Modeling, Analysis and Optimization of Networks (Part 2: design)

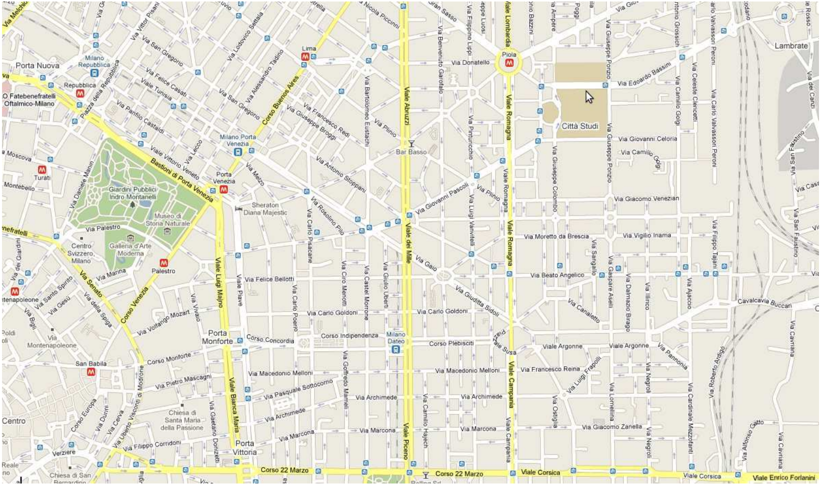
Lecture 3

Alberto Ceselli
alberto.ceselli@unimi.it

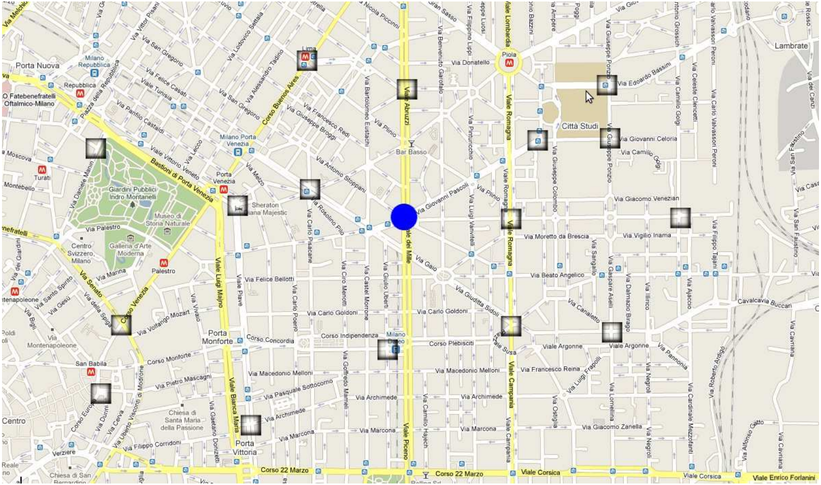
Dipartimento di Informatica
Università degli Studi di Milano

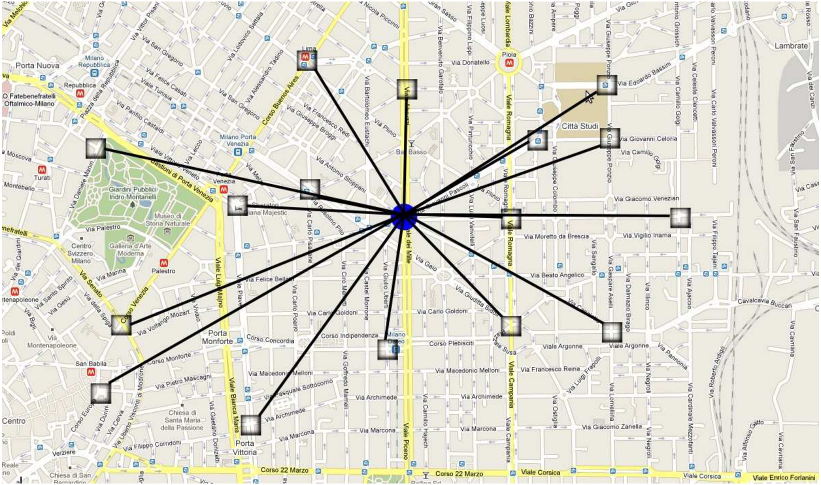
March 29, 2017

Connecting nodes on a network

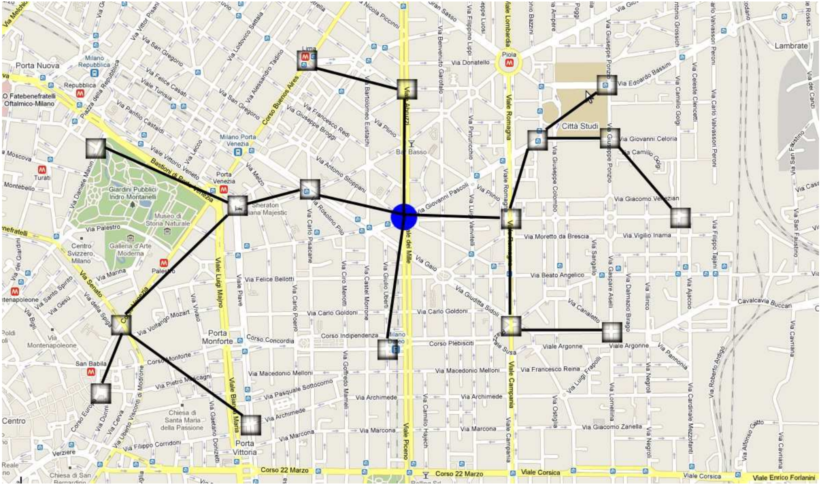


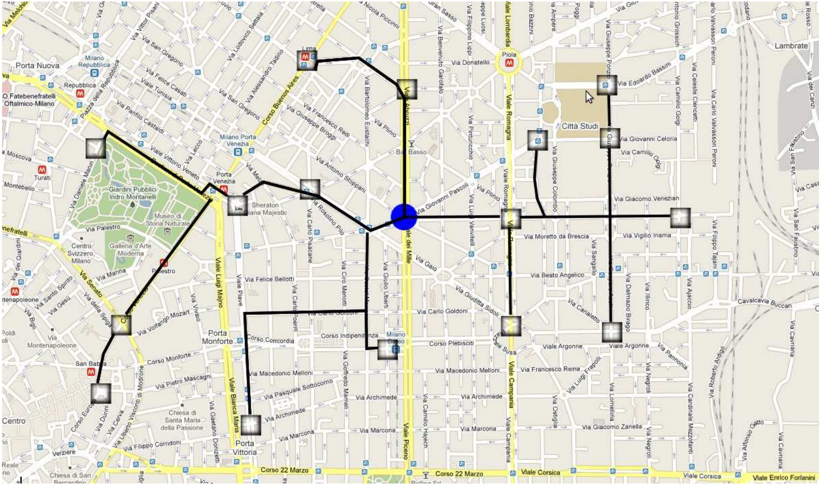
Connecting nodes on a network




Porta Nuova
Repubblica
Fatebenefratelli
Oftalmico-Milano
Moscova
Turati
Montebello
Giardini Pubblici Indro Montanelli
Museo di Storia Naturale
Galleria d'Arte Moderna
Palestro
Centro Svizzero Milano
Milano Porta Venezia
Venezia
Lima
Piola
Città Studi
Lambrate
Porta Monforte
San Babila
Chiesa di Santa Maria della Passione
Porta Vittoria
Centro
Corso Buenos Aires
Viale Lombardia
Viale Romagna
Viale dei Mille
Viale Piceno
Viale Campania
Viale Argonne
Corso Plebisciti
Corso Indipendenza
Corso Concordia
Corso Monforte
Corso Venezia
Corso 22 Marzo
Viale Corsica
Viale Enrico Forlanini
Cavalcavia Buccari
Viale Bianca Maria
Viale Luigi Majno
Via Carlo Goldoni
Via Archimede
Via Marcona
Via Macedonio Melloni
Via Giovanni Celoria
Via Giacomo Venezian
Via Vigilio Inama
Via Beato Angelico
Via Moretto da Brescia
Via Francesco Reina
Via Giacomo Zanella
Via Edoardo Bassini
Via Donatello
Bastioni di Porta Venezia

Connecting nodes on a network




Porta Nuova
Repubblica
Lambrate
Piola
Città Studi
Giardini Pubblici Indro Montanelli
Museo di Storia Naturale
Galleria d'Arte Moderna
Porta Venezia
Porta Monforte
Porta Vittoria
Corso Buenos Aires
Viale Abruzzi
Viale Romagna
Viale Lombardia
Viale dei Mille
Viale Piceno
Viale Campania
Viale Argonne
Viale Corsica
Viale Enrico Forlanini
Corso 22 Marzo
Corso Indipendenza
Corso Concordia
Corso Plebisciti
Corso Monforte
Via Giovanni Pascoli
Via Giacomo Venezian
Via Giovanni Celoria
Via Edoardo Bassini
Via Moretto da Brescia
Via Beato Angelico
Via Carlo Goldoni
Via Macedonio Melloni
Via Archimede
Via Marcona
Via Giacomo Zanella
Cavalcavia Buccari
Centro

The problem

Given

- a set of terminal nodes,
- a set of bridge nodes,
- a set of potential links connecting them,

I want to

- decide how to link nodes,
- in such a way that transmissions can be performed between each pair of terminal nodes,
- minimizing the network cost.

Problem features:

Given:

- A graph $G(V, E)$ (telecomunication network: $V =$ sites, $E =$ links).

Problem features:

Given:

- A graph $G(V, E)$ (telecomunication network: V = sites, E = links).
- A subset T of vertices of the graph, which correspond to terminals.
- A subset $B = V \setminus T$ of vertices of the graph, which correspond to bridges.

Problem features:

Given:

- A graph $G(V, E)$ (telecomunication network: V = sites, E = links).
- A subset T of vertices of the graph, which correspond to terminals.
- A subset $B = V \setminus T$ of vertices of the graph, which correspond to bridges.
- Each edge connecting vertices $i \in V$ and $j \in V$ has a cost c_{ij}.

Problem features:

Given:

- A graph $G(V, E)$ (telecomunication network: V = sites, E = links).
- A subset T of vertices of the graph, which correspond to terminals.
- A subset $B = V \setminus T$ of vertices of the graph, which correspond to bridges.
- Each edge connecting vertices $i \in V$ and $j \in V$ has a cost c_{ij}.
- Find a **tree** in G of minimum total cost

Problem features:

Given:

- A graph $G(V, E)$ (telecomunication network: V = sites, E = links).
- A subset T of vertices of the graph, which correspond to terminals.
- A subset $B = V \setminus T$ of vertices of the graph, which correspond to bridges.
- Each edge connecting vertices $i \in V$ and $j \in V$ has a cost c_{ij}.
- Find a **tree** in G of minimum total cost
- ...containg **all** terminals ($i \in T$) and **any subset** of the bridges ($i \in B$).

It is called the **Steiner Tree Problem (STP)** (Gauss, 1777-1855).

Solving the STP

Some considerations:

- Is it like a Minimum Spanning Tree?

Solving the STP

Some considerations:

- Is it like a Minimum Spanning Tree?
- ... (it's not): MST is polynomially solvable, STP is NP-Hard.

Solving the STP

Some considerations:

- Is it like a Minimum Spanning Tree?
- ... (it's not): MST is polynomially solvable, STP is NP-Hard.
- We'll see how to **approximate** it.

Approximation

What does it mean approximation?

Exact algorithms	a-priori guarantee of global optimality
Heuristics	no quality guarantee
Upper and lower bounds	a-posteriori quality guarantee
Approximation algorithms	a-priori quality guarantee

An α-approx algorithm **always** gives a solution of cost **at most** α times worse than the optimum.

A heuristic for the STP

Idea:

- STP asks to find a *minimum cost tree*
- ...connecting vertices in the set T
- $\rightarrow$ let's build a MST on the set T only!

A heuristic for the STP

Idea:

- STP asks to find a *minimum cost tree*
- ... connecting vertices in the set T
- $\rightarrow$ let's build a MST on the set T only!
- Good news: easy to compute.

A heuristic for the STP

Idea:

- STP asks to find a *minimum cost tree*
- ...connecting vertices in the set T
- $\rightarrow$ let's build a MST on the set T only!
- Good news: easy to compute.
- Bad news: such a tree might not be optimal (on the whiteboard, AA page 28).

A heuristic for the STP

Idea:

- STP asks to find a *minimum cost tree*
- ...connecting vertices in the set T
- $\rightarrow$ let's build a MST on the set T only!
- Good news: easy to compute.
- Bad news: such a tree might not be optimal (on the whiteboard, AA page 28).
- Question: what if an element of T exists having no neighbors in T?

The metric STP

The **metric** STP is a STP whose edge costs satisfy the **triangle inequality**: given three vertices $i, j, k \in V$

$$c_{ij} \leq c_{ik} + c_{kj}$$

Theorem: there is an approximation factor preserving reduction from the STP to the metric STP (proof on the whiteboard, AA page 27).

Approximating the metric STP

Metric STPs have a better structure:

Theorem: (for the metric STP), the cost of an MST on T is within 2-OPT (proof on the whiteboard, AA page 28).

Approximating the STP

A 2-approx algorithm for the STP is the following:

- given a STP instance on a graph G, build an (equivalent) instance of the metric STP on a graph G'
- find a MST on terminals in graph G'
- map edges of G' in this MST to edges in G

Steiner forests

Let us generalize the STP as follows: given

- A graph $G(V, E)$ (telecomunication network: V = sites, E = links).
- A subset T of vertices of the graph, which correspond to terminals.
- A subset $B = V \setminus T$ of vertices of the graph, which correspond to bridges.
- Each edge connecting vertices $i \in V$ and $j \in V$ has a cost c_{ij}.

Steiner forests

Let us generalize the STP as follows: given

- A graph $G(V, E)$ (telecomunication network: V = sites, E = links).
- A subset T of vertices of the graph, which correspond to terminals.
- A subset $B = V \setminus T$ of vertices of the graph, which correspond to bridges.
- Each edge connecting vertices $i \in V$ and $j \in V$ has a cost c_{ij}.
- A set of **connection requests** between terminals: for each pair of terminals $s, t \in T$, coefficients $r_{st} = 1$ if s and t must be connected, $r_{st} = 0$ otherwise.

Steiner forests

Let us generalize the STP as follows: given

- A graph $G(V, E)$ (telecomunication network: V = sites, E = links).
- A subset T of vertices of the graph, which correspond to terminals.
- A subset $B = V \setminus T$ of vertices of the graph, which correspond to bridges.
- Each edge connecting vertices $i \in V$ and $j \in V$ has a cost c_{ij}.
- A set of **connection requests** between terminals: for each pair of terminals $s, t \in T$, coefficients $r_{st} = 1$ if s and t must be connected, $r_{st} = 0$ otherwise.
- Find a **forest** in G of minimum total cost
- ...containing **at least one path** connecting each pair of terminals s and t having $r_{st} = 1$.

It is called the **Steiner Forest Problem (SFP)**.

Mathematical programming models

Let us consider a *cut function*: for each $S \subseteq V$

$$f(S) = \begin{cases} 1 & \text{if } S \text{ contains } s \text{ and } V \setminus S \text{ contains } t \text{ such that } r_{st} = 1 \\ 0 & \text{otherwise} \end{cases}$$

Let us consider *crossing sets*: for each $S \subseteq V$

$$\delta(S) = \text{set of edges crossing the cut } (S, V \setminus S)$$

Mathematical programming models

Primal:

$$\begin{aligned}
\text{minimize} \quad & \sum_{(i,j)\in E} c_{ij} x_{ij} & \\
\text{subject to} \quad & \sum_{(i,j)\in\delta(S)} x_{ij} \geq f(S) & \forall S \subseteq V \\
& x_{ij} \in \{0,1\} & \forall (i,j) \in E
\end{aligned}$$

Mathematical programming models

Primal:

$$\begin{aligned} \text{minimize} \quad & \sum_{(i,j)\in E} c_{ij} x_{ij} & \\ \text{subject to} \quad & \sum_{(i,j)\in\delta(S)} x_{ij} \geq f(S) & \forall S \subseteq V \\ & x_{ij} \geq 0 & \forall (i,j) \in E \end{aligned}$$

Dual:

$$\begin{aligned} \text{maximize} \quad & \sum_{S\subseteq V} f(S) y_S & \\ \text{subject to} \quad & \sum_{S:(i,j)\in\delta(S)} y_S \leq c_{ij} & \forall (i,j) \in E \\ & y_S \geq 0 & \forall S \subseteq V \end{aligned}$$

Figurative terminology

- set S *has been raised* if $y_S > 0$ (remark: it is never convenient to raise sets S having $f(S) = 0$)
- edge (i, j) *feels* dual y_S if $(i, j) \in \delta(S)$ and $y_S > 0$
- edge (i, j) *is tight* (resp overtight) if the sum of duals it feels equals (resp exceeds) its cost

Optimality conditions

Primal (slackness) conditions: for each $(i,j) \in E$,

$$x_{ij} \neq 0 \rightarrow \sum_{S:(i,j)\in\delta(S)} y_S = c_{ij}$$

Dual (slackness) conditions: for each $S \subseteq V$,

$$y_S \neq 0 \rightarrow \sum_{(i,j)\in\delta(S)} x_{ij} = 1$$

Dual (relaxed slackness) conditions: for each $S \subseteq V$,

$$y_S \neq 0 \rightarrow \sum_{(i,j)\in\delta(S)} x_{ij} \leq 2 \cdot f(S) \text{ "on the average"}$$

(every raised cut has degree at most 2).

Primal-dual algorithm

Idea:

- start with a super-optimal (infeasible) primal and a sub-optimal (feasible) dual
- iteratively improve the feasibility of the primal and the optimality of the dual, until a feasible primal is obtained
- x_{ij} vars indicate which cuts need to be raised
- y_S vars indicate which edges need to be picked

Invariant: the set of vars x_{ij} always identifies a *forest*.

Primal-dual algorithm

A key step: **unsatisfied** and **active** sets. Given a primal solution, a set S is **unsatisfied** if

- has $f(S) = 1$
- there is no picked edge crossing the cut $(S, V \setminus S)$;

a set S is **active** if

- it is unsatisfied
- it does not contain unsatisfied sets (i.e. it is minimal wrt inclusion)

Lemma: A set S is active iff it is a connected component in the currently picked forest (and $f(S) = 1$).
(proof on the whiteboard, AA page 200).

Primal-dual algorithm for SFP

Primal-dual algorithm for SFP:

- (init) $x_{ij} := 0$; $y_S := 0$
- (augmentation) while there exists an unsatisfied set S do
  find active sets (by listing connected components)
  simultaneously raise y_S for each active set S
  until some edge (ij) becomes tight
  set $x_{ij} := 1$ for each tight edge
- (pruning) for each (i, j) such that $x_{ij} = 1$, set $x_{ij} := 0$ if the primal solution remains feasible

(example on the dashboard, AA pages 202-204).

Analysis

Theorem: Primal-dual algorithm for the SFP achieves an approximation guarantee of 2.
(proof on the whiteboard, AA pages 204-206)

Tightness of the analysis

Are the analyses tight?

- try to find a STP (or SFP) instance in which our algorithms reach the worst case guarantee ...

example: page 30 AA.

Further remarks

Some final observations. Both STP and MST are special cases of SFP:

- when run on a STP instance, the primal-dual algorithm *builds a Spanning Tree* on set T
- $\rightarrow$ the MST algorithm for STP is a special case of the primal-dual;
- when run on a MST instance (i.e. $T = V$), the primal-dual algorithm is essentially Kruskal's algorithm.