Solving Bulk-Robust Assignment Problems to Optimality

Matthias Walter (RWTH Aachen)

Joint work with

David Adjiashvili (ETH Zürich), Viktor Bindewald & Dennis Michaels (TU Dortmund)

SCIP Workshop 2018, Aachen, March 8 2018
**Assignment Problem:**

- Input: Bipartite graph $G = (V, E)$ with $V = A \cup B$, edge costs $c \in \mathbb{R}^E$
- Feasible sets: Perfect matchings $M \subseteq E$ (assuming $|A| = |B|$)
- Goal: Minimize cost $c(M) := \sum_{e \in M} c_e$
Bulk-Robustness for Assignment Problems

Assignment Problem:
- Input: Bipartite graph $G = (V, E)$ with $V = A \cup B$, edge costs $c \in \mathbb{R}^E$
- Feasible sets: Perfect matchings $M \subseteq E$ (assuming $|A| = |B|$)
- Goal: Minimize cost $c(M) := \sum_{e \in M} c_e$

Bulk-Robustness:
- Possible (or likely) failure scenarios are given (explicitly or implicitly).
- Goal: Buy edges such that for every scenario, there still exists a perfect matching using the (bought) edges that survived.
Bulk-Robustness for Assignment Problems

**Assignment Problem:**
- Input: Bipartite graph $G = (V, E)$ with $V = A \cup B$, edge costs $c \in \mathbb{R}^E$
- Feasible sets: Perfect matchings $M \subseteq E$ (assuming $|A| = |B|$)
- Goal: Minimize cost $c(M) := \sum_{e \in M} c_e$

**Bulk-Robustness:**
- Possible (or likely) failure scenarios are given (explicitly or implicitly).
- Goal: Buy edges such that for every scenario, there still exists a perfect matching using the (bought) edges that survived.

**Literature:**
- Concept formally introduced by Adjiashvili, Stiller & Zenklusen (MPA 2015)
- Classical related problems: $k$-edge connected spanning subgraph problem robustifies spanning-tree problem against failure of any $(k - 1)$-edge set.
- LP-based $O(\log(|V|))$-approximation algorithm by Adjiashvili, Bindewald & Michaels (ICALP 2016)
Input:

- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{\{f_1\}, \ldots, \{f_\ell\}\}$ with $f_i \in E$.
- Edge costs $c \in \mathbb{R}^E$
Input:

- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{\{f_1\}, \ldots, \{f_\ell\}\}$ with $f_i \in E$.
- Edge costs $c \in \mathbb{R}^E$

Goal:

- Find $X \subseteq E$ with minimum $c(X)$ such that for all $F \in \mathcal{F}$, the subgraph $(V, X \setminus F)$ contains a perfect matching.
Input:
- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{\{f_1\}, \ldots, \{f_\ell\}\}$ with $f_i \in E$.
- Edge costs $c \in \mathbb{R}^E$

Goal:
- Find $X \subseteq E$ with minimum $c(X)$ such that for all $F \in \mathcal{F}$, the subgraph $(V, X \setminus F)$ contains a perfect matching.

Example:

Hardness:
- SetCover reduces to the problem.
- For any $d < 1$, it admits no $(d \log |V|)$-approximation, unless $\text{NP} \subseteq \text{DTIME}(|V|^{\log \log |V|})$. 
Bulk-Robust Assignments with Edge Failures

Input:
- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{\{f_1\}, \ldots, \{f_\ell\}\}$ with $f_i \in E$.
- Edge costs $c \in \mathbb{R}^E$

Goal:
- Find $X \subseteq E$ with minimum $c(X)$ such that for all $F \in \mathcal{F}$, the subgraph $(V, X \setminus F)$ contains a perfect matching.

Example:

Hardness:
- SetCover reduces to the problem.
- For any $d < 1$, it admits no $(d \log |V|)$-approximation, unless $\text{NP} \subseteq \text{DTIME}(|V|^{\log \log |V|})$. 
**Bulk-Robust Assignments with Edge Failures**

**Input:**
- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{\{f_1\}, \ldots, \{f_\ell\}\}$ with $f_i \in E$.
- Edge costs $c \in \mathbb{R}^E$

**Goal:**
- Find $X \subseteq E$ with minimum $c(X)$ such that for all $F \in \mathcal{F}$, the subgraph $(V, X \setminus F)$ contains a perfect matching.

**Example:**

**Hardness:**
- SetCover reduces to the problem.
- For any $d < 1$, it admits no $(d \log |V|)$-approximation, unless $\text{NP} \subseteq \text{DTIME}(|V|^\log \log |V|)$. 
Bulk-Robust Assignments with Edge Failures

Input:
- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{\{f_1\}, \ldots, \{f_\ell\}\}$ with $f_i \in E$.
- Edge costs $c \in \mathbb{R}^E$

Goal:
- Find $X \subseteq E$ with minimum $c(X)$ such that for all $F \in \mathcal{F}$, the subgraph $(V, X \setminus F)$ contains a perfect matching.

Example:

Hardness:
- SetCover reduces to the problem.
- For any $d < 1$, it admits no $(d \log |V|)$-approximation, unless $\text{NP} \subseteq \text{DTIME}(|V|^\log \log |V|)$. 
Bulk-Robust Assignments with Edge Failures

Input:
- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{\{f_1\}, \ldots, \{f_\ell\}\}$ with $f_i \in E$.
- Edge costs $c \in \mathbb{R}^E$

Goal:
- Find $X \subseteq E$ with minimum $c(X)$ such that for all $F \in \mathcal{F}$, the subgraph $(V, X \setminus F)$ contains a perfect matching.

Example:

Hardness:
- SetCover reduces to the problem.
- For any $d < 1$, it admits no $(d \log |V|)$-approximation, unless $\text{NP} \subseteq \text{DTIME}(|V|^\log \log |V|)$. 
Bulk-Robust Assignments with Edge Failures

Input:
- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{\{f_1\}, \ldots, \{f_\ell\}\}$ with $f_i \in E$.
- Edge costs $c \in \mathbb{R}^E$

Goal:
- Find $X \subseteq E$ with minimum $c(X)$ such that for all $F \in \mathcal{F}$, the subgraph $(V, X \setminus F)$ contains a perfect matching.

Example:

Hardness:
- SetCover reduces to the problem.
- For any $d < 1$, it admits no $(d \log |V|)$-approximation, unless $\text{NP} \subseteq \text{DTIME}(|V|^{\log \log |V|})$. 
Bulk-Robust Assignments with Edge Failures

Input:
- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{\{f_1\}, \ldots, \{f_\ell\}\}$ with $f_i \in E$.
- Edge costs $c \in \mathbb{R}^E$

Goal:
- Find $X \subseteq E$ with minimum $c(X)$ such that for all $F \in \mathcal{F}$, the subgraph $(V, X \setminus F)$ contains a perfect matching.

Example:

Hardness:
- SetCover reduces to the problem.
- For any $d < 1$, it admits no $(d \log |V|)$-approximation, unless $\text{NP} \subseteq \text{DTIME}(|V|^{\log \log |V|})$. 
Bulk-Robust Assignments with Edge Failures

Input:
- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{\{f_1\}, \ldots, \{f_\ell\}\}$ with $f_i \in E$.
- Edge costs $c \in \mathbb{R}^E$

Goal:
- Find $X \subseteq E$ with minimum $c(X)$ such that for all $F \in \mathcal{F}$, the subgraph $(V, X \setminus F)$ contains a perfect matching.

Example:

Hardness:
- SetCover reduces to the problem.
- For any $d < 1$, it admits no $(d \log |V|)$-approximation, unless $\text{NP} \subseteq \text{DTIME}(|V|^\log \log |V|)$. 
Bulk-Robust Assignments with Node Failures

Input:
- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{\delta(b_1), \ldots, \delta(b_\ell)\}$ with $b_i \in B$.
- Edge costs $c \in \mathbb{R}^E$
Bulk-Robust Assignments with Node Failures

Input:
- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{\delta(b_1), \ldots, \delta(b_\ell)\}$ with $b_i \in B$.
- Edge costs $c \in \mathbb{R}^E$

Goal:
- Find $X \subseteq E$ with minimum $c(X)$ such that
- for all $F \in \mathcal{F}$, the subgraph $(V, X \setminus F)$ contains an $A$-perfect matching (a matching that covers $A$).
Bulk-Robust Assignments with Node Failures

Input:
- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{\delta(b_1), \ldots, \delta(b_\ell)\}$ with $b_i \in B$.
- Edge costs $c \in \mathbb{R}^E$

Goal:
- Find $X \subseteq E$ with minimum $c(X)$ such that
- for all $F \in \mathcal{F}$, the subgraph $(V, X \setminus F)$ contains an $A$-perfect matching (a matching that covers $A$).

Example:

Related Problem:
- Related version where nodes from $B$ are bought (in contrast to edges) has approximation algorithm by Adjiashvili, Bindewald & Michaels (2017).
Bulk-Robust Assignments with Node Failures

Input:
- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{\delta(b_1), \ldots, \delta(b_\ell)\}$ with $b_i \in B$.
- Edge costs $c \in \mathbb{R}^E$

Goal:
- Find $X \subseteq E$ with minimum $c(X)$ such that
- for all $F \in \mathcal{F}$, the subgraph $(V, X\setminus F)$ contains an $A$-perfect matching (a matching that covers $A$).

Example:

Related Problem:
- Related version where nodes from $B$ are bought (in contrast to edges) has approximation algorithm by Adjiashvili, Bindewald & Michaels (2017).
Bulk-Robust Assignments with Node Failures

Input:
- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{\delta(b_1), \ldots, \delta(b_\ell)\}$ with $b_i \in B$.
- Edge costs $c \in \mathbb{R}^E$

Goal:
- Find $X \subseteq E$ with minimum $c(X)$ such that
  - for all $F \in \mathcal{F}$, the subgraph $(V, X \setminus F)$ contains an $A$-perfect matching (a matching that covers $A$).

Example:

Related Problem:
- Related version where nodes from $B$ are bought (in contrast to edges) has approximation algorithm by Adjiashvili, Bindewald & Michaels (2017).
Bulk-Robust Assignments with Node Failures

Input:
- Bipartite graph \( G = (V, E) \) with \( V = A \cup B \)
- Failure scenarios \( \mathcal{F} = \{ \delta(b_1), \ldots, \delta(b_\ell) \} \) with \( b_i \in B \).
- Edge costs \( c \in \mathbb{R}^E \)

Goal:
- Find \( X \subseteq E \) with minimum \( c(X) \) such that
- for all \( F \in \mathcal{F} \), the subgraph \( (V, X \setminus F) \) contains an \( A \)-perfect matching (a matching that covers \( A \)).

Example:

Related Problem:
- Related version where nodes from \( B \) are bought (in contrast to edges) has approximation algorithm by Adjiashvili, Bindewald & Michaels (2017).
Bulk-Robust Assignments with Node Failures

Input:
- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{\delta(b_1), \ldots, \delta(b_\ell)\}$ with $b_i \in B$.
- Edge costs $c \in \mathbb{R}^E$

Goal:
- Find $X \subseteq E$ with minimum $c(X)$ such that
- for all $F \in \mathcal{F}$, the subgraph $(V, X \setminus F)$ contains an $A$-perfect matching (a matching that covers $A$).

Example:

Related Problem:
- Related version where nodes from $B$ are bought (in contrast to edges) has approximation algorithm by Adjiashvili, Bindewald & Michaels (2017).
Bulk-Robust Assignments with Node Failures

Input:
- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{\delta(b_1), \ldots, \delta(b_\ell)\}$ with $b_i \in B$.
- Edge costs $c \in \mathbb{R}^E$

Goal:
- Find $X \subseteq E$ with minimum $c(X)$ such that
- for all $F \in \mathcal{F}$, the subgraph $(V, X \setminus F)$ contains an $A$-perfect matching (a matching that covers $A$).

Example:

Related Problem:
- Related version where nodes from $B$ are bought (in contrast to edges) has approximation algorithm by Adjiashvili, Bindewald & Michaels (2017).
Bulk-Robust Assignments with Node Failures

Input:
- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{\delta(b_1), \ldots, \delta(b_\ell)\}$ with $b_i \in B$.
- Edge costs $c \in \mathbb{R}^E$

Goal:
- Find $X \subseteq E$ with minimum $c(X)$ such that
- for all $F \in \mathcal{F}$, the subgraph $(V, X \setminus F)$ contains an $A$-perfect matching (a matching that covers $A$).

Example:

Related Problem:
- Related version where nodes from $B$ are bought (in contrast to edges) has approximation algorithm by Adjiashvili, Bindewald & Michaels (2017).
Bulk-Robust Assignments with Node Failures

Input:
- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{\delta(b_1), \ldots, \delta(b_\ell)\}$ with $b_i \in B$.
- Edge costs $c \in \mathbb{R}^E$

Goal:
- Find $X \subseteq E$ with minimum $c(X)$ such that
  - for all $F \in \mathcal{F}$, the subgraph $(V, X \setminus F)$ contains an $A$-perfect matching (a matching that covers $A$).

Example:

Related Problem:
- Related version where nodes from $B$ are bought (in contrast to edges) has approximation algorithm by Adjiashvili, Bindewald & Michaels (2017).
Bulk-Robust Assignments with Node Failures

Input:
- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{\delta(b_1), \ldots, \delta(b_\ell)\}$ with $b_i \in B$.
- Edge costs $c \in \mathbb{R}^E$

Goal:
- Find $X \subseteq E$ with minimum $c(X)$ such that
- for all $F \in \mathcal{F}$, the subgraph $(V, X \setminus F)$ contains an $A$-perfect matching (a matching that covers $A$).

Example:

Related Problem:
- Related version where nodes from $B$ are bought (in contrast to edges) has approximation algorithm by Adjiashvili, Bindewald & Michaels (2017).
Bulk-Robust Assignments with Node Failures

Input:
- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{\delta(b_1), \ldots, \delta(b_\ell)\}$ with $b_i \in B$.
- Edge costs $c \in \mathbb{R}^E$

Goal:
- Find $X \subseteq E$ with minimum $c(X)$ such that
  - for all $F \in \mathcal{F}$, the subgraph $(V, X \setminus F)$ contains an $A$-perfect matching (a matching that covers $A$).

Example:

Related Problem:
- Related version where nodes from $B$ are bought (in contrast to edges) has approximation algorithm by Adjiashvili, Bindewald & Michaels (2017).
Bulk-Robust Assignments with Node Failures

Input:
- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{\delta(b_1), \ldots, \delta(b_\ell)\}$ with $b_i \in B$.
- Edge costs $c \in \mathbb{R}^E$

Goal:
- Find $X \subseteq E$ with minimum $c(X)$ such that
- for all $F \in \mathcal{F}$, the subgraph $(V, X \setminus F)$ contains an $A$-perfect matching (a matching that covers $A$).

Example:

Related Problem:
- Related version where nodes from $B$ are bought (in contrast to edges) has approximation algorithm by Adjiashvili, Bindewald & Michaels (2017).
General Case

Input:
- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{F_1, \ldots, F_\ell\}$ with $F_i \subseteq E$
  with cardinalities $k(F)$ for all $F \in \mathcal{F}$
- Edge costs $c \in \mathbb{R}^E$

Goal:
- Find $X \subseteq E$ with minimum $c(X)$ such that for all $F \in \mathcal{F}$, the subgraph $(V, X \setminus F)$ contains a matching of size $k(F)$. 
General Case

Input:

- Bipartite graph $G = (V, E)$ with $V = A \cup B$
- Failure scenarios $\mathcal{F} = \{F_1, \ldots, F_\ell\}$ with $F_i \subseteq E$
  with cardinalities $k(F)$ for all $F \in \mathcal{F}$
- Edge costs $c \in \mathbb{R}^E$

Goal:

- Find $X \subseteq E$ with minimum $c(X)$ such that
  for all $F \in \mathcal{F}$, the subgraph $(V, X \setminus F)$ contains a matching of size $k(F)$.

Special Cases:

- Edge failures: Set $k(F_i) := |A| = |B|$ and $F_i := \{f_i\}$ for all $i \in [\ell]$.
- Node failures: Set $k(F_i) := |A|$ and $F_i := \delta(b_i)$ for all $i \in [\ell]$. 
**Integer Programming Models**

**Straight-forward model (see Adjiashvili et al., ICALP 2016):**

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad x \geq y^{(F)} \quad \text{for all } F \in \mathcal{F} \\
& \quad y^{(F)} \in P_{k(F)\text{-match}}(G - F) \quad \text{for all } F \in \mathcal{F} \\
& \quad x_e \in \mathbb{Z}_+ \quad \text{for all } e \in E
\end{align*}
\]

- Has $\mathcal{O}(|\mathcal{F}| \cdot |E|)$ variables and constraints.
### Integer Programming Models

**Straight-forward model (see Adjiashvili et al., ICALP 2016):**

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad x \geq y^{(F)} \quad \text{for all } F \in \mathcal{F} \\
& \quad y^{(F)} \in P_{k(F)}\text{-match}(G - F) \quad \text{for all } F \in \mathcal{F} \\
& \quad x_e \in \mathbb{Z}_+ \quad \text{for all } e \in E
\end{align*}
\]

- Has \( O(|\mathcal{F}| \cdot |E|) \) variables and constraints.

**Polyhedral combinatorics helps:**

- What does this mean for \( x \)?

\[
\exists y : x \geq y, \ y \in P_{k(F)}\text{-match}(G')
\]
Integer Programming Models

Straight-forward model (see Adjiashvili et al., ICALP 2016):

\[
\begin{align*}
\min & \quad c^\top x \\
\text{s.t.} & \quad x \geq y^{(F)} & \text{for all } F \in \mathcal{F} \\
& \quad y^{(F)} \in P_{k(F)}\text{-match}(G - F) & \text{for all } F \in \mathcal{F} \\
& \quad x_e \in \mathbb{Z}_+ & \text{for all } e \in E
\end{align*}
\]

- Has \(\mathcal{O}(|\mathcal{F}| \cdot |E|)\) variables and constraints.

Polyhedral combinatorics helps:

- What does this mean for \(x\)?

\[\exists y : x \geq y, \ y \in P_{k(F)}\text{-match}(G')\]

- Projection onto \(x\) is the dominant of the \(k(F)\)-matching polytope.
Integer Programming Models

Straight-forward model (see Adjiashvili et al., ICALP 2016):

\[
\begin{align*}
\text{min} & \quad c^\top x \\
\text{s.t.} & \quad x \geq y^{(F)} & \text{for all } F \in \mathcal{F} \\
 & \quad y^{(F)} \in P_{k(F)}\text{-match}(G - F) & \text{for all } F \in \mathcal{F} \\
 & \quad x_e \in \mathbb{Z}_+ & \text{for all } e \in E
\end{align*}
\]

- Has \( O(|\mathcal{F}| \cdot |E|) \) variables and constraints.

Polyhedral combinatorics helps:

- What does this mean for \( x \)?

\[ \exists y : x \geq y, \ y \in P_{k(F)}\text{-match}(G') \]

- Projection onto \( x \) is the dominant of the \( k(F) \)-matching polytope.

- Inequalities known (Fulkerson 1970):

\[ \sum_{e \in E[S]} x_e \geq |S| - |V| + k(F) \quad \text{for all } S \subseteq V \]
Integer Programming Models

Straight-forward model (see Adjiashvili et al., ICALP 2016):

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad x \geq y^{(F)} \quad \text{for all } F \in \mathcal{F} \\
& \quad y^{(F)} \in P_{k(F)\text{-match}}(G - F) \quad \text{for all } F \in \mathcal{F} \\
& \quad x_e \in \mathbb{Z}_+ \quad \text{for all } e \in E
\end{align*}
\]

- Has \( O(|\mathcal{F}| \cdot |E|) \) variables and constraints.

Equivalent (derived from dominant):

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad \sum_{e \in E[S] \setminus F} x_e \geq |S| - |V| + k(F) \quad \text{for all } S \subseteq V \text{ for all } F \in \mathcal{F} \\
& \quad x_e \in \mathbb{Z}_+ \quad \text{for all } e \in E
\end{align*}
\]

- Has \( O(|E|) \) variables and \( O(|\mathcal{F}| \cdot 2^{|V|}) \) constraints.
- For every \( F \in \mathcal{F} \), separation problem reduces to a minimum \( s-t \)-cut problem.
Computational Setup

Setup:
- System: KUbuntu on 2.6 GHz with 16 GB RAM
- Software: SCIPOptSuite 5.0.0 and LEMON Graph Library

Parameters:
- General purpose cuts off
- Heuristics off in root node
- Time limit of 600 s
- Different further settings depending on experiment.
Models in Practice: LP Relaxation

Setup:

- Complete bipartite graphs with \(|A| = |B| = n\)
- Uniform failures \(\mathcal{F} = \{\{e\} \mid e \in E\}\), unit costs \(c = 1\)
- Settings: only root node

![Running times for LP relaxation](image-url)
Robust Assignments

Models

CG Cuts

Models in Practice: LP Relaxation

Setup:

- Erdős-Rényi graphs with $|A| = |B| = n$, $p = 0.5$
- Uniform failures $\mathcal{F} = \{\{e\} \mid e \in E\}$, unit costs $c = 1$
- Settings: only root node

Running times for LP relaxation

Running time [s]

$n = |A| = |B|$
Models in Practice: IP Bounds

Setup:
- Complete bipartite graphs with \(|A| = |B| = n\)
- Uniform failures \(F = \{\{e\} \mid e \in E\}\), unit costs \(c = 1\)

Results for compact vs. dominant model (IP)

<table>
<thead>
<tr>
<th>n</th>
<th>Opt</th>
<th>Compact model</th>
<th></th>
<th>Dominant model</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Root time</td>
<td>Final bnd</td>
<td>Time</td>
<td>Root time</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0.0</td>
<td>10.0</td>
<td>6.7</td>
<td>0.0</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>0.1</td>
<td>12.0</td>
<td>252.5</td>
<td>0.0</td>
</tr>
<tr>
<td>7</td>
<td>14</td>
<td>0.4</td>
<td>10.5</td>
<td>600.0</td>
<td>0.0</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>1.7</td>
<td>10.8</td>
<td>600.0</td>
<td>0.0</td>
</tr>
<tr>
<td>9</td>
<td>18</td>
<td>5.8</td>
<td>11.1</td>
<td>600.0</td>
<td>0.0</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>14.8</td>
<td>11.4</td>
<td>600.0</td>
<td>0.0</td>
</tr>
<tr>
<td>11</td>
<td>22</td>
<td>41.2</td>
<td>12.2</td>
<td>600.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Solving with Gurobi

Setup:

- Combination of complete graph, singleton failures and unit costs has lots of symmetry.
- Gurobi detects this and can prove lower bound earlier.
- For $n = 8$, we observe the following behavior:

### Results for compact model with Gurobi (IP)

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Expl Unexpl</td>
<td>Obj</td>
<td>Depth</td>
<td>IntInf</td>
<td>Incumbent</td>
<td>BestBd</td>
<td>Gap</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>9.14286</td>
<td>0 3648</td>
<td>64.00000</td>
<td>9.14286</td>
<td>85.7%</td>
<td>-</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>9.14286</td>
<td>0 3648</td>
<td>64.00000</td>
<td>9.14286</td>
<td>85.7%</td>
<td>-</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>9.14286</td>
<td>0 3648</td>
<td>64.00000</td>
<td>9.14286</td>
<td>85.7%</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>10.00000</td>
<td>1 3521</td>
<td>64.00000</td>
<td>10.00000</td>
<td>84.4%</td>
<td>5738</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>525</td>
<td>355</td>
<td>13.00000</td>
<td>16 1267</td>
<td>16.00000</td>
<td>11.73810</td>
<td>26.6%</td>
<td>1526</td>
</tr>
<tr>
<td>542</td>
<td>349</td>
<td>infeasible</td>
<td>16</td>
<td>16.00000</td>
<td>13.42857</td>
<td>16.1%</td>
<td>1509</td>
</tr>
<tr>
<td>559</td>
<td>345</td>
<td>15.00000</td>
<td>19 1923</td>
<td>16.00000</td>
<td>14.00000</td>
<td>12.5%</td>
<td>1525</td>
</tr>
</tbody>
</table>

Explored 568 nodes (899200 simplex iterations) in 223.95 seconds
Thread count was 1 (of 4 available processors)

Optimal solution found (tolerance 1.00e-04)
Best objective 1.60000000000000e+01, best bound 1.60000000000000e+01, gap 0.0%
Solving with Gurobi

**Setup:**
- Combination of complete graph, singleton failures and unit costs has lots of symmetry.
- Gurobi detects this and can prove lower bound earlier.
- For \( n = 8 \), we observe the following behavior:

**Results for compact model with Gurobi (IP)**

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Current Node</th>
<th>Objective Bounds</th>
<th>Work</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Expl Unexpl</td>
<td>Obj Depth IntInf</td>
<td>Incumbent</td>
</tr>
<tr>
<td>0 0</td>
<td>9.14286</td>
<td>0 3648</td>
<td>64.00000</td>
</tr>
<tr>
<td>0 0</td>
<td>9.14286</td>
<td>0 3648</td>
<td>64.00000</td>
</tr>
<tr>
<td>0 2</td>
<td>9.14286</td>
<td>0 3648</td>
<td>64.00000</td>
</tr>
<tr>
<td>2 2</td>
<td>10.00000</td>
<td>1 3521</td>
<td>64.00000</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>525 355</td>
<td>13.00000</td>
<td>16 1267</td>
<td>16.00000</td>
</tr>
<tr>
<td>542 349</td>
<td>infeasible</td>
<td>16</td>
<td>16.00000</td>
</tr>
<tr>
<td>559 345</td>
<td>15.00000</td>
<td>19 1923</td>
<td>16.00000</td>
</tr>
</tbody>
</table>

Explored 568 nodes (899200 simplex iterations) in 223.95 seconds

Thread count was 1 (of 4 available processors)

Optimal solution found (tolerance 1.00e-04)
Best objective 1.6000000000000e+01, best bound 1.6000000000000e+01, gap 0.0%

**Note:** This effect vanishes as soon as the graph is not symmetric anymore.
Strengthening the Model

Chváatal-Gomory cuts:

- Consider $F_1, \ldots, F_\ell$ with constant $k(F_i) = k$ for all $i \in [\ell]$ ($\ell \geq 2$).
- Sum up all inequalities for fixed $S$ with $|S| - |V| + k \geq 1$. 
Strengthening the Model

Chvátal-Gomory cuts:

- Consider $F_1, \ldots, F_\ell$ with constant $k(F_i) = k$ for all $i \in [\ell]$ ($\ell \geq 2$).
- Sum up all inequalities for fixed $S$ with $|S| - |V| + k \geq 1$.

$$\sum_{e \in E[S]} \left| \{ i \in [\ell] \mid e \in E \setminus F_i \} \right| x_e \geq \ell(|S| - |V| + k)$$
**Strengthening the Model**

**Chvátal-Gomory cuts:**
- Consider $F_1, \ldots, F_\ell$ with constant $k(F_i) = k$ for all $i \in [\ell]$ ($\ell \geq 2$).
- Sum up all inequalities for fixed $S$ with $|S| - |V| + k \geq 1$.

$$\sum_{e \in E[S]} \{ i \in [\ell] \mid e \in E \setminus F_i \} x_e \geq \ell(|S| - |V| + k)$$

- Scale it by $1/(\ell - 1)$.
Strengthening the Model

Chvátal-Gomory cuts:

- Consider $F_1, \ldots, F_\ell$ with constant $k(F_i) = k$ for all $i \in [\ell]$ ($\ell \geq 2$).
- Sum up all inequalities for fixed $S$ with $|S| - |V| + k \geq 1$.

$$\sum_{e \in E[S]} \left| \{i \in [\ell] \mid e \in E \setminus F_i \} \right| x_e \geq \ell(|S| - |V| + k)$$

- Scale it by $1/((\ell - 1))$.

$$\sum_{e \in E[S]} \frac{\left| \{i \in [\ell] \mid e \in E \setminus F_i \} \right|}{\ell - 1} x_e \geq \frac{\ell}{\ell - 1}(|S| - |V| + k)$$
Strengthening the Model

**Chvátal-Gomory cuts:**

- Consider $F_1, \ldots, F_\ell$ with constant $k(F_i) = k$ for all $i \in [\ell]$ ($\ell \geq 2$).
- Sum up all inequalities for fixed $S$ with $|S| - |V| + k \geq 1$.

$$\sum_{e \in E[S]} \left| \left\{ i \in [\ell] \mid e \in E \setminus F_i \right\} \right| x_e \geq \ell(\ell - 1)$$

- Scale it by $1/(\ell - 1)$.

$$\sum_{e \in E[S]} \frac{\left| \left\{ i \in [\ell] \mid e \in E \setminus F_i \right\} \right|}{\ell - 1} x_e \geq \frac{\ell}{\ell - 1}(|S| - |V| + k)$$

- $x$ is integer and nonnegative, so round up coefficients and right-hand side.
Strengthening the Model

Chvátal-Gomory cuts:

- Consider $F_1, \ldots, F_\ell$ with constant $k(F_i) = k$ for all $i \in [\ell]$ ($\ell \geq 2$).
- Sum up all inequalities for fixed $S$ with $|S| - |V| + k \geq 1$.

$$\sum_{e \in E(S)} \left| \left\{ i \in [\ell] \mid e \in E \setminus F_i \right\} \right| x_e \geq \ell(|S| - |V| + k)$$

- Scale it by $1/((\ell - 1))$.

$$\sum_{e \in E(S)} \left| \left\{ i \in [\ell] \mid e \in E \setminus F_i \right\} \right| \frac{x_e}{\ell - 1} \geq \frac{\ell}{\ell - 1}(|S| - |V| + k)$$

- $x$ is integer and nonnegative, so round up coefficients and right-hand side.

$$\sum_{e \in E(S)} \begin{cases} 2 & \text{if } e \text{ in no } F_i \\ 0 & \text{if } e \text{ in all } F_i \\ 1 & \text{otherwise} \end{cases} x_e \geq |S| - |V| + k + 1$$
Strengthening the Model

Chvátal-Gomory cuts:

- Consider $F_1, \ldots, F_\ell$ with constant $k(F_i) = k$ for all $i \in [\ell]$ ($\ell \geq 2$).
- Sum up all inequalities for fixed $S$ with $|S| - |V| + k \geq 1$.

$$
\sum_{e \in E[S]} \left\{ \begin{array}{ll}
 i \in [\ell] & |e \in E \setminus F_i| \geq \ell(|S| - |V| + k) \\
 \end{array} \right.
$$

- Scale it by $1/(\ell - 1)$.

$$
\sum_{e \in E[S]} \left\{ \begin{array}{ll}
 i \in [\ell] & |e \in E \setminus F_i| \geq \frac{\ell}{\ell - 1}(|S| - |V| + k) \\
 \end{array} \right.
$$

- $x$ is integer and nonnegative, so round up coefficients and right-hand side.

$$
\sum_{e \in E[S]} \left\{ \begin{array}{ll}
 2 & \text{if } e \text{ in no } F_i \\
 0 & \text{if } e \text{ in all } F_i \\
 1 & \text{otherwise} \\
 \end{array} \right. x_e \geq |S| - |V| + k + 1
$$

- **Weakened** for coefficients with $e$ in no $F_i$.
- **Strengthened** for coefficients with $e$ in all $F_i$.
- **Stronger** right-hand side.
Separation Problem

Input:
- Bipartite graph $G = (V, E)$ with bipartition $V = A \cup B$.
- Edge weights $w \in \mathbb{R}_+^E$
- Parameter $k$.

Goal:
- Find $S \subseteq V$ with $|S| \geq |V| - k + 1$ minimizing $w(E[S]) - |S| + |V| - k$
Separation Problem

**Input:**
- Bipartite graph $G = (V, E)$ with bipartition $V = A \cup B$.
- Edge weights $w \in \mathbb{R}_+^E$
- Parameter $k$.

**Goal:**
- Find $S \subseteq V$ with $|S| \geq |V| - k + 1$ minimizing $w(E[S]) - |S| + |V| - k$

**IP Model:**
- Variables $y$ and $z$ with
- $y_v = 1 \iff v \in S$
- $z_e = 1 \iff e \in E[S]$

$$\begin{align*}
\text{min} & \quad - \sum_{v \in V} y_v + \sum_{e \in E} w_e z_e \\
\text{s.t.} & \quad -y_a - y_b + z_{a,b} \geq -1 \quad \text{for all } \{a, b\} \in E \\
& \quad y(A) + y(B) \geq |V| - k + 1 \\
& \quad y, z \text{ binary}
\end{align*}$$

**Observe:** TU system plus a single inequality.
Bad News: NP-hardness

Separation problem:
- Input: bipartite graph $G = (V, E)$, a nonnegative vector $w \in \mathbb{Q}^E_+$ and a number $\ell \in \mathbb{N}$.
- Goal: find a set $S \subseteq V$ with $|S| \geq \ell$ that minimizes $w(E[S]) - |S|$.

Some NP-hard problem:
- Input: bipartite Graph $G = (V, E)$, numbers $m, n \in \mathbb{N}$.
- Goal: is there a set of at most $n$ nodes that cover at least $m$ of $G$'s edges?
- Hardness: Apollonio & Simeone (2014)
Bad News: NP-hardness

Separation problem:
- Input: bipartite graph $G = (V, E)$, a nonnegative vector $w \in \mathbb{Q}_+^E$ and a number $\ell \in \mathbb{N}$.
- Goal: find a set $S \subseteq V$ with $|S| \geq \ell$ that minimizes $w(E[S]) - |S|$.

Some NP-hard problem:
- Input: bipartite Graph $G = (V, E)$, numbers $m, n \in \mathbb{N}$.
- Goal: is there a set of at most $n$ nodes that cover at least $m$ of $G$’s edges?
- Hardness: Apollonio & Simeone (2014)

Reduction idea:
- **Node complementing** ($\ell := |V| - n$) and **proper scaling** ($w := (|V| + 1)1_E$)
Bad News: NP-hardness

**Separation problem:**
- Input: bipartite graph \( G = (V, E) \), a nonnegative vector \( w \in \mathbb{Q}_+^E \) and a number \( \ell \in \mathbb{N} \).
- Goal: find a set \( S \subseteq V \) with \( |S| \geq \ell \) that minimizes \( w(E[S]) - |S| \).

**Some NP-hard problem:**
- Input: bipartite Graph \( G = (V, E) \), numbers \( m, n \in \mathbb{N} \).
- Goal: is there a set of at most \( n \) nodes that cover at least \( m \) of \( G \)'s edges?
- Hardness: Apollonio & Simeone (2014)

**Reduction idea:**
- **Node complementing** \( (\ell := |V| - n) \) and **proper scaling** \( (w := (|V| + 1)\mathbb{1}_E) \)
- Existence of \( S \) with \( |S| \leq n \) and \( |\{ e \in E \mid e \cap S \neq \emptyset \}| \geq m \) is equivalent to existence of \( \overline{S} \) with \( |\overline{S}| \geq \ell \) and

\[
|E \setminus E[\overline{S}]| \geq m \iff |E[\overline{S}]| \leq (|E| - m) \\
\iff (|V| + 1)|E[\overline{S}]| \leq (|V| + 1)(|E| - m) \\
\iff (|V| + 1)|E[\overline{S}]| - |\overline{S}| \leq (|V| + 1)(|E| - m) \\
\iff w(E[\overline{S}]) - |\overline{S}| \leq (|V| + 1)(|E| - m).
\]

(note that \( 0 \leq |\overline{S}| < |V| + 1 \))
Main idea:

- Let’s move $y(A) + y(B) \geq |V| - k + 1$ into the objective function!
- Lagrange multiplier is one-dimensional: (binary) search for good values.
- Subproblem again reduces to minimum $s-t$-cut problem.
- If it returns a set $S$ then we have a most-violated inequality among all inequalities with this $|S|$. 
Good News: Nice Heuristic Approach

Main idea:
- Let’s move $y(A) + y(B) \geq |V| - k + 1$ into the objective function!
- Lagrange multiplier is one-dimensional: (binary) search for good values.
- Subproblem again reduces to minimum s-t-cut problem.
- If it returns a set $S$ then we have a most-violated inequality among all inequalities with this $|S|$.

Desirable side-effect:

$$\sum_{e \in E[S]} \{0, 1, 2\} x_e \geq |S| - |V| + k + 1$$

- Chvátal-Gomory strengthening is stronger for small right-hand sides.
- We can control $|S|$ via Lagrange multipliers to get a small right-hand side.
- Experimentally best strategy: aim for violated cuts with minimum $|S|$.
Models in Practice: CG Cuts

Setup:

- Complete bipartite graphs with $|A| = |B| = n$
- Uniform failures $\mathcal{F} = \{\{e\} \mid e \in E\}$, unit costs $c = 1$

**Running times for IP**

![Graph showing running times for CG and CG+degree]

Running time [s]

$n = |A| = |B|$
Models in Practice: CG Cuts

Setup:

- Complete bipartite graphs with $|A| = |B| = n$
- Uniform failures $\mathcal{F} = \{\{e\} \mid e \in E\}$, unit costs $c = 1$

Running times for IP

- Note that we are solving the **IP** and not just the relaxation!
Models in Practice: CG Cuts

Setup:

- Complete bipartite graphs with $|A| = |B| = n$
- Uniform failures $\mathcal{F} = \{\{e\} \mid e \in E\}$, unit costs $c = 1$
- Special case of CG cuts are strengthened degree inequalities $x(\delta(v)) \geq 2$.

Running times for IP

- Note that we are solving the IP and not just the relaxation!
Models in Practice: CG Cuts

Setup:

- Erdős-Rényi graphs with $|A| = |B| = n$, $p = 0.5$
- Uniform failures $\mathcal{F} = \{\{e\} \mid e \in E\}$, unit costs $c = 1$

Running times for IP

![Running time graph]

- CG
- CG+degree

$n = |A| = |B|$
Models in Practice: CG Cuts

Setup:

- Complete bipartite graphs with $|A| = |B| = n$
- Uniform failures $\mathcal{F} = \{\{e\} \mid e \in E\}$
- Random costs $c_e \in \{1, \ldots, 2\}$ for all $e \in E$ independently.

Running times for IP

![Graph showing running times for CG and CG+degree methods. The x-axis represents $n = |A| = |B|$ ranging from 0 to 100, and the y-axis represents running time in seconds ranging from 0 to 300. The graph compares CG and CG+degree methods.]
Models in Practice: CG Cuts

Setup:
- Complete bipartite graphs with $|A| = |B| = n$
- Uniform failures $\mathcal{F} = \{\{e\} \mid e \in E\}$
- Random costs $c_e \in \{1, \ldots, 4\}$ for all $e \in E$ independently.

Running times for IP

![Graph showing running times for CG and CG+degree algorithms as a function of $n = |A| = |B|$]
Models in Practice: IP Bounds with CG Cuts

Setup:

- Complete bipartite graphs with $|A| = n$ and $|B| = [1.5n]$
- Node failures $\mathcal{F} = \{\delta(b) \mid b \in B\}$, unit costs $c = 1$

Remark: Problem is on primal side, i.e., finding an optimal solution!
Thanks!

Things you’ve seen:

- Speed-up of dominant formulation vs. compact one.
- Derivation of Chvátal-Gomory (CG) cuts.
- Fast heuristic separation with Lagrange multiplier.
- Strength of CG cuts, in particular strengthened degree.
Thanks!

Things you’ve seen:
- Speed-up of dominant formulation vs. compact one.
- Derivation of Chvátal-Gomory (CG) cuts.
- Fast heuristic separation with Lagrange multiplier.
- Strength of CG cuts, in particular strengthened degree.

Things you might see in the future:
- Generalization to arbitrary $s$-$t$-flows (thanks to Britta Peis).
- Structured instances:
  - ...obtained from the SetCover reduction
  - ...obtained from other sources (QAPLIB?)
  - ...yours?
- Implementation of / comparison with approximation algorithm