Partitioning a Graph into Connected Components

Hande Yaman

Research Center for Operations Research & Statistics (ORSTAT)
Faculty of Economics and Business (FEB), KU Leuven
Leuven Institute of Mobility

February 13, 2026, AIROYoung

Agenda

- Connected (sub)partition polytope
  joint work with Phablo F. S. Moura and Roel Leus

- Power system restoration
  joint work with H. Çalık and D. Van Hertem

- Balanced connected partitions of edge-weighted graphs
  joint work with Morteza Davari and Phablo F. S. Moura

The connected (sub)partition polytope

Co-authors:

Phablo F. S. Moura
KU Leuven, ORSTAT

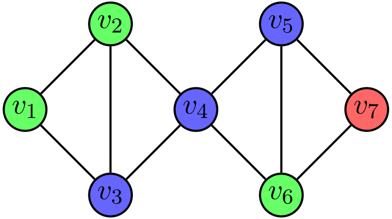
Roel Leus
KU Leuven, ORSTAT

Paper:
P. F. S. Moura, H. Yaman, R. Leus. On the connected (sub)partition polytope. Mathematical Programming.

Connectivity in combinatorial problems

Convex Recoloring Problem

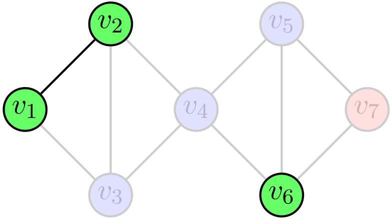
- Input: Graph and **arbitrary** coloring (may not be proper)
- Output: Convex recoloring (i.e., coloring s.t. vertices with a same color induce a connected subgraph)



Connectivity in combinatorial problems

Convex Recoloring Problem

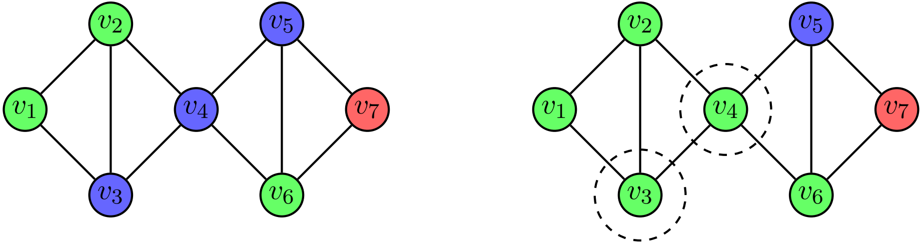
- Input: Graph and **arbitrary** coloring (may not be proper)
- Output: Convex recoloring (i.e., coloring s.t. vertices with a same color induce a connected subgraph)



Connectivity in combinatorial problems

Convex Recoloring Problem

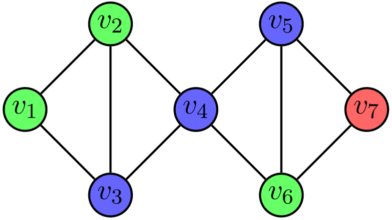
- Input: Graph and **arbitrary** coloring (may not be proper)
- Output: Convex recoloring (i.e., coloring s.t. vertices with a same color induce a connected subgraph)
- Goal: Minimize the number of recolored vertices

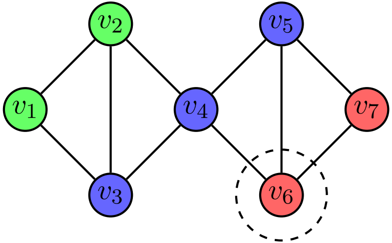


Connectivity in combinatorial problems

Convex Recoloring Problem

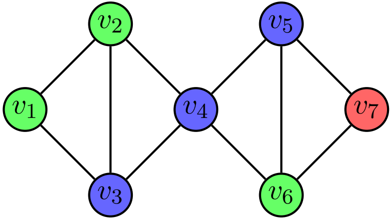
- Input: Graph and **arbitrary** coloring (may not be proper)
- Output: Convex recoloring (i.e., coloring s.t. vertices with a same color induce a connected subgraph)
- Goal: Minimize the number of recolored vertices

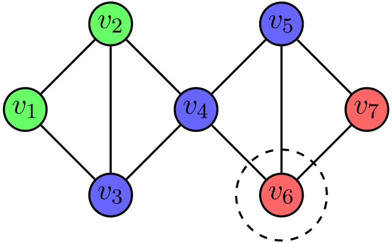




Convex Recoloring Problem

- Input: Graph and **arbitrary** coloring (may not be proper)
- Output: Convex recoloring (i.e., coloring s.t. vertices with a same color induce a connected subgraph)
- Goal: Minimize the number of recolored vertices
- NP-hard



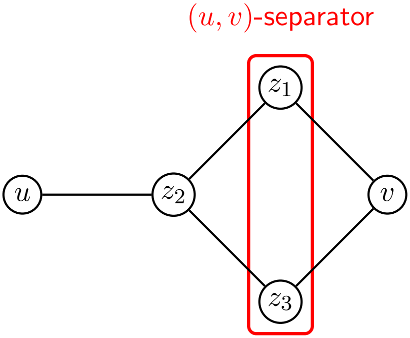


Definitions, notation and some background

- $k \in \mathbb{Z}_{>}$, $[k] = \{1, \ldots, k\}$
- $G = (V, E)$ is a connected undirected graph
- **Connected k-subpartition** $\mathcal{V} = \{V_1, \ldots, V_k\}$ of G:
  - pairwise disjoint subsets of V
  - $G[V_i]$, the subgraph induced by V_i, is connected $\forall i \in [k]$
- For connected subgraph polytope (i.e., case $k = 1$), see Wang et al. (2017).
- For polyhedral results for $k \geq 1$ in the context of Convex Recoloring and Balanced Connected Partition, see Campêlo et al. (2016) and Miyazawa et al. (2021).

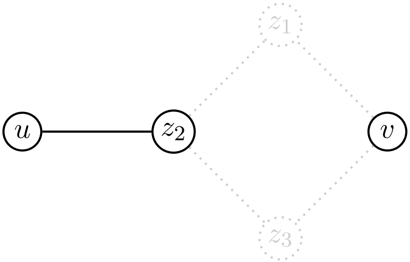
Vertices separator

- Let $\{u, v\} \notin E$. A (u, v)-**separator** is a set $Z \subseteq V \setminus \{u, v\}$ such that u and v belong to different components of $G - Z$.



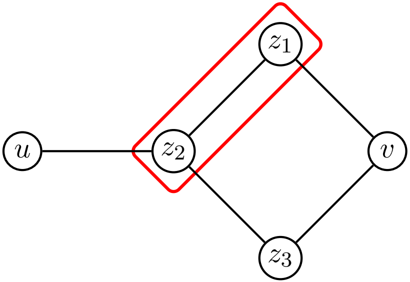
Vertices separator

- Let $\{u,v\} \notin E$. A (u,v)-**separator** is a set $Z \subseteq V \setminus \{u,v\}$ such that u and v belong to different components of $G - Z$.



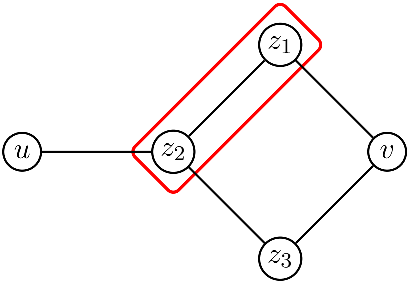
Vertices separator

- Let $\{u, v\} \notin E$. A (u, v)-**separator** is a set $Z \subseteq V \setminus \{u, v\}$ such that u and v belong to different components of $G - Z$.



Vertices separator

- Let $\{u, v\} \notin E$. A (u, v)-**separator** is a set $Z \subseteq V \setminus \{u, v\}$ such that u and v belong to different components of $G - Z$.



- $\{z_1, z_2\}$ is a (u, v)-separator but it is not minimal since $\{z_2\}$ is also a (u, v)-separator.
- $\{z_1, z_3\}$ is a minimal (u, v)-separator

ILP formulation

- $x_{v,c}$ is 1 if vertex $v \in V$ is in class $c \in [k]$ and 0 otherwise.
- $\Gamma(u, v)$ is the set of all **minimal** (u, v)-separators in G.

$$
\begin{aligned}
&\sum_{c \in [k]} x_{v,c} \le 1 && \forall v \in V, \\
&x_{u,c} + x_{v,c} - \sum_{z \in Z} x_{z,c} \le 1 && \forall \{u, v\} \notin E, Z \in \Gamma(u, v), c \in [k], \\
&x_{v,c} \in \{0, 1\} && \forall v \in V \text{ and } c \in [k].
\end{aligned}
$$

- Let $\mathcal{P}(G, k)$ be the convex hull of the set of solutions of this system.
- We are interested in finding strong valid inequalities for $\mathcal{P}(G, k)$.

Single-class inequalities

- Let $\pi x \leq \pi_0$ be a nontrivial valid inequality for $\mathcal{P}(G,1)$ different from $x_v \leq 1$ for all $v \in V$.
- Let $k \geq 2$ be an integer, let $c \in [k]$, and let $\pi^c x \leq \pi_0$ be a valid inequality of $\mathcal{P}(G,k)$, where the (possibly) non-zero entries of π^c are precisely $\pi^c_{v,c} = \pi_v$ for all $v \in V$.
- It holds that $\pi x \leq \pi_0$ is facet defining for $\mathcal{P}(G,1)$ if and only if $\pi^c x \leq \pi_0$ is facet defining for $\mathcal{P}(G,k)$.

Connectivity inequalities

- Let $u, v \in V$ with $\{u, v\} \notin E$, Z be a (u, v)-separator and $c \in [k]$. The connectivity inequality

$$x_{u,c} + x_{v,c} - \sum_{z \in Z} x_{z,c} \leq 1$$

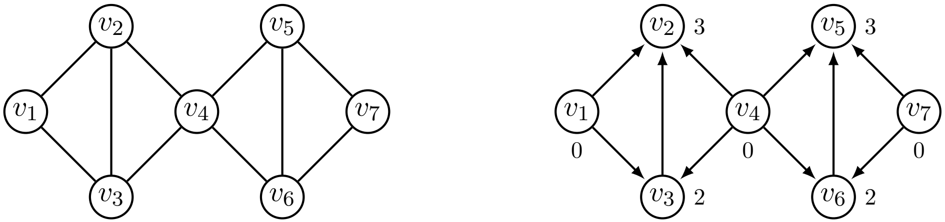
is facet-defining for $\mathcal{P}(G, 1)$ if and only if Z is minimal [Wang et al. (2017)].

- So it is facet-defining for $\mathcal{P}(G, k)$ if and only if Z is minimal.

- Separation problem can be solved in polynomial time ($\mathcal{O}(n^2)$ max-flow instances).

Indegree inequalities

- Consider an **orientation** of the edges of a graph and let $d(v)$ be the **indegree** of v, i.e, the number of arcs entering v.



- For every orientation of G and $c \in [k]$, the following is valid:

$$\sum_{v \in V} (1 - d(v)) x_{v,c} \leq 1 \text{ OR } \sum_{v \in V} x_{v,c} - 1 \leq \sum_{v \in V} d(v) x_{v,c}$$

- In the example above, we get

$$x_{v_1 c} + x_{v_4 c} + x_{v_7 c} \leq 1 + 2x_{v_2 c} + x_{v_3 c} + 2x_{v_5 c} + x_{v_6 c}.$$

Indegree inequalities

- Separation problem can be solved in linear time [Wang et al. (2017)].

$$\sum_{v \in V} x_{v,c} - 1 \leq \sum_{v \in V} d(v) x_{v,c}$$

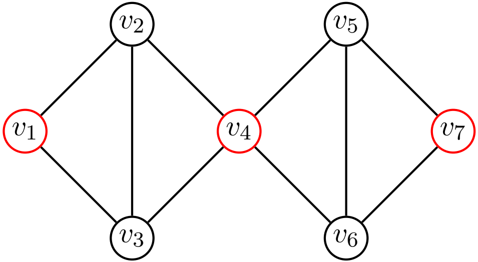
- Describe the convex hull if $k = 1$ and G is a tree [Korte et al. (1991)].

- Facet-defining for $\mathcal{P}(G, 1)$ under conditions [Wang et al. (2017)].

- Hence also facet-defining for $\mathcal{P}(G, k)$ under the same conditions.

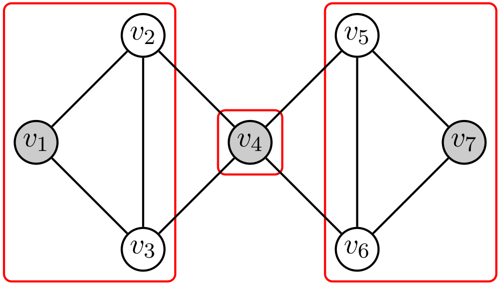
Generalized connectivity (indegree) inequalities

- Let $S \subseteq V$ with $|S| = \ell$ vertices



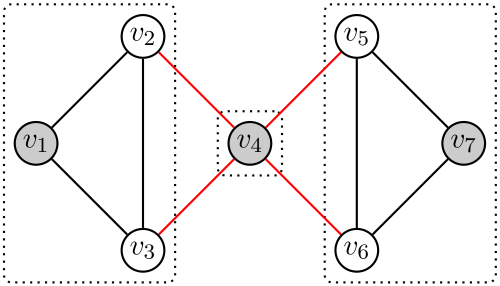
Generalized connectivity (indegree) inequalities

- Let $S \subseteq V$ with $|S| = \ell$ vertices
- Partition $\mathcal{W} = \{W_1, \ldots, W_\ell\}$ of V with $|W_i \cap S| = 1$



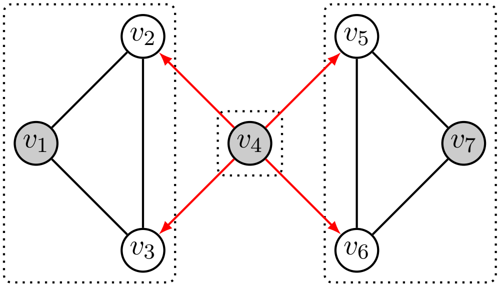
Generalized connectivity (indegree) inequalities

- Let $S \subseteq V$ with $|S| = \ell$ vertices
- Partition $\mathcal{W} = \{W_1, \ldots, W_\ell\}$ of V with $|W_i \cap S| = 1$
- $E(\mathcal{W}) = \{uv \in E \colon u \in W_i, v \in W_j \text{ with } i, j \in [\ell], i \neq j\}$



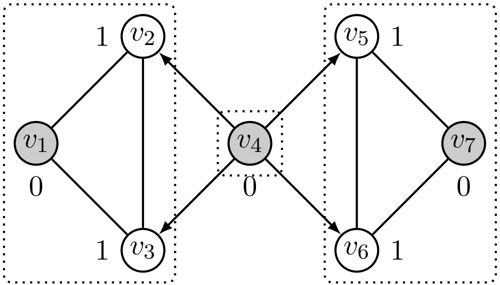
Generalized connectivity (indegree) inequalities

- Let $S \subseteq V$ with $|S| = \ell$ vertices
- Partition $\mathcal{W} = \{W_1, \dots, W_\ell\}$ of V with $|W_i \cap S| = 1$
- $E(\mathcal{W}) = \{uv \in E \colon u \in W_i, v \in W_j \text{ with } i, j \in [\ell], i \neq j\}$
- $\vec{E}(\mathcal{W})$ denotes an orientation of $E(\mathcal{W})$



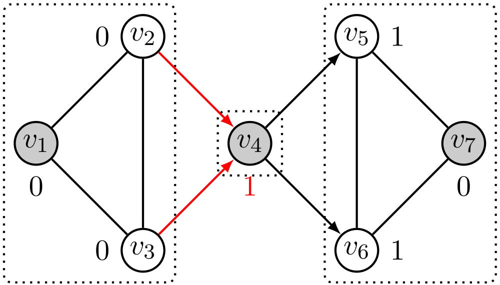
Generalized connectivity (indegree) inequalities

- Let $S \subseteq V$ with $|S| = \ell$ vertices
- Partition $\mathcal{W} = \{W_1, \ldots, W_\ell\}$ of V with $|W_i \cap S| = 1$
- $E(\mathcal{W}) = \{uv \in E \colon u \in W_i, v \in W_j \text{ with } i, j \in [\ell], i \neq j\}$
- $\vec{E}(\mathcal{W})$ denotes an orientation of $E(\mathcal{W})$
- Define $\hat{d}(v) = |\{j \in [\ell] : (u, v) \in \vec{E}(\mathcal{W}) \text{ and } u \in W_j\}|$



Generalized connectivity (indegree) inequalities

- Let $S \subseteq V$ with $|S| = \ell$ vertices
- Partition $\mathcal{W} = \{W_1, \dots, W_\ell\}$ of V with $|W_i \cap S| = 1$
- $E(\mathcal{W}) = \{uv \in E \colon u \in W_i, v \in W_j$ with $i, j \in [\ell], i \neq j\}$
- $\vec{E}(\mathcal{W})$ denotes an orientation of $E(\mathcal{W})$
- Define $\hat{d}(v) = |\{j \in [\ell] : (u, v) \in \vec{E}(\mathcal{W})$ and $u \in W_j\}|$

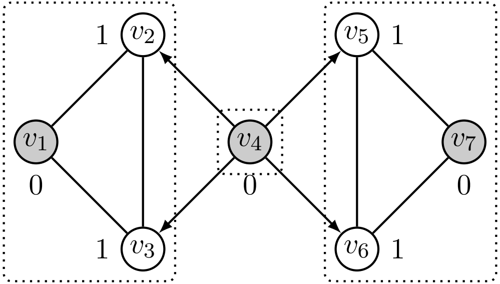


Generalized connectivity (indegree) inequalities

- The generalized connectivity inequality is

$$\sum_{v \in S} (1 - \hat{d}(v)) x_{v,c} - \sum_{v \in V \setminus S} \hat{d}(v) x_{v,c} \leq 1$$

and is valid for $\mathcal{P}(G, k)$.



$$x_{v_1,c} + x_{v_4,c} + x_{v_7,c} \leq 1 + x_{v_2,c} + x_{v_3,c} + x_{v_5,c} + x_{v_6,c}$$

Generalized connectivity (indegree) inequalities

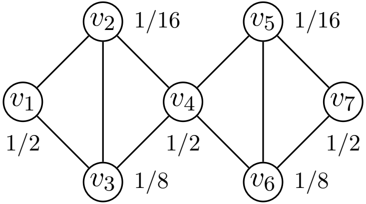
$$\sum_{v \in S} (1 - \hat{d}(v)) x_{v,c} - \sum_{v \in V \setminus S} \hat{d}(v) x_{v,c} \leq 1$$

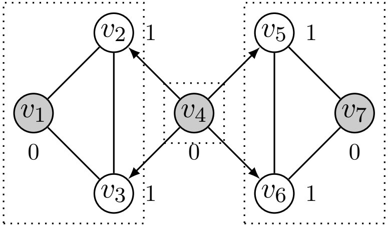
- Generalizes both connectivity ($S = \{u, v\}$) and indegree inequalities ($S = V$).
- Facet-defining for $\mathcal{P}(G, k)$ under some conditions.
- Separation problem can be solved in $\mathcal{O}(k|V||E|)$ (if $\mathcal{W}$ is given) via min vertex cover of bipartite graphs.

Generalized connectivity (indegree) inequalities

- Fractional solution x satisfies all connectivity and indegree inequalities.
- But it violates the generalized inequality

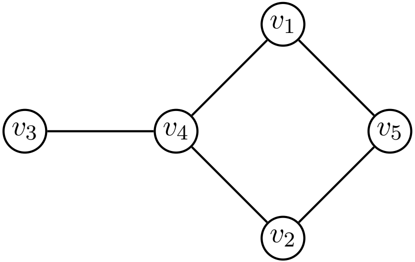
$$x_{v_1,c} + x_{v_4,c} + x_{v_7,c} \leq 1 + x_{v_2,c} + x_{v_3,c} + x_{v_5,c} + x_{v_6,c}$$





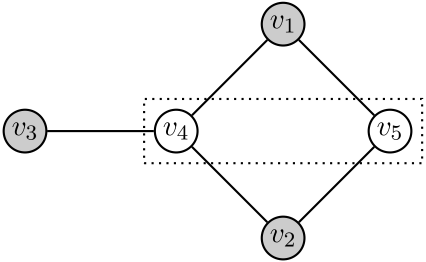
Multiway inequalities

- a stable set $S \subset V$
- a *multiway cut* of S in G is set of vertices $Z \subseteq V \setminus S$ s.t. each component of $G - Z$ contains at most one vertex in S



Multiway inequalities

- a stable set $S \subset V$
- a *multiway cut* of S in G is set of vertices $Z \subseteq V \setminus S$ s.t. each component of $G - Z$ contains at most one vertex in S
- stable set $S = \{v_1, v_2, v_3\}$ and multiway cut $Z = \{v_4, v_5\}$



- The corresponding multiway inequality for $C = \{a, b\}$ is

$$\sum_{i=1}^{3} (x_{v_i,a} + x_{v_i,b}) \leq 2 + \sum_{i=4}^{5} (x_{v_i,a} + x_{v_i,b})$$

Multiway inequalities

Proposition

Let $C \subseteq [k]$ with $|C| \geq 1$, $S \subseteq V$ be a stable set of G, Z be a multiway cut of S in G, and $\beta := \max\{|S| - |C|, 0\}$. The following inequality is valid for $\mathcal{P}(G, k)$.

$$\sum_{v \in S} \sum_{c \in C} x_{v,c} - \sum_{z \in Z} \sum_{c \in C} \beta x_{z,c} \leq |C|$$

- Characterization of the facet-defining inequalities
- Separation problem is NP-hard (if S is given)

Computation

Maximum-Weight Subgraph Problem
INPUT: A graph G, a vertex-weight function $w\colon V \to \mathbb{Z}$ (possibly with negative weights), and $k \in \mathbb{Z}_{\geq}$.
OUTPUT: A subgraph H of G with at most k connected components.
OBJECTIVE: Maximize $w(H) := \sum_{v \in V(H)} w(v)$.

- random and bipartite graphs with 100 vertices
- $k \in \{5, 10, 15, 20, 25\}$
- different densities
- five instances for each k and density
- implemented in C++ using LEMON Graph Lib. 1.3.1, and Gurobi 10
- time limit 900 seconds

Computation

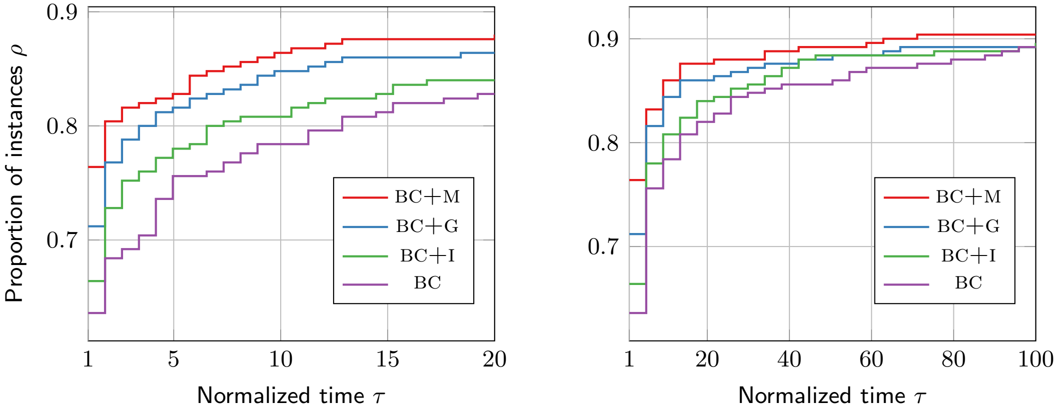


Figure: Performance profiles for the random graph instances.

Final gaps

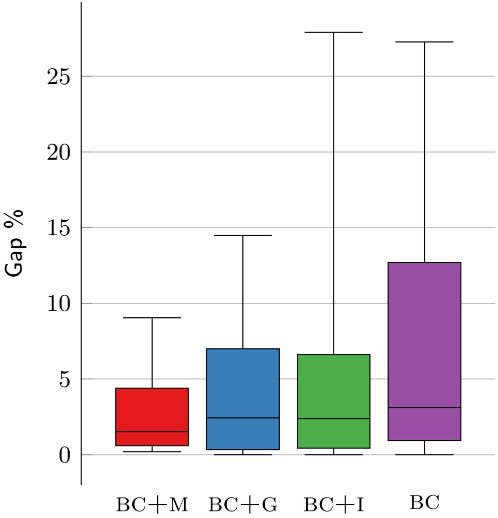


Figure: Gaps for the random graph instances.

Power system restoration

Co-authors:

Hatice Çalık
KU Leuven, Department of Electrical Engineering, ELECTA

Dirk Van Hertem
KU Leuven, Department of Electrical Engineering, ELECTA

Paper:

H. Çalık, D. Van Hertem and H. Yaman (2025). Novel exact solution methods for efficient parallel power system restoration.

Iberian Blackout 2025

- ‘Multifactorial’ technical reasons
- No electricity or mobile network
- 35000 rail passengers evacuated
- 500 flights cancelled
- At least 5 deaths
- Restoration: +16 hours (Spain)
- +1.6 billion euro

Power System Restoration - PSR

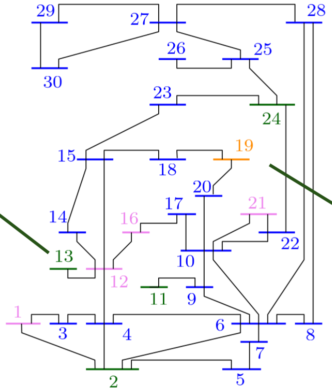
NBS generators

External (cranking) power

Ramping period

BS generators

Can restart themselves



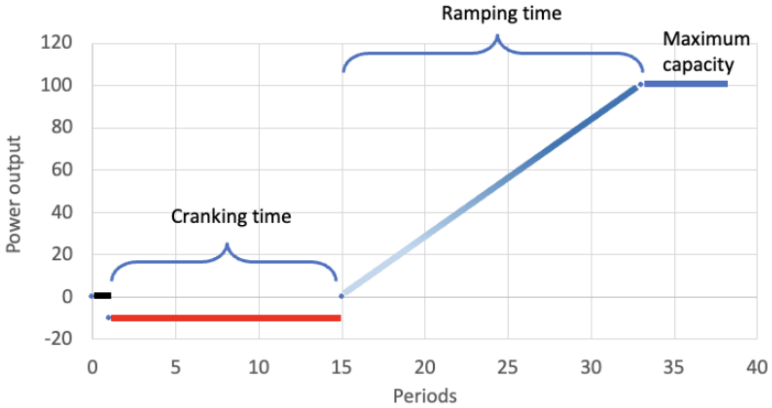
TR – transshipment

No power demand or generation

CL – critical load

External power

NBS generators



Power System Restoration

- Minimize the latest start-up period
- Net power is nonnegative at each period
- Partition the network into islands
- One island per BS generator
- Each component assigned to exactly one island
- Each island is connected

Generator start-up sequencing problem

Single island problem:

$$
\begin{aligned}
\min \quad & \sum_{k \in K} r_k u_k \\
\text{s.t.} \quad & \sum_{k \in K} u_k = 1, \\
& \sum_{t \in T} x_{it} = 1 && \forall i \in N, \\
& \sum_{m=1}^{r_k} x_{im} \geq \sum_{j \in K : j \leq k} u_j && \forall i \in N, k \in K, \\
& \sum_{b \in B} p_{t0}^b + \sum_{i \in N} \sum_{m=1}^{t} p_{tm}^i x_{im} \geq \sum_{i \in L} c_i \sum_{k \in K : r_k \leq t} u_k && \forall t \in T, \\
& x_{it} \in \{0, 1\} && \forall i \in N, t \in T, \\
& u_k \in \{0, 1\} && \forall k \in K.
\end{aligned}
$$

Binary search

For a given radius r, we have a feasibility problem:

$$
\begin{aligned}
F(r) \qquad & \sum_{m=1}^{r} x_{im} = 1 && \forall i \in N, \\
& \sum_{b \in B} p_{t0}^{b} + \sum_{i \in N} \sum_{m=1}^{t} p_{tm}^{i} x_{im} \geq 0 && \forall t \in [r-1], \\
& \sum_{b \in B} p_{r0}^{b} + \sum_{i \in N} \sum_{m=1}^{r} p_{rm}^{i} x_{im} \geq \sum_{i \in L} c_i, \\
& x_{it} \in \{0,1\} && \forall i \in N, t \in [r].
\end{aligned}
$$

The critical loads start at the last period.

Islanding and connectivity

Assign nodes to islands

$$
\begin{aligned}
& a_{bb} = 1, && \forall b \in B \\
& a_{vb} = 0, && \forall b, v \in B : v \neq b \\
& \sum_{b \in B} a_{ib} = 1, && \forall i \in I \\
& \sum_{b \in B} a_{ib} \leq 1, && \forall i \in H \\
& a_{ib} \in \{0, 1\}, && \forall i \in V, b \in B
\end{aligned}
$$

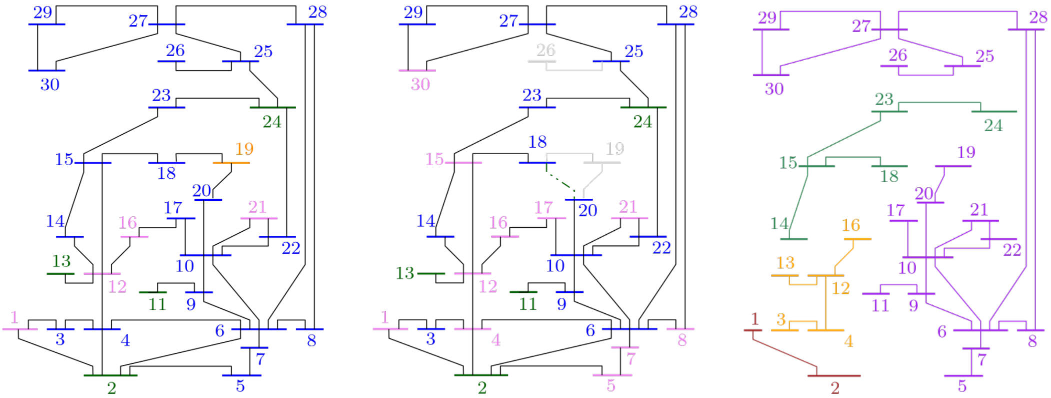
Ensure connectivity for each island

$$
a_{ub} - \sum_{j \in Z} a_{jb} \leq 0, \qquad \forall b \in B, u \in V : (u, b) \notin E, Z \subset \mathcal{Z}_{ub}
$$

Solution method and results

- Enhancement strategies:
  - Valid inequalities, optimality cuts, variable fixing and elimination
  - Network reduction rules
- Improved bounds and solving times for PPSR
  - We solve IEEE 30, 118 and 300 instances much faster
  - We obtain the best-known bounds for 500-bus much faster
  - New improved lower bounds for 1354- and 1888-bus instances

IEEE 30 network before and after reduction and an optimal solution



(a) Before reduction. (b) After reduction. (c) An optimal solution.

$B = \{2, 11, 13, 24\}$. (a) $L = \{1, 12, 16, 21\}$, $H = \{19\}$. (b) $L = \{1, 4, 5, 7, 8, 12, 15, 16, 17, 21, 30\}$.

Balanced connected partitions of edge-weighted graphs

Co-authors:

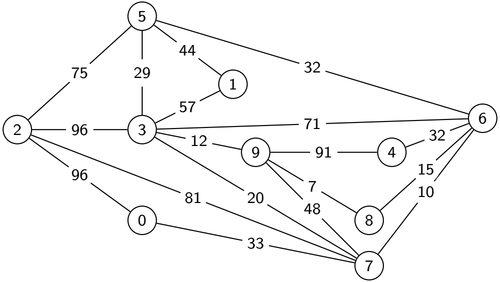
Morteza Davari
SKEMA Business School

Phablo Moura
KU Leuven, ORSTAT

Paper:

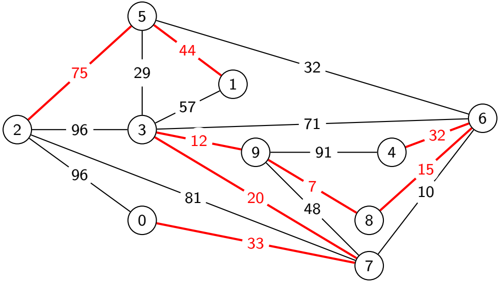
M. Davari, P. F. S. Moura, H. Yaman (2025). Balanced connected partitions of edge-weighted graphs: Hardness and solving methods. arXiv [Cs.DS].

Balanced Spanning Forest



We want to find a collection of k trees (a spanning forest) while minimizing the weight of the heaviest tree.

Balanced Spanning Forest



Tree 1: (1 5)(2 5) w: 119

Tree 2: (8 9)(3 9)(6 8)(3 7)(4 6)(0 7) w: 119

Complexity

Min-Max Balanced Spanning Forest (min-max-bsf)
INSTANCE: Connected graph G, $k \in \mathbb{Z}_{\geq}$, and $w\colon E \to \mathbb{Q}_{\geq}$.
OBJECTIVE: Collection of k vertex-disjoint trees $\{T_i\}_{i\in[k]}$ spanning V that minimizes $\max_{i\in[k]} w(T_i)$, where $w(T_i) := \sum_{e\in E(T)} w(e)$ for all $i \in [k]$.

- The decision version of MIN-MAX-BSF is weakly $\mathcal{NP}$-complete on complete graphs with $k = 2$.

- The decision version of MIN-MAX-BSF is $\mathcal{NP}$-complete on unweighted bipartite graphs for each fixed $k \geq 2$.

Approximation

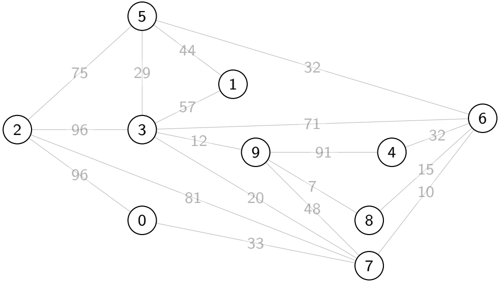
MIN-MAX-BSF admits a tight k-approximation algorithm, which runs in $\mathcal{O}(m \log n)$ time on graphs with n vertices and m edges.

Steps:

1. Compute a *minimum spanning tree*.
2. Remove the heaviest $k - 1$ edges in the tree.

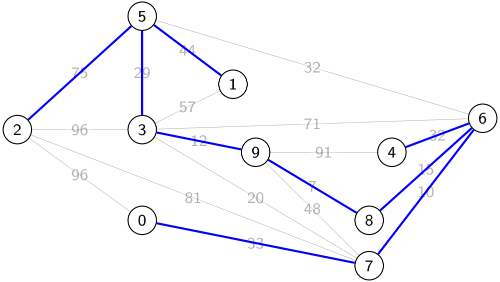
A simple k-approximation

For $k = 2$



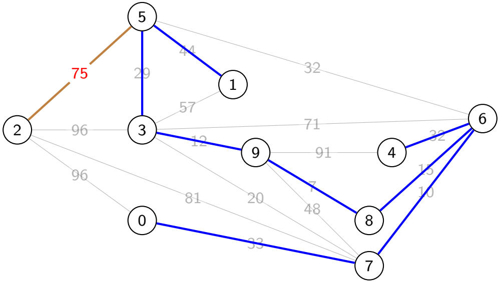
A simple k-approximation

For $k = 2$



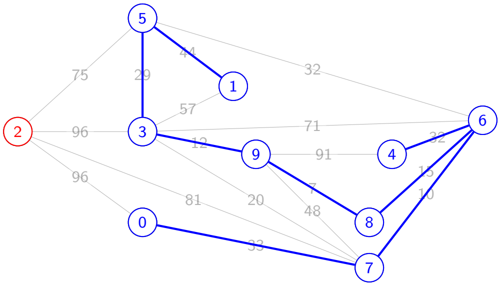
A simple k-approximation

For $k = 2$



A simple k-approximation

For $k = 2$



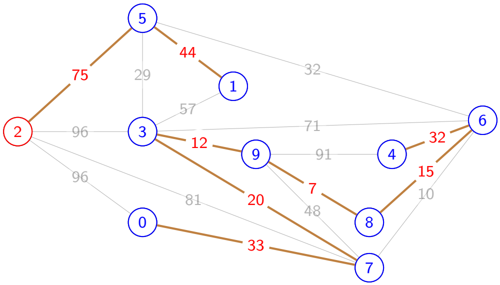
the resulting forest:

one single node tree the other tree covering the remaining nodes

tree weights: 0 182

A simple k-approximation

For $k = 2$



the resulting forest:

one single node tree the other tree covering the remaining nodes

tree weights: 0 182

optimal forest: 119, 119

Heuristic

- Starting from the initial graph with no forbidden edges, compute a minimum spanning tree T excluding the forbidden set of edges H.
- If the lower bound $\mathrm{LB} = \lceil w(T)/k \rceil$ is less than the current global upper bound UB, we use Vaishali et al.'s algorithm to find a minimum spanning k-forest F for T, from which an upper bound UB is computed.
- The algorithm proceeds by generating child nodes where each edge $e \in T$ is added to the forbidden set.

Flow formulation

$$
\begin{aligned}
\min\ & \omega & \\
\text{s.t. } & \omega \geq f_{0v} & \forall v \in V, \\
& \sum_{v \in V} y_v = k, & \\
& \sum_{a \in \delta^-(v)} x_a + y_v = 1 & \forall v \in V, \\
& f_{0v} + \sum_{a \in \delta^-(v)} f_a - \sum_{a \in \delta^+(v)} f_a = \sum_{a \in \delta^-(v)} w_a \bar{x}_a & \forall v \in V, \\
& w_a x_a \leq f_a \leq UB x_a & \forall a \in A, \\
& 0 \leq f_{0v} \leq UB y_v & \forall v \in V, \\
& x_a \in \{0,1\} & \forall a \in A, \\
& y_v \in \{0,1\} & \forall v \in V.
\end{aligned}
$$

Set partitioning formulation

$$
\begin{aligned}
&\min \omega \\
\text{s.t. } &\sum_{T\in\mathcal{T}} x_T \le k, \\
&\sum_{T\in\mathcal{T}: v\in V(T)} x_T = 1 && \forall v \in V, \\
&\sum_{T\in\mathcal{T}: v\in V(T)} w(T)\, x_T \le \omega && \forall v \in V, \\
&x_T \in \{0,1\} && \forall T \in \mathcal{T}.
\end{aligned}
$$

Pricing problem

Let $(\hat{\theta}, \hat{\eta}, \hat{\zeta})$ denote an optimal dual solution.

$$\mathrm{PP}: \max \quad \sum_{u \in V} \hat{\eta}_u y_u - \sum_{e \in E} w_e x_e \sum_{u \in V} \hat{\zeta}_u y_u$$

s.t.

$$\sum_{e \in E} x_e = \sum_{u \in V} y_u - 1,$$

$$x_e \leq y_u \qquad \forall e \in E, u \in e,$$

$$\sum_{e \in E} x_e \leq |S| - 1 \qquad \forall S \subseteq V \text{ with } S \neq \emptyset,$$

$$y_u \in \{0, 1\} \qquad \forall u \in V,$$

$$x_e \in \{0, 1\} \qquad \forall e \in E.$$

Solving the pricing problem

Idea 1 : Linearization

- Since $x_e, y_u \in \{0,1\}$, we linearize $x_e y_u$ with a new variable z_{eu} and constraints:

$$z_{eu} \leq x_e, \quad z_{eu} \leq y_u, \quad z_{eu} \geq x_e + y_u - 1.$$

- So

$$\max \quad \sum_{u \in V} \hat{\eta}_u y_u \; - \sum_{e \in E} \sum_{u \in V} w_e \hat{\zeta}_u z_{eu}.$$

- This leads to a super slow column generation.

Solving the pricing problem

Idea 2: Fixing vertices

- Let $B := \{ u \in V \mid \hat{\zeta}_u \neq 0 \}$.
- $|B|$ is usually very small (often ≤ 4).
- For each subset $S \subseteq B$, we obtain a subproblem by enforcing the inclusion of all vertices in S in the tree, while ensuring that all vertices in $B \setminus S$ do not belong to this tree.
- The objective function becomes

$$\max \quad \sum_{u \in V} \hat{\eta}_u y_u - \sum_{e \in E} \left(w_e \sum_{u \in S} \hat{\zeta}_u \right) x_e.$$

- This is a Prize-Collecting Steiner Tree (PCST) problem.
- Fast unless B is large.

Solving the pricing problem

Idea 3: Fixing the weight

- Replace the tree-dependent term $w(T)$ with a fixed parameter $W \in [0, \mathrm{UB}]$.
- We define an approximate reduced cost:

$$\rho'(W, T) = -\hat{\theta} + \sum_{u \in V(T)} \hat{\eta}_u - W \sum_{v \in V(T)} \hat{\zeta}_v.$$

- We add an additional budget constraint $w(T) \leq W$.
- Pricing reduces to solving a sequence of Budgeted Prize-Collecting Steiner Tree (BPCST) instances for $W \in [1, \mathrm{UB}]$.

Computation

- Total of **380 instances** generated.
- Parameters:

$$n \in \{20, 30, 40, 50\}, \quad p \in \{0.1, 0.2, 0.3, 0.4, 0.5\}, \quad k \in \{2, 4, 6, 8, 10\}.$$

- For each (n, p, k): 4 random graphs generated with

$$m = \left\lfloor p \cdot \frac{n(n-1)}{2} \right\rfloor$$

  edges using `boost::generate_random_graph()`.

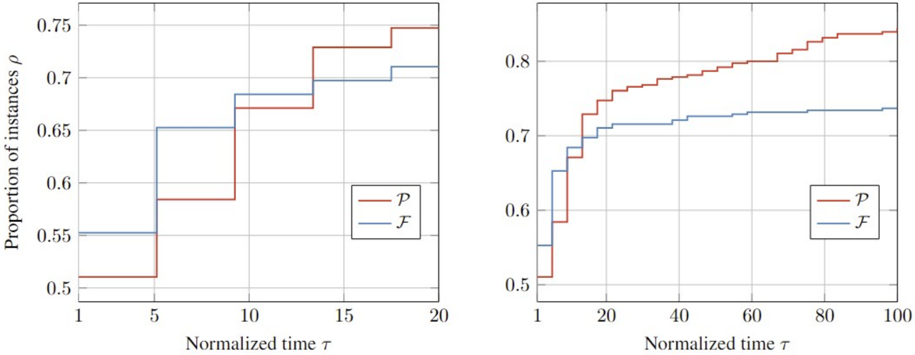
- $\mathcal{F}$: flow formulation
- $\mathcal{P}$: Branch-and-price with set partitioning formulation

Average LP Gaps

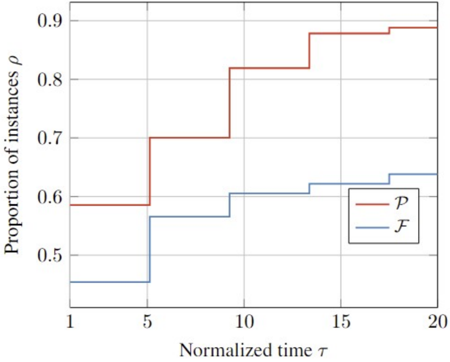
Table: Average relative optimality gaps (%) for different bounds across instance classes. The column generation does not finish within the time limit for the number of instances indicated in parentheses, and so these instances are excluded from the average.

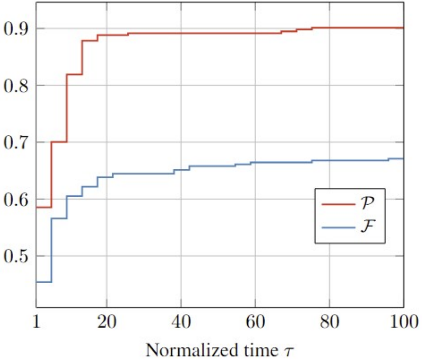
n	# instances	F_{gap}	P_{gap}
20	80	38.92	1.74
30	100	26.06	2.67
40	100	20.65	(3) 3.52
50	100	15.94	(9) 2.46

Performance Profiles (All Instances)

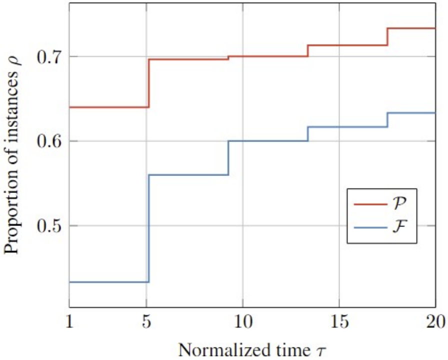


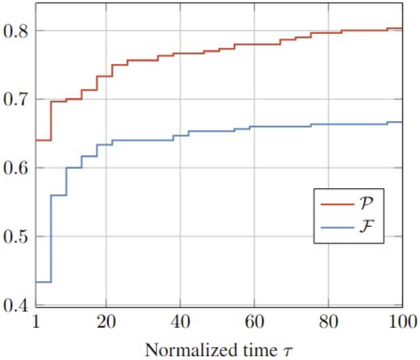
Performance Profiles ($k \geq 4$)





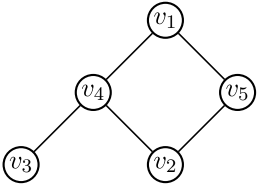
Performance Profiles ($n \geq 30$)





Concluding remarks

- Other inequalities for connected sub-partition polytope: a graph inducing a polytope (with $k = 2$) that has facets not yet identified



- Other definitions of connectivity: length-bounded, signed

- Vertex-weighted version of balanced partitions

- Power restoration networks: location of new BSs, uncertainty regarding renewables, damage, minimizing impact during recovery