Planning and Optimizing
Large Scale Telecommunication Networks

CO@Work
October 5, 2009

ZIB

Maren Martens
Andreas Bley, Christian Raack
A Telecommunication Network
A Telecommunication Network
A Telecommunication Network
A Telecommunication Network
Planning Problems

Access Networks:
- How to connect customers to given routers?

Core Networks:
- Which links to use between routers?

Survivability:
- How to avoid interruptions?

Demands:
- Given customer demands, how to route them?
Access Networks
The Minimum Cost Flow Problem (MCFP)

**Given:** Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{R}_0^+$ ($v \in V$) with $\sum_{v \in V} b_v = 0$, arc cost $c_a$ and capacities $u_a$ ($a \in A$)
The Minimum Cost Flow Problem (MCFP)

**Given:** Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{R}_0^+$ ($v \in V$) with $\sum_{v \in V} b_v = 0$, arc cost $c_a$ and capacities $u_a$ ($a \in A$)
The Minimum Cost Flow Problem (MCFP)

**Given:** Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{R}_0^+$ $(v \in V)$ with $\sum_{v \in V} b_v = 0$, arc cost $c_a$ and capacities $u_a$ $(a \in A)$

Find: Flow $f : A \to \mathbb{R}_0^+$ of minimum cost, satisfying all demands
The Minimum Cost Flow Problem (MCFP)

Given: Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{R}_0^+$ ($v \in V$) with $\sum_{v \in V} b_v = 0$, arc cost $c_a$ and capacities $u_a$ ($a \in A$)

Find: Flow $f : A \rightarrow \mathbb{R}_0^+$ of minimum cost, satisfying all demands
The Minimum Cost Flow Problem (MCFP)

**Given:** Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{R}_0^+$ ($v \in V$) with $\sum_{v \in V} b_v = 0$, arc cost $c_a$ and capacities $u_a$ ($a \in A$)

**Find:** Flow $f : A \to \mathbb{R}_0^+$ of minimum cost, satisfying all demands

![Diagram of the Minimum Cost Flow Problem](image)
The Minimum Cost Flow Problem (MCFP)

Given: Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{R}_0^+$ ($v \in V$) with $\sum_{v \in V} b_v = 0$, arc cost $c_a$ and capacities $u_a$ ($a \in A$)

Find: Flow $f : A \rightarrow \mathbb{R}_0^+$ of minimum cost, satisfying all demands

$$\min \sum_{a \in A} c_a f(a)$$

s.t. $$\sum_{a \in \delta^+(v)} f(a) - \sum_{a \in \delta^-(v)} f(a) = b_v \quad \forall v \in V$$

$$0 \leq f(a) \leq u_a \quad \forall a \in A$$
Access Networks and the MCFP

**Given:** Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{R}_0^+$ ($v \in V$) with $\sum_{v \in V} b_v = 0$, arc cost $c_a$ and capacities $u_a$ ($a \in A$)

**Find:** Flow $f : A \rightarrow \mathbb{R}_0^+$ of minimum cost, satisfying all demands
**Access Networks and the MCFP**

**Given:** Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{R}_0^+ (v \in V)$ with $\sum_{v \in V} b_v = 0$, arc cost $c_a$ and capacities $u_a (a \in A)$

**Find:** Flow $f : A \rightarrow \mathbb{R}_0^+$ of minimum cost, satisfying all demands

\[ \text{cost} = \text{length} \]
\[ \text{capacities} = \infty \]
Access Networks and the MCFP

**Given:** Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{R}_0^+$ ($v \in V$) with $\sum_{v \in V} b_v = 0$, arc cost $c_a$ and capacities $u_a$ ($a \in A$)

**Find:** Flow $f : A \to \mathbb{R}_0^+$ of minimum cost, satisfying all demands

\[
\text{cost} = \text{length capacities} = \infty
\]
Access Networks and the MCFP

**Given:** Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{R}_0^+$ ($v \in V$) with $\sum_{v \in V} b_v = 0$, arc cost $c_a$ and capacities $u_a$ ($a \in A$)

**Find:** Flow $f : A \rightarrow \mathbb{R}_0^+$ of minimum cost, satisfying all demands

\[
\text{cost} = \text{lengthcapacities} = \infty
\]
Access Networks and the MCFP

**Given:** Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{Z}_0^+$ ($v \in V$) with $\sum_{v \in V} b_v = 0$, arc cost $c_a$ and capacities $u_a$ ($a \in A$)

**Find:** Flow $f : A \rightarrow \mathbb{R}_0^+$ of minimum cost, satisfying all demands

$\text{cost} = \text{length} \quad \text{capacities} = \infty$
Access Networks and the MCFP

**Given:** Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{Z}_0^+$ ($v \in V$) with $\sum_{v \in V} b_v = 0$, arc cost $c_a$ and capacities $u_a$ ($a \in A$)

**Find:** Flow $f : A \rightarrow \mathbb{R}_0^+$ of minimum cost, satisfying all demands
Access Networks and the MCFP

**Given:** Digraph $D = (V, A)$, demands/supplies $b_v \in \mathbb{Z}_0^+$ ($v \in V$) with $\sum_{v \in V} b_v = 0$, arc cost $c_a$ and capacities $u_a$ ($a \in A$)

**Find:** Flow $f : A \rightarrow \mathbb{Z}_0^+$ of minimum cost, satisfying all demands

$\rightarrow f_e \text{ integral!}$
Algorithms for MCFP

\[
\begin{align*}
\min \quad & \sum_{a \in A} c_a f(a) \\
\text{s.t.} \quad & \sum_{a \in \delta^+(v)} f(a) - \sum_{a \in \delta^-(v)} f(a) = b_v \quad \forall \ v \in V \\
& 0 \leq f(a) \leq u_a \quad \forall \ a \in A \\
\rightarrow \quad & \text{Solvable in polynomial time for } f(a) \in \mathbb{R}_0^+ !
\end{align*}
\]
Algorithms for MCFP

\[
\begin{align*}
\min & \quad \sum_{a \in A} c_a f(a) \\
\text{s.t.} & \quad \sum_{a \in \delta^+(v)} f(a) - \sum_{a \in \delta^-(v)} f(a) = b_v \quad \forall \ v \in V \\
& \quad 0 \leq f(a) \leq u_a \quad \forall \ a \in A
\end{align*}
\]

→ Solvable in polynomial time for \( f(a) \in \mathbb{R}_0^+ \)!

How do we quickly find integral solutions?
Algorithms for MCFP

$$\min \sum_{a \in A} c_a f(a)$$

s.t. $$\sum_{a \in \delta^+(v)} f(a) - \sum_{a \in \delta^-(v)} f(a) = b_v \quad \forall \ v \in V$$

$$0 \leq f(a) \leq u_a \quad \forall \ a \in A$$

→ Solvable in polynomial time for $$f(a) \in \mathbb{R}_0^+$$!

How do we quickly find integral solutions?

Polynomial combinatorial algorithms:

- Minimum mean cycle cancelling algorithm
- Successive shortest path algorithm
Algorithms for MCFP

\[
\min \sum_{a \in A} c_a f(a)
\]

s.t. \[
\sum_{a \in \delta^+(v)} f(a) - \sum_{a \in \delta^-(v)} f(a) = b_v \quad \forall \ v \in V
\]

\[
0 \leq f(a) \leq u_a \quad \forall \ a \in A
\]

→ Solvable in polynomial time for \( f(a) \in \mathbb{R}^+_0 \)!

How do we quickly find integral solutions?

Polynomial combinatorial algorithms:

- Minimum mean cycle cancelling algorithm
- Successive shortest path algorithm

Both give integral optimal solution, if \( b \) and \( u \) are integral!
Core Networks
A Core Network
A Core Network

Connection Point
Steiner Trees

Definition

Given a graph $G = (V, E)$ with terminals $T \subseteq V$, a Steiner tree is a tree $S \subseteq E$ that connects all terminals in $T$. 
Steiner Trees

**Definition**

Given a graph $G = (V, E)$ with terminals $T \subseteq V$, a Steiner tree is a tree $S \subseteq E$ that connects all terminals in $T$. 

$T = \{ \text{Steiner node} \}$
Steiner Trees

Definition

Given a graph $G = (V, E)$ with terminals $T \subseteq V$, a Steiner tree is a tree $S \subseteq E$ that connects all terminals in $T$. 

$T = \{ \text{Steiner node} \}$
The Steiner Tree Problem

**Definition**

**Given:** $G = (V, E)$, terminals $T \subseteq V$, edge weights $c_e \ (e \in E)$

**Find:** Steiner tree $S \subseteq E$ of minimum weight $c(S) = \sum_{e \in S} c_e$
The Steiner Tree Problem

Definition

**Given:** \( G = (V, E) \), terminals \( T \subseteq V \), edge weights \( c_e \ (e \in E) \)

**Find:** Steiner tree \( S \subseteq E \) of minimum weight \( c(S) = \sum_{e \in S} c_e \)

\[ T = \{ \text{Steiner node} \} \]
The Steiner Tree Problem

Definition

**Given:** \( G = (V, E) \), terminals \( T \subseteq V \), edge weights \( c_e \ (e \in E) \)

**Find:** Steiner tree \( S \subseteq E \) of minimum weight \( c(S) = \sum_{e \in S} c_e \)

Theorem (Garey & Johnson, 1979)

*The Steiner Tree Problem is NP-complete.*
The Steiner Tree Problem

**Definition**

Given: \( G = (V, E) \), terminals \( T \subseteq V \), edge weights \( c_e \ (e \in E) \)

Find: Steiner tree \( S \subseteq E \) of minimum weight \( c(S) = \sum_{e \in S} c_e \)

**Theorem (Garey & Johnson, 1979)**

*The Steiner Tree Problem is NP-complete.*

Simple special cases:

- \(|T| = 2\): Shortest Path Problem
- \(T = V\): Minimum Spanning Tree Problem
Formulation via Cuts (undirected)

Aneja 1980:

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad x(\delta(U)) \geq 1 \quad \forall U \subset V : \emptyset \neq U \cap T \neq T \\
& \quad x_e \in \mathbb{Z}_0^+ \quad \forall e \in E
\end{align*}
\]

This formulation has $|E|$ variables and $O(2|V|)$ constraints. The separation problem is a minimum cut problem and can therefore be solved in polynomial time. The LP relaxation can also be solved in polynomial time (separation time $\approx$ optimization time, [GLS1988]).
Aneja 1980:

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad x(\delta(U)) \geq 1 & \forall U \subset V : \emptyset \neq U \cap T \neq T \\
& \quad x_e \in \mathbb{Z}_0^+ & \forall e \in E
\end{align*}
\]

- This formulation has \(|E|\) variables and \(O(2^{|V|})\) constraints.
**Formulation via Cuts (undirected)**

Aneja 1980:

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad x(\delta(U)) \geq 1 \quad \forall \ U \subset V : \emptyset \neq U \cap T \neq T \\
& \quad x_e \in \mathbb{Z}_0^+ \quad \forall \ e \in E
\end{align*}
\]

- This formulation has $|E|$ variables and $O(2^{|V|})$ constraints.
- The separation problem is a minimum cut problem and can therefore be solved in polynomial time.
Formulation via Cuts (undirected)

Aneja 1980:

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad x(\delta(U)) \geq 1 \quad \forall U \subset V : \emptyset \neq U \cap T \neq T \\
& \quad x_e \in \mathbb{Z}_0^+ \quad \forall e \in E
\end{align*}
\]

- This formulation has \(|E|\) variables and \(O(2^{|V|})\) constraints.
- The separation problem is a minimum cut problem and can therefore be solved in polynomial time.
- The LP relaxation can also be solved in polynomial time. (separation time \(\approx\) optimization time, [GLS1988])
Formulation via Directed Cuts

**Formulation**

Given a graph $G = (V,E)$, terminals $T \subseteq V$, and edge weights $c_e$: 

1. Build digraph $D = (V,A)$ with $A := \{(i,j), (j,i) | \{i,j\} \in E\}$.
2. Choose arc weights $c((i,j)) = c_{\{i,j\}}$ for all $(i,j) \in A$.
3. Choose a root $r \in T$ and (other) terminals $T' := T \setminus \{r\}$.

**Planning and Optimizing Large Scale Networks**

Maren Martens

15/29
From $G = (V, E)$, terminals $T \subseteq V$, and edge weights $c_e$:

- Build digraph $D = (V, A)$ with $A := \{(i, j), (j, i) | \{i, j\} \in E\}$.
- Choose arc weights $c_{(i,j)} := c_{\{i,j\}}$ for all $(i, j) \in A$. 

Formulation via Directed Cuts
Formulation via Directed Cuts

From $G = (V, E)$, terminals $T \subseteq V$, and edge weights $c_e$:

- Build digraph $D = (V, A)$ with $A := \{(i, j), (j, i) | \{i, j\} \in E\}$.
- Choose arc weights $c_{(i, j)} := c_{\{i, j\}}$ for all $(i, j) \in A$.
- Choose a root $r \in T$ and (other) terminals $T' := T \setminus \{r\}$. 
Formulation via Directed Cuts

From $G = (V, E)$, terminals $T \subseteq V$, and edge weights $c_e$:

- Build digraph $D = (V, A)$ with $A := \{(i, j), (j, i) | \{i, j\} \in E\}$.
- Choose arc weights $c_{(i,j)} := c_{\{i,j\}}$ for all $(i, j) \in A$.
- Choose a root $r \in T$ and (other) terminals $T' := T \setminus \{r\}$.

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad y(\delta^+(U)) \geq 1 & \forall U \subseteq V : \ r \in U, U \cap T' \neq T' \\
& \quad y(i,j) + y(j,i) \leq x_{\{i,j\}} & \forall \{i, j\} \in E \\
& \quad y(i,j) \in \{0, 1\} & \forall (i, j) \in A \\
& \quad x_{\{i,j\}} \in \mathbb{Z}_0^+ & \forall \{i, j\} \in E
\end{align*}
\]
Formulation via Directed Cuts

From $G = (V, E)$, terminals $T \subseteq V$, and edge weights $c_e$:

- Build digraph $D = (V, A)$ with $A := \{(i, j), (j, i) \mid \{i, j\} \in E\}$.
- Choose arc weights $c_{(i, j)} := c_{\{i, j\}}$ for all $(i, j) \in A$.
- Choose a root $r \in T$ and (other) terminals $T' := T \setminus \{r\}$.

\[
\begin{align*}
\min \quad & c^T x \\
\text{s.t.} \quad & y(\delta^+(U)) \geq 1 \quad \forall \ U \subset V : \ r \in U, U \cap T' \neq T' \\
& y(i, j) + y(j, i) \leq x_{\{i, j\}} \quad \forall \ \{i, j\} \in E \\
& y(i, j) \geq 0 \quad \forall \ (i, j) \in A \\
& x_{\{i, j\}} \in \mathbb{Z}^+_0 \quad \forall \ \{i, j\} \in E
\end{align*}
\]
Directed vs. Undirected

- The directed cut formulation reflects all feasible Steiner trees.
Directed vs. Undirected

- The directed cut formulation reflects all feasible Steiner trees.
- It permits fewer solutions than the undirected cut formulation.
Directed vs. Undirected

- The directed cut formulation reflects all feasible Steiner trees.
- It permits fewer solutions than the undirected cut formulation.

\[ T = V \]
Directed vs. Undirected

- The directed cut formulation reflects all feasible Steiner trees.
- It permits fewer solutions than the undirected cut formulation.

\[ T = V \]
Directed vs. Undirected

- The directed cut formulation reflects all feasible Steiner trees.
- It permits fewer solutions than the undirected cut formulation.

\[ T = V \]
Directed vs. Undirected

- The directed cut formulation reflects all feasible Steiner trees.
- It permits fewer solutions than the undirected cut formulation.

\[ T = V \]
Directed vs. Undirected

- The directed cut formulation reflects all feasible Steiner trees.
- It permits fewer solutions than the undirected cut formulation.

\[ r = V \]

\[
\begin{align*}
T &= V \\
\end{align*}
\]
Directed vs. Undirected

- The directed cut formulation reflects all feasible Steiner trees.
- It permits fewer solutions than the undirected cut formulation.

\[ T = V \]
Directed vs. Undirected

- The directed cut formulation reflects all feasible Steiner trees.
- It permits fewer solutions than the undirected cut formulation.

\[ T = V \]
Directed vs. Undirected

- The directed cut formulation reflects all feasible Steiner trees.
- It permits fewer solutions than the undirected cut formulation.

\[ T = V \]

Theorem

The directed cut formulation is stronger than the undirected cut formulation.
Formulation via Flows

Problem with cuts: There are too many!
Formulation via Flows

Problem with cuts: There are too many!

On digraph $D = (V, A)$ with root $r \in T$ and $T' := T \setminus \{r\}$:

- $\forall t \in T', a \in A$: flow variables $f_t(a) = \begin{cases} 1, & a \text{ on } r-t\text{-path} \\ 0, & \text{otherwise} \end{cases}$
Formulation via Flows

Problem with cuts: There are too many!

On digraph $D = (V, A)$ with root $r \in T$ and $T' := T \setminus \{r\}$:

- $\forall t \in T', a \in A$: flow variables $f_t(a) = \begin{cases} 1, & a \text{ on } r-t\text{-path} \\ 0, & \text{otherwise} \end{cases}$

$$\min \ c^T x$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(v)} f_t(a) - \sum_{a \in \delta^-(v)} f_t(a) = \begin{cases} 1, & v = r \\ 0, & \text{else} \end{cases} \quad \forall t \in T', v \neq t$$

$$f_s(i, j) + f_t(j, i) \leq x_{\{i,j\}} \quad \forall s, t \in T', \{i, j\} \in E$$

$$f_t(a) \in \{0, 1\} \quad \forall t \in T', a \in A$$

$$x_e \in \mathbb{Z}^+_0 \quad \forall e \in E$$
Formulation via Flows

**Problem with cuts:** There are too many!

On digraph $D = (V, A)$ with root $r \in T$ and $T' := T \setminus \{r\}$:

- $\forall t \in T', a \in A$: flow variables $f_t(a) = \begin{cases} 1, & a \text{ on } r-t\text{-path} \\ 0, & \text{otherwise} \end{cases}$

$$\min c^T x$$

$$\text{s.t. } \sum_{a \in \delta^+(v)} f_t(a) - \sum_{a \in \delta^-(v)} f_t(a) = \begin{cases} 1, & v = r \\ 0, & \text{else} \end{cases} \forall t \in T', v \neq t$$

$$f_s(i, j) + f_t(j, i) \leq x_{\{i,j\}} \forall s, t \in T', \{i, j\} \in E$$

$$f_t(a) \geq 0 \forall t \in T', a \in A$$

$$x_e \in \mathbb{Z}_0^+ \forall e \in E$$
Survivability
Failures happen...
Failures happen...
Failures happen...
Failures happen...
Failures happen...
2-Connectivity

**Given:** $G = (V, E)$, edge cost $c_e (e \in E)$

**Find:** Minimum cost subgraph containing all $v \in V$ and having 2 node/edge-disjoint $s$-$t$-paths for all $s, t \in V$
2-Connectivity

**Given:** \( G = (V, E) \), edge cost \( c_e (e \in E) \)

**Find:** Minimum cost subgraph containing all \( v \in V \) and having 2 node/edge-disjoint \( s-t \)-paths for all \( s, t \in V \)
2-Connectivity

**Given:** $G = (V, E)$, edge cost $c_e$ ($e \in E$)

**Find:** Minimum cost subgraph containing all $v \in V$ and having 2 node/edge-disjoint $s$-$t$-paths for all $s, t \in V$
{0,1,2}-Survivable Network Design ({0,1,2}-SND)

**Given:** \( G = (V, E) \), connectivity request \( \rho_v \in \{0, 1, 2\} \) \((v \in V)\), edge cost \( c_e \) \((e \in E)\)

**Find:** Minimum cost subgraph containing all \( v \in V \) with \( \rho_v > 0 \), having \( \min\{\rho_s, \rho_t\} \) node/edge-disjoint \( s-t \)-paths \((s, t \in V)\)
\textbf{0,1,2}-Survivable Network Design (\{0,1,2\}-SND)

Given: $G = (V, E)$, connectivity request $\rho_v \in \{0, 1, 2\}$ ($v \in V$),
equation{edge cost $c_e$ ($e \in E$)}

Find: Minimum cost subgraph containing all $v \in V$ with $\rho_v > 0$,
having $\min\{\rho_s, \rho_t\}$ node/edge-disjoint $s$-$t$-paths ($s, t \in V$)
**{0,1,2}-Survivable Network Design (\{0,1,2\)-SND)**

**Given:** \( G = (V, E) \), connectivity request \( \rho_v \in \{0, 1, 2\} \) \((v \in V)\), edge cost \( c_e \) \((e \in E)\)

**Find:** Minimum cost subgraph containing all \( v \in V \) with \( \rho_v > 0 \), having \( \min\{\rho_s, \rho_t\} \) node/edge-disjoint \( s-t \)-paths \((s, t \in V)\)
Algorithms for \( \{0,1,2\}\)-SND

Theorem

\( \{0,1,2\}\)-SND is NP-hard.
Algorithms for $\{0,1,2\}$-SND

**Theorem**

$\{0,1,2\}$-SND is NP-hard.

IP formulations:

- via undirected cuts (Grötschel, Monma, Stoer 1989)
- via directed cuts (Chimani, Kandyba, Ljubic, Mutzel 2007)
Theorem

\{0,1,2\}-SND is NP-hard.

IP formulations:

- via undirected cuts (Grötschel, Monma, Stoer 1989)
- via directed cuts (Chimani, Kandyba, Ljubic, Mutzel 2007)

Algorithms approximating the minimum cost:

- Factor 2-approximation for edge-connectivity (Jain 1998)
- Node-connectivity is NP-hard to approximate within factor $2^\log^{1-\epsilon} |V|$ (Kortsarz, Krauthgamer, Lee 2003)
Algorithms for \{0,1,2\}-SND

Theorem
\{0,1,2\}-SND is NP-hard.

IP formulations:
- via undirected cuts (Grötschel, Monma, Stoer 1989)
- via directed cuts (Chimani, Kandyba, Ljubic, Mutzel 2007)

Algorithms approximating the minimum cost:
- Factor 2-approximation for edge-connectivity (Jain 1998)
- Node-connectivity is NP-hard to approximate within factor \(2^{\log^{1-\epsilon}|V|}\) (Kortsarz, Krauthgamer, Lee 2003)*
  * 2-approximation for 2-node-connected subgraph (Khuller 1997)
DEMANDS AND CAPACITIES
Are we done yet?
Are we done yet?
Are we done yet?
Are we done yet?
Are we done yet?
Connectivity and Flows

**Given:** $G = (V, E)$, commodities $(s, t) \in V \times V$ with demands $d_{st}$, capacity modules $k \in K$ with capacities $u_k$, cost $c_k$
Given: $G = (V, E)$, commodities $(s, t) \in V \times V$ with demands $d_{st}$, capacity modules $k \in K$ with capacities $u_k$, cost $c_k$

Find: Minimum cost subgraph with capacity installations such that we can route all demands.
Connectivity and Flows

**Given:** \( G = (V, E) \), commodities \((s, t) \in V \times V\) with demands \( d_{st} \), capacity modules \( k \in K \) with capacities \( u_k \), cost \( c_k \)

**Find:** Minimum cost subgraph with capacity installations such that we can route all demands
Connectivity and Flows

**Given:** $G = (V, E)$, commodities $(s, t) \in V \times V$ with demands $d_{st}$, capacity modules $k \in K$ with capacities $u_k$, cost $c_k$

**Find:** Minimum cost subgraph with capacity installations such that we can route all demands

Capacity modules:
1: $u_1 = 2$, $c_1 = 20$
2: $u_2 = 3$, $c_2 = 25$
3: $u_3 = 5$, $c_3 = 30$
Connectivity and Flows

**Given:** \( G = (V, E) \), commodities \((s, t) \in V \times V\) with demands \(d_{st}\), capacity modules \(k \in K\) with capacities \(u_k\), cost \(c_k\)

**Find:** Minimum cost subgraph with capacity installations such that we can route all demands

**Capacity modules:**
1: \(u_1 = 2, c_1 = 20\)
2: \(u_2 = 3, c_2 = 25\)
3: \(u_3 = 5, c_3 = 30\)
Connectivity and Flows

**Given:** \( G = (V, E) \), commodities \((s, t) \in V \times V\) with demands \( d_{st} \), capacity modules \( k \in K \) with capacities \( u_k \), cost \( c_k \)

**Find:** Minimum cost subgraph with capacity installations such that we can route all demands

Capacity modules:
1: \( u_1 = 2, c_1 = 20 \)
2: \( u_2 = 3, c_2 = 25 \)
3: \( u_3 = 5, c_3 = 30 \)

Total cost for core: 55
IP Formulation (Connectivity and Flows)

**Given:** $G = (V, E)$, commodities $(s, t) \in V \times V$ with demands $d_{st}$, capacity modules $k \in K$ with capacities $u_k$, cost $c_k$
IP Formulation (Connectivity and Flows)

**Given:** $G = (V, E)$, commodities $(s, t) \in V \times V$ with demands $d_{st}$, capacity modules $k \in K$ with capacities $u_k$, cost $c_k$

Network flow on arcs: $f(a) \geq 0$, for $a \in \{(i, j), (j, i)\mid \{i, j\} \in E\}$

Capacity decisions for edges: $y_e^e \in \mathbb{Z}_0^+$, for $e \in E, k \in K$
IP Formulation (Connectivity and Flows)

**Given:** $G = (V, E)$, commodities $(s, t) \in V \times V$ with demands $d_{st}$, capacity modules $k \in K$ with capacities $u_k$, cost $c_k$

Network flow on arcs: $f(a) \geq 0$, for $a \in \{(i, j), (j, i)|\{i, j\} \in E\}$

Capacity decisions for edges: $y_k^e \in \mathbb{Z}_0^+$, for $e \in E, k \in K$

$$\min \sum_{e \in E} \sum_{k \in K} c_k y_k^e$$

s.t. $\sum_{a \in \delta^+(v)} f_{st}(a) - \sum_{a \in \delta^-(v)} f_{st}(a) = \begin{cases} d_{st}, & v = s \\ 0, & \text{else} \end{cases} \quad \forall (s, t), v \neq t$

$$\sum_{(s,t)} a_{st}(f(a_e) + f(\overleftarrow{a_e})) \leq \sum_{k \in K} u_k y_k^e \quad \forall e \in E$$

[for $e = \{i, j\} \in E$: directed arcs $a_e = (i, j)$ and $\overleftarrow{a_e} = (j, i)$]
How to realize survivability?

(1+1)-protection:
Send every demand twice on two node/edge-disjoint paths.
How to realize survivability?

(1+1)-protection:
Send every demand twice on two node/edge-disjoint paths.
How to realize survivability?

(1+1)-protection:
Send every demand twice on two node/edge-disjoint paths.
How to realize survivability?

(1+1)-protection:
Send every demand twice on two node/edge-disjoint paths.
How to realize survivability?

(1+1)-protection:
Send every demand twice on two node/edge-disjoint paths.

total cost for core: 155
IP Formulation ((1+1)-Protection)

**Given:** $G = (V, E)$, commodities $(s, t) \in V \times V$ with demands $d_{st}$, capacity modules $k \in K$ with capacities $u_k$, cost $c_k$

**Network flow on arcs:** $f(a) \geq 0$, for $a \in \{(i, j), (j, i) | \{i, j\} \in E\}$

**Capacity decisions for edges:** $y_{ke} \in \mathbb{Z}_0^+$, for $e \in E, k \in K$
IP Formulation \(((1+1)\text{-Protection})\)

**Given:** \(G = (V, E)\), commodities \((s, t) \in V \times V\) with demands \(d_{st}\), capacity modules \(k \in K\) with capacities \(u_k\), cost \(c_k\)

**Network flow on arcs:** \(f(a) \geq 0\), for \(a \in \{(i, j), (j, i)|\{i, j\} \in E\}\)

**Capacity decisions for edges:** \(y_{ek}^c \in \mathbb{Z}_0^+,\) for \(e \in E, k \in K\)

\[
\begin{align*}
\min & \quad \sum_{e \in E} \sum_{k \in K} c_k y_{ek}^c \\
\text{s.t.} & \quad \sum_{a \in \delta^+(v)} f_{st}(a) - \sum_{a \in \delta^-(v)} f_{st}(a) = \begin{cases} 2, & v = s \\ 0, & \text{else} \end{cases} \quad \forall (s, t), v \neq t \\
& \quad \sum_{(s,t)} d_{st}(f(a_e) + f(\bar{a_e})) \leq \sum_{k \in K} u_k y_{ek}^c \quad \forall e \in E \\
& \quad \sum_{a \in \delta^-(v)} f_{st}(a) \leq 1 \quad \forall (s, t), v \neq t^* 
\end{align*}
\]
**IP Formulation ((1+1)-Protection)**

**Given:** $G = (V, E)$, commodities $(s, t) \in V \times V$ with demands $d_{st}$, capacity modules $k \in K$ with capacities $u_k$, cost $c_k$

Network flow on arcs: $f(a) \geq 0$, for $a \in \{(i, j), (j, i)\} \{i, j\} \in E$

Capacity decisions for edges: $y_{ek}^c \in \mathbb{Z}_0^+$, for $e \in E, k \in K$

\[
\begin{align*}
\min \quad & \sum_{e \in E} \sum_{k \in K} c_k y_{ek}^c \\
\text{s.t.} \quad & \sum_{a \in \delta^+(v)} f_{st}(a) - \sum_{a \in \delta^-(v)} f_{st}(a) = \begin{cases} 2, & v = s \\ 0, & \text{else} \end{cases} \quad \forall (s, t), v \neq t \\
& \sum_{(s,t)} d_{st}(f(a_e) + f(\overleftarrow{a_e})) \leq \sum_{k \in K} u_k y_{ek}^c \quad \forall e \in E \\
& \sum_{a \in \delta^-(v)} f_{st}(a) \leq 1 \quad \forall (s, t), v \neq t^* \\
\end{align*}
\]

* for node-disjointness
To be continued with Exercises at 2PM!
To be continued with Exercises at 2PM!
Enjoy lunch!