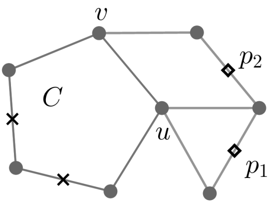
# Modelling of piece-wise linear concave constraints in continous covering problems

Liding Xu, Claudia D'Ambrosio

OptimiX, LIX, École Polytechnique

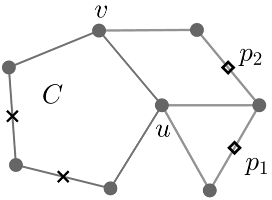
November 14, 2024

# Set covering problems on networks




- Consider a network $N = (V, A)$,
- **Vars:** Put points in edges or nodes.

# Set covering problems on networks




- Consider a network $N = (V, A)$,
- **Vars:** Put points in edges or nodes.
- Each point has the same covering radius $\delta$.
- **Constraints:** Cover all edges and nodes.

# Extensions of set covering problems on networks

$\mathcal{P}$ denotes the set of covering points, and $D$ denotes the set of points to cover. Let $C(N)$ be the set of points in $N$.

# Extensions of set covering problems on networks

$\mathcal{P}$ denotes the set of covering points, and $D$ denotes the set of points to cover. Let $C(N)$ be the set of points in $N$.

- *Discrete:* $\mathcal{P} = D = V$, classical set covering.
- *Semi-continuous:* either $\mathcal{P} = C(N)$ or $D = C(N)$, reduced to the classical set covering.

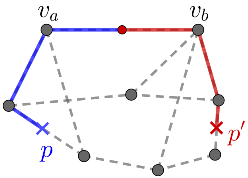
# Extensions of set covering problems on networks

$\mathcal{P}$ denotes the set of covering points, and $D$ denotes the set of points to cover. Let $C(N)$ be the set of points in $N$.

- *Discrete:* $\mathcal{P} = D = V$, classical set covering.
- *Semi-continuous:* either $\mathcal{P} = C(N)$ or $D = C(N)$, reduced to the classical set covering.
- *Continuous:* both $\mathcal{P} = D = C(N)$, the *continuous set covering on networks*.

Objective: we use a subset $\mathcal{P}' \subset \mathcal{P}$ and minimize its cardinality P'.

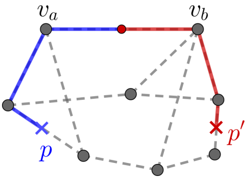
# Set covering problems on networks



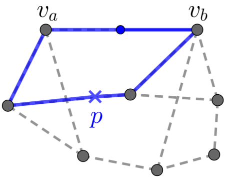

(a) two points $p$ and $p'$

- $d(p, v_a)$: the shortest path distance between points $p$ and $v_a$.
- if $d(p, v_a) \leq \delta$, $v_a$ is covered by $p$.

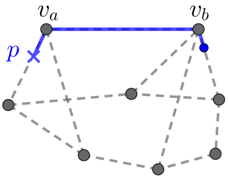
# How to continuously cover an edge




(a) By two points




(b) By one point, through both ends




(c) By one point, trough one end

Figure: Covering of an edge $e = (v_a, v_b) \in E$

# Exact approaches

- (Edge) MILP formulation by Fröhlich et al.: $\mathcal{P}$ is indexed by $A$.
- (Edge-vertex) MILP formulations by Mercedes et al.: $\mathcal{P}$ indexed by $A \cup V$, preprocessing, long-edge reformulation.

# Exact approaches

- (Edge) MILP formulation by Fröhlich et al.: $\mathcal{P}$ is indexed by $A$.
- (Edge-vertex) MILP formulations by Mercedes et al.: $\mathcal{P}$ indexed by $A \cup V$, preprocessing, long-edge reformulation.
- **This work**: compare (edge) reformulations (big-M, disjunctive, indicator).

# The difference comes from

How to model the covering constraint for an edge $e = (v_a, v_b)$:

$$\max_{p \in \mathcal{P}'_{v_a}} (\delta - d(v_a, p)) + \max_{p \in \mathcal{P}'_{v_b}} (\delta - d(v_b, p)) \geq \ell_e, \qquad (1)$$

where $\max_{p \in \mathcal{P}'_v} (\delta - d(v, p))$ is the residual cover truncated at an end node $v$, and $\mathcal{P}'_v$ is a subset of $\mathcal{P}$ that can cover $v$.

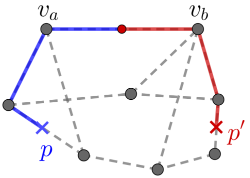
# The difference comes from

How to model the covering constraint for an edge $e = (v_a, v_b)$:

$$\max_{p \in \mathcal{P}'_{v_a}} (\delta - d(v_a, p)) + \max_{p \in \mathcal{P}'_{v_b}} (\delta - d(v_b, p)) \geq \ell_e, \tag{1}$$

where $\max_{p \in \mathcal{P}'_v} (\delta - d(v, p))$ is the residual cover truncated at an end node $v$, and $\mathcal{P}'_v$ is a subset of $\mathcal{P}$ that can cover $v$.

# Affine component of distance function




(a) residual cover of $v_a$ in $p$

Identify the point $p$ by its "home" edge $e' = (v'_a, v'_b)$, label $p$ by its coordinate, a continuous variable $q_{e'} \in [0, \ell_{e'}]$. Define the "residual cover of $v$" function through $v'_a$ or $v'_b$ in $q_{e'}$:

$$\tau_{ve'i'} : q_{e'} \mapsto \tau_{ve'i'}(q_{e'}) := d(v, v'_i) + \mathbf{1}_{i'=a} q_{e'} + \mathbf{1}_{i'=b}(\ell_{e'} - q_{e'}). \quad (2)$$

The function is affine in $q_{e'}$.

# Max-affine representation of distance function

Then, we can express the $\delta - d(v,p)$ as the maximum of two affine functions:

$$\delta - d(v,p) = \max_{(e',a'),(e',b')} (\delta - \tau_{ve'i'}(q_{e'}))$$

# Max-affine representation of distance function

Then, we can express the $\delta - d(v, p)$ as the maximum of two affine functions:

$$\delta - d(v, p) = \max_{(e', a'), (e', b')} (\delta - \tau_{ve'i'}(q_{e'}))$$

Let $\mathcal{S}$ be a subset of $\mathcal{P}'_v \times \{a, b\}$ such that for all $(e', i') \in \mathcal{S}$, $d(v, v'_i) \leq \delta$.

# Piece-wise linear concavity

The covering constraint has a reformulation:

$$\max_{(e',i')\in\mathcal{S}_{v_a}} (\delta - \tau_{v_a e' i'}(q_{e'}) + \max_{(e',i')\in\mathcal{S}_{v_b}} (\delta - \tau_{v_b e' i'}(q_{e'}) \geq \ell_e, \quad (3)$$

This constraint is a piece-wise linear concave constraint.
We try to overestimate the convex function in the left.

# First approach: indicator constraint

Represent each max function by an indicator constraint.
Then, the covering constraint reads as:

$$
\begin{aligned}
& \sum_{(e',i') \in \mathcal{S}_v} z_{ve'i'} = 1 & & \\
& z_{ve'i'} \Rightarrow r_v \leq \delta - \tau_{ve'i'}(q_{e'}) & (e', i') \in \mathcal{S}_v & \\
& z_{ve'i'} \in \{0, 1\} & (e', i') \in \mathcal{S}_v & \qquad (4) \\
& \ell_e \leq r_{v_a} + r_{v_b} & e \in E & \\
& q_{e'} \in [0, \ell_{e'}] & e' \in E & \\
& r_v \geq 0 & v \in V. &
\end{aligned}
$$

Let solvers handle it.

# Second approach: big-M technique

Using big-M to represent each possible argmax.

$$r_v \leq M_{ve'i'}(1 - z_{ve'i'}) + \delta - \tau_{ve'i'}(q_{e'}) \quad (5)$$

One can set $M \approx \delta$, we have a method to search tight big-Ms such that $M \approx \ell_e$!

# Third approach: disjunctive technique

Using disjunctive technique to model the max of affine functions.
Where is the disjunctive system?

# Third approach: disjunctive technique

Using disjunctive technique to model the max of affine functions.
Where is the disjunctive system?
$\max_{(e',i')\in\mathcal{S}_v} (\delta - \tau_{ve'i'}(q_{e'})$ is equivalent to:

$$\bigvee_{(e',i')\in\mathcal{S}_v} \left[ \begin{array}{c} 0 \le r_v \le \delta - \tau_{ve'i'}(q_{e'}) \\ 0 \le q_{e'} \le \ell_{e'} \end{array} \right]. \qquad (6)$$

Only one clause could be true.

# Third approach: disjunctive technique

Disjunctive programming gives a MILP reformulation (in lifted space) without big-M:

- whose LP relaxation is tight (convex hull);
- but with additional variables.

# Third approach: disjunctive technique

We define the following affine function, similar to $\tau_{ve'i'}$ in (2):

$$R_{ve'i'} : (w, y) \mapsto R_{ve'i'}(w, y) := (\delta - d(v, v_i') - \mathbf{1}_{i'=b}\ell_{e'})y + (\mathbf{1}_{i'=b} - \mathbf{1}_{i'=a})w. \quad (7)$$

Define

$$\mathcal{S}^{-1}(e') := \{(v, i') : \exists i' \in \{a, b\}, (e', i') \in \mathcal{S}(v)\}. \quad (8)$$

# Third approach: disjunctive technique

The reformulation:

$$
\begin{aligned}
& r_v = \sum_{(e',i')\in\mathcal{S}_v} r_{ve'i'} \\
& q_{e'} = q_{ve'} + \sum_{i':(e',i')\in\mathcal{S}_v} q_{ve'i'} && e' \in (v) \\
& r_{ve'i'} \le R_{ve'i'}(q_{ve'i'}, z_{ve'i'}) && (e',i') \in \mathcal{S}_v \\
& q_{ve'i'} \le z_{ve'i'}\ell_{e'} && (e',i') \in \mathcal{S}_v \\
& q_{ve'} \le \left(1 - \sum_{i':(e',i')\in\mathcal{S}_v} z_{ve'i'}\right)\ell_{e'} && e' \in (v) \\
& r_{ve'i'}, q_{ve'i'} \ge 0 && (e',i') \in \mathcal{S}_v \\
& q_{e'}, q_{ve'} \ge 0 && e' \in (v)
\end{aligned}
\tag{9}
$$

# Experiments

- Implementation in Julia-JuMP and using CPLEX.

# Experiments

- Implementation in Julia-JuMP and using CPLEX.
- Algorithmic options:
  - EF-P: the big-M formulation for the edge model;
  - EF-PI: the indicator formulation of the edge model;
  - EF-PD: the disjunctive programming formulation.

# Performance metric

- the relative dual gap is defined as:

$$\sigma := \frac{\overline{v} - \underline{v}}{\overline{v}},$$

  where $\overline{v}$ is an upper-bound and $\underline{v}$ is a lower-bound.
- the relative primal bound

$$v_r := \frac{\overline{v}}{n_{sd}},$$

- $t$: the total running time in CPU seconds.
- S/A: the number of solved instances/ the number of affected instances.

# Benchmarks

Two benchmarks: Small and Large.

| Statistics | Small | Large |
|---|---|---|
| Number of instances | 32 | 24 |
| Min number of edges | 9 | 185 |
| Medium number of edges | 69 | 699 |
| Max number of edges | 148 | 1035 |
| Average number of edges | 71 | 584 |
| Average graph density | 137.5 | 1162.1 |

Table: The statistics of the benchmarks

# Experimental results I

| Formulation | Small radius | | | | Large radius | | | |
|---|---|---|---|---|---|---|---|---|
| | t | $\sigma$ | $v_r$ | S/A | t | $\sigma$ | $v_r$ | S/A |
| EF-P | 269.4 | 21.0% | 30.3% | 11/31 | 21.1 | 18.0% | 15.4% | 28/32 |
| EF-PI | 287.6 | 20.2% | 28.9% | 10/32 | 24.1 | 20.3% | 15.7% | 28/32 |
| EF-PD | 284.4 | 17.4% | 30.2% | 12/31 | 42.9 | 9.6% | 15.5% | 26/32 |

Table: Results for the Small benchmark (32 instances)

Disjunctive formulation is the best.

# Experimental results II

| Formulation | Small radius | | | | Large radius | | | |
|---|---|---|---|---|---|---|---|---|
| | t | $\sigma$ | $v_r$ | S/A | t | $\sigma$ | $v_r$ | S/A |
| EF-P | 1800.8 | 57.0% | 63.3% | 0/24 | 1631.5 | 53.4% | 39.2% | 3/24 |
| EF-PI | 1801.0 | 61.3% | 65.4% | 0/24 | 1608.0 | 55.2% | 56.8% | 1/24 |
| EF-PD | 1800.3 | 67.9% | 69.5% | 0/13 | 1630.8 | 63.3% | 40.4% | 2/13 |

Table: Results for the Large benchmark

Big-M formulation is the best. In practice, solvers may use big-M to reformulate indicator constrains, but their values are usually too loose.

# Conclusion

- New edge model formulations for continuous set-covering on networks.
- For different scales of problems, a receipt for choosing the best formulations.
- The solver may not handle concave piece-wise linear functions properly, though lot of techniques have been studied.