Network Design and Optimization course

Lecture 7

Alberto Ceselli
alberto.ceselli@unimi.it

Dipartimento di Tecnologie dell'Informazione
Università degli Studi di Milano

November 16, 2011

Università degli Studi di Milano

The problem

We are now able to cope with costs and capacities at the same time (Min Cost Flows). Given

- a set of nodes,
- a set of links connecting them,
- a **set** of connection requests, between **pairs of nodes** of the network,

that is, an existing network, with realistic traffic.
I want to

- decide which links to use in the connections (route)
- maximizing the quality of service (e.g. minimizing delay time)

Assumptions

Some assumptions:

1. cost matters,
2. the capacity of links may not be enough for all connection requests,
3. different connections can be routed on the same links ...
4. ... provided the capacity of each link is enough.

Recognizing a known problem ...

We observe

- when a single connection request is made on the network, the problem is to compute a Min Cost Flow ...

we are facing a *multicommodity Min Cost Flow (MCF) problem*.

Application Areas

See Orlin's slides, 22-7

Graph model

Given a network, build a *directed* graph $G = (V, A)$ having

- one vertex $i \in V$ for each node of the network,
- one arc $a \in A \subseteq V \times V$ for each link of the network,
- capacities $u_{(i,j)}$ on each arc $(i, j) \in A$.

Then, consider the set K of connection requests (commodities), and enrich the graph with

- costs $c^k_{(i,j)}$ on each arc $(i, j) \in A$ for each commodity $k \in K$,
- flow excess b^k_i for each node $i \in V$ and for each commodity $k \in K$.

Assumptions

We assume that

- **Homogeneous commodities:** each unit of flow of uses one unit of capacity on each arc, independently of k,
- **No congestion:** cost is linear in the amount of flow on each arc (until capacity limit is reached),
- **Fractional flows:** no integrality condition is imposed on flows.

WLOG we assume also that

- $b_i^k > 0$ for a unique $i \in V$ (origin of commodity $k \rightarrow s_k$),
- $b_i^k < 0$ for a unique $i \in V$ (destination of commodity $k \rightarrow t_k$).

We search for k min cost flows on the network, one for each commodity.

Mathematical Programming model

Let $x^k_{(i,j)}$ 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.

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

$$\text{subject to } \sum_{j \in V} x^k_{(i,j)} = \sum_{k \in V} x^k_{(k,i)} + b^k_i \quad \forall i \in V, i \neq s,t; \quad \forall k \in K$$

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

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

An example of multicommodity flow

See Orlin's slides 22,3-4

Discrete and fractional flows

See Orlin's slides 22,8-10

Ways of solving MMCF

There are many ways of solving MMCF:

- price (cost) directed decompositions,
- resource (capacity) directed decompositions,
- simplex based approaches.

Price directed decompositions

Idea behind price directed decompositions:

- modify costs on arcs ...
- ... such that solving k MCF independently gives a full MMCF solution ...
- ... that automatically satisfies capacity constraints.

We will see:

- Lagrangean relaxation,
- column generation.

Optimality conditions: partial dualization

Theorem: The multicommodity flow (x_{ij}^k) is *optimal* if there exist non-negative prices (w_{ij}) on the arcs, so that the following is true:

- if $w_{ij} > 0$ then $\sum_{k \in K} x_{ij}^k = u_{ij}$,
- each flow k is (independently) optimal for commodity k if each cost c_{ij}^k is replaced by

$$c_{ij}^{w,k} = c_{ij}^k + w_{ij}$$

.

Recall: flow k is optimal for commodity k if there is no negative cost cycle in the residual network for commodity k.

See Orlin's slides 22,14-16

Lagrangean algorithm for MMCF

Idea: update w and solve MCF until the partial dualization conditions are satisfied.

Lagrangean algorithm for MMCF

BEGIN
$x := 0; w := 0$
$\theta := 1$
while **partial dualization optimality conditions** are not satisfied
begin
    set $c_{ij}^{w,k} := c_{ij}^k + w_{ij}$ for each $k \in K$ and for each $(i,j) \in A$
    for each $k \in K$
        build a residual network $G^k(x)$
        solve a MCF problem on $G^k(x)$ using costs $c_{ij}^{w,k}$
        obtain a flow x_{ij}^k
    update prices w: for each $(i,j) \in A$
        $w_{ij} := \max\{0, w_{ij} + \theta \cdot (\sum_{k \in K} x_{ij}^k - u_{ij})\}$
    reduce θ
end
END

Solving MMCF by Lagrangean relaxation

See Orlin's slides 22,21-28

A path-based model

Idea: represent overall flow as sum of partial flows, each following a single path, and combine them in a feasible way.

A path-based model

$$\begin{aligned}
\text{minimize } v = & \sum_{k \in K} \sum_{P \in \mathcal{P}^k} c^P \cdot f^P & \\
\text{subject to } & \sum_{k \in K} \sum_{P \in \mathcal{P}^k} a_{ij}^P \cdot f^P \leq u_{ij} & \forall (i,j) \in A \\
& \sum_{P \in \mathcal{P}^k} f^P = b_{s_k}^k & \forall k \in K \\
& f^P \geq 0 & \forall k \in K, \forall P \in \mathcal{P}^k
\end{aligned}$$

where:

- $\mathcal{P}^k$ is the set of *all* paths from s_k to t_k
- c^P is the cost of path P
- f^P is the amount of flow sent on path P
- $a_{ij}^P = 1$ if path P includes arc (i,j), and $= 0$ otherwise

A path-based model

Orlin's slides 23,8-10

A path-based model

$$\text{minimize } v = \sum_{k \in K} \sum_{P \in \mathcal{P}^k} c^P \cdot f^P$$

$$\text{subject to } \sum_{k \in K} \sum_{P \in \mathcal{P}^k} a_{ij}^P \cdot f^P \leq u_{ij} \qquad \forall (i,j) \in A$$

$$\sum_{P \in \mathcal{P}^k} f^P = b_{s_k}^k \qquad \forall k \in K$$

$$f^P \geq 0 \qquad \forall k \in K, \forall P \in \mathcal{P}^k$$

Is it possible to straightly optimize it?

A path-based model

$$\begin{aligned}
\text{minimize } v = \sum_{k\in K} \sum_{P\in\mathcal{P}^k} c^P \cdot f^P & \\
\text{subject to } \sum_{k\in K} \sum_{P\in\mathcal{P}^k} a_{ij}^P \cdot f^P \leq u_{ij} & \qquad \forall (i,j) \in A \\
\sum_{P\in\mathcal{P}^k} f^P = b_{s_k}^k & \qquad \forall k \in K \\
f^P \geq 0 & \qquad \forall k \in K, \forall P \in \mathcal{P}^k
\end{aligned}$$

Is it possible to straightly optimize it?
$|\mathcal{P}^k|$ grows combinatorially with problem dimension: we need an iterative approach (column generation).

A path-based model

$$
\begin{aligned}
\text{minimize } v = & \sum_{k \in K} \sum_{P \in \mathcal{P}^k} c^P \cdot f^P \\
\text{subject to } & \sum_{k \in K} \sum_{P \in \mathcal{P}^k} a_{ij}^P \cdot f^P \leq u_{ij} && \forall (i,j) \in A \\
& \sum_{P \in \mathcal{P}^k} f^P = b_{s_k}^k && \forall k \in K \\
& f^P \geq 0 && \forall k \in K, \forall P \in \mathcal{P}^k
\end{aligned}
$$

Idea: in a good MMCF solution, only *very few good paths* are chosen.

A path-based model

$$\text{minimize } v = \sum_{k \in K} \sum_{P \in \mathcal{S}^k} c^P \cdot f^P$$

$$\text{subject to } \sum_{k \in K} \sum_{P \in \mathcal{S}^k} a_{ij}^P \cdot f^P \leq u_{ij} \qquad \forall (i,j) \in A$$

$$\sum_{P \in \mathcal{S}^k} f^P = b_{s_k}^k \qquad \forall k \in K$$

$$f^P \geq 0 \qquad \forall k \in K, \forall P \in \mathcal{S}^k$$

Idea: in a good MMCF solution, only *very few good paths* are chosen.
We replace each $\mathcal{P}^k$ with a "well chosen" subset $\mathcal{S}^k \subset \mathcal{P}^k$
If we are lucky, all useful paths are in $\mathcal{S}^k$, otherwise we iteratively enlarge it.

A path-based model

$$\text{minimize } v = \sum_{k \in K} \sum_{P \in \mathcal{P}^k} c^P \cdot f^P$$

$$\text{subject to } \sum_{k \in K} \sum_{P \in \mathcal{P}^k} -a_{ij}^P \cdot f^P \geq -u_{ij} \qquad \forall (i,j) \in A (\lambda_{ij})$$

$$\sum_{P \in \mathcal{P}^k} f^P = b_{s_k}^k \qquad \forall k \in K (\mu_k)$$

$$f^P \geq 0 \qquad \forall k \in K, \forall P \in \mathcal{P}^k$$

This is a Linear Programming model:

- let $\lambda_{ij} \geq 0$ and μ_k be the *dual variables*
- the *reduced cost* of each variable f^P is $\bar{c}^P := c^P - \sum_{(i,j) \in A} (-\lambda_{ij} \cdot a_{ij}^P) - \mu_k$
- ... that is $\bar{c}^P := \sum_{(i,j) \in A} (c_{ij}^k + \lambda_{ij}) \cdot a_{ij}^P - \mu_k$
- a solution f is optimal if all variables have *non-negative reduced cost*

A path-based model

$$\begin{aligned}
\text{minimize } v = & \sum_{k \in K} \sum_{P \in \mathcal{P}^k} c^P \cdot f^P \\
\text{subject to } & \sum_{k \in K} \sum_{P \in \mathcal{P}^k} -a_{ij}^P \cdot f^P \geq -u_{ij} && \forall (i,j) \in A (\lambda_{ij}) \\
& \sum_{P \in \mathcal{P}^k} f^P = b_{s_k}^k && \forall k \in K (\mu_k) \\
& f^P \geq 0 && \forall k \in K, \forall P \in \mathcal{P}^k
\end{aligned}$$

This is a Linear Programming model:

- I am free to replace each $\mathcal{P}^k$ with a subset $\mathcal{S}^k$, and optimize a *restricted* problem
- then, I search for the variable having *most negative reduced cost*
    - if even this reduced cost is non-negative, then all of them are
    - otherwise, the variable corresponds to a good path ...
    - I enlarge an $\mathcal{S}^k$ with this path, and iterate.

A path-based model

$$\begin{aligned}
\text{minimize } v = & \sum_{k \in K} \sum_{P \in \mathcal{P}^k} c^P \cdot f^P \\
\text{subject to } & \sum_{k \in K} \sum_{P \in \mathcal{P}^k} -a_{ij}^P \cdot f^P \geq -u_{ij} && \forall (i,j) \in A (\lambda_{ij}) \\
& \sum_{P \in \mathcal{P}^k} f^P = b_{s_k}^k && \forall k \in K (\mu_k) \\
& f^P \geq 0 && \forall k \in K, \forall P \in \mathcal{P}^k
\end{aligned}$$

This is a Linear Programming model: Very good news:

- searching for the variable with most negative reduced cost is like searching for a *minimum cost s-t path*

Column Generation Algorithm for MMCF

BEGIN
Initialize $\mathcal{S}^k$
do
    solve the **restricted** LP model, considering $\mathcal{S}^k$
    get the values of dual variables $\lambda_{ij} \geq 0$ and μ_k
    for each $k \in K$
        find a **shortest path** on G using (red.) costs
            $\bar{c}_{ij} = c_{ij}^k + \lambda_{ij}$
        obtain a path P of (reduced) cost $\bar{c}^P$
        if $\bar{c}^P - \mu_k < 0$, add P to $\mathcal{S}^k$
while (*any new path has been added to* $\mathcal{S}^k$)
END

Column Generation Example

Orlin's slides 23,21-31

MMCF lab session

Implementing Lagrangean Relaxation and Column Generation MMCF algorithms in AMPL.