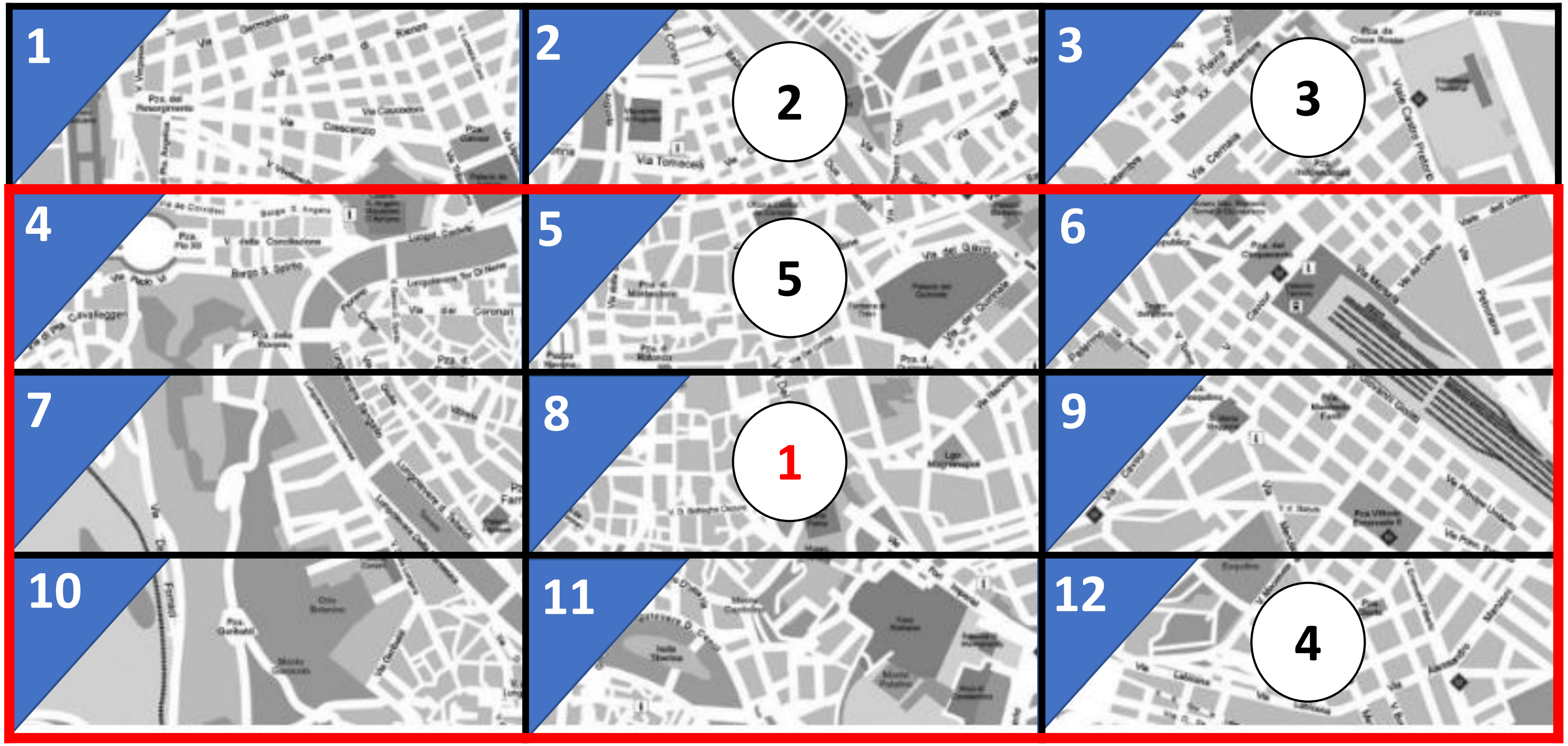
An example of SCLP

- *I = set of* **sub-areas of the city center** *(or demand nodes)*
- *J = set of candidate* **ATM locations**
- **Coverage Distance***:* each ATM may serve the sub-area in which it is located and all the sub-areas adjacent to it
- There is also a **cost of construction** of an ATM which depends on the sub-area

An example of SCLP

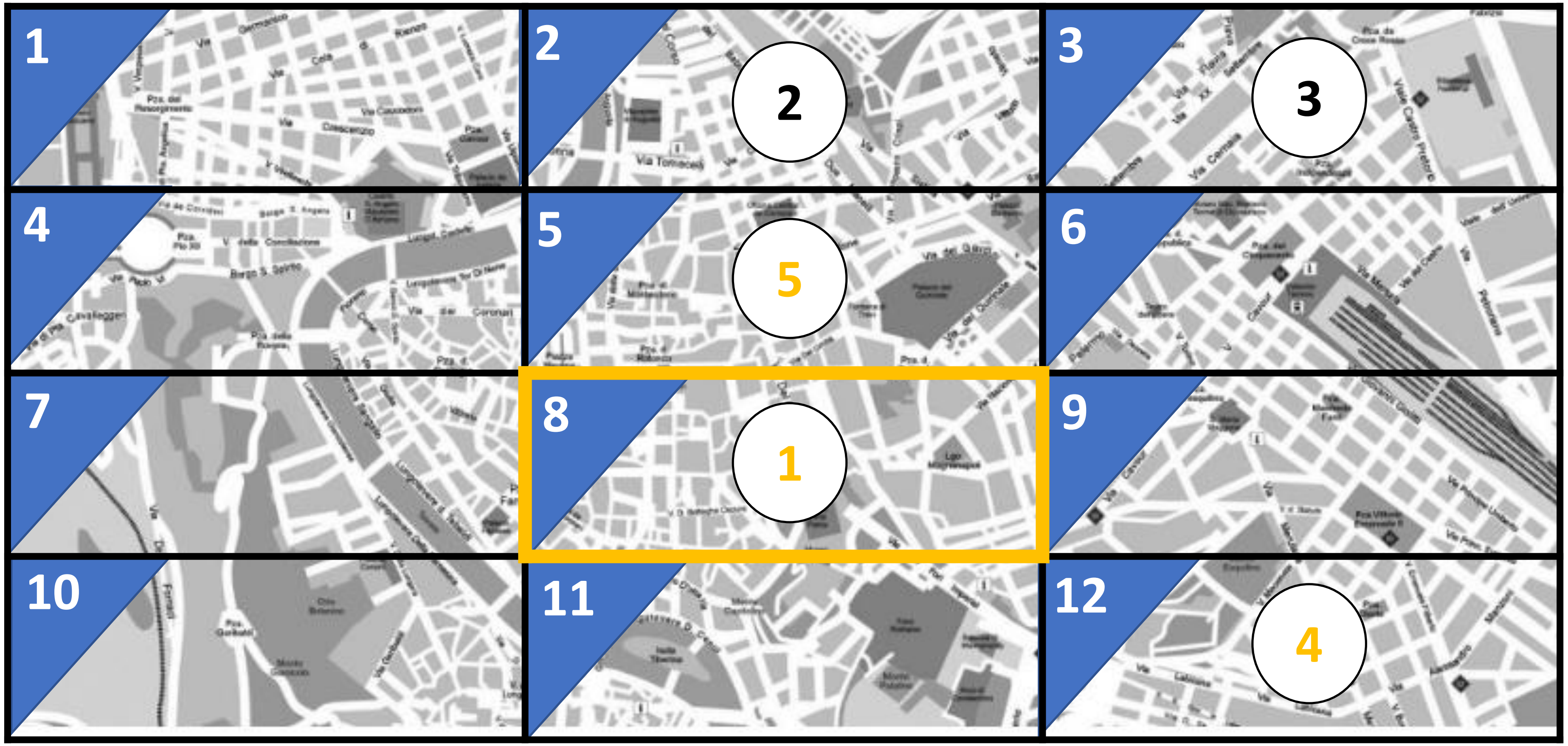


An example of SCLP



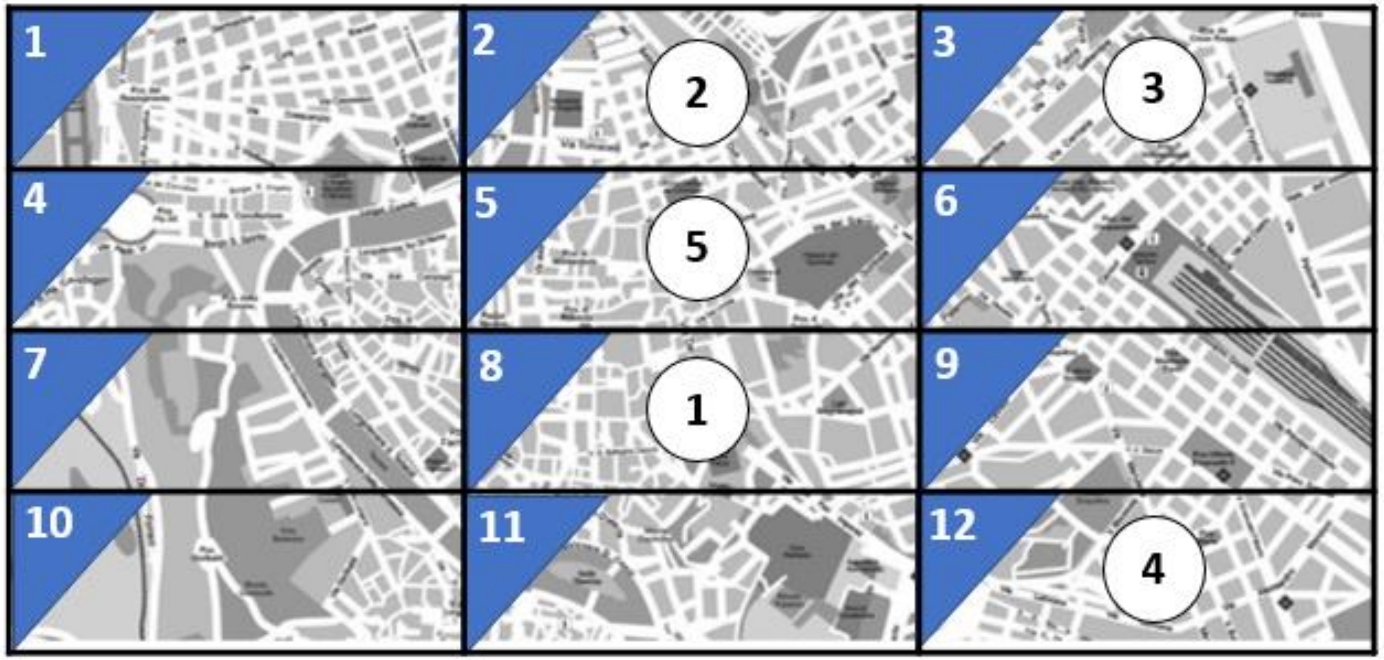
ATM 1 serves its sub-area and the adjacent sub-areas {4, 5, 6, 7, 8, 9, 10, 11, 12}

An example of SCLP



Sub-area 8 could be served by ATM {1, 5, 4}

An example of SCLP



ATM	Cost of Construction
1	4
2	5
3	5
4	4
5	3

Example of SCLP: Set Definition

- *I = set of* **sub-areas of the city center** (indexed by i)
- *J = set of candidate* **ATM locations** (indexed by j)

```
# model
param TotATM;
param TotSubArea;

set ATM:= 1.. TotATM;
set SubArea:= 1.. TotSubArea;
```

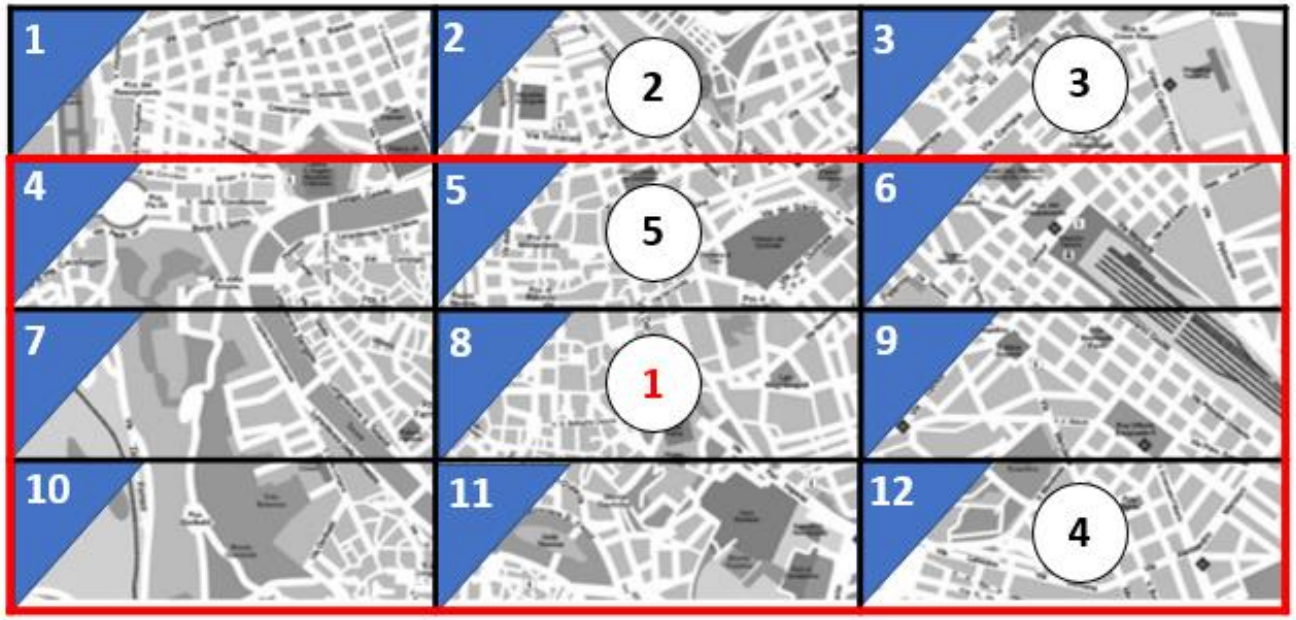
```
# data
param TotATM = 5;
param TotSubArea = 12;
```

Example of SCLP: Parameters Definition

- *Coverage Distance:* each ATM may serve the sub-area in which it is located and all the sub-areas adjacent to it
- There is also a *cost of construction* of an ATM which depends on the sub-area

Example of SCLP: Coverage Matrix

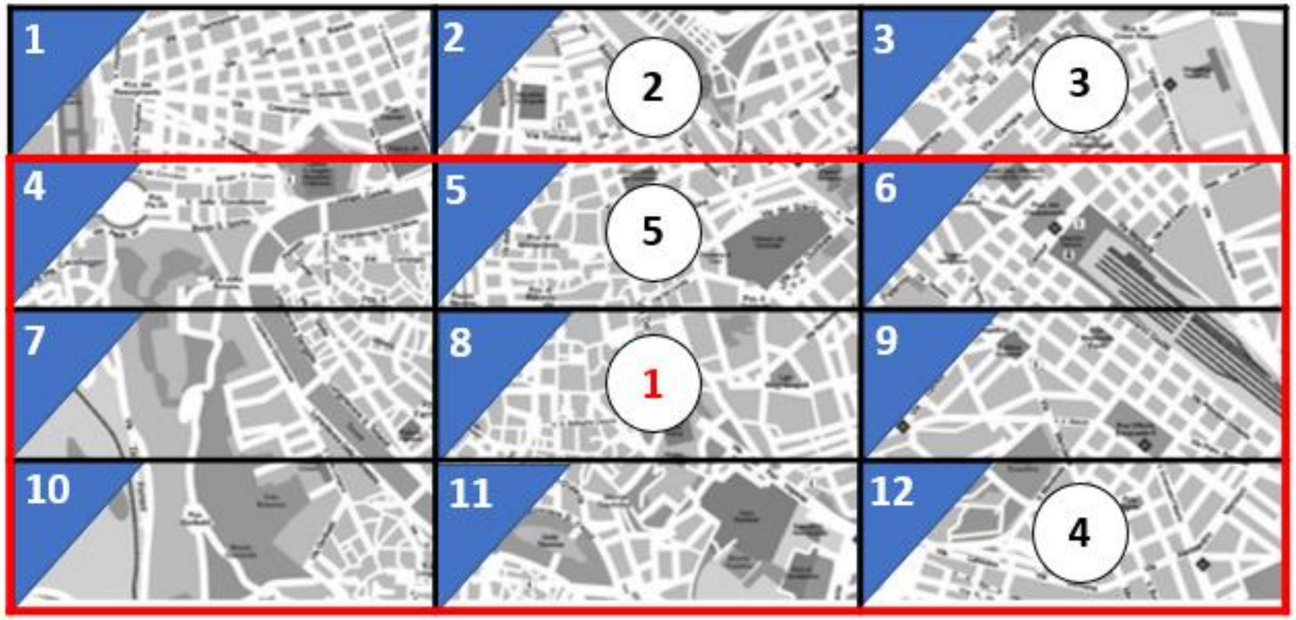
Coverage Distance: **each ATM may serve the sub-area in which it is located** and all the sub-areas adjacent to it




```
# ATM          1 2 3 4 5 :=
# sub-areas
1
2
3
4
5
6
7
8              1
9
10
11
12
;
```

Example of SCLP: Coverage Matrix

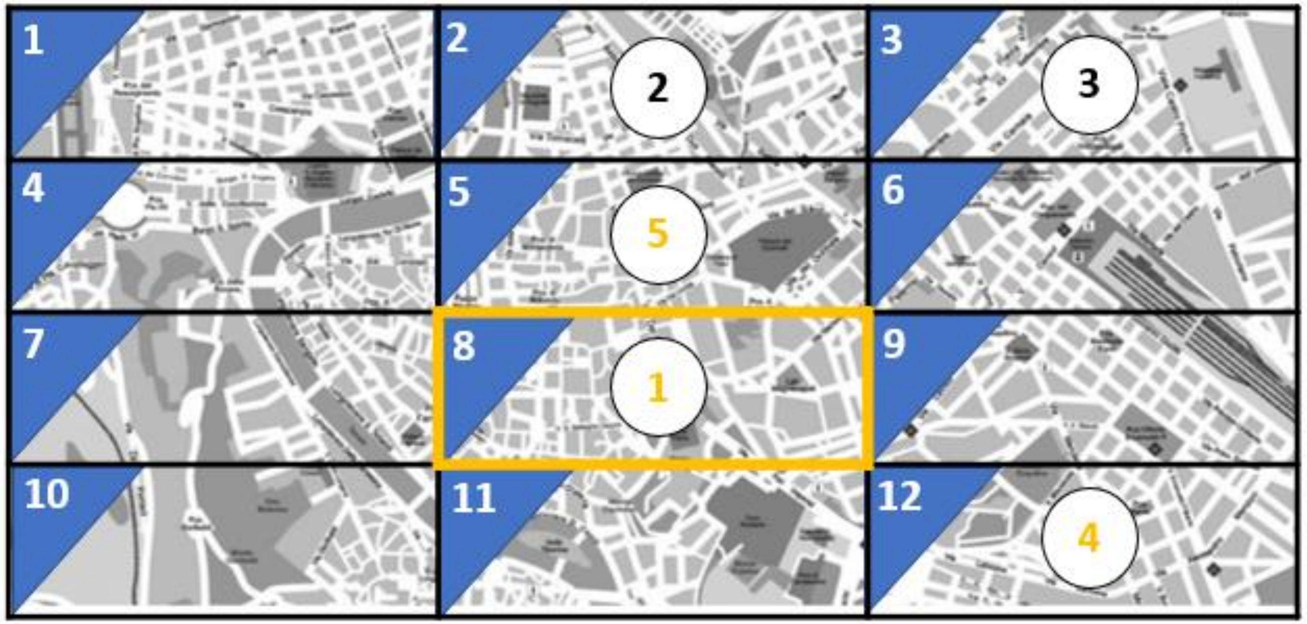
Coverage Distance: each ATM may serve the sub-area in which it is located **and all the sub-areas adjacent to it**




```
# ATM            1 2 3 4 5 :=
# sub-areas
1                0
2                0
3                0
4                1
5                1
6                1
7                1
8                1
9                1
10               1
11               1
12               1
;
```

Example of SCLP: Coverage Matrix

Coverage Distance: each ATM may serve the sub-area in which it is located and all the sub-areas adjacent to it




```
# ATM          1 2 3 4 5 :=
# sub-areas
1              0 1 0 0 1
2              0 1 1 0 1
3              0 1 1 0 1
4              1 1 0 0 1
5              1 1 1 0 1
6              1 1 1 0 1
7              1 0 0 0 1
8              1 0 0 1 1
9              1 0 0 1 1
10             1 0 0 0 0
11             1 0 0 1 0
12             1 0 0 1 0
;
```

Example of SCLP: Parameters Definition

Coverage Distance: each ATM may serve the sub-area in which it is located and all the sub-areas adjacent to it

```
# model
param TotATM;
param TotSubArea;

set ATM:= 1.. TotATM;
set SubArea:= 1.. TotSubArea;

param Coverage {SubArea, ATM};
```

```
# data
param Coverage:
# ATM          1 2 3 4 5 :=
# sub-areas
1              0 1 0 0 1
2              0 1 1 0 1
3              0 1 1 0 1
4              1 1 0 0 1
5              1 1 1 0 1
6              1 1 1 0 1
7              1 0 0 0 1
8              1 0 0 1 1
9              1 0 0 1 1
10             1 0 0 0 0
11             1 0 0 1 0
12             1 0 0 1 0;
```

Example of SCLP: Parameters Definition

There is a *cost of construction* of an ATM which depends on the sub-area

```
# model
param TotATM;
param TotSubArea;

set ATM:= 1.. TotATM;
set SubArea:= 1.. TotSubArea;

param Coverage {SubArea, ATM};
param OpeningCost {ATM};
```

```
# data
param Coverage:
 [ … ]

param OpeningCost:=
1 4
2 5
3 5
4 4
5 3
;
```

Example of SCLP: Variables Definition

$$x_j = \begin{cases} 1 & \text{if we locate at site } j \\ 0 & \text{otherwise} \end{cases}, \ \forall j \in J.$$

```
# model
param TotATM;
param TotSubArea;

set ATM:= 1.. TotATM;
set SubArea := 1.. TotSubArea;
param Coverage {SubArea, ATM};
param OpeningCost {ATM};

var Opening{ATM} binary;
```

Example of SCLP: Objective Definition

The director wants to *minimize the total cost* of ATM construction

```
# model
param TotATM;
param TotSubArea;
set ATM:= 1.. TotATM;
set SubArea := 1.. TotSubArea;
param Coverage {SubArea, ATM};
param OpeningCost {ATM};

var Opening{ATM} binary;

minimize Total_Opening:  sum {j in ATM} OpeningCost[j]*Opening[j];
```

Example of SCLP: Constraint Definition

The director of a new bank needs to decide where to open ATM so as to *cover entirely the city center*

```
# model
param TotATM;
param TotSubArea;
set ATM:= 1.. TotATM;
set SubArea := 1.. TotSubArea;
param Coverage {SubArea, ATM};
param OpeningCost {ATM};

var Opening{ATM} binary;

minimize Total_Opening:  sum {j in ATM} OpeningCost[j]*Opening[j];

subject to CoveringConstr{i in SubArea}: sum {j in ATM} Coverage[i,j] * Opening[j] >= 1;
```

Example of SCLP: Model and Data Files

```
# model
param TotATM;
param TotSubArea;
set ATM:= 1.. TotATM;
set SubArea := 1.. TotSubArea;
param Coverage {SubArea, ATM};
param OpeningCost {ATM};

var Opening{ATM} binary;

minimize Total_Opening:  sum {j in ATM}
OpeningCost[j]*Opening[j];

subject to CoveringConstr{i in SubArea}: sum {j in ATM}
Coverage[i,j] * Opening[j] >= 1;
```

```
data;
param TotATM = 5;
param TotSubArea = 12;
param Coverage:
# sub-areas # ATM   1 2 3 4 5 :=
1                   0 1 0 0 1
2                   0 1 1 0 1
3                   0 1 1 0 1
[…]
10                  1 0 0 0 0
11                  1 0 0 1 0
12                  1 0 0 1 0;
param OpeningCost:=
1 4
2 5
3 5
4 4
5 3;
```

Example of SCLP: Launching with Cplex

```
# Console
ampl: reset;
ampl: model ATM.mod;
ampl: data ATM.dat;
ampl: option solver cplexamp;
ampl: solve;
CPLEX 12.6.1.0: optimal integer solution;
objective 7
0 MIP simplex iterations
0 branch-and-bound nodes
ampl: display Opening;
Opening [*] :=
1  1
2  0
3  0
4  0
5  1;
```

Running scripts: *.run file* and *include*

AMPL provides the ***include*** command that causes input to be taken from a file with extension *.run:*

1. Create a file with extension *.run*
2. Write all the instructions you need
3. Run the model using the keyword ***include*** *filename.run* in the console

Running scripts: *.run file* and *include*

```
# Console
ampl: include ATM.run;
CPLEX 12.6.1.0: optimal integer solution; objective 7
0 MIP simplex iterations
0 branch-and-bound nodes
Opening [*] :=
1  1
2  0
3  0
4  0
5  1
;
```

```
# ATM.run file
reset;
model ATM.mod;
data ATM.dat;
option solver cplexamp;
solve;
display Opening;
```

AMPL Main Commands:

- reset; # reset the environment
- model *modelfilename*.mod; # model upload
- data *datafilename*.dat; # data upload
- option solver *nameofsolver*; # optimizer selection
- solve; # solve
- display *nameofvariables*; # displays variables