Network Design and Optimization course

Lecture 11

Alberto Ceselli
alberto.ceselli@unimi.it

Dipartimento di Tecnologie dell'Informazione
Università degli Studi di Milano

December 21, 2011

Università degli Studi di Milano

The problem

Given

- a set of nodes,
- a set of links connecting them,

(that is, an existing network), I want to

- evaluate the robustness of the network
- ... with respect to some adverse event.

The problem

Example of an adverse event: two nodes are not connected anymore.

Assumptions

Let us consider

- a coefficient c_e to indicate "the robustness' of each link e
- given a set of links S, we assume that the "robustness" of the set of S is

$$\sum_{e \in S} c_e$$

and we search for the set S

- of minimum robustness
- whose removal splits the network in two

we are facing a *Global Min Cut Problem* (GMCP).

A very simple GMCP algorithm

How to solve a GMCP?

A very simple GMCP algorithm

How to solve a GMCP?

- Finding a s-t min cut can be done by flow computations

A very simple GMCP algorithm

How to solve a GMCP?

- Finding a s-t min cut can be done by flow computations
- let us consider all s-t pairs in the network
  - compute each min s-t cut (max s-t flow)
  - pick the minimum among all s-t pairs

A very simple GMCP algorithm

How to solve a GMCP?

- Finding a s-t min cut can be done by flow computations
- let us consider all s-t pairs in the network
  - compute each min s-t cut (max s-t flow)
  - pick the minimum among all s-t pairs

  running time $O(n^2 \cdot (nm + n^2 \log U))$ with preflow-push flow algorithms.

A simple GMCP algorithm

How to solve a GMCP?

A simple GMCP algorithm

How to solve a GMCP?

- Let's select a random node k

A simple GMCP algorithm

How to solve a GMCP?

- Let's select a random node k
- k is either on the "left" or on the "right" side of the cut

A simple GMCP algorithm

How to solve a GMCP?

- Let's select a random node k
- k is either on the "left" or on the "right" side of the cut
- phase 1: assume k to be on the left side: for each node t
  - compute the min k-t cut (max k-t flow)
  - pick the minimum among them

A simple GMCP algorithm

How to solve a GMCP?

- Let's select a random node k
- k is either on the "left" or on the "right" side of the cut
- phase 1: assume k to be on the left side: for each node t
  - compute the min k-t cut (max k-t flow)
  - pick the minimum among them
- phase 2: assume k to be on the right side: for each node t
  - compute the min t-k cut (max t-k flow)
  - pick the minimum among them

A simple GMCP algorithm

How to solve a GMCP?

- Let's select a random node k
- k is either on the "left" or on the "right" side of the cut
- phase 1: assume k to be on the left side: for each node t
  - compute the min k-t cut (max k-t flow)
  - pick the minimum among them
- phase 2: assume k to be on the right side: for each node t
  - compute the min t-k cut (max t-k flow)
  - pick the minimum among them

running time $O(n \cdot (nm + n^2 \log U))$ with preflow-push flow algorithms.

A simple GMCP algorithm

Orlin's slide 10.

A dedicated GMCP algorithm

Better GMCP algorithms exist:

A dedicated GMCP algorithm

Better GMCP algorithms exist:
Orlin's slides 8, 10-22

Failure free routing

Problem: how to design routing schemes that *still works even if k links (or nodes) of the network fail*?

Failure free routing

Problem: how to design routing schemes that *still works even if k links (or nodes) of the network fail*?

- edge-disjoint shortest path problems
- vertext-disjoint shortest path problems

Modeling edge-disjoint shortest path problems

Using flows to model edge-disjoint SPPs (on the whiteboard).

Modeling node-disjoint shortest path problems

Using flows to model node-disjoint SPPs (on the whiteboard).

Review of location problems

Given

- a set of nodes,
- a set of links connecting them,
- a set of **service requests**, **one for each node** of the network,
- a set of **devices**, able to provide service, to be installed in the network,

I want to

- decide where to place the service provider devices,
- decide how to satisfy service requests,
- maximizing the quality of service (e.g. minimizing delay time)

Review of location problems

Given

- a set of nodes,
- a set of links connecting them,
- a set of **service requests**, **one for each node** of the network,
- a set of **devices**, able to provide service, to be installed in the network,

I want to

- decide where to place the service provider devices,
- decide how to satisfy service requests,
- maximizing the quality of service (e.g. minimizing delay time)

in such a way that the resulting network is tolerant to faults in links or devices

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 I of vertices of the graph, which correspond to sites in which servers can be installed.
- A subset J of vertices of the graph, in which terminals are placed.

Problem features:

Given:

- A graph $G(V, E)$ (telecomunication network: $V =$ sites, $E =$ links).
- A subset I of vertices of the graph, which correspond to sites in which servers can be installed.
- A subset J of vertices of the graph, in which terminals are placed.

- Installing a server in each site $i \in I$ has a cost f_i.
- Connecting a terminal in site $j \in J$ to a server in $i \in I$ has a cost c_{ij}.

Problem features:

Given:

- A graph $G(V, E)$ (telecomunication network: $V =$ sites, $E =$ links).
- A subset I of vertices of the graph, which correspond to sites in which servers can be installed.
- A subset J of vertices of the graph, in which terminals are placed.

- Installing a server in each site $i \in I$ has a cost f_i.
- Connecting a terminal in site $j \in J$ to a server in $i \in I$ has a cost c_{ij}.

- Choose if and where to intall the servers (binary variables y_i) and how to connect terminals to servers (variables x_{ij})...

Problem features:

Given:

- A graph $G(V, E)$ (telecomunication network: $V =$ sites, $E =$ links).
- A subset I of vertices of the graph, which correspond to sites in which servers can be installed.
- A subset J of vertices of the graph, in which terminals are placed.

- Installing a server in each site $i \in I$ has a cost f_i.
- Connecting a terminal in site $j \in J$ to a server in $i \in I$ has a cost c_{ij}.

- Choose if and where to intall the servers (binary variables y_i) and how to connect terminals to servers (variables x_{ij})...
- ...in such a way that each terminal is connected to a server.

A Uncapacitated Facility Location Problem (UFLP)

$$
\begin{aligned}
\min \ & \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} & \\
\text{s.t. } & \sum_{i \in I} x_{ij} = 1 & \forall j \in J \\
& x_{ij} \leq y_i & \forall i \in I, \forall j \in J \\
& x_{ij} \in \{0, 1\} & \forall i \in I, \forall j \in J \\
& y_i \in \{0, 1\} & \forall i \in I
\end{aligned}
$$

N.B. variables x_{ij} take integer values as soon as they are constrained to be non-negative.

A Uncapacitated Facility Location Problem (UFLP)

$$
\begin{aligned}
\min \; & \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} & \\
\text{s.t.} \; & \sum_{i \in I} x_{ij} = \; \mathbf{2} & \forall j \in J \\
& x_{ij} \leq y_i & \forall i \in I, \forall j \in J \\
& x_{ij} \in \{0, 1\} & \forall i \in I, \forall j \in J \\
& y_i \in \{0, 1\} & \forall i \in I
\end{aligned}
$$

N.B. variables x_{ij} take integer values as soon as they are constrained to be non-negative.

A constructive heuristic for the UFLP

No polynomial type algorithm is known for the UFLP: for large scale problems we can use *heuristics*

- (construction) build a feasible UFLP solution
- (improvement) iteratively change the structure of the initial solution through *local search*.

Greedy construction

An example of initial solution:

- fix a parameter k
- define pseudo opening costs $\tilde{f}_i$
- install k servers in the k sites of minimum pseudo opening cost
- connect each terminal to the nearest server

Greedy construction

An example of initial solution:

- fix a parameter k
- define pseudo opening costs $\tilde{f}_i$
- install k servers in the k sites of minimum pseudo opening cost
- connect each terminal to the nearest server

How to choose k?

$$k = \lceil \frac{\sum_{i \in I, j \in J} c_{ij}}{\sum_{i \in I} f_i} \rceil$$

How to choose $\tilde{f}_i$?

$$\tilde{f}_i = f_i + \frac{|J|}{k} \cdot \frac{\sum_{j \in J} c_{ij}}{|J|}$$

Location-allocation procedure

An example of alternating heuristic:

- (location) for each cluster of terminals, move the server to the site of the cluster having minimum cost
- (allocation) assign each terminal to the nearest server
- and iterate ...

(for instance, this procedure is the basis of k-means clustering)

ADD procedure

Classical ADD move:

- try to open a new server in each site i
- assign to its cluster each terminal having i as the site with the nearest server
- if the solution is not improved, backtrack the move

DROP procedure

Classical DROP move:

- try to close in turn each open server
- assign unassigned terminal to its nearest open server
- if the solution is not improved, backtrack the move

A Local Search (LS) framework:

1. perform greedy construction,
2. iteratively perform location-allocation procedure, until no further improving changes can be made,
3. iteratively perform ADD moves, until no further improving ADD can be made,
4. iteratively perform DROP moves, until no further improving DROP can be made,
5. if any improving move was made, go back to step 2, otherwise stop.

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 I of vertices of the graph, which correspond to sites in which servers can be installed.
- A subset J of vertices of the graph, in which terminals are placed.

Problem features:

Given:

- A graph $G(V, E)$ (telecomunication network: $V =$ sites, $E =$ links).
- A subset I of vertices of the graph, which correspond to sites in which servers can be installed.
- A subset J of vertices of the graph, in which terminals are placed.

- Installing a server in each site $i \in I$ has a cost f_i.
- Connecting a terminal in site $j \in J$ to a server in $i \in I$ has a cost c_{ij}.
- **Each terminal has a service request d_j; each server has a service capacity Q_i.**

Problem features:

Given:

- A graph $G(V, E)$ (telecomunication network: V = sites, E = links).
- A subset I of vertices of the graph, which correspond to sites in which servers can be installed.
- A subset J of vertices of the graph, in which terminals are placed.

- Installing a server in each site $i \in I$ has a cost f_i.
- Connecting a terminal in site $j \in J$ to a server in $i \in I$ has a cost c_{ij}.
- **Each terminal has a service request d_j; each server has a service capacity Q_i.**
- Choose if and where to intall the servers (binary variables y_i) and how to connect terminals to servers (variables x_{ij})...

Problem features:

Given:

- A graph $G(V, E)$ (telecomunication network: V = sites, E = links).
- A subset I of vertices of the graph, which correspond to sites in which servers can be installed.
- A subset J of vertices of the graph, in which terminals are placed.
- Installing a server in each site $i \in I$ has a cost f_i.
- Connecting a terminal in site $j \in J$ to a server in $i \in I$ has a cost c_{ij}.
- **Each terminal has a service request d_j; each server has a service capacity Q_i.**
- Choose if and where to intall the servers (binary variables y_i) and how to connect terminals to servers (variables x_{ij})...
- ...in such a way that each terminal is connected to a server.
- **...and the service requests associated to each server do not exceed its capacity.**

A Single-Source Capacitated Facility Location Problem (SS-CFLP)

$$
\begin{aligned}
\min\ & \sum_{i\in I} f_i y_i + \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} & \\
\text{s.t.}\ & \sum_{i\in I} x_{ij} = 1 & \forall j \in J \\
& \sum_{j\in J} d_j x_{ij} \leq Q_i & \forall j \in J \\
& x_{ij} \leq y_i & \forall i \in I, \forall j \in J \\
& x_{ij} \in \{0,1\} & \forall i \in I, \forall j \in J \\
& y_i \in \{0,1\} & \forall i \in I
\end{aligned}
$$

A Single-Source Capacitated Facility Location Problem (SS-CFLP)

$$\begin{aligned}
\min\ & \sum_{i\in I} f_i y_i + \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} & \\
\text{s.t.}\ & \sum_{i\in I} x_{ij} = \ \mathbf{2} & \forall j \in J \\
& \sum_{j\in J} d_j x_{ij} \le Q_i & \forall j \in J \\
& x_{ij} \le y_i & \forall i \in I, \forall j \in J \\
& x_{ij} \in \{0,1\} & \forall i \in I, \forall j \in J \\
& y_i \in \{0,1\} & \forall i \in I
\end{aligned}$$

k-SWAP procedure

k-SWAP means:

- disconnect k terminals from their servers (update residual capacities)
- connect them back in the best possible way

The VNS method

A Variable Neighborhood Search (VNS) procedure:

- perform greedy construction,
- iteratively perform Location-Allocation, until no further improving changes can be made,
- perform ADD: if an improvement is made go to 2, otherwise go to 4
- perform DROP: if an improvement is made go to 2, otherwise go to 5
- perform 2-SWAP: if an improvement is made go to 2, otherwise go to 5
- perform 3-SWAP: if an improvement is made go to 2, otherwise stop

Review of multicommodity flows

Let $f^k_{(i,j)}$ be *decision variables* representing the *amount of flow for commodity* k sent on arc (i,j). Let v represent the total cost of routing packets in the network.

$$\textbf{minimize } v = \sum_{(i,j)\in A} \sum_{k \in K} c^k_{(i,j)} f^k_{(i,j)}$$

$$\textbf{subject to } \sum_{(i,j)\in A} f^k_{(i,j)} = \sum_{(j,i)\in A} f^k_{(j,i)} + b^k_i \qquad \forall i \in V; \quad \forall k \in K$$

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

$$0 \leq f^k_{(i,j)} \leq u_{(i,j)} \qquad \forall (i,j) \in A \quad \forall k \in K$$

Review of multicommodity flows

Let $f^k_{(i,j)}$ be *decision variables* representing the *amount of flow for commodity* k sent on arc (i,j). Let v represent the total cost of routing packets in the network. **Let $x_{(i,j)}$ be decision variables taking value 1 if link (i,j) is installed, 0 otherwise.**

$$
\begin{aligned}
\textbf{minimize } & v = \sum_{(i,j)\in A} \sum_{k\in K} c^k_{(i,j)} f^k_{(i,j)} + \sum_{\mathbf{(i,j)\in A}} \mathbf{d_{(i,j)} x_{(i,j)}} \\
\textbf{subject to } & \sum_{(i,j)\in A} f^k_{(i,j)} = \sum_{(j,i)\in A} f^k_{(j,i)} + b^k_i && \forall i \in V;\quad \forall k \in K \\
& \sum_{k\in K} f^k_{(i,j)} \le u_{(i,j)} \mathbf{x_{(i,j)}} && \forall (i,j) \in A \\
& 0 \le f^k_{(i,j)} \le u_{(i,j)} && \forall (i,j) \in A \quad \forall k \in K \\
& \mathbf{x_{(i,j)}} \in \{\mathbf{0}, \mathbf{1}\} && \forall (i,j) \in A;\quad \forall k \in K
\end{aligned}
$$

Link dimensioning

Let $f^k_{(i,j)}$ be *decision variables* representing the *amount of flow for commodity* k sent on arc (i,j). Let v represent the total cost of routing packets in the network. Let $x_{(i,j)}$ be *decision variables* taking value 1 if link (i,j) is installed, 0 otherwise.

$$\text{minimize } v = \sum_{(i,j)\in A} \sum_{k\in K} c^k_{(i,j)} f^k_{(i,j)} + \sum_{(i,j)\in A} \mathbf{d}_{\mathbf{(i,j)}} \mathbf{x}_{\mathbf{(i,j)}}$$

$$\text{subject to } \sum_{(i,j)\in A} f^k_{(i,j)} = \sum_{(j,i)\in A} f^k_{(j,i)} + b^k_i \qquad \forall i \in V; \forall k \in K$$

$$\sum_{k\in K} f^k_{(i,j)} \leq \mathbf{u}_{\mathbf{(i,j)}} \mathbf{x}_{\mathbf{(i,j)}} \qquad \forall (i,j) \in A \forall \ell \in L$$

$$0 \leq f^k_{(i,j)} \leq \mathbf{u}_{\mathbf{(i,j)}} \qquad \forall (i,j) \in A; \forall k \in K$$

$$x_{(i,j)} \in \{0,1\} \qquad \forall (i,j) \in A; \forall \ell \in L$$

Link dimensioning

Let $f^k_{(i,j)}$ be *decision variables* representing the *amount of flow for commodity* k sent on arc (i,j). Let v represent the total cost of routing packets in the network. Let $x_{(i,j)}$ be *decision variables* taking value 1 if link (i,j) is installed, 0 otherwise.

$$\text{minimize } v = \sum_{(i,j)\in A} \sum_{k\in K} c^k_{(i,j)} f^k_{(i,j)} + \sum_{(i,j)\in A} \sum_{\ell\in \mathbf{L}} \mathbf{d}^{\ell}{}_{(\mathbf{i},\mathbf{j})} \mathbf{x}^{\ell}{}_{(\mathbf{i},\mathbf{j})}$$

$$\text{subject to } \sum_{(i,j)\in A} f^k_{(i,j)} = \sum_{(j,i)\in A} f^k_{(j,i)} + b^k_i \qquad \forall i \in V; \forall k \in K$$

$$\sum_{k\in K} f^k_{(i,j)} \leq \sum_{\ell\in\mathbf{L}} \mathbf{u}^{\ell}{}_{(\mathbf{i},\mathbf{j})} \mathbf{x}^{\ell}{}_{(\mathbf{i},\mathbf{j})} \qquad \forall (i,j) \in A \forall \ell \in L$$

$$0 \leq f^k_{(i,j)} \leq \sum_{\ell\in\mathbf{L}} \mathbf{u}^{\ell}{}_{(\mathbf{i},\mathbf{j})} \qquad \forall (i,j) \in A; \forall k \in K$$

$$x^{\ell}{}_{(i,j)} \in \{0,1\} \qquad \forall (i,j) \in A; \forall \ell \in L$$

Handling protection schemes

Let us assume that each connection request from s to t is modeled as a commodity k by setting $b^k_s = 1$ and $b^k_t = -1$. In this case $f^k_{(i,j)}$ is always 0 or 1.

$$
\begin{aligned}
\text{minimize } v = & \sum_{(i,j)\in A}\sum_{k\in K} c^k_{(i,j)} f^k_{(i,j)} + \sum_{(i,j)\in A}\sum_{\ell\in L} d^\ell_{(i,j)} x^\ell_{(i,j)} & \\
\text{subject to } & \sum_{(i,j)\in A} f^k_{(i,j)} = \sum_{(j,i)\in A} f^k_{(j,i)} + b^k_i & \forall i \in V; \forall k \in K \\
& \sum_{k\in K} f^k_{(i,j)} \leq \sum_{\ell\in L} u^\ell_{(i,j)} x^\ell_{(i,j)} & \forall (i,j) \in A \forall \ell \in L \\
& 0 \leq f^k_{(i,j)} \leq \sum_{\ell\in L} u^\ell_{(i,j)} & \forall (i,j) \in A; \forall k \in K \\
& x^\ell_{(i,j)} \in \{0,1\} & \forall (i,j) \in A; \forall \ell \in L
\end{aligned}
$$

Handling protection schemes

Let us assume that each connection request from s to t is modeled as a commodity k by setting $b_s^k = 1$ and $b_t^k = -1$. In this case $f_{(i,j)}^k$ is always 0 or 1. Consider two commodities for each connection request from s to t: $k1(s,t) \rightarrow$ the actual connection, and $k2(s,t) \rightarrow$ the backup path.

$$\text{minimize } v = \sum_{(i,j)\in A} \sum_{k\in K} c_{(i,j)}^k f_{(i,j)}^k + \sum_{(i,j)\in A} \sum_{\ell\in L} d_{(i,j)}^\ell x_{(i,j)}^\ell$$

$$\text{subject to } \sum_{(i,j)\in A} f_{(i,j)}^k = \sum_{(j,i)\in A} f_{(j,i)}^k + b_i^k \qquad \forall i \in V; \forall k \in K$$

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

$$0 \leq f_{(i,j)}^k \leq \sum_{\ell\in L} u_{(i,j)}^\ell \qquad \forall (i,j) \in A; \forall k \in K$$

$$x_{(i,j)}^\ell \in \{0,1\} \qquad \forall (i,j) \in A; \forall \ell \in L$$

Handling protection schemes

Let us assume that each connection request from s to t is modeled as a commodity k by setting $b_s^k = 1$ and $b_t^k = -1$. In this case $f_{(i,j)}^k$ is always 0 or 1. Consider two commodities for each connection request from s to t: $k1(s,t) \rightarrow$ the actual connection, and $k2(s,t) \rightarrow$ the backup path.

$$
\begin{aligned}
\text{minimize } v = & \sum_{(i,j)\in A} \sum_{k\in K} c_{(i,j)}^k f_{(i,j)}^k + \sum_{(i,j)\in A} \sum_{\ell\in L} d_{(i,j)}^\ell x_{(i,j)}^\ell \\
\text{subject to } & \sum_{(i,j)\in A} f_{(i,j)}^k = \sum_{(j,i)\in A} f_{(j,i)}^k + b_i^k && \forall i \in V; \forall k \in K \\
& \sum_{k\in K} f_{(i,j)}^k \leq \sum_{\ell\in L} u_{(i,j)}^\ell x_{(i,j)}^\ell && \forall (i,j) \in A \forall \ell \in L \\
& 0 \leq f_{(i,j)}^k \leq \sum_{\ell\in L} u_{(i,j)}^\ell && \forall (i,j) \in A; \forall k \in K \\
& \mathbf{f_{(i,j)}^{k1(s,t)} + f_{(i,j)}^{k2(s,t)} \leq 1} && \forall \text{connection} \\
& x_{(i,j)}^\ell \in \{0,1\} && \forall (i,j) \in A; \forall \ell \in L
\end{aligned}
$$

Università degli Studi di Milano

Exercises

- Implement the min global cut algorithm
- Try the link / node protected shortest paths model
- implement LS for UFLP
- implement VNS for SS-CFLP