Université Paris Sud

Laboratoire de Recherche en Informatique

Network Optimization

Fabio Martignon

Lecture overview

- Multi-commodity flow problem

- Network design problem
  - Node positioning
  - Users' coverage (Assignment problem)
  - Traffic routing

- Radio/coverage planning

Multi-commodity flow problem

Multi-commodity flow problem

Given:

An oriented graph $G = (N, A)$.

The capacity u_{ij} and the cost c_{ij} are associated with each arc $(i , j) \in A$.

A set of demands K, where each demand k is characterized by:

- ➔Source $s_k \in N$
- ➔Destination $t_k \in N$
- ➔An amount of flow d_k

Multi-commodity flow problem (cont.)

- *Problem:*
- Route all the demands at the least cost, taking into account the capacity constraints of the arcs.

Model

Decision variables:

The amount of flow (x^k_{ij}) of demand *k* routed on arc *(i , j)*:

$$x_{ij}^k \geq 0 \qquad \forall (i,j) \in A, \forall k \in K$$

Objective function:

$$\min \sum_{(i,j)\in A} \sum_{k\in K} c_{ij} x_{ij}^k$$

Model

Constraints:

(1) Flow Balance constraints:

$$\sum_{(i,j)\in A} x_{ij}^k - \sum_{(j,i)\in A} x_{ji}^k = \begin{cases} d_k & \text{if } i = s_k, \\ -d_k & \text{if } i = t_k \\ 0 & \text{if } i \neq s_k, t_k, \end{cases} \qquad \forall i \in N, \forall k \in K$$

(2) Capacity constraints:

$$\sum_{k\in K} x_{ij}^k \leq u_{ij}, \qquad \forall (i,j) \in A$$

Model

Constraints:

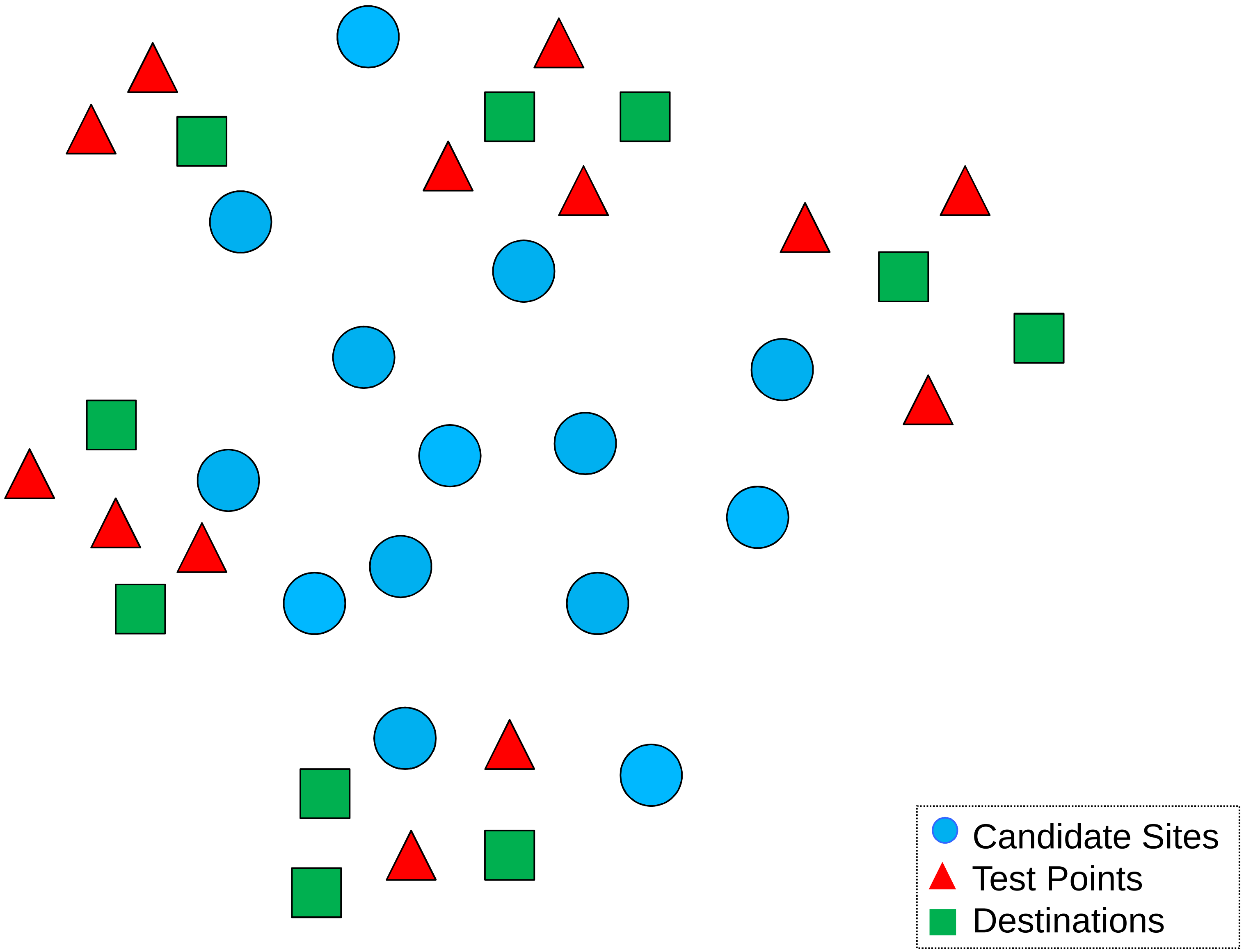
(3) Positivity constraints:

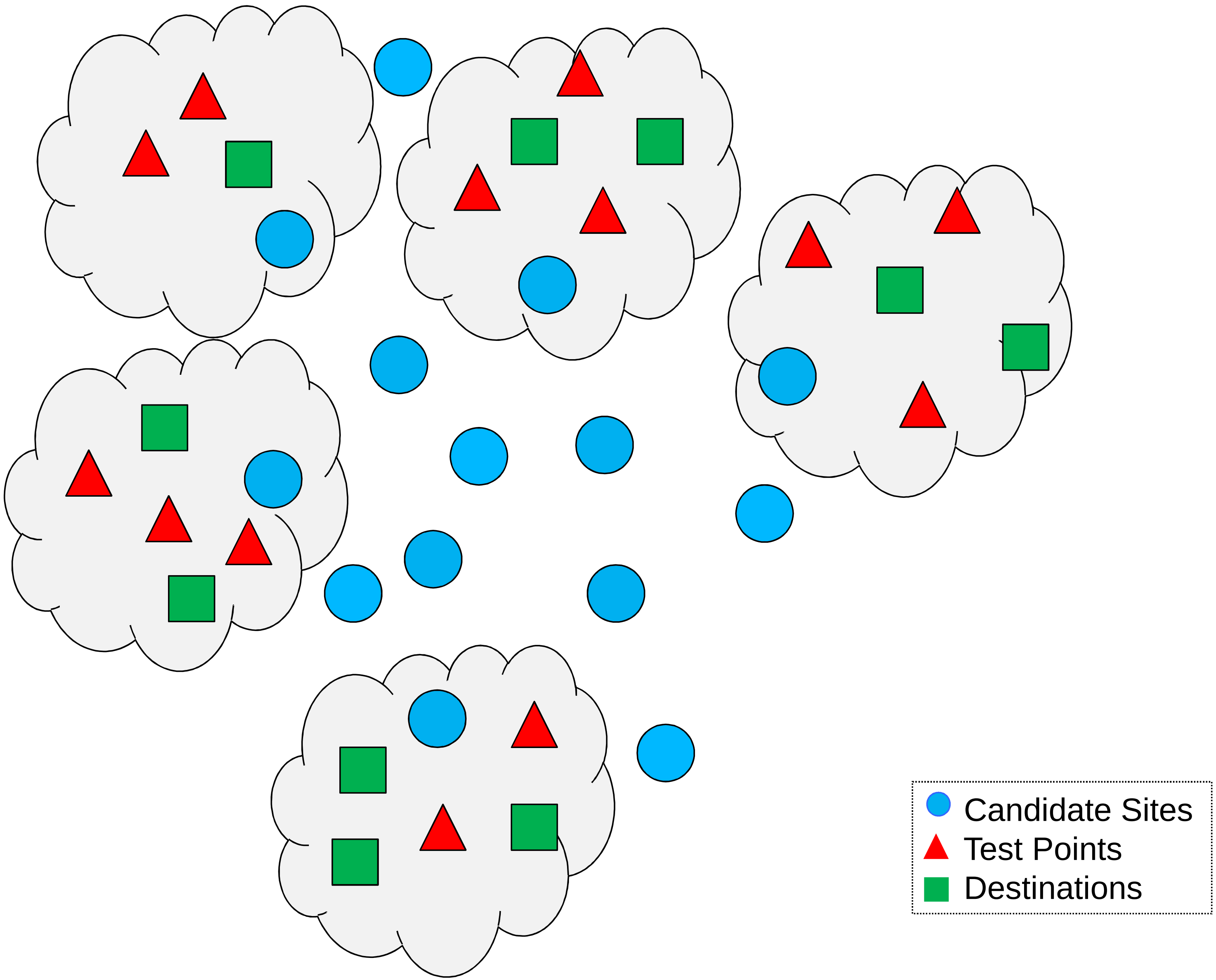
$$x_{ij}^{k} \geq 0, \qquad \forall (i,j) \in A$$

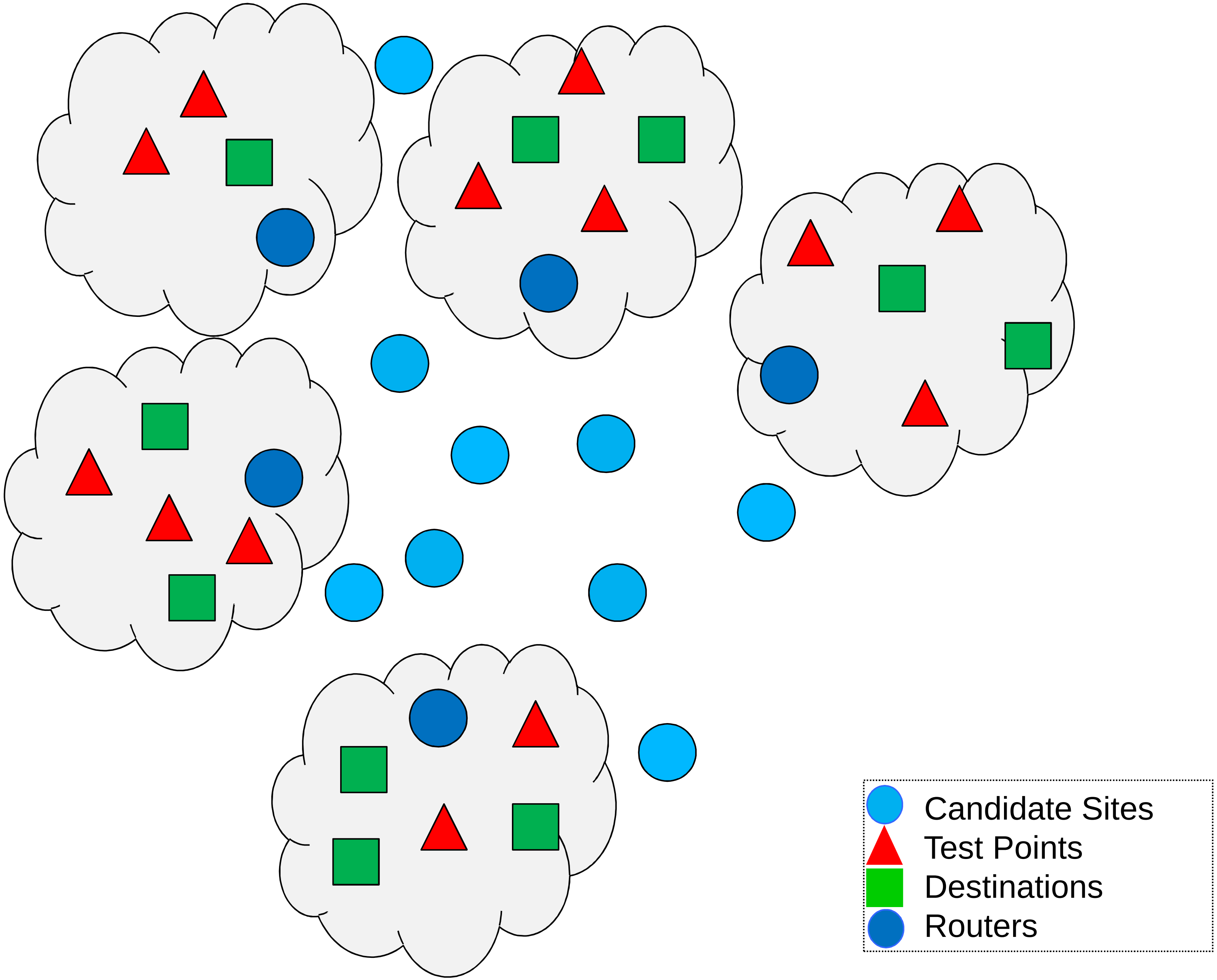
Formulation dimension

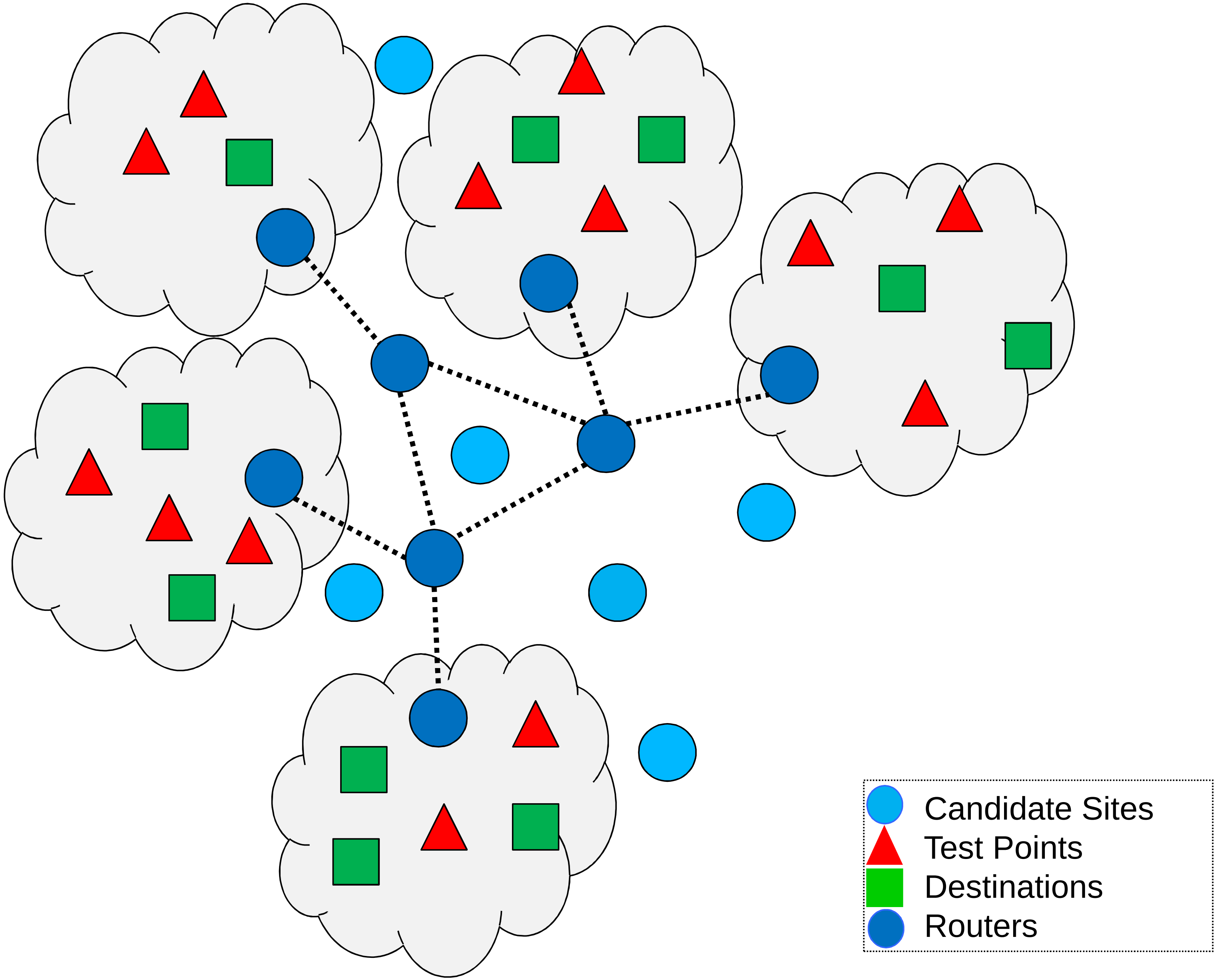
- Number of variables: $|A||K|$
- Number of constraints: $|N||K| + |A|$

Network design problem


Candidate Sites
Test Points
Destinations


Candidate Sites
Test Points
Destinations


Candidate Sites
Test Points
Destinations
Routers


Candidate Sites
Test Points
Destinations
Routers

Network design problem

Given

- A set of Candidate Sites (CSs, where to install nodes)
- A set of test points (TPs) and a set of destinations (DNs)
  – source-destination traffic pairs (s_k, t_k)

Problem

- Install nodes, links, and route traffic demands minimizing the total network installation cost

Network model

Notations and parameters:

- S: the set of CSs
- I: the set of TPs
- D: the set of DNs
- c^I_j: cost for installing a node in CS j
- c^B_{jl}: cost for buying one bandwidth unit between CSs j and l
- c^A_{ij}: Access cost per bandwidth unit between TP i and CS j
- c^E_{jk}: Egress cost per bandwidth unit between CS j and DN k

Network model (cont.)

Notations and parameters:

- d_{ik}: traffic generated by TP i towards DN k
- u_{jl}: maximum capacity that can be reserved on link (j,l)
- v_j: maximum capacity of the access link of CS j
- h_{jk}: maximum capacity that can be reserved on egress link (j,k)
- a_{ij}: 0-1 parameter that indicates if TP i can access the network through CS j
- b_{jl}: 0-1 parameter that indicates if CS j can be connected to CS l
- e_{jk}: 0-1 parameter that indicates if CS j can be connected to DN k

Network model (cont.)

Decision variables:

- x_{ij}: 0-1 variable that indicates if TP i is assigned to CS j
- z_j: 0-1 variable that indicates if a node is installed in CS j
- w_{jk}: 0-1 variable that indicates if CS j is connected to DN k
- f^k_{jl}: flow variable which denotes the traffic flow routed on link (j,l) destined to DN k
- f_{jk}: flow variable which denotes the traffic flow routed on egress link (j, k)

Network model (cont.)

Objective function:

The objective function accounts for the total network cost, including installation costs and the costs related to the connection of nodes, users' access and egress costs.

$$Minimize \quad \left\{ \sum_{j \in S} c_j^I z_j + \sum_{j,l \in S} \sum_{k \in D} c_{jl}^B f_{jl}^k + \right.$$

$$\left. + \sum_{i \in I, j \in S, k \in D} c_{ij}^A d_{ik} x_{ij} + \sum_{j \in S, k \in D} c_{jk}^E f_{jk} \right\}$$

Network model (cont.)

Constraints:

$$\sum_{j \in S} x_{ij} = 1, \quad \forall i \in I$$

$$x_{ij} \leq z_j a_{ij}, \quad \forall i \in I, j \in S$$

$$\sum_{i \in I} d_{ik} x_{ij} + \sum_{l \in S} (f_{lj}^k - f_{jl}^k) - f_{jk} = 0, \quad \forall j \in S, k \in D$$

$$\sum_{k \in D} f_{jl}^k \leq u_{jl} b_{jl} z_j, \sum_{k \in D} f_{jl}^k \leq u_{jl} b_{jl} z_l, \quad \forall j, l \in S$$

Network model (cont.)

Constraints:

$$\sum_{i \in I, k \in D} d_{ik} x_{ij} \leq v_j, \quad \forall j \in S$$

$$f_{jk} \leq h_{jk} w_{jk}, \quad \forall j \in S, k \in D$$

$$w_{jk} \leq e_{jk} z_j, \quad \forall j \in S, k \in D$$

$$x_{ij}, z_j, w_{jk} \in \{0, 1\}, \quad \forall i \in I, j \in S, k \in D$$

AMPL basics

- AMPL means "A Mathematical Programming Language"
- AMPL is an implementation of the Mathematical Programming language
- Many solvers can work with AMPL
- AMPL works as follows:
  - translates a user-defined model to a low-level formulation (called ***flat form***) that can be understood by a solver
  - passes the *flat form* to the solver
  - reads a solution back from the solver and interprets it within the higher-level model (called ***structured form)***

AMPL basics (cont.)

- AMPL usually requires three files:
  - the model file (extension **.mod**) holding the MP formulation
  - the data file (extension **.dat**), which lists the values to be assigned to each parameter symbol
  - the run file (extension **.run**), which contains the (imperative) commands necessary to solve the problem
- The model file is written in the MP language
- The data file simply contains numerical data together with the corresponding parameter symbols
- The run file is written in an imperative C-like language (many notable differences from C, however)
- Sometimes, MP language and imperative language commands can be mixed in the same file (usually the run file)
- To run AMPL, type ***ampl my-runfile.run*** from the command line

costModel.mod

```
set D;                      # set of destinations
set TP;                     # set of TPs
set CS;                     # set of CSs

param ITP{TP,CS};           # Matrix aij (TP/CS)
param ID{CS,D};             # Matrix eij (CS/D)
param ICS{CS,CS};           # Matrix bjl (CS/CS)
param d{TP,D};              # Traffic generated by each TP, destined to destination D
param U{CS,CS};             # Capacity on the link CS/CS
param costU{CS,CS};         # Transport cost (per unit of bandwidth) for traffic on the transport link
between                       the CSs
param V{CS};                # Capacity of the link between each TP and CS
param costR{CS};            # Router installation cost
param costD{CS,D};          # Cost (per bandwidth unit) for traffic on the link between the CS and the
                             destination D
param costTP{TP,CS};        # Cost (per bandwidth unit) for traffic on the link between the TP and the
                              CS
param H{CS,D};              # Capacity of the egress link CS/D

var x{TP,CS} binary;        # Binary variable of assignment of each TP to a CS
var z{CS} binary;           # Binary variable of installation of a router in a CS
var f{CS,CS,D} >=0;         # Flow variable per destination D on the link between CSs
var w{CS,D} binary;         # Binary variable of connection of CS to a destination node D
var fw{CS,D} >=0;           # Flow variable on the link between a CS and a destination D
```

costModel.mod (cont.)

```
minimize total_cost: sum {j in CS} (costR[j] * z[j]) +
             sum {j in CS, l in CS, k in D} (costU[j,l] * f[j,l,k]) +
             sum {j in CS, k in D} (costD[j,k] * fw[j,k]) +
             sum {j in CS, i in TP, k in D} d[i,k] * x[i,j] * costTP[i,j];

subject to assignment {i in TP}: sum {j in CS} x[i,j] = 1;
subject to existence {i in TP, j in CS}: x[i,j] <= ITP[i,j] * z[j];

subject to flow_balance_constraints {j in CS, k in D}: sum {i in TP} d[i,k]*x[i,j] + sum
{l in CS} (f[l,j,k] - f[j,l,k]) - fw[j,k] = 0;
subject to max_flow_per_TP_CS {j in CS}: sum {i in TP, k in D} d[i,k] * x[i,j] <= V[j];

subject to connect_CS_D {j in CS, k in D}: w[j,k] <= ID[j,k] * z[j];
subject to flow_CS_D {j in CS, k in D}: fw[j,k] <= H(j,k) * w[j,k];

subject to link_existence_1 {j in CS, l in CS: j!=l}: sum {k in D} f[j,l,k] <= U[j,l] *
ICS[j,l] * z[j];
subject to link_existence_2 {j in CS, l in CS: j!=l}: sum {k in D} f[j,l,k] <= U[j,l] *
ICS[j,l] * z[l];
```

runfile_costModel.run

```
model costModel.mod;
data outfile.dat;
option solver 'cplexamp';
option log_file 'ffile.log';
option cplex_options 'timing 1' 'mipdisplay=1' 'integrality=1e-09';
option display_1col 1000000;
solve;

display _solve_user_time > results_processingTime.out;
display (sum {i in TP, j in CS} x[i,j] + (sum {j in CS, k in D: fw[j,k]!= 0} 1) + sum {j in CS, l in
CS: (sum {k in D} f[j,l,k]) != 0} 1) > results_nbrOfLinks.out;
display x > results_xy.out;
display z > results_z.out;
display f > results_perk_f.out;
display sum {j in CS} z[j] > results_nbrOfRouters.out;
display w > results_w.out;
display {j in CS, l in CS} (sum {k in D} f[j,l,k]) > results_f.out;
display fw > results_fw.out;
display total_cost > results_totalCost.out;
display sum {j in CS} (costoR[j] * z[j]) > results_zcost.out;
display solve_result_num > solve.tmp;
quit;
```

Solution

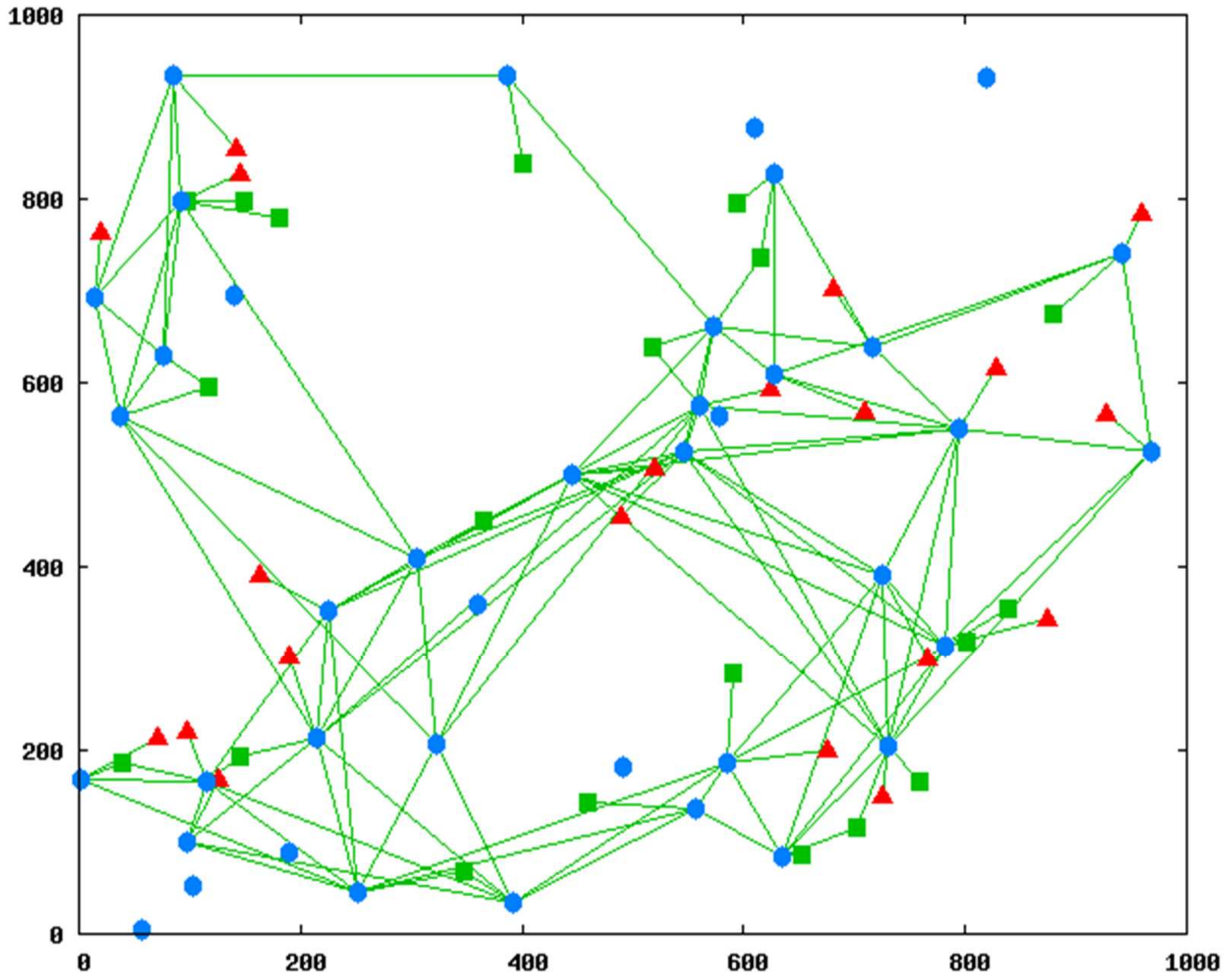
Node log . . .

Best integer = 4.390008e+03 Node = 0 Best node = 4.046214e+03
Best integer = 4.238774e+03 Node = 0 Best node = 4.052842e+03
Best integer = 4.099293e+03 Node = 0 Best node = 4.057592e+03
Best integer = 4.096009e+03 Node = 40 Best node = 4.072417e+03
Best integer = 4.094250e+03 Node = 138 Best node = 4.085538e+03
Best integer = 4.093422e+03 Node = 178 Best node = 4.089841e+03

Implied bound cuts applied: 5
Flow cuts applied: 708
Mixed integer rounding cuts applied: 1

Times (seconds):
Input = 0.084005
Solve = 106.719
Output = 0.48003
CPLEX 11.0.1: optimal integer solution within mipgap or absmipgap; ***objective 4093.422***
22401 MIP simplex iterations
204 branch-and-bound nodes
absmipgap = 0.279608, relmipgap = 6.83066e-05
708 flow-cover cuts
5 implied-bound cuts
1 mixed-integer rounding cut

Example of a planned network

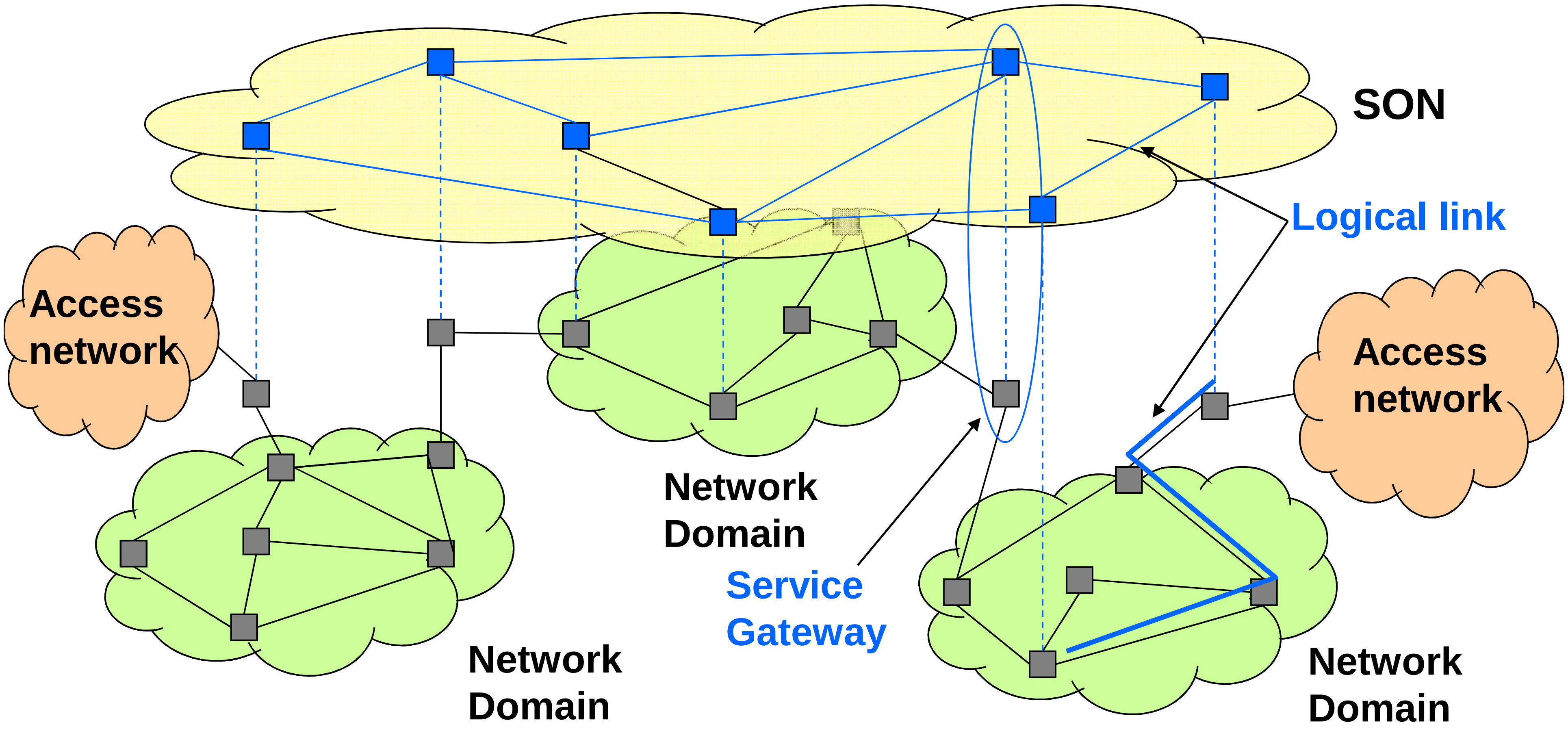


- Randomly generated topology
- *LxL* square, with *L*=1000

● 40 Candidate Sites
▲ 20 Test Points
■ 20 Destination Nodes

Network Design Applications: Service Overlay Network

- SON is an application-layer network built on top of the traditional IP-layer networks

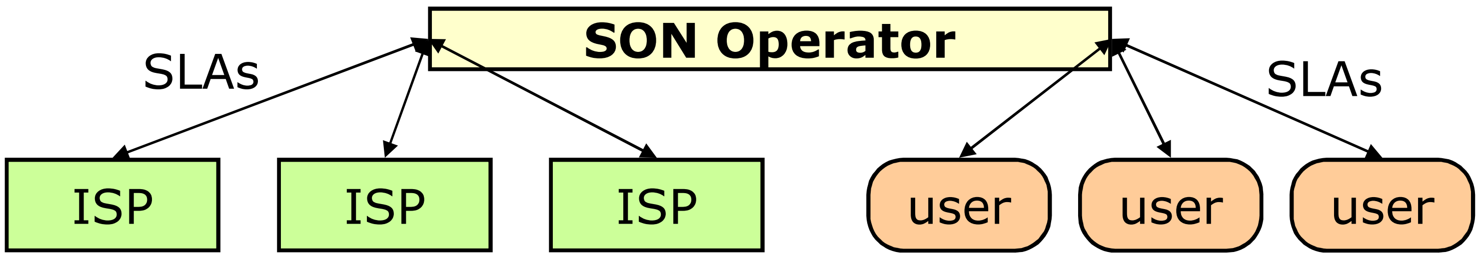


What is a Service Overlay Network?

- SON is operated by an "overlay ISP"
- The SON operator owns one or more overlay nodes (also called "service gateways") hosted in the underlying ISP domains
- Overlay nodes are interconnected by virtual overlay links that are mapped into paths of the underlying network
- SON operator purchases bandwidth for virtual links from ISPs with bilateral SLAs
- SON provides QoS guarantees to customers implementing application specific traffic management mechanisms

Why using SONs ?

- SONs provide a simple solution to end-to-end QoS both from a technical and an economical perspective



- SONs don't require any changes in the underlying networks
- SONs provide a unified framework that can be shared by different applications

Topology Design & Bandwidth Provisioning of SONs

- **Problem Statement:**

  Given a set of Candidate Sites (where to install overlay nodes) and source-destination traffic pairs:
- **Goals:**

Deploy a SON that:

1. Minimizes the total network installation cost
2. Maximizes the profit of the SON operator
   - Taking into account the SON operator's budget

- **Critical issues:**
  - Revenue: the model must take explicitly into account the SON operator's revenue in the optimization procedure
  - The number and location of overlay nodes are not pre-determined
  - Capacity constraints on overlay links are considered
  - Fast and efficient heuristics must be developed to deal with large-scale network optimization and to support periodical SON redesign based on traffic statistics measured on-line

Topology Design & Bandwidth Provisioning of SONs

We now illustrate an optimization framework for planning SONs

- Two mathematical programming models:
  1. The first model (**FCSD**) minimizes the network installation cost while providing full coverage to all users
  2. The second model (**PMSD**) maximizes the SON profit choosing which users to serve based on the expected gain and taking into account the budget constraint

Topology Design & Bandwidth Provisioning of SONs

- Two efficient heuristics to get near-optimal solutions for large-size network instances with a short computing time

1. The Cost Minimization SON Design Heuristic (**H-FCSD**)

2. The Profit Maximization SON Design Heuristic (**H-PMSD**)

Mathematical Models

FCSD

Objective Function:

(FCSD: Full-Coverage SON Design model)

$$\underbrace{Minimize \sum_{j \in S} c_j^I z_j}_{\text{Node Installation cost}} + \underbrace{\sum_{j,l \in S} \sum_{k \in D} c_{jl}^B f_{jl}^k}_{\text{Overlay links bandwidth cost}} +$$

$$+ \underbrace{\sum_{i \in I, j \in S, k \in D} c_{ij}^A d_{ik} x_{ij}}_{\text{Access cost}} + \underbrace{\sum_{j \in S, k \in D} c_{jk}^E f_{jk}}_{\text{Egress cost}}$$

PMSD

Objective Function:

(PMSD: Profit Maximization SON Design model)

$$Maximize \underbrace{\sum_{i \in I, j \in S, k \in D} g_i d_{ik} x_{ij}}_{\text{SON revenue}} - \{\sum_{j \in S} c_j^I z_j +$$

$$+ \sum_{j,l \in S} \sum_{k \in D} c_{jl}^B f_{jl}^k + \sum_{i \in I, j \in S, k \in D} c_{ij}^A d_{ik} x_{ij} + \sum_{j \in S, k \in D} c_{jk}^E f_{jk}\}$$

Subject to:

Flow Conservation constraints

Access and Egress coverage

Coherence and Integrality constraints

Profit Maximization Model

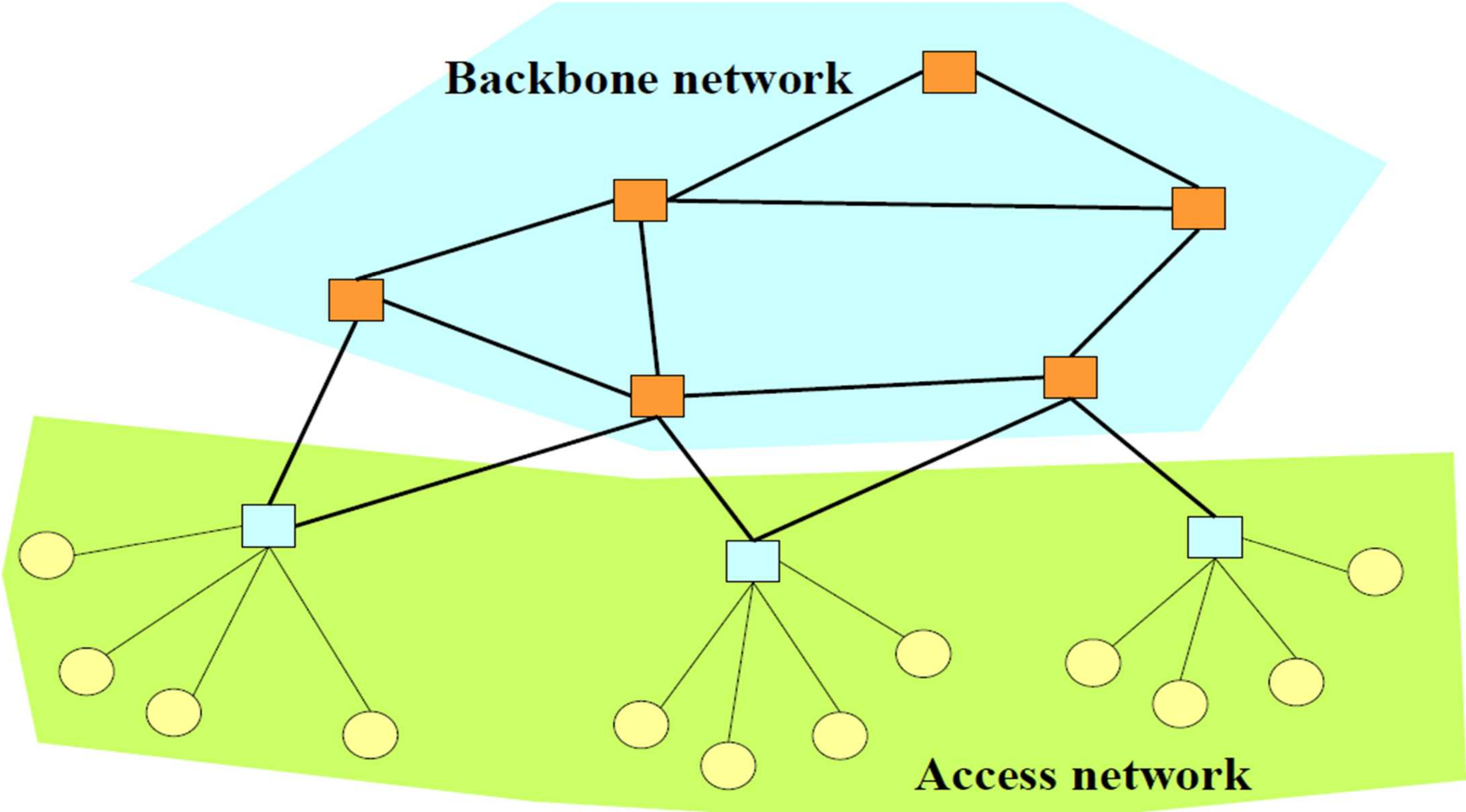
Budget constraint

- The SON planner may define a budget (B) to limit the economic risks in the deployment of its network:

$$\underbrace{\sum_{j \in S} c_j^I z_j + \sum_{j,l \in S} \sum_{k \in D} c_{jl}^B f_{jl}^k + \sum_{i \in I, j \in S, k \in D} c_{ij}^A d_{ik} x_{ij} + \sum_{j \in S, k \in D} c_{jk}^E f_{jk} \leq B}_{\text{Budget Constraint}}$$

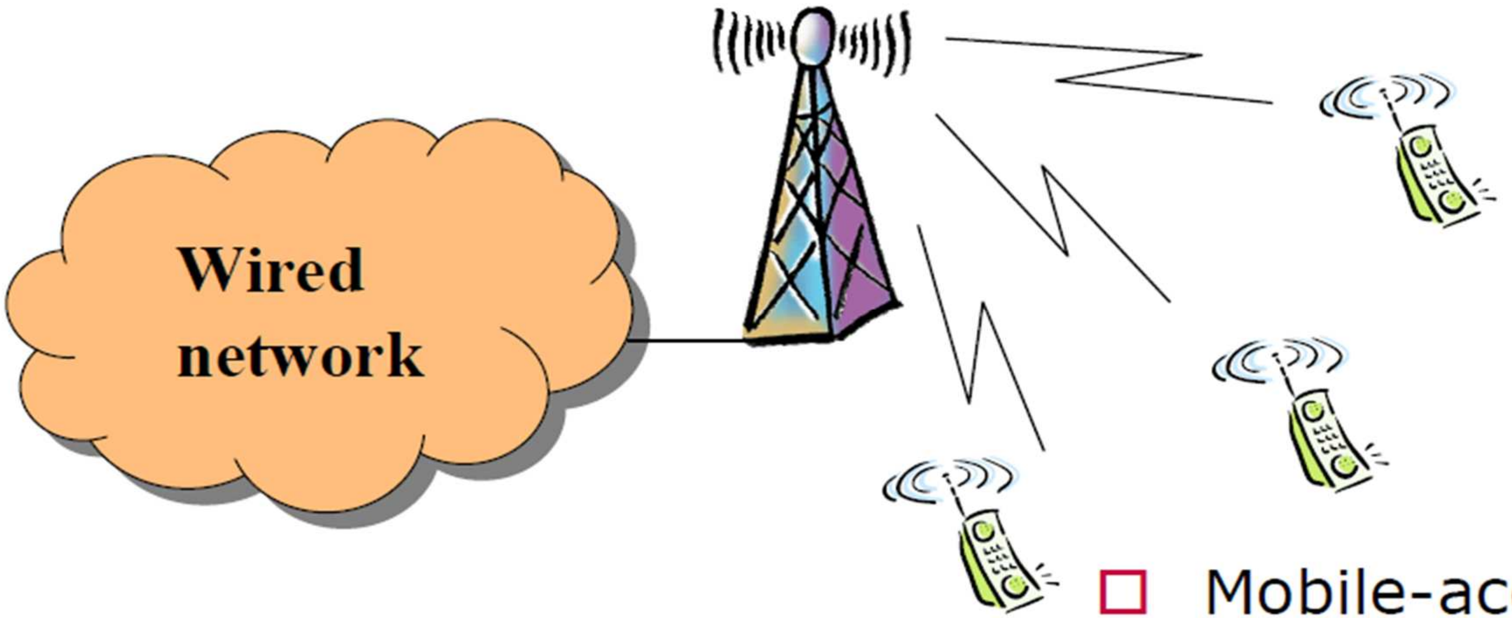
Radio planning

Network architecture



Wireless Network

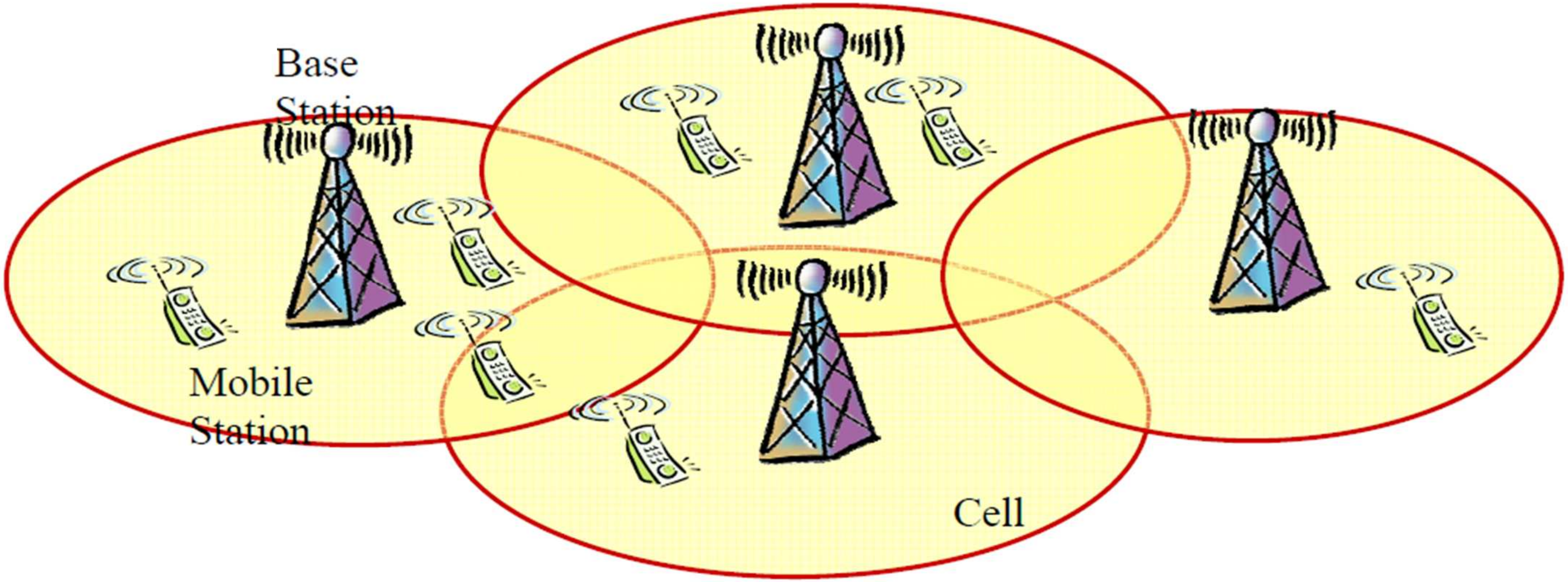
- Wireless networks are mainly access networks
- Fixed access point (cellular systems, WLAN, WMAN)



- Mobile-access point connection

Wireless Network

Cellular coverage: the territory coverage is obtained by Base Stations–BS (or Access Points) that provide radio access to Mobile Stations (MSs) within a service area called CELL

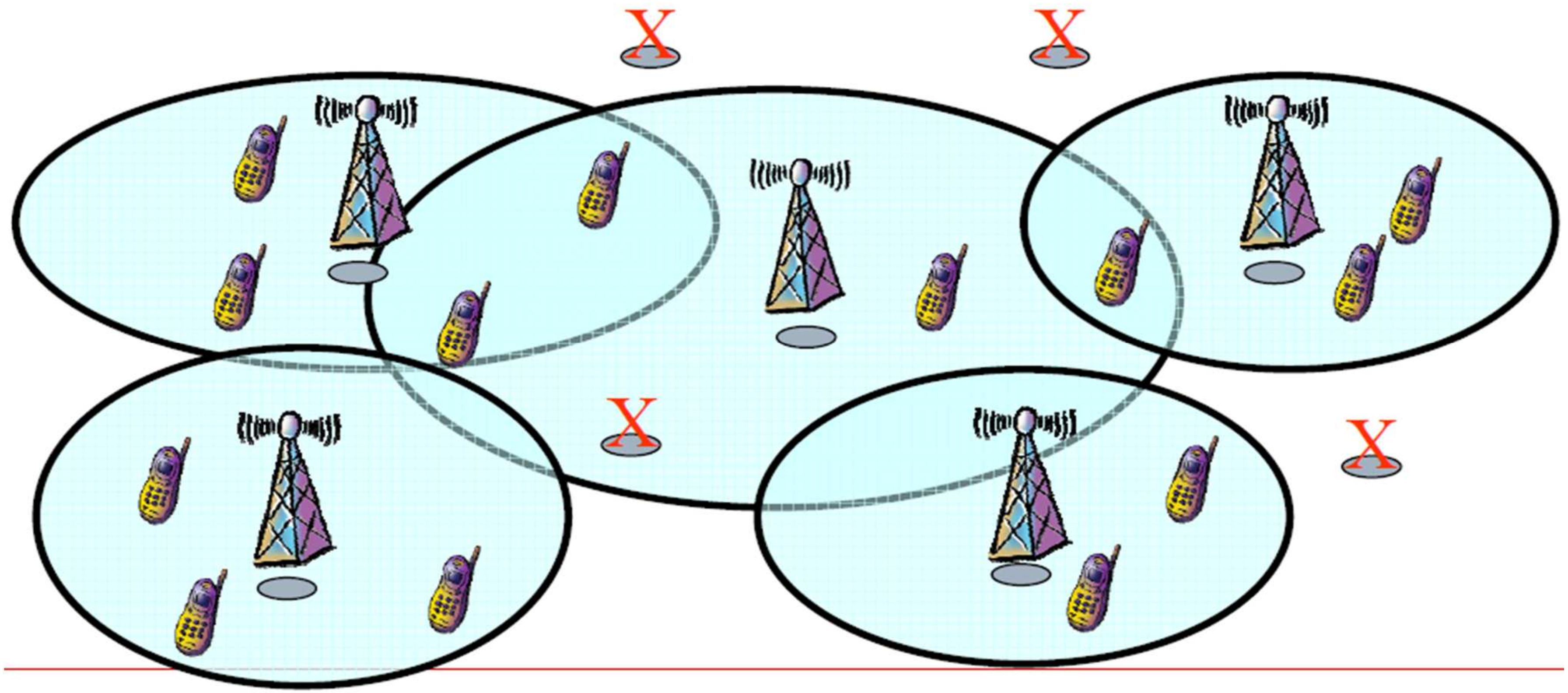


What is radio planning?

- When we have to install a new wireless network or extend an existing one into a new area, we need to design the fixed and the radio parts of the network. This phase is called ***radio planning***.
- The basic decisions that must be taken during the radio planning phase are:
  - Where to install base stations (or access points, depending on the technology)
  - How to configure base stations (antenna type, height, sectors orientation, maximum power, device capacity, etc.)

What is radio planning?

- The basic decisions that must be taken during the radio planning phase are:
  - Where to install base stations (or access points, depending on the technology)
  - How to configure base stations (antenna type, height, sectors orientation, maximum power, device capacity, etc.)



Antenna positioning

- The selection of possible antenna sites depends on several technical (traffic density and distribution, ground morphology, etc.) and non-technical (electromagnetic pollution, local authority rules, agreements with building owners, etc.) issues.

- We denote with S the set of Css

- We can assume that the channel gain g_{ij} between TP i and CS j is provided by a propagation prediction tool.

Antenna positioning

- The antenna configuration affects the signal level received at TPs
- For each CS j we can define a set of possible antenna configurations K_j
- We can assume that the channel gain g_{ijk} between TP i and CS j depends also on configuration k.
- Based on signal quality requirement and channel gain we can evaluate if TP i can be covered by CS j with an antenna with configuration k, and define coefficients:

$$a_{ijk} = \begin{cases} 1 & \text{if TP } i \text{ can be covered by CS } j \text{ with conf. } k \\ 0 & \text{otherwise} \end{cases}$$

Coverage planning

- The goal of the coverage planning is to:
  – Select where to install base stations
  – Select antenna configurations
- To ensure that the signal level in all TPs is high enough to guarantee a good communication quality
- *Note that interference is not considered here*

Decision variables and parameters

- *Decision variables:*
- y_{jk}: 0-1 decision variable that indicates if a base station with configuration k is installed in CS j

- *Installation cost:*
- c_{jk}:cost related to the installation of a base station in CS j with configuration k

Set covering problem

$$\min \sum_{j\in S}\sum_{k\in K_j} c_{jk} y_{jk}$$

Objective function: Total cost

$$\sum_{j\in S}\sum_{k\in K_j} a_{ijk} y_{jk} \geq 1 \quad \forall i \in I$$

Full coverage constraints

$$\sum_{k\in K_j} y_{jk} \leq 1 \quad \forall j \in S$$

One configuration per site

$$y_{jk} \in \{0,1\} \quad \forall j \in S, k \in K_j$$

Integrality constraints