Minimum-Power Routing: Part I

Peng-Jun Wan

wan@cs.iit.edu

Outline

- Introduction
- Aggregation Routing
- Minimum Submodular Cover With Submodular Cost
- Broadcast Routing
  - Plane Geometric Networks
  - Symmetric Networks
  - Asymmetric Networks

Network model

- Communication topology: $D = (V, A; c)$
- Adjustable transmission power
- Fixed received power cost: $q \in \mathbb{R}_+^V$

Routings for Primitive Communication Tasks

- Unicast: a directed path

Routings for Primitive Communication Tasks

- Unicast: a directed path
- Aggregation: a spanning in-arborescence

Routings for Primitive Communication Tasks

- Unicast: a directed path
- Aggregation: a spanning in-arborescence
- Broadcast: a spanning out-arborescence

Routings for Primitive Communication Tasks

- Unicast: a directed path
- Aggregation: a spanning in-arborescence
- Broadcast: a spanning out-arborescence
- Multicast: a Steiner out-arborescence

Power Consumption

For $D' = (V', A') \subseteq D$,

- Transmission power consumption
  - For each $u \in V'$: $p_{D'}(u) = \max\left\{c(a) : a \in \delta_{D'}^{out}(u)\right\}$
  - Total: $p(D') = \sum_{u \in V'} p_{D'}(u)$

Power Consumption

For $D' = (V', A') \subseteq D$,

- Transmission power consumption
  - For each $u \in V'$: $p_{D'}(u) = \max\{c(a) : a \in \delta^{out}_{D'}(u)\}$
  - Total: $p(D') = \sum_{u \in V'} p_{D'}(u)$
- Receiving power consumption
  - For each $u \in V'$: $q_{D'}(u) = \deg^{in}_{D'}(u) \cdot q(u)$
  - Total: $q(D') = \sum_{u \in V'} q_{D'}(u)$

Power Consumption

For $D' = (V', A') \subseteq D$,

- Transmission power consumption
  - For each $u \in V'$: $p_{D'}(u) = \max\left\{c(a) : a \in \delta_{D'}^{out}(u)\right\}$
  - Total: $p(D') = \sum_{u \in V'} p_{D'}(u)$
- Receiving power consumption
  - For each $u \in V'$: $q_{D'}(u) = \deg_{D'}^{in}(u) \cdot q(u)$
  - Total: $q(D') = \sum_{u \in V'} q_{D'}(u)$
- Power consumption of D': $p(D') + q(D')$

Three Types of Networks

- Asymmetric Networks
  - for some $(u, v) \in A$, either $(v, u) \notin A$ or $(v, u) \in A$ but $c(u, v) \neq c(v, u)$

Three Types of Networks

- Asymmetric Networks
  - for some $(u, v) \in A$, either $(v, u) \notin A$ or $(v, u) \in A$ but $c(u, v) \neq c(v, u)$
- Symmetric Networks
  - $(u, v) \in A$ implies $(v, u) \in A$ and $c(u, v) = c(v, u)$

Three Types of Networks

- Asymmetric Networks
  - for some $(u, v) \in A$, either $(v, u) \notin A$ or $(v, u) \in A$ but $c(u, v) \neq c(v, u)$
- Symmetric Networks
  - $(u, v) \in A$ implies $(v, u) \in A$ and $c(u, v) = c(v, u)$
- Plane Geometric Networks
  - symmetric
  - $\overline{D}$ is a R-disk graph for some $R > 0$
  - $c(u, v) = c(v, u) = \|uv\|^{\kappa}$

Summary on Routing Algorithms

- Receiving power consumption is ignored
  - for clarity and simplicity
  - minor modification

Summary on Routing Algorithms

- Receiving power consumption is ignored
  - for clarity and simplicity
  - minor modification
- Unicast: shortest path, and hence not covered

Summary on Routing Algorithms

- Receiving power consumption is ignored
  - for clarity and simplicity
  - minor modification
- Unicast: shortest path, and hence not covered
- Aggregation: polynomial optimal algorithms

Summary on Routing Algorithms

- Receiving power consumption is ignored
  - for clarity and simplicity
  - minor modification
- Unicast: shortest path, and hence not covered
- Aggregation: polynomial optimal algorithms
- Broadcast: approximations in three types of networks

Summary on Routing Algorithms

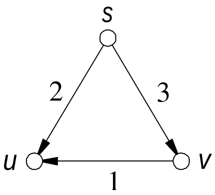
- Receiving power consumption is ignored
  - for clarity and simplicity
  - minor modification
- Unicast: shortest path, and hence not covered
- Aggregation: polynomial optimal algorithms
- Broadcast: approximations in three types of networks
- Multicast: approximations in three types of networks

Roadmap

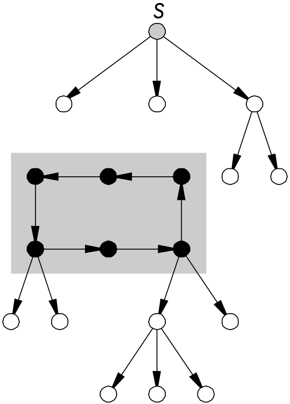
- Introduction
- **Aggregation Routing**
- Minimum Submodular Cover With Submodular Cost
- Broadcast Routing
    - Plane Geometric Networks
    - Symmetric Networks
    - Asymmetric Networks

Aggregation Routing

- Min-power aggregation routing: Min-weight spanning in-arborescence
  - symmetric networks: min-weight spanning tree
- Equivalent to min-weight spanning out-arborescence by taking reverse
  - subsequently all arborescences are out-arborescences
- Failure of the greedy method



Orphan Strong Component



A Recursive Algorithm: Base

$A_0 \leftarrow \{a \in A : c(a) = 0\}$;
If (V, A_0) contains no orphan strong component,
    $T \leftarrow$ an out-arborescence of (V, A_0) rooted at v;
    output T;

A Recursive Algorithm: Recursion

$K \leftarrow$ an orphan strong component of (V, A_0);
$\mu \leftarrow \min\left\{c(a) : a \in \delta_D^{in}(K)\right\}$;
for each $a \in \delta^{in}(K)$, $c'(a) \leftarrow c(a) - \mu$;
for each $a \notin \delta^{in}(K)$, $c'(a) \leftarrow c(a)$;
find **recursively** a MSA T' w.r.t. c';
modify T' to a MSA T w.r.t. c' s.t. T enters K *exactly once*;
output T.

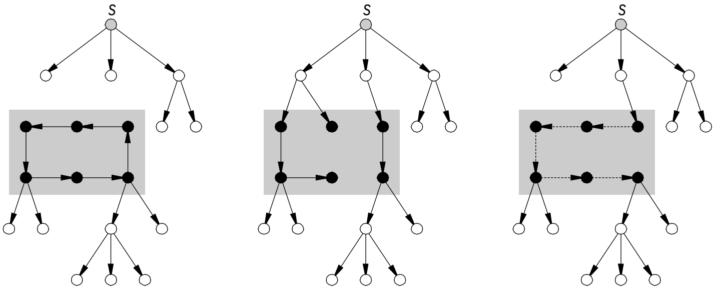
A Recursive Algorithm: Recursion

$K \leftarrow$ an orphan strong component of (V, A_0);
$\mu \leftarrow \min\left\{c(a) : a \in \delta_D^{in}(K)\right\}$;
for each $a \in \delta^{in}(K)$, $c'(a) \leftarrow c(a) - \mu$;
for each $a \notin \delta^{in}(K)$, $c'(a) \leftarrow c(a)$;
find **recursively** a MSA T' w.r.t. c';
modify T' to a MSA T w.r.t. c' s.t. T enters K *exactly once*;
output T.

Optimality: for a MSA $T^* = (V, F^*)$ w.r.t. c,

$$\begin{aligned} c(T^*) &= c'(T^*) + \mu\left|F^* \cap \delta_D^{in}(K))\right| \\ &\geq c'(T^*) + \mu \geq c'(T) + \mu = c(T). \end{aligned}$$

$v \leftarrow$ a node in K of minimum depth in T';
$T'' \leftarrow$ an out-arborescence of K rooted at v;
$T \leftarrow T' - \left\{\delta^{in}_{T'}(u) : u \in K \setminus \{v\}\right\} + T''$;



(a) (b) (c)

Roadmap

- Introduction
- Aggregation Routing
- **Minimum Submodular Cover With Submodular Cost**
- Broadcast Routing
    - Plane Geometric Networks
    - Symmetric Networks
    - Asymmetric Networks

Submodular Set Functions

$f : 2^E \to \mathbb{R}$ for some ground set E.
Marginal values:

$$\Delta_B f(A) = f(A \cup B) - f(A),$$
$$\Delta_x f(A) = f(A \cup \{x\}) - f(A)$$

Submodular Set Functions

$f : 2^E \to \mathbb{R}$ for some ground set E.
Marginal values:

$$\Delta_B f(A) = f(A \cup B) - f(A),$$
$$\Delta_x f(A) = f(A \cup \{x\}) - f(A)$$

Submodular Set Functions

$f : 2^E \to \mathbb{R}$ for some ground set E.
Marginal values:

$$\Delta_B f(A) = f(A \cup B) - f(A),$$
$$\Delta_x f(A) = f(A \cup \{x\}) - f(A)$$

f is submodular if for any $A \subseteq E$ and *distinct* $x, y \in E \setminus A$.

$$\Delta_x f(A) \geq \Delta_x f(A \cup \{y\}).$$

Submodular Set Functions

$f : 2^E \to \mathbb{R}$ for some ground set E.
Marginal values:

$$\Delta_B f(A) = f(A \cup B) - f(A),$$
$$\Delta_x f(A) = f(A \cup \{x\}) - f(A)$$

f is submodular if for any $A \subseteq E$ and *distinct* $x, y \in E \setminus A$.

$$\Delta_x f(A) \geq \Delta_x f(A \cup \{y\}).$$

Submodular Set Functions

$f : 2^E \to \mathbb{R}$ for some ground set E.
Marginal values:

$$\Delta_B f(A) = f(A \cup B) - f(A),$$
$$\Delta_x f(A) = f(A \cup \{x\}) - f(A)$$

f is submodular if for any $A \subseteq E$ and *distinct* $x, y \in E \setminus A$.

$$\Delta_x f(A) \geq \Delta_x f(A \cup \{y\}).$$

Decreasing marginal values: diminishing returns

Submodular Set Functions

$f : 2^E \to \mathbb{R}$ for some ground set E.
Marginal values:

$$\Delta_B f(A) = f(A \cup B) - f(A),$$
$$\Delta_x f(A) = f(A \cup \{x\}) - f(A)$$

f is submodular if for any $A \subseteq E$ and *distinct* $x, y \in E \setminus A$.

$$\Delta_x f(A) \geq \Delta_x f(A \cup \{y\}).$$

Decreasing marginal values: diminishing returns
Alternatively:

$$f(X \cup Y) + f(X \cap Y) \leq f(X) + f(Y), \forall X, Y \subseteq E.$$

Increasing And Submodular Functions

f is *increasing* if $\Delta_x f(A) \geq 0$ for any $A \subseteq E$ and $x \in E \setminus A$.

Increasing And Submodular Functions

f is *increasing* if $\Delta_x f(A) \geq 0$ for any $A \subseteq E$ and $x \in E \setminus A$.
$\Leftrightarrow$ for any $A \subseteq B \subseteq E$, $f(A) \leq f(B)$

Increasing And Submodular Functions

f is *increasing* if $\Delta_x f(A) \geq 0$ for any $A \subseteq E$ and $x \in E \setminus A$.
$\Leftrightarrow$ for any $A \subseteq B \subseteq E$, $f(A) \leq f(B)$

f is increasing and submodular $\Leftrightarrow$ for any $A, B \subseteq E$,

$$f(B) - f(A) \leq \sum_{x \in B \setminus A} \Delta_x f(A).$$

Increasing And Submodular Functions

f is *increasing* if $\Delta_x f(A) \geq 0$ for any $A \subseteq E$ and $x \in E \setminus A$.
$\Leftrightarrow$ for any $A \subseteq B \subseteq E$, $f(A) \leq f(B)$

f is increasing and submodular $\Leftrightarrow$ for any $A, B \subseteq E$,

$$f(B) - f(A) \leq \sum_{x \in B \setminus A} \Delta_x f(A).$$

f is a *polymatroid* function if f is increasing, submodular, and $f(\emptyset) = 0$.

Power

$D = (V, A; c)$: a weighted digraph.
$f : 2^A \longrightarrow \mathbb{R}$ defined by

$$f(B) = \text{power of } (V, B), \forall B \subseteq A.$$

Power

$D = (V, A; c)$: a weighted digraph.
$f : 2^A \to \mathbb{R}$ defined by

$$f(B) = \text{power of } (V, B), \forall B \subseteq A.$$

For each $a \in A$ with tail u,

$$\Delta_a f(B) = \max \left\{ 0, c(a) - \max_{b \in B \cap \delta_D^{out}(u)} c(b) \right\}.$$

Graph Matroid Rank

$D = (V, A)$: a digraph.
$f : 2^A \to \mathbb{R}$ defined by

$$f(B) = |V| - \# \text{ of weak components of } (V, B), \forall B \subseteq A.$$

Graph Matroid Rank

$D = (V, A)$: a digraph.
$f : 2^A \to \mathbb{R}$ defined by

$$f(B) = |V| - \#\text{ of weak components of } (V, B), \forall B \subseteq A.$$

For each $a = (u, v) \in A$, $\Delta_a f(B) = 0$ if u and v belong to the same weak component in (V, B), and 1 otherwise.

Graph Matroid Rank

$G = (V, E)$: an undirected graph.
$f : 2^E \to \mathbb{R}$ defined by

$$f(B) = |V| - \# \text{ of components of } (V, B), \forall B \subseteq A.$$

Graph Matroid Rank

$G = (V, E)$: an undirected graph.
$f : 2^E \to \mathbb{R}$ defined by

$$f(B) = |V| - \# \text{ of components of } (V, B), \forall B \subseteq A.$$

For each $e \in E$, $\Delta_e f(B) = 0$ if both ends of e belong to the same component in (V, B), and 1 otherwise.

Vertex Domination

$G = (V, E)$: an undirected graph.
$f : 2^V \to \mathbb{R}$

$$f(U) = |N[U]|, \forall U \subseteq V$$

Vertex Domination

$G = (V, E)$: an undirected graph.
$f : 2^V \to \mathbb{R}$

$$f(U) = |N[U]|, \forall U \subseteq V$$

$$\Delta_v f(U) = |N[v] \setminus N[U]|.$$

Undirected cut

$G = (V, E)$: an undirected graph.
$f : 2^V \to \mathbb{R}$

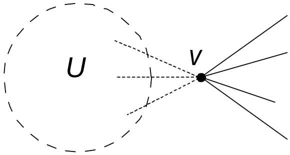
$$f(U) = |\delta(U)|, \forall U \subseteq V$$

Undirected cut

$G = (V, E)$: an undirected graph.
$f : 2^V \to \mathbb{R}$

$$f(U) = |\delta(U)|, \forall U \subseteq V$$

$$\Delta_v f(U) = |N(v) \setminus U| - |N(v) \cap U| = \deg(v) - 2|N(v) \cap U|.$$



Directed cut

$D = (V, A)$: a digraph.
$f : 2^V \to \mathbb{R}$

$$f(U) = \left|\delta^{out}(U)\right|, \forall U \subseteq V$$

Directed cut

$D = (V, A)$: a digraph.
$f : 2^V \to \mathbb{R}$

$$f(U) = \left|\delta^{out}(U)\right|, \forall U \subseteq V$$

$$\begin{aligned}\Delta_v f(U) &= \left|N^{out}(v) \setminus U\right| - \left|N^{in}(v) \cap U\right| \\ &= \deg^{out}(v) - \left(\left|N^{out}(v) \cap U\right| + \left|N^{out}(v) \cap U\right|\right).\end{aligned}$$

Affine Combination

f, g: submodular on 2^E. c: a nonnegative constant

- Multiplication by positive scalar: cf.

Affine Combination

f, g: submodular on 2^E. c: a nonnegative constant

- Multiplication by positive scalar: cf.
- Summation: $f + g$

Mapping

f: submodular on 2^E.
$g : S \to 2^E$. Define $g(X) = \cup_{x \in X} g(x)$ for any $X \subseteq S$.
Then, $f \circ g$ is submodular on 2^S.

Mapping

f: submodular on 2^E.
$g : S \rightarrow 2^E$. Define $g(X) = \cup_{x \in X} g(x)$ for any $X \subseteq S$.
Then, $f \circ g$ is submodular on 2^S.

$$
\begin{aligned}
& f \circ g(A) + f \circ g(B) \\
& = f(g(A)) + f(g(B)) \\
& \geq f((g(A) \cup g(B))) + f(g(A) \cap g(B)) \\
& \geq f(g(A \cup B)) + f(g(A \cap B)).
\end{aligned}
$$

Integral Submodular Cover with Submodular Cost

f: a integer-valued polymatroid function on 2^E;
g: a polymatroid function on 2^E.
The minimization problem

$$\min \{g(X) : X \subseteq E, f(X) = f(E)\}.$$

Greedy Algorithm

GSC:

$X \leftarrow \emptyset$;

While $f(X) < f(E)$ do

select $x \in E$ with minimum $g(x) / \Delta_x f(X)$;

$X \leftarrow X \cup \{x\}$;

Output X.

Approximation Ratio

Define the curvature of c to be

$$\rho = \min_{S\text{: min-cost cover}} \frac{\sum_{e \in S} g(e)}{g(S)}.$$

Approximation Ratio

Define the curvature of c to be

$$\rho = \min_{S\text{: min-cost cover}} \frac{\sum_{e \in S} g(e)}{g(S)}.$$

Theorem

The approximation ratio of **GSC** *is at most* $\rho H(\gamma)$ *where* $\gamma = \max_{e \in E} f(e)$.

Proof by A Charging Argument

S: a cover of min-cost cover satisfying that

$$\sum_{e \in S} g(e) = \rho \cdot g(S).$$

$x_1, x_2, \cdots, x_k$: the sequence of elements selected by the greedy algorithm. Let $X_0 = \emptyset$, and for each $1 \leq i \leq k$ let X_i be the first i elements. $\mu_i = \frac{g(x_i)}{\Delta_{x_i} f(X_{i-1})}$: the average price per increment of coverage by the i-th element. Then

$$\mu_1 \leq \mu_2 \leq \cdots \leq \mu_k.$$

Proof by A Charging Argument

S: a cover of min-cost cover satisfying that

$$\sum_{e \in S} g(e) = \rho \cdot g(S).$$

$x_1, x_2, \cdots, x_k$: the sequence of elements selected by the greedy algorithm. Let $X_0 = \emptyset$, and for each $1 \leq i \leq k$ let X_i be the first i elements. $\mu_i = \frac{g(x_i)}{\Delta_{x_i} f(X_{i-1})}$: the average price per increment of coverage by the i-th element. Then

$$\mu_1 \leq \mu_2 \leq \cdots \leq \mu_k.$$

In the iteration i, we charge each $e \in S$ with $\mu_i (\Delta_e f(X_{i-1}) - \Delta_e f(X_i))$.

Proof by A Charging Argument

- $\sum_{i=1}^{k} g(x_i)$ is no more than the total charge on S.
- The total charge on $e \in S$ is at most $H(\gamma) g(e)$.

Proof by A Charging Argument

- $\sum_{i=1}^{k} g(x_i)$ is no more than the total charge on S.
- The total charge on $e \in S$ is at most $H(\gamma) g(e)$.

Hence,

$$g(X) \leq \sum_{i=1}^{k} g(x_i) \leq H(\gamma) \sum_{e \in S} g(e) = \rho H(\gamma) \cdot g(S).$$

Roadmap

- Introduction
- Aggregation Routing
- Minimum Submodular Cover With Submodular Cost
- **Broadcast Routing**
  - Plane Geometric Networks
  - Symmetric Networks
  - Asymmetric Networks

Min-Power Broadcast Routing

- NP-hard

Min-Power Broadcast Routing

- NP-hard
- Approximality:
  - Plane geometric: $O(1)$
  - Symmetric/Asymmetric: no $(1-\varepsilon)\ln n$-approx. for any $\varepsilon > 0$ unless P=NP

Min-Power Broadcast Routing

- NP-hard
- Approximality:
    - Plane geometric: $O(1)$
    - Symmetric/Asymmetric: no $(1-\varepsilon)\ln n$-approx. for any $\varepsilon > 0$ unless P=NP
- Approx. bounded achieved:
    - Plane geometric: 6
    - Symmetric: $2H(\Delta)$
    - Asymmetric: $2H(n-1)-1$

Roadmap

- Introduction
- Aggregation Routing
- Minimum Submodular Cover With Submodular Cost
- **Broadcast Routing**
  - **Plane Geometric Networks**
  - Symmetric Networks
  - Asymmetric Networks

Min-Power Broadcast Routing in Plane Geometric Networks

- NP-hard
- Two constant-approximations
    - MST-based heuristic
    - Broadcast Incremental power (BIP)

A Lower Bound on Minimum Power

opt: the power required by broadcast from s.
MST: an Eucildean minimum spanning tree

Lemma

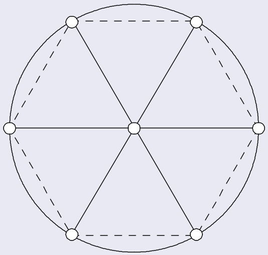
$opt \geq c\left(MST\right)/6$.

A Property of Euclidean MST

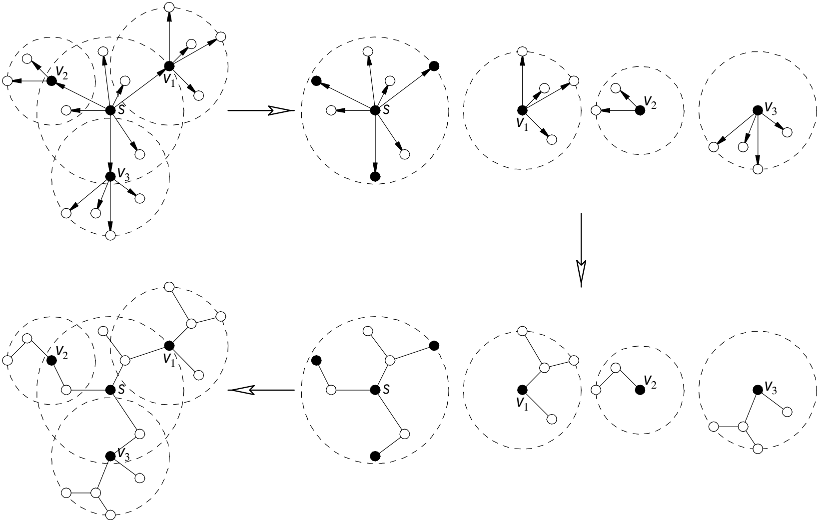
Theorem

Let $\mathcal{D}$ is a unit-disk centered at u. Then, for any finite point set $V \subseteq \mathcal{D}$ which contains u and any Euclidean MST (V, E) of V,

$$\sum_{e \in E} \|e\|^2 \leq 6.$$



Proof of The Lower Bound



MST-Based Broadcast Routing

MSA: the out-arborescence rooted at s oriented from MST.

$$p(MSA) \leq c(MSA) = c(MST) \leq 6opt.$$

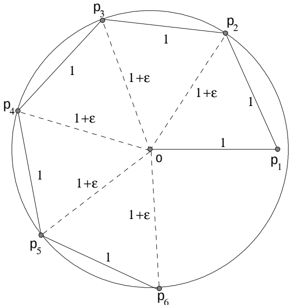
MST-Based Broadcast Routing

MSA: the out-arborescence rooted at s oriented from *MST*.

$$p\,(MSA) \leq c\,(MSA) = c\,(MST) \leq 6opt.$$

Theorem

The approximation ratio of the MSA heuristic is 6



Broadcasting Incremental Power (BIP)

BIP:
$T \leftarrow (\{s\}, \emptyset)$;
$U \leftarrow \{s\}$;
For $i = 1$ to $|V| - 1$,
  Choose $a = (u, v) \in \delta^{out}(U)$ minimizing $c(a) - p_T(u)$;
  $T \leftarrow T + a$;
  $U \leftarrow U \cup \{v\}$;
Output T.

Approx. Ratio of BIP

Theorem

The approximation ratio of the BIP heuristic is between $\frac{13}{3}$ *and* 6.

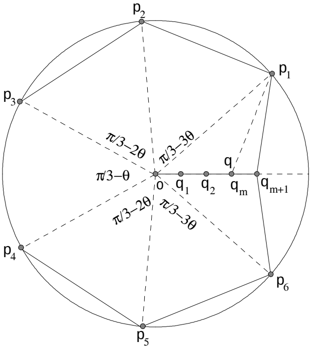


Figure: A bad instance for BIP.

Approx. Ratio of BIP

Lemma

The power of the BIP arborescence is at most $c(MST)$.

Approx. Ratio of BIP

Lemma

The power of the BIP arborescence is at most $c(MST)$.

- T: the BIP arborescence.
- $a_1, a_2, \cdots, a_{n-1}$: the sequence of the arcs selected.
- e_i for $1 \leq i \leq n-1$: the edge between the ends of a_i
- $\overline{D} = (V, E; c)$

Approx. Ratio of BIP

- Define $c' \in R_+^E$ by
  - $c'(e_i) =$ the incremental power of a_i, for $1 \leq i \leq n-1$
  - $c'(e) = c(e)$ for any $e \in E \setminus \{e_i : 1 \leq i \leq n-1\}$

- Define $c' \in R_+^E$ by
  - $c'(e_i)$ = the incremental power of a_i, for $1 \leq i \leq n-1$
  - $c'(e) = c(e)$ for any $e \in E \setminus \{e_i : 1 \leq i \leq n-1\}$

Then, $\overline{T}$ is a minimum spanning tree of $(V, E; c')$.

Approx. Ratio of BIP

- Define $c' \in R_{+}^{E}$ by
  - $c'(e_i) =$ the incremental power of a_i, for $1 \leq i \leq n-1$
  - $c'(e) = c(e)$ for any $e \in E \setminus \{e_i : 1 \leq i \leq n-1\}$

Then, $\overline{T}$ is a minimum spanning tree of $(V, E; c')$.
Hence,

$$p(T) = c'(T) = c'\left(\overline{T}\right) \leq c'(MST) \leq c(MST).$$

Roadmap

- Introduction
- Aggregation Routing
- Minimum Submodular Cover With Submodular Cost
- **Broadcast Routing**
  - Plane Geometric Networks
  - **Symmetric Networks**
  - Asymmetric Networks

Two-Phased Approximation

Phase 1: Compute a weakly-connected spanning digraph D' with $p(D') \leq H(\Delta)\, opt$

Two-Phased Approximation

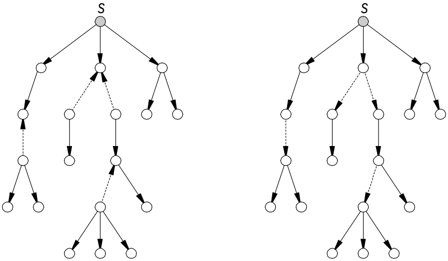
Phase 1: Compute a weakly-connected spanning digraph D' with $p(D') \leq H(\Delta)\, opt$

Phase 2: Compute a spanning tree of $\overline{D'}$ and orient it to an arborescence T rooted at s

Two-Phased Approximation

Phase 1: Compute a weakly-connected spanning digraph D' with $p(D') \leq H(\Delta)\, opt$

Phase 2: Compute a spanning tree of $\overline{D'}$ and orient it to an arborescence T rooted at s



(a) (b)

$$p(T) \leq 2p(D') \leq 2H(\Delta)\, opt$$

Phase 1: Formulation into MSC

For any $a = (u, v) \in A$, define

$$S(a) = \left\{ b \in \delta^{out}(u) : c(b) \leq c(a) \right\}.$$

Phase 1: Formulation into MSC

For any $a = (u, v) \in A$, define

$$S(a) = \left\{ b \in \delta^{out}(u) : c(b) \leq c(a) \right\}.$$

For each $B \subseteq A$,define

$$\begin{aligned} S(B) &= \cup_{a \in B} S(a), \\ f(B) &= n - \# \text{ of weak components of } (V, S(B)), \\ g(B) &= \text{power of } (V, S(B)). \end{aligned}$$

Phase 1: Formulation into MSC

For any $a = (u, v) \in A$, define

$$S(a) = \left\{ b \in \delta^{out}(u) : c(b) \leq c(a) \right\}.$$

For each $B \subseteq A$,define

$$\begin{aligned} S(B) &= \cup_{a \in B} S(a), \\ f(B) &= n - \# \text{ of weak components of } (V, S(B)), \\ g(B) &= \text{power of } (V, S(B)). \end{aligned}$$

MSC formulation:

$$\min \{ g(B) : B \subseteq A, f(B) = f(A) \}$$

.

Approximation Bound

$\max_{a \in A} f(a) = \Delta.$

Approximation Bound

$\max_{a \in A} f(a) = \Delta$.

Curvature of g: $\rho = 1$, as there is an optimal solution which is the union of directed stars with distinct roots.

Approximation Bound

$\max_{a \in A} f(a) = \Delta$.

Curvature of g: $\rho = 1$, as there is an optimal solution which is the union of directed stars with distinct roots.

Theorem

The approximation ratio of the greedy algorithm is at most $H(\Delta)$.

Roadmap

- Introduction
- Aggregation Routing
- Minimum Submodular Cover With Submodular Cost
- **Broadcast Routing**
  - Plane Geometric Networks
  - Symmetric Networks
  - **Asymmetric Networks**

Greedy Algorithm

GBA:

$B \leftarrow \emptyset$;

while $f(B) > 0$,

  find a cheapest $T \in \mathcal{T}(B)$;

  $B \leftarrow B \cup T$;

output BFS arborescence of (V, B) rooted at s.

Greedy Algorithm

GBA:
$B \leftarrow \emptyset$;
while $f(B) > 0$,
find a cheapest $T \in \mathcal{T}(B)$;
$B \leftarrow B \cup T$;
output BFS arborescence of (V, B) rooted at s.

$f(B) = \#$ of orphan components of (V, B)

Greedy Algorithm

GBA:
$B \leftarrow \emptyset$;
while $f(B) > 0$,
find a cheapest $T \in \mathcal{T}(B)$;
$B \leftarrow B \cup T$;
output BFS arborescence of (V, B) rooted at s.

$f(B) = \#$ of orphan components of (V, B)
Head of an orphan component: node of smallest ID.

Greedy Algorithm

GBA:
  $B \leftarrow \emptyset$;
  while $f(B) > 0$,
    find a cheapest $T \in \mathcal{T}(B)$;
    $B \leftarrow B \cup T$;
  output BFS arborescence of (V, B) rooted at s.

$f(B) = \#$ of orphan components of (V, B)
Head of an orphan component: node of smallest ID.

$$\text{price of } T = \frac{p(T)}{\# \text{ of heads in } T}$$

Greedy Algorithm

> **GBA**:
> $B \leftarrow \emptyset$;
> while $f(B) > 0$,
> find a cheapest $T \in \mathcal{T}(B)$;
> $B \leftarrow B \cup T$;
> output BFS arborescence of (V, B) rooted at s.

$f(B) = \#$ of orphan components of (V, B)
Head of an orphan component: node of smallest ID.

$$\text{price of } T = \frac{p(T)}{\# \text{ of heads in } T}$$

$\mathcal{T}(B)$: a set of at most $|A| \cdot f(B)$ candidates

Candidate Arborescences

For each $u \in V$, denote

$$L(u) = \left\{ c(a) : a \in \delta^{out}(u) \right\}.$$

$\mathcal{T}(B, u, l)$: at most $f(B)$ candidates supplied by $u \in V$ at the power level $l \in L(u)$

Candidate Arborescences

For each $u \in V$, denote

$$L(u) = \left\{ c(a) : a \in \delta^{out}(u) \right\}.$$

$\mathcal{T}(B, u, l)$: at most $f(B)$ candidates supplied by $u \in V$ at the power level $l \in L(u)$

$\mathcal{T}(B)$: the set of all candidates given by

$$\mathcal{T}(B) = \bigcup_{u \in V} \bigcup_{l \in L(u)} \mathcal{T}(B, u, l).$$

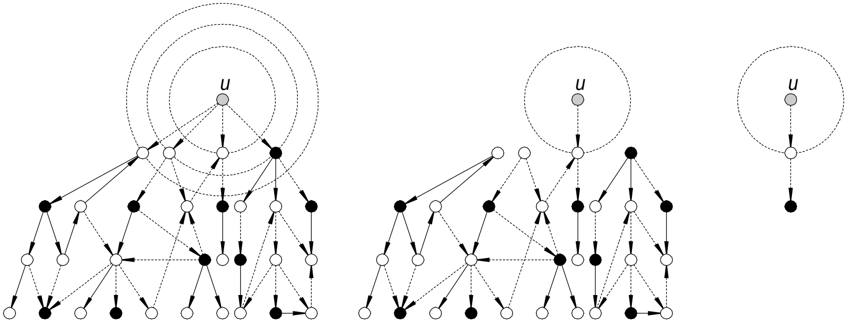
Candidate Arborescences Supplied by u at Power Level l

- $D(B, u, l)$: a weighted digraph obtained from D by:
  - removing all arcs $a \in \delta^{out}(u)$ with $c(a) > l$ and
  - resetting the weights of all arcs $a \in \delta^{out}(u)$ with $c(a) \leq l$ to zero.

Candidate Arborescences Supplied by u at Power Level l

- $D(B, u, l)$: a weighted digraph obtained from D by:
  - removing all arcs $a \in \delta^{out}(u)$ with $c(a) > l$ and
  - resetting the weights of all arcs $a \in \delta^{out}(u)$ with $c(a) \leq l$ to zero.
- U: the set of heads reachable from u in $D(B, u, l)$.

- $D(B, u, l)$: a weighted digraph obtained from D by:
  - removing all arcs $a \in \delta^{out}(u)$ with $c(a) > l$ and
  - resetting the weights of all arcs $a \in \delta^{out}(u)$ with $c(a) \leq l$ to zero.
- U: the set of heads reachable from u in $D(B, u, l)$.
- If $U = \emptyset$ or if $|U| = 1$ and $u \neq s$, then $\mathcal{T}(B, u, l) = \emptyset$.



(a) (b) (c)

Candidate Arborescences Supplied by u at Power Level l

- Otherwise,
  - $T(B, u, l)$: the shortest-path arborescence in $D(B, u, l)$ from u to U

Candidate Arborescences Supplied by u at Power Level l

- Otherwise,
  - $T(B,u,l)$: the shortest-path arborescence in $D(B,u,l)$ from u to U
  - sort U in the increasing order of distance in $T(B,u,l)$

Candidate Arborescences Supplied by u at Power Level l

- Otherwise,
  - $T(B,u,l)$: the shortest-path arborescence in $D(B,u,l)$ from u to U
  - sort U in the increasing order of distance in $T(B,u,l)$
  - $T_h(B,u,l)$ for each $1 \leq h \leq |U|$: the minimal aborescence in $T(B,u,l)$ spanning u and the first h heads.

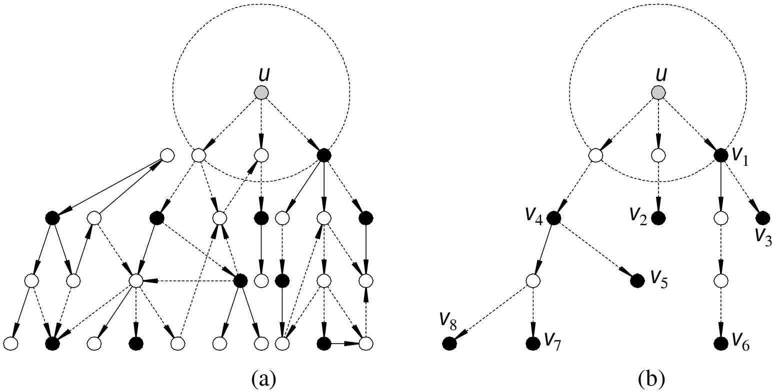
- Otherwise,
  - $T(B, u, l)$: the shortest-path arborescence in $D(B, u, l)$ from u to U
  - sort U in the increasing order of distance in $T(B, u, l)$
  - $T_h(B, u, l)$ for each $1 \leq h \leq |U|$: the minimal aborescence in $T(B, u, l)$ spanning u and the first h heads.
  - If $u = s$, then,

$$\mathcal{T}(B, u, l) = \{T_h(B, u, l) : 1 \leq h \leq |U|\};$$

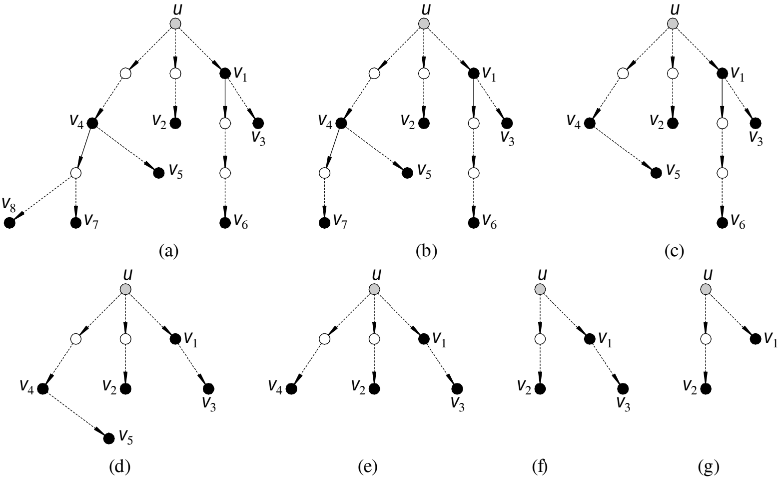
otherwise,

$$\mathcal{T}(B, u, l) = \{T_h(B, u, l) : 2 \leq h \leq |U|\}.$$

Candidate Arborescences Supplied by u at Power Level l



Candidate Arborescences Supplied by u at Power Level l



Approximation Ratio

Theorem

The approximation ratio of **GBA** *is at most* $2H(n-1)-1$.

Gain of A Candidate Arborescence

Lemma

Suppose that $f(B) > 0$. *Then for any* $T \in \mathcal{T}(B)$ *containing* h *heads w.r.t.* B,

$$f(B) - f(B \cup T) \geq h/2.$$

Gain of A Candidate Arborescence

Lemma

Suppose that $f(B) > 0$. Then for any $T \in \mathcal{T}(B)$ containing h heads w.r.t. B,

$$f(B) - f(B \cup T) \geq h/2.$$

u: root of T.

If $u = s$, then

$$f(B) - f(B \cup T) \geq h \geq h/2.$$

Gain of A Candidate Arborescence

Lemma

Suppose that $f(B) > 0$. Then for any $T \in \mathcal{T}(B)$ containing h heads w.r.t. B,

$$f(B) - f(B \cup T) \geq h/2.$$

u: root of T.

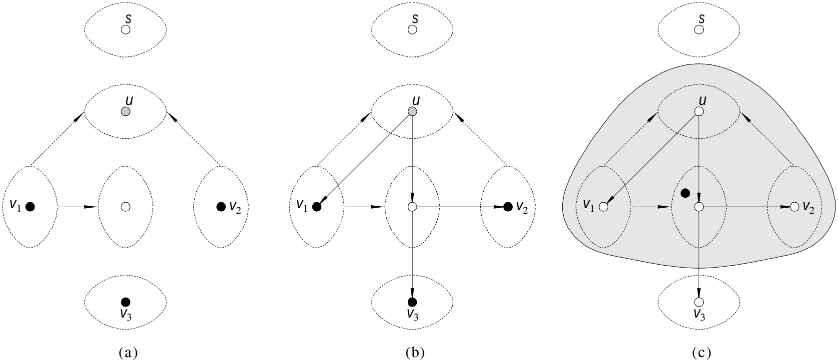
If $u = s$, then

$$f(B) - f(B \cup T) \geq h \geq h/2.$$

If $u \neq s$, then $h \geq 2$ and

$$f(B) - f(B \cup T) \geq h - 1 \geq h/2.$$

Gain of A Candidate Arborescence

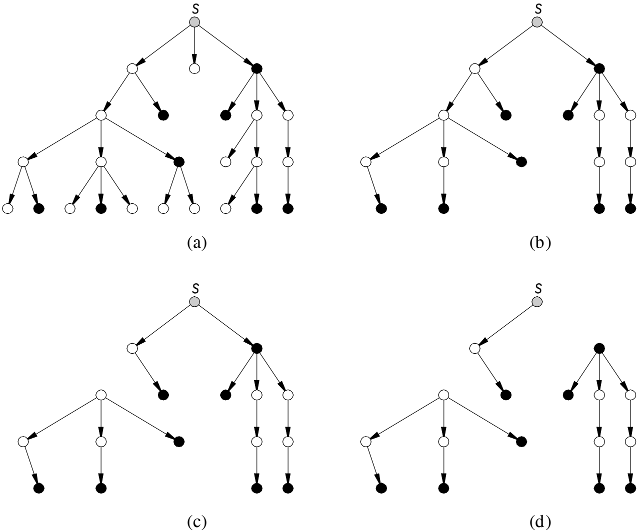


Price of The Cheapest Candidate Arborescence

Lemma

Suppose that $f(B) > 0$. Then the price of the cheapest arborescence in $\mathcal{T}(B)$ is at most $\frac{opt}{f(B)}$.

Spider Decomposition of An Optimal Arborescence



Price of The Cheapest Spider

$S_1, S_2, \cdots, S_t$: spiders .
h_j for each $1 \leq j \leq t$: the number of heads in S_j.
S: the cheapest one among $S_1, S_2, \cdots, S_t$.

$$\sum_{j=1}^{t} h_j = f(B) \text{ and } \sum_{j=1}^{t} p(S_j) \leq opt$$

$$\Rightarrow \text{price of } S \leq \min_{1 \leq j \leq t} \frac{p(S_j)}{h_j} \leq \frac{\sum_{j=1}^{t} p(S_j)}{\sum_{j=1}^{t} h_j} \leq \frac{opt}{f(B)}$$

.

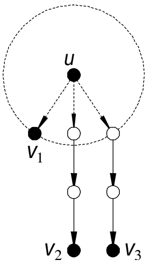
Power of The Cheapest Spider

u: root of S

$v_1, v_2, \cdots, v_h$: heads in S.

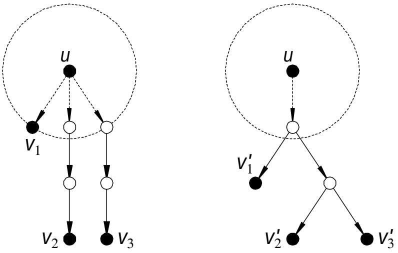
$l = p_S(u) \Rightarrow S \subseteq D(B, u, l)$

$$p(S) = l + \sum_{i=1}^{h} dist_S(u, v_i).$$



$v_1', v_2', \cdots, v_h'$: heads in $T_h(B, u, l)$.

$$
\begin{aligned}
& p\left(T_h\left(B, u, l\right)\right) \\
& \leq l + \sum_{i=1}^{h} dist_{D(B,u,l)}\left(u, v_i'\right) \\
& \leq l + \sum_{i=1}^{h} dist_{D(B,u,l)}\left(u, v_i\right) \\
& \leq l + \sum_{i=1}^{h} dist_{S}\left(u, v_i\right) \\
& = p\left(S\right).
\end{aligned}
$$



(a) (b)

Price of The Cheapest Candidate Arborescence

price of the cheapest candidate arboresence

$$
\begin{aligned}
&\leq \text{price of } T_h(B,u,l) = \frac{p(T_h(B,u,l))}{h} \\
&\leq \frac{p(S)}{h} = \text{price of } S \\
&\leq \frac{opt}{f(B)}.
\end{aligned}
$$

Power of The Cheapest Candidate Arborescence

$T_1, T_2, \cdots, T_k$: the sequence of selected arborescences
$\ell_0 = n - 1$
ℓ_i: # of orphan components just after iteration i.
h_i: # of heads in T_i.

Power of The Cheapest Candidate Arborescence

$T_1, T_2, \cdots, T_k$: the sequence of selected arborescences
$\ell_0 = n - 1$
ℓ_i: # of orphan components just after iteration i.
h_i: # of heads in T_i.
For each $1 \leq i \leq k$,

$$\ell_{i-1} - \ell_i \geq h_i/2 \Rightarrow h_i \leq 2\left(\ell_{i-1} - \ell_i\right);$$
$$\frac{p\left(T_i\right)}{h_i} \leq \frac{opt}{\ell_{i-1}}.$$

Power of The Cheapest Candidate Arborescence

$T_1, T_2, \cdots, T_k$: the sequence of selected arborescences
$\ell_0 = n - 1$
ℓ_i: # of orphan components just after iteration i.
h_i: # of heads in T_i.
For each $1 \leq i \leq k$,

$$\ell_{i-1} - \ell_i \geq h_i/2 \Rightarrow h_i \leq 2\left(\ell_{i-1} - \ell_i\right);$$
$$\frac{p\left(T_i\right)}{h_i} \leq \frac{opt}{\ell_{i-1}}.$$

Hence,

$$p\left(T_k\right) \leq opt,$$
$$p\left(T_i\right) \leq \frac{h_i}{\ell_{i-1}} opt \leq \frac{2\left(\ell_{i-1} - \ell_i\right)}{\ell_{i-1}} opt, \forall 1 \leq i \leq k-1$$

Upper Bound on Greedy Solution

$$\begin{aligned}
\sum_{i=1}^{k} p\left(T_i\right) &\leq 2opt \sum_{i=1}^{k-1} \frac{\ell_{i-1} - \ell_i}{\ell_{i-1}} + opt \\
&\leq 2opt \sum_{i=1}^{k-1} \sum_{j=\ell_i+1}^{\ell_{i-1}} \frac{1}{j} + opt \\
&= 2opt \sum_{j=\ell_{k-1}+1}^{\ell_0} \frac{1}{j} + opt \\
&\leq 2opt \sum_{j=2}^{n-1} \frac{1}{j} + opt \\
&= \left(2H\left(n-1\right) - 1\right) opt.
\end{aligned}$$

Roadmap

- Introduction
- Aggregation Routing
- Minimum Submodular Cover With Submodular Cost
- Broadcast Routing
- **Multicast Routing**
  - Plane Geometric Networks
  - Symmetric Networks
  - Asymmetric Networks

Min-Power Multicast Routing

- NP-hard

Min-Power Multicast Routing

- NP-hard
- Approximality:
  - Plane geometric: $O(1)$
  - Symmetric: no $(1-\varepsilon)\ln n$-approx. for any $\varepsilon > 0$ unless P=NP
  - Asymmetric: no $\log^{2-\varepsilon} n$ approximation for any $\varepsilon > 0$ unless NP has quasi-polynomial Las Vegas algorithms

Min-Power Multicast Routing

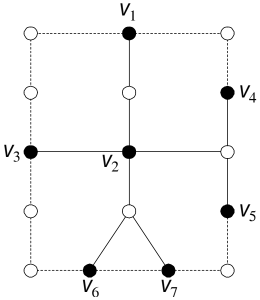
- NP-hard
- Approximality:
  - Plane geometric: $O(1)$
  - Symmetric: no $(1-\varepsilon)\ln n$-approx. for any $\varepsilon > 0$ unless P=NP
  - Asymmetric: no $\log^{2-\varepsilon} n$ approximation for any $\varepsilon > 0$ unless NP has quasi-polynomial Las Vegas algorithms
- Approx. bounded achieved:
  - Plane geometric: ≤ 12
  - Symmetric: $4H(k) - 2$
  - Asymmetric: $O(k^{\varepsilon})$ for any fixed $\varepsilon > 0$

Roadmap

- Introduction
- Aggregation Routing
- Minimum Submodular Cover With Submodular Cost
- Broadcast Routing
- **Multicast Routing**
  - **Plane Geometric Networks**
  - Symmetric Networks
  - Asymmetric Networks

Min-Power Multicast Routing in Plane Geometric Networks

- NP-hard
- Steiner Tree Based Approach



Min-Weight Steiner Tree

- NP-hard
- Constant-approximations:
  - Robins-Zelikovsky algorithm: approx. ratio $\leq 1 + \frac{ln3}{2} + \varepsilon \approx 1.55 + \varepsilon$.
  - Takahashi-Matsuyama algorithm: 2-approximation
    - Prim's-like
    - iteratively expanding a tree by a shortest path between terminals in the current tree and terminal outside the current tree

A Lower Bound on Minimum Power

opt: the power required by broadcast from s.
SMT: a Min-Weight Steiner tree

Lemma

$opt \geq c\left(SMT\right)/6.$

ST-Based Multicast Routing

$\mathcal{A}$: an α-approximation algorithm for Min-Weight Steiner Tree

ST-Based Multicast Routing

$\mathcal{A}$: an α-approximation algorithm for Min-Weight Steiner Tree

$\mathcal{A}^*$ for Min-Power Multicast Routing:

1. Step 1: Apply $\mathcal{A}$ to $\overline{D}$ to produce a Steiner tree ST for $\{s\} \cup X$
2. Step 2: Orient ST to an arborescence SA rooted at s

ST-Based Multicast Routing

$\mathcal{A}$: an α-approximation algorithm for Min-Weight Steiner Tree

$\mathcal{A}^*$ for Min-Power Multicast Routing:

1. Step 1: Apply $\mathcal{A}$ to $\overline{D}$ to produce a Steiner tree ST for $\{s\} \cup X$
2. Step 2: Orient ST to an arborescence SA rooted at s

$$p(SA) \leq c(SA) = c(ST) \leq \alpha \cdot c(SMT) \leq 6\alpha \cdot opt.$$

Theorem

The approximation ratio of the algorithm $\mathcal{A}^*$ is at most 6α.

Roadmap

- Introduction
- Aggregation Routing
- Minimum Submodular Cover With Submodular Cost
- Broadcast Routing
- **Multicast Routing**
  - Plane Geometric Networks
  - **Symmetric Networks**
  - Asymmetric Networks

Two-Phased Approximation

Phase 1: Compute a weak Steiner Connector D' with

$$p\left(D'\right) \leq \left(2H\left(k\right) - 1\right) opt$$

Two-Phased Approximation

Phase 1: Compute a weak Steiner Connector D' with

$$p\left(D'\right) \leq \left(2H\left(k\right) - 1\right) opt$$

Phase 2: Compute a spanning tree of $\overline{D'}$ and orient it to an arborescence T rooted at s

Two-Phased Approximation

Phase 1: Compute a weak Steiner Connector D' with

$$p\left(D'\right) \leq \left(2H\left(k\right) - 1\right) opt$$

Phase 2: Compute a spanning tree of $\overline{D'}$ and orient it to an arborescence T rooted at s

$$p\left(T\right) \leq 2p\left(D'\right) \leq \left(4H\left(k\right) - 2\right) opt$$

Greedy Algorithm for Phase 1

GWSC:

$B \leftarrow \emptyset$;

while $f(B) > 0$,

find a cheapest $T \in \mathcal{T}(B)$;

$B \leftarrow B \cup T$;

output (V, B).

GWSC:
$B \leftarrow \emptyset$;
while $f(B) > 0$,
find a cheapest $T \in \mathcal{T}(B)$;
$B \leftarrow B \cup T$;
output (V, B).

piece w.r.t. B: a weak component of (V, B) containing at least one destination but not s

Greedy Algorithm for Phase 1

GWSC:
$B \leftarrow \emptyset$;
while $f(B) > 0$,
find a cheapest $T \in \mathcal{T}(B)$;
$B \leftarrow B \cup T$;
output (V, B).

piece w.r.t. B: a weak component of (V, B) containing at least one destination but not s

$f(B) = \#$ of pieces w.r.t. B

Greedy Algorithm for Phase 1

GWSC:

$B \leftarrow \emptyset$;

while $f(B) > 0$,

  find a cheapest $T \in \mathcal{T}(B)$;

  $B \leftarrow B \cup T$;

output (V, B).

piece w.r.t. B: a weak component of (V, B) containing at least one destination but not s

$f(B) = \#$ of pieces w.r.t. B

Head of a piece: node of smallest ID.

$$\text{price of } T = \frac{p(T)}{\# \text{ of heads in } T}$$

Greedy Algorithm for Phase 1

GWSC:
$B \leftarrow \emptyset$;
while $f(B) > 0$,
find a cheapest $T \in \mathcal{T}(B)$;
$B \leftarrow B \cup T$;
output (V, B).

piece w.r.t. B: a weak component of (V, B) containing at least one destination but not s

$f(B) = \#$ of pieces w.r.t. B

Head of a piece: node of smallest ID.

$$\text{price of } T = \frac{p(T)}{\# \text{ of heads in } T}$$

$\mathcal{T}(B)$: a set of at most $|A| \cdot f(B)$ candidates, defined exactly as in **GBA**

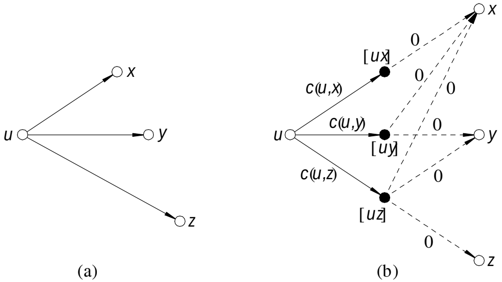
Roadmap

- Introduction
- Aggregation Routing
- Minimum Submodular Cover With Submodular Cost
- Broadcast Routing
- **Multicast Routing**
  - Plane Geometric Networks
  - Symmetric Networks
  - **Asymmetric Networks**

Reduction to Min-Weight Steiner Arborescence

Construction of a weighted digraph $D' = (V \cup V', A'; c')$ from D:

- For each $(u, v) \in A$, add to V' a node $[uv]$, and add to A' two arcs $(u, [uv])$ and $([uv], v)$ of weight $c(u, v)$ and 0 respectively.
- For each node u, sort arcs in $\delta^{out}(u)$ in the increasing order of weight:
$$c(u, v_1) \leq c(u, v_2) \leq \cdots \leq c(u, v_k),$$
and add to A' the arcs $([uv_j], [uv_i])$ of zero weight for $1 \leq i < j \leq k$.

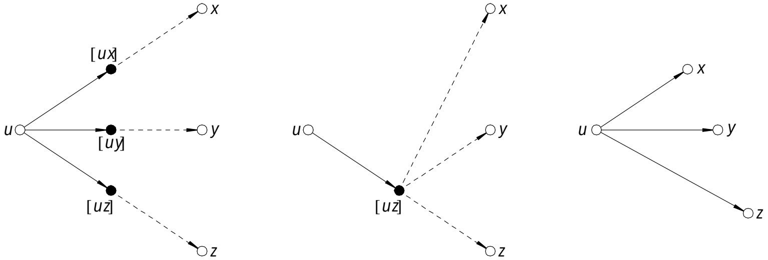


SA-Based Algorithm

$\mathcal{A}$: an α-approximation algorithm for Min-Weight Steiner Arborescence

- Best known approximation α: $O\left(k^{\varepsilon}\right)$ for any fixed $\varepsilon > 0$

1. Apply $\mathcal{A}$ to D' to construct a SA $T' \subseteq D'$ for s and X.
2. Construct another SA $T'' \subseteq D'$ such that $c'\left(T''\right) \leq c'\left(T'\right)$ and $\deg_{T''}^{out}\left(u\right) \leq 1$ for each $u \in V$.
3. Construct a SA $T \subseteq D$ from T'' satisfying that $p\left(T\right) = c'\left(T''\right)$.



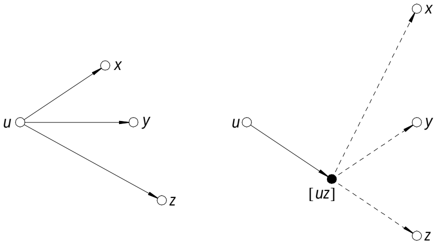
(a) (b) (c)

Approximation Ratio

OPT: a min-power SA in D for s and X
opt: power of T^*.

Construct a SA $OPT' \subseteq D'$ such that $c'(OPT') = p(OPT) = opt$.

$$p(T) = c'(T'') \leq c'(T') \leq \alpha \cdot c'(OPT') \leq \alpha \cdot opt.$$



(a) (b)