Minimum Connected Dominating Set in Multihop Wireless Networks

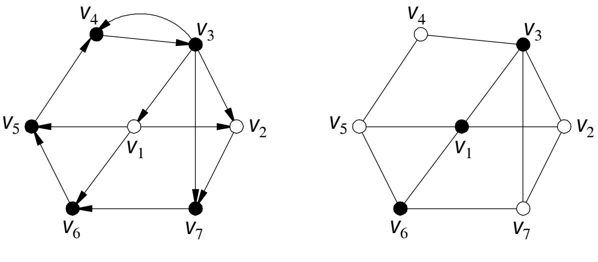
Peng-Jun Wan

Outline

- Overview
- MCDS in Arbitrary Networks
- MCDS in Symmetric Networks
- MCDS in Plane Geometric Networks

Virtual Backbone (CDS)

Virtual backbone/CDS: a subset U of nodes such that any pair of non-adjacent nodes can communicate with each other though the nodes in U.



(a) (b)

Minimum CDS (MCDS)

MCDS: compute a CDS of the smallest size in a network.

Minimum CDS (MCDS)

MCDS: compute a CDS of the smallest size in a network.

- **MCDS** in plane geometric networks: NP-hard, but admits PTAS.

Minimum CDS (MCDS)

MCDS: compute a CDS of the smallest size in a network.

- **MCDS** in plane geometric networks: NP-hard, but admits PTAS.
- **MCDS** in symmetric networks: admits no $(1-\varepsilon)\ln n$-approximation for any fixed $\varepsilon > 0$ unless $NP \subset DTIME\left[n^{O(\log\log n)}\right]$.

Minimum CDS (MCDS)

MCDS: compute a CDS of the smallest size in a network.

- **MCDS** in plane geometric networks: NP-hard, but admits PTAS.
- **MCDS** in symmetric networks: admits no $(1-\varepsilon)\ln n$-approximation for any fixed $\varepsilon > 0$ unless $NP \subset DTIME\left[n^{O(\log\log n)}\right]$.
- **MCDS** in arbitrary (or asymmetric) networks: at least as hard as MCDS in symmetric networks.

- **MCDS** in arbitrary networks: $(4H(n-2)-2)$-approximation

Summary on Algorithms

- **MCDS** in arbitrary networks: $(4H(n-2)-2)$-approximation
- **MCDS** in symmetric networks: $(2+\ln(\Delta-2))$-approximation

- **MCDS** in arbitrary networks: $(4H(n-2)-2)$-approximation
- **MCDS** in symmetric networks: $(2+\ln(\Delta-2))$-approximation
- **MCDS** in plane geometric networks: 6.075-approximation

Roadmap

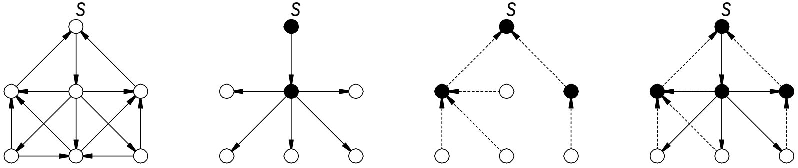
- Overview
- **MCDS in Arbitrary Networks**
- MCDS in Symmetric Networks
- MCDS in Plane Geometric Networks

T_1: a spanning s-arborescence in D

T_2: a spanning inward s-arborescence in D

Lemma

All non-sink nodes of T_1 and all non-source nodes of T_2 form a CDS of D.



(a) (b) (c) (d)

Spanning Arborescence with Fewest Internal Nodes

SAFIN: Given a digraph $D = (V, A)$ and a root node $s \in V$, compute a spanning s-arborescence T with minimum $|I(T) \setminus \{s\}|$, where $I(T)$ denote the set of non-sink nodes of T.

Spanning Arborescence with Fewest Internal Nodes

SAFIN: Given a digraph $D = (V, A)$ and a root node $s \in V$, compute a spanning s-arborescence T with minimum $|I(T) \setminus \{s\}|$, where $I(T)$ denote the set of non-sink nodes of T.

Reduction to **Min-Power Routing for Broadcast**

From SAFIN to MCDS

$\mathcal{A}$: a μ-approximation algorithm for SAFIN
s: a node in $OPT \cap S$.
D^R: reverse of D
T_1: spanning s-arborescence of D output by $\mathcal{A}$
T_2: spanning s-arborescence of D^R output by $\mathcal{A}$

$$\begin{aligned} |I(T_1) \cup I(T_2)| &\leq 1 + |I(T_1) \setminus \{s\}| + |I(T_2) \setminus \{s\}| \\ &\leq 1 + 2\mu(\gamma_c - 1) \\ &= 2\mu \cdot \gamma_c - 2\mu + 1. \end{aligned}$$

Candidates of Root

Fact: for any node u, any CDS must contain at least one node in $N^{in}[u]$ and at least one node in $N^{out}[u]$.

Candidates of Root

Fact: for any node u, any CDS must contain at least one node in $N^{in}[u]$ and at least one node in $N^{out}[u]$.
S: candidates of root

$$\boxed{\begin{array}{l} u \leftarrow \arg\min_{v \in V} \min\left(\delta^{in}(v), \delta^{out}(v)\right); \\ \text{If } \delta^{in}(u) \leq \delta^{out}(u) \text{ then } S \leftarrow N^{in}[u]; \\ \quad \text{else } S \leftarrow N^{out}[u]; \end{array}}$$

Outline of The Algorithm for MCDS

Algorithm $\mathcal{A}^*$:

  $U \leftarrow V$;

  for each $s \in S$,

    $T_1 \leftarrow$ spanning s-arborescence in D output by $\mathcal{A}$;

    $T_2 \leftarrow$ spanning s-arborescence in D^R output by $\mathcal{A}$;

    if $|I(T_1) \cup I(T_2)| < |U|$ then

      $U \leftarrow I(T_1) \cup I(T_2)$;

  output U.

Greedy Algorithm for SAFIN

GBA2:

$B \leftarrow \{s\}$;

while $f(B) > 0$,

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

$B \leftarrow B \cup I(T)$;

output a BFS arborescence of $D\langle B \rangle$ rooted at s.

Greedy Algorithm for SAFIN

GBA2:
$B \leftarrow \{s\}$;
while $f(B) > 0$,
find a cheapest $T \in \mathcal{T}(B)$;
$B \leftarrow B \cup I(T)$;
output a BFS arborescence of $D\langle B\rangle$ rooted at s.

$f(B) = \#$ of orphan components of $D\langle B\rangle = \left(V, \bigcup_{v\in B} \delta^{out}(v)\right)$.

Greedy Algorithm for SAFIN

GBA2:
$B \leftarrow \{s\}$;
while $f(B) > 0$,
find a cheapest $T \in \mathcal{T}(B)$;
$B \leftarrow B \cup I(T)$;
output a BFS arborescence of $D\langle B\rangle$ rooted at s.

$f(B) = \#$ of orphan components of $D\langle B\rangle = (V, \bigcup_{v \in B} \delta^{out}(v))$.
Head of an orphan component: node of smallest ID.

Greedy Algorithm for SAFIN

GBA2:
$B \leftarrow \{s\}$;
while $f(B) > 0$,
find a cheapest $T \in \mathcal{T}(B)$;
$B \leftarrow B \cup I(T)$;
output a BFS arborescence of $D\langle B\rangle$ rooted at s.

$f(B) = \#$ of orphan components of $D\langle B\rangle = \left(V, \bigcup_{v\in B} \delta^{out}(v)\right)$.
Head of an orphan component: node of smallest ID.

$$\text{price of } T = \frac{|I(T) \setminus B|}{\#\text{ of heads in } T}$$

Greedy Algorithm for SAFIN

GBA2:
$B \leftarrow \{s\}$;
while $f(B) > 0$,
find a cheapest $T \in \mathcal{T}(B)$;
$B \leftarrow B \cup I(T)$;
output a BFS arborescence of $D\langle B \rangle$ rooted at s.

$f(B) = \#$ of orphan components of $D\langle B \rangle = \left(V, \bigcup_{v \in B} \delta^{out}(v)\right)$.
Head of an orphan component: node of smallest ID.

$$\text{price of } T = \frac{|I(T) \setminus B|}{\# \text{ of heads in } T}$$

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

Candidate Arborescences Supplied by u

- *BFS* (u): a BFS u-arborescence in D.

Candidate Arborescences Supplied by u

- $BFS(u)$: a BFS u-arborescence in D.
- Sort all the heads in the increasing order of depth in $BFS(u)$.

Candidate Arborescences Supplied by u

- $BFS(u)$: a BFS u-arborescence in D.
- Sort all the heads in the increasing order of depth in $BFS(u)$.
- $T_h(B,u)$ for $1 \leq h \leq f(B)$: the minimal arborescence in $BFS(u)$ spanning u and the first h heads.

Candidate Arborescences Supplied by u

- $BFS\left(u\right)$: a BFS u-arborescence in D.
- Sort all the heads in the increasing order of depth in $BFS\left(u\right)$.
- $T_h\left(B,u\right)$ for $1 \leq h \leq f\left(B\right)$: the minimal arborescence in $BFS\left(u\right)$ spanning u and the first h heads.
- If $u = s$, then,

$$\mathcal{T}\left(B,u\right) = \left\{T_h\left(B,u\right) : 1 \leq h \leq f\left(B\right)\right\};$$

otherwise,

$$\mathcal{T}\left(B,u\right) = \left\{T_h\left(B,u\right) : 2 \leq h \leq f\left(B\right)\right\}.$$

Approximation Bound

Theorem

The approximation ratio of the algorithm **GBA2** *is at most* $2H(n-2)-1$.

Approximation Bound

Theorem

The approximation ratio of the algorithm **GBA2** *is at most* $2H(n-2)-1$.

Theorem

The approximation ratio of the algorithm $\mathbf{GBA2}^*$ *is at most* $4H(n-2)-2$.

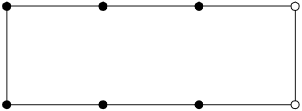
Roadmap

- Overview
- MCDS in Arbitrary Networks
- **MCDS in Symmetric Networks**
- MCDS in Plane Geometric Networks

MCDS in Graphs with Max. Degree at Most 2

- If $\Delta \leq 2$, G is either a path or a cycle.

(a)

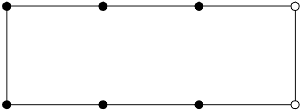


(b)

MCDS in Graphs with Max. Degree at Most 2

- If $\Delta \leq 2$, G is either a path or a cycle.
- When G is a path, the MCDS consists of all internal vertices.

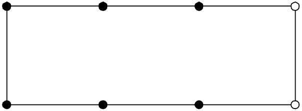
(a)



(b)

- If $\Delta \leq 2$, G is either a path or a cycle.
- When G is a path, the MCDS consists of all internal vertices.
- When G is a cycle, a MCDS can be obtained by deleting two adjacent vertices.

(a)

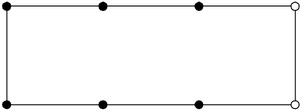


(b)

MCDS in Graphs with Max. Degree at Most 2

- If $\Delta \leq 2$, G is either a path or a cycle.
- When G is a path, the MCDS consists of all internal vertices.
- When G is a cycle, a MCDS can be obtained by deleting two adjacent vertices.

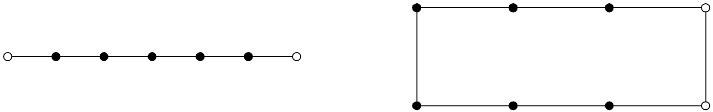
(a)



(b)

MCDS in Graphs with Max. Degree at Most 2

- If $\Delta \leq 2$, G is either a path or a cycle.
- When G is a path, the MCDS consists of all internal vertices.
- When G is a cycle, a MCDS can be obtained by deleting two adjacent vertices.

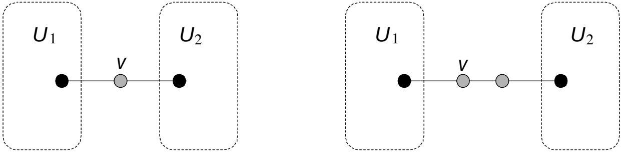


(a) (b)

So we assume that $\Delta \geq 3$ from now on.

A Two-Phased Algorithm

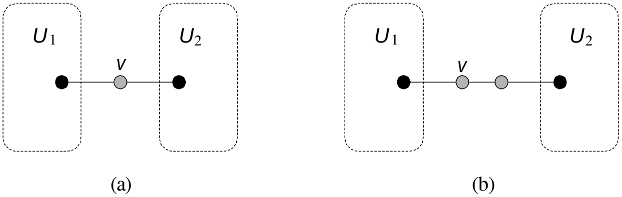
1. Phase 1: apply the greedy algorithm for minimum submodular cover to produce a dominating set U with $|U| \leq H(\Delta)\gamma_c$.



(a) (b)

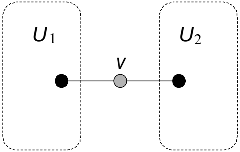
A Two-Phased Algorithm

1. Phase 1: apply the greedy algorithm for minimum submodular cover to produce a dominating set U with $|U| \leq H(\Delta)\gamma_c$.
2. Phase 2: select a set W of "connectors" to interconnect U.

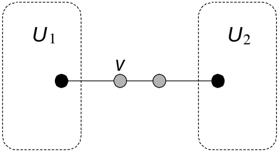


A Two-Phased Algorithm

1. Phase 1: apply the greedy algorithm for minimum submodular cover to produce a dominating set U with $|U| \leq H(\Delta)\gamma_c$.
2. Phase 2: select a set W of "connectors" to interconnect U.
   - start with an empty W



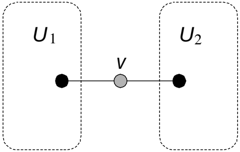
(a)



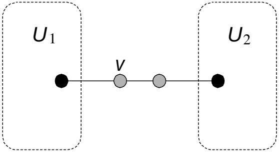
(b)

A Two-Phased Algorithm

1. Phase 1: apply the greedy algorithm for minimum submodular cover to produce a dominating set U with $|U| \leq H(\Delta)\gamma_c$.
2. Phase 2: select a set W of "connectors" to interconnect U.
   - start with an empty W
   - iteratively reduce the # of components of $G[U \cup W]$ by adding at most two connectors to W until $G[U \cup W]$ is connected.



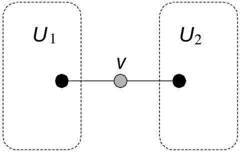
(a)



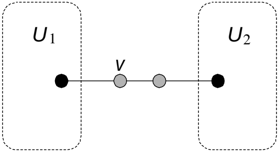
(b)

A Two-Phased Algorithm

1. Phase 1: apply the greedy algorithm for minimum submodular cover to produce a dominating set U with $|U| \leq H(\Delta)\gamma_c$.
2. Phase 2: select a set W of "connectors" to interconnect U.
   - start with an empty W
   - iteratively reduce the $\#$ of components of $G[U \cup W]$ by adding at most two connectors to W until $G[U \cup W]$ is connected.



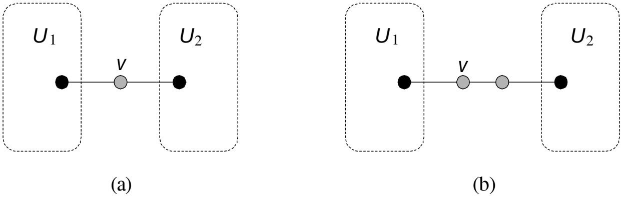
(a)



(b)

A Two-Phased Algorithm

1. Phase 1: apply the greedy algorithm for minimum submodular cover to produce a dominating set U with $|U| \leq H(\Delta)\gamma_c$.
2. Phase 2: select a set W of "connectors" to interconnect U.
   - start with an empty W
   - iteratively reduce the # of components of $G[U \cup W]$ by adding at most two connectors to W until $G[U \cup W]$ is connected.



$$|W| \leq 2(|U|-1) \Rightarrow |U \cup W| \leq 3|U| - 2 \leq 3H(\Delta)\gamma_c - 2.$$

A Single-Phased Greedy Algorithm

- Single phase with proper potential function

A Single-Phased Greedy Algorithm

- Single phase with proper potential function
- Better approximation bound: $2 + \ln(\Delta - 2)$

Potential Function

$\forall U \subseteq V$,

$$f_1(U) = \# \text{ of components in } G[U],$$
$$f_2(U) = \# \text{ of components in } G\langle U\rangle,$$

where

$$G\langle U\rangle = \left(V, \bigcup_{u\in U} \delta(u)\right).$$

Potential Function

$\forall U \subseteq V$,

$$f_1(U) = \#\text{ of components in } G[U],$$
$$f_2(U) = \#\text{ of components in } G\langle U\rangle,$$

where

$$G\langle U\rangle = \left(V, \bigcup_{u\in U} \delta(u)\right).$$

Then, the potential of U is $f(U) = f_1(U) + f_2(U) - 1$.

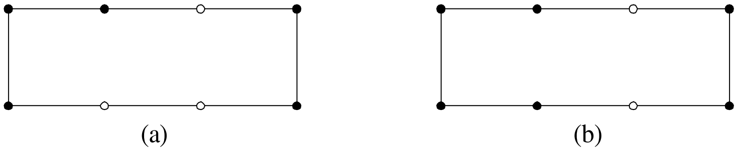


Figure: U consists of black nodes. In both (a) and (b), $f_1(U) = 2$, $f_2(U) = 1$, and $f(U) = 2$.

Potential Function

Clearly,

$$f_1(\emptyset) = 0, f_2(\emptyset) = n \Rightarrow f(\emptyset) = n - 1;$$

Potential Function

Clearly,

$$f_1(\emptyset) = 0, f_2(\emptyset) = n \Rightarrow f(\emptyset) = n - 1;$$

$\forall \emptyset \neq U \subseteq V$,

$$f_1(U) \geq 1, f_2(U) \geq 1 \Rightarrow f(U) \geq 1.$$

$$U \text{ is a CDS } \Leftrightarrow f(U) = 1$$

Gain

The *gain* of $v \in V$ w.r.t. $U \subseteq V$ is defined to be

$$\partial_v f(U) = f(U) - f(U \cup \{v\}).$$

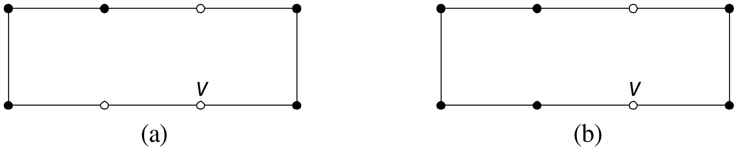


Figure: U consists of black nodes. (a) $\partial_v f(U) = 0$; (b) $\partial_v f(U) = 1$.

Gain

Denote

$$\partial_v f_1(U) = f_1(U) - f_2(U \cup \{v\}),$$
$$\partial_v f_2(U) = f_2(U) - f_2(U \cup \{v\}),$$

Then

$$\partial_v f(U) = \partial_v f_1(U) + \partial_v f_2(U).$$

Gain

Denote

$$\partial_v f_1(U) = f_1(U) - f_2(U \cup \{v\}),$$
$$\partial_v f_2(U) = f_2(U) - f_2(U \cup \{v\}),$$

Then

$$\partial_v f(U) = \partial_v f_1(U) + \partial_v f_2(U).$$

If $v \in U$ then

$$\partial_v f_1(U) = \partial_v f_2(U) = \partial_v f(U) = 0;$$

Gain

Denote

$$\partial_v f_1(U) = f_1(U) - f_2(U \cup \{v\}),$$
$$\partial_v f_2(U) = f_2(U) - f_2(U \cup \{v\}),$$

Then

$$\partial_v f(U) = \partial_v f_1(U) + \partial_v f_2(U).$$

If $v \in U$ then

$$\partial_v f_1(U) = \partial_v f_2(U) = \partial_v f(U) = 0;$$

else

$$\partial_v f_1(U) = (\#\text{ of components of } G[U] \text{ adj. to } v) - 1,$$
$$\partial_v f_2(U) = \#\text{ of components of } G\langle U\rangle \text{ adj. to } v.$$

Greedy Algorithm

GCDS

$B \leftarrow \emptyset$;

While $f(B) > 1$ do

select $v \in V \setminus B$ with maximum $\partial_v f(B)$;

$B \leftarrow B \cup \{v\}$;

Output B.

Running Time And Approximation Bound

Theorem

The algorithm **GCDS** *runs in at most* $n-2$ *iterations and has approximation bound of at most* $2+\ln(\Delta-2)$.

Extensibility

Lemma

If U is not a CDS, then at least one node v has a positive gain w.r.t. U.

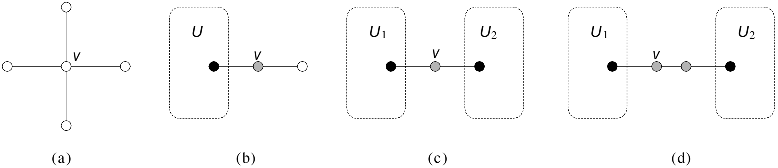


Figure: A node v with positive gain w.r.t. U. (a) $U = \emptyset$; (b) $U \neq \emptyset$ but is not a DS; (c) and (d) U is a DS.

"Shifted" Supermodularity

Lemma

Suppose that U and W are two subsets of V satisfying that $G[W]$ is connected. Then, for any node $v \in V$,

$$\partial_v f(U \cup W) \leq \partial_v f(U) + 1.$$

"Shifted" Supermodularity

Lemma

Suppose that U and W are two subsets of V satisfying that $G[W]$ is connected. Then, for any node $v \in V$,

$$\partial_v f(U \cup W) \leq \partial_v f(U) + 1.$$

$$\partial_v f_1(U \cup W) \leq \partial_v f_1(U) + 1,$$
$$\partial_v f_2(U \cup W) \leq \partial_v f_2(U).$$

Lower Bound on The Gain

Lemma

If U is not a CDS, then then at least one node v has gain at least $\max\left\{1, \frac{f(U)}{\gamma_c} - 1\right\}$ *w.r.t. U.*

Lower Bound on The Gain

Lemma

If U is not a CDS, then then at least one node v has gain at least $\max\left\{1, \frac{f(U)}{\gamma_c} - 1\right\}$ w.r.t. U.

$W = \{v_i : 1 \leq i \leq \gamma_c\}$: a MCDS sorted in the BFS order in $G[W]$.
$W_j = \{v_i : 1 \leq i \leq j\}$ with $1 \leq j \leq \gamma_c$.

Lower Bound on The Gain

$$f(U) - 1 = f(U) - f(U \cup W) = \partial_{v_1} f(U) + \sum_{j=2}^{\gamma_c} \partial_{v_j} f(U \cup W_{j-1})$$

$$\leq \partial_{v_1} f(U) + \sum_{j=2}^{\gamma_c} \left(\partial_{v_j} f(U) + 1 \right) = \gamma_c - 1 + \sum_{j=1}^{\gamma_c} \partial_{v_j} f(U)$$

$$\leq \gamma_c - 1 + \gamma_c \cdot \max_{1 \leq j \leq \gamma_c} \partial_{v_j} f(U).$$

Lower Bound on The Gain

$$
\begin{aligned}
f(U) - 1 &= f(U) - f(U \cup W) = \partial_{v_1} f(U) + \sum_{j=2}^{\gamma_c} \partial_{v_j} f(U \cup W_{j-1}) \\
&\leq \partial_{v_1} f(U) + \sum_{j=2}^{\gamma_c} \left(\partial_{v_j} f(U) + 1 \right) = \gamma_c - 1 + \sum_{j=1}^{\gamma_c} \partial_{v_j} f(U) \\
&\leq \gamma_c - 1 + \gamma_c \cdot \max_{1 \leq j \leq \gamma_c} \partial_{v_j} f(U).
\end{aligned}
$$

Hence,

$$
\max_{1 \leq j \leq \gamma_c} \partial_{v_j} f(U) \geq \frac{f(U)}{\gamma_c} - 1.
$$

Lower Bound on Optimum

Lemma

$\gamma_c \geq \frac{n-2}{\Delta-1}$.

Lower Bound on Optimum

Lemma

$\gamma_c \geq \frac{n-2}{\Delta-1}$.

$W = \{v_i : 1 \leq i \leq \gamma_c\}$: a MCDS sorted in the BFS order in $G[W]$.

Lower Bound on Optimum

Lemma

$\gamma_c \geq \frac{n-2}{\Delta-1}$.

$W = \{v_i : 1 \leq i \leq \gamma_c\}$: a MCDS sorted in the BFS order in $G[W]$.

v_1 dominates at most $\Delta + 1$ nodes. Each v_i with $2 \leq i \leq \gamma_c$ dominates $\Delta - 1$ additional nodes.

Lower Bound on Optimum

Lemma

$\gamma_c \geq \frac{n-2}{\Delta-1}$.

$W = \{v_i : 1 \leq i \leq \gamma_c\}$: a MCDS sorted in the BFS order in $G[W]$.

v_1 dominates at most $\Delta + 1$ nodes. Each v_i with $2 \leq i \leq \gamma_c$ dominates $\Delta - 1$ additional nodes.

$$\begin{aligned} n &\leq (\Delta + 1) + (\Delta - 1)(\gamma_c - 1) \\ &= (\Delta - 1)\gamma_c + 2. \end{aligned}$$

Upper Bound on Greedy Solution

B: output of the greedy algorithm.
B_i for $1 \leq i \leq |B|$, : the sequence of the first i nodes in B.
$B_0 = \emptyset$.
k: be the first (smallest) nonnegative integer such that $f(B_k) < 2\gamma_c + 2$.

Upper Bound on Greedy Solution

B: output of the greedy algorithm.
B_i for $1 \leq i \leq |B|$, : the sequence of the first i nodes in B.
$B_0 = \emptyset$.
k: be the first (smallest) nonnegative integer such that $f(B_k) < 2\gamma_c + 2$.

Claim 1: $|B \setminus B_k| \leq 2\gamma_c - 1$.
Claim 2: $k - 1 \leq \gamma_c \ln(\Delta - 2)$.

Upper Bound on Greedy Solution

B: output of the greedy algorithm.
B_i for $1 \leq i \leq |B|$, : the sequence of the first i nodes in B.
$B_0 = \emptyset$.
k: be the first (smallest) nonnegative integer such that $f(B_k) < 2\gamma_c + 2$.

Claim 1: $|B \setminus B_k| \leq 2\gamma_c - 1$.
Claim 2: $k - 1 \leq \gamma_c \ln(\Delta - 2)$.

$$|B| = k + |B \setminus B_k| \leq k + 2\gamma_c - 1 \leq (2 + \ln(\Delta - 2))\gamma_c.$$

Proof of Claim 1

Each node in $B \setminus B_k$ has gain ≥ 1.

Case 1: $f(B_k) \leq 2\gamma_c$.

$$|B \setminus B_k| \leq f(B_k) - 1 \leq 2\gamma_c - 1.$$

Case 2: $f(B_k) = 2\gamma_c + 1$. Then the first node in $B \setminus B_k$ has gain ≥ 2.

$$2 + (|B \setminus B_k| - 1) \leq f(B_k) - 1 = 2\gamma_c \Rightarrow |B \setminus B_k| \leq 2\gamma_c - 1.$$

Proof of Claim 2

$\ell_i = f(B_i) - \gamma_c$ for $0 \leq i < k$: "shifted" uncoverage

Proof of Claim 2

$\ell_i = f(B_i) - \gamma_c$ for $0 \leq i < k$: "shifted" uncoverage

$$n - 1 - \gamma_c = \ell_0 > \ell_1 > \cdots > \ell_{k-1} \geq \gamma_c + 2,$$

$$\ell_{i-1} - \ell_i \geq \frac{f(C_{i-1})}{\gamma_c} - 1 = \frac{\ell_{i-1}}{\gamma_c} \Rightarrow \frac{\ell_{i-1} - \ell_i}{\ell_{i-1}} \geq \frac{1}{\gamma_c}.$$

Proof of Claim 2

$\ell_i = f(B_i) - \gamma_c$ for $0 \le i < k$: "shifted" uncoverage

$$n - 1 - \gamma_c = \ell_0 > \ell_1 > \cdots > \ell_{k-1} \ge \gamma_c + 2,$$
$$\ell_{i-1} - \ell_i \ge \frac{f(C_{i-1})}{\gamma_c} - 1 = \frac{\ell_{i-1}}{\gamma_c} \Rightarrow \frac{\ell_{i-1} - \ell_i}{\ell_{i-1}} \ge \frac{1}{\gamma_c}.$$

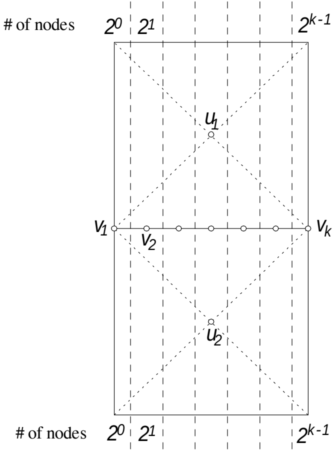
Therefore,

$$\begin{aligned}
\frac{k-1}{\gamma_c} &\le \sum_{i=1}^{k-1} \frac{\ell_{i-1} - \ell_i}{\ell_{i-1}} \le \ln \frac{\ell_0}{\ell_{k-1}} \le \ln \frac{n - 1 - \gamma_c}{\gamma_c + 2} \\
&\le \ln \frac{(\Delta - 1)\gamma_c + 2 - 1 - \gamma_c}{\gamma_c + 2} \\
&= \ln \left(\Delta - 2 - \frac{2\Delta - 5}{\gamma_c + 2} \right) < \ln (\Delta - 2).
\end{aligned}$$

Roadmap

- Overview
- MCDS in Arbitrary Networks
- MCDS in Symmetric Networks
- **MCDS in Plane Geometric Networks**

Performance of Greedy Algorithms in Plane Geometric Networks

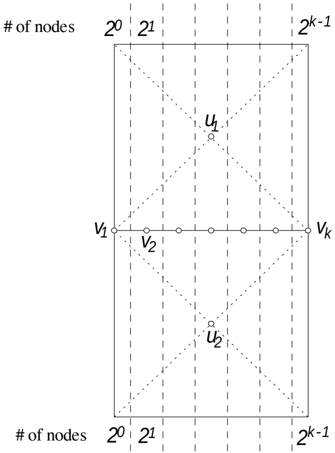


$$n = k + 2 + 2\sum_{i=1}^{k} 2^{i-1} = k + 2^{k+1},$$

$$\Delta = 2^k + k + 1,$$

$$\gamma_c = 2.$$

Performance of Greedy Algorithms in Plane Geometric Networks



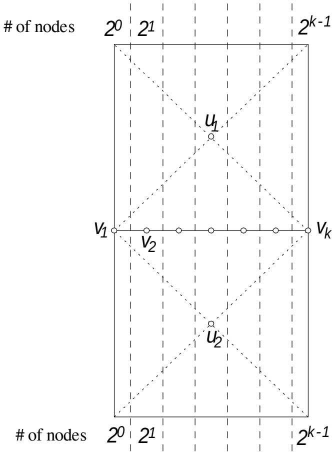
$$n = k + 2 + 2 \sum_{i=1}^{k} 2^{i-1} = k + 2^{k+1},$$

$$\Delta = 2^k + k + 1,$$

$$\gamma_c = 2.$$

Greedy solution: $v_k, v_{k-1}, \cdots, v_1$

Performance of Greedy Algorithms in Plane Geometric Networks



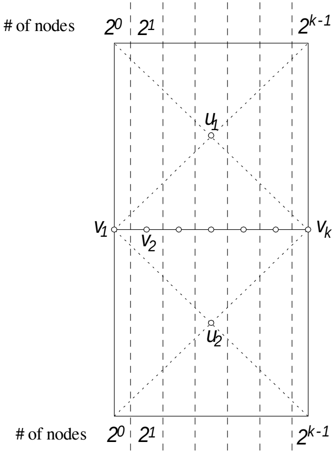
$$n = k + 2 + 2\sum_{i=1}^{k} 2^{i-1} = k + 2^{k+1},$$

$$\Delta = 2^k + k + 1,$$

$$\gamma_c = 2.$$

Greedy solution: $v_k, v_{k-1}, \cdots, v_1$
Approx. bound: $\geq \frac{k}{2} \geq \frac{\log \Delta}{2} - \frac{1}{2}$

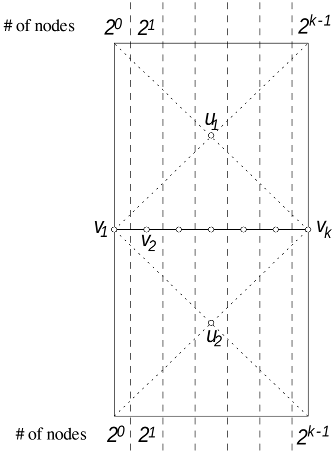
Greedy Solution



Initial degree:

- v_i: $2 \cdot 2^{i-1} + (k-1) + 2 = 2^i + k + 1$;

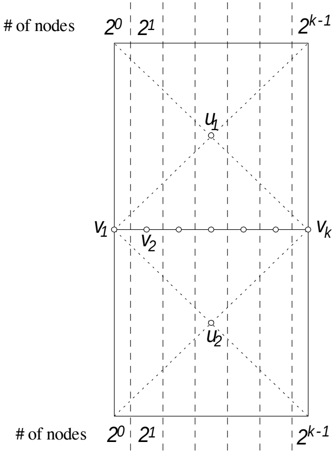
Greedy Solution



Initial degree:

- v_i: $2 \cdot 2^{i-1} + (k-1) + 2 = 2^i + k + 1$;
- u_1 and u_2: $\sum_{i=1}^{k} 2^{i-1} + k + 1 = 2^k + k$;

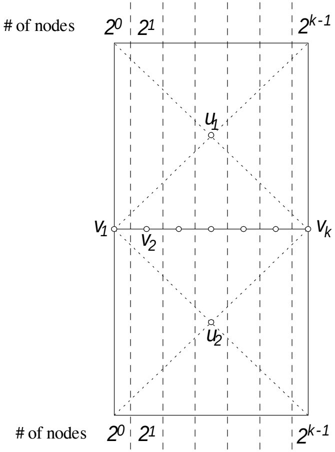
Greedy Solution



Initial degree:

- v_i: $2 \cdot 2^{i-1} + (k-1) + 2 = 2^i + k + 1$;
- u_1 and u_2: $\sum_{i=1}^{k} 2^{i-1} + k + 1 = 2^k + k$;
- others: $\sum_{i=1}^{k} 2^{i-1} - 1 + 1 + 1 = 2^k$.

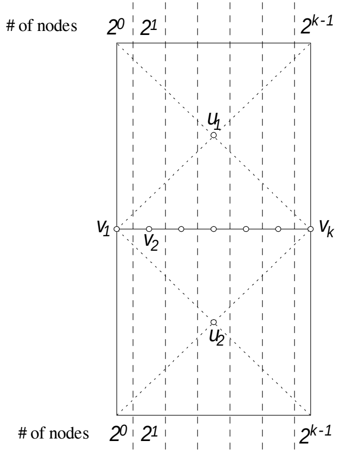
Greedy Solution



Initial degree:

- v_i: $2 \cdot 2^{i-1} + (k-1) + 2 = 2^i + k + 1$;
- u_1 and u_2: $\sum_{i=1}^{k} 2^{i-1} + k + 1 = 2^k + k$;
- others: $\sum_{i=1}^{k} 2^{i-1} - 1 + 1 + 1 = 2^k$.

Greedy Solution

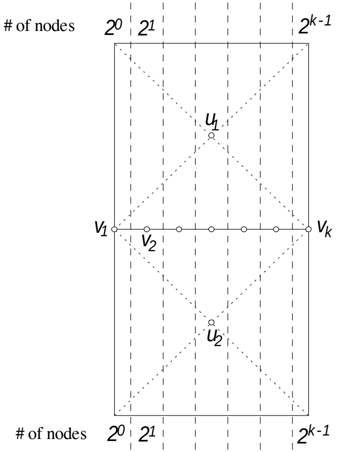


Initial degree:

- v_i: $2 \cdot 2^{i-1} + (k-1) + 2 = 2^i + k + 1$;
- u_1 and u_2: $\sum_{i=1}^{k} 2^{i-1} + k + 1 = 2^k + k$;
- others: $\sum_{i=1}^{k} 2^{i-1} - 1 + 1 + 1 = 2^k$.

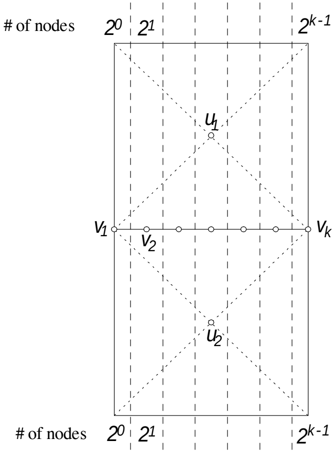
So v_k is selected as the first dominator.

Greedy Solution



After the selection of $v_k, v_{k-1}, \cdots, v_j$, the "residue" degree:

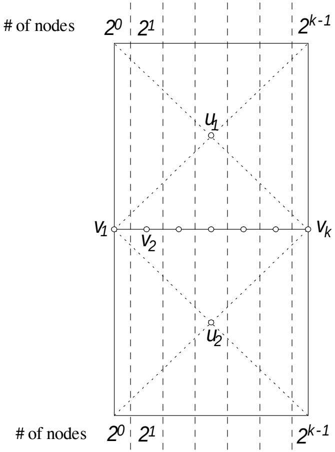
- v_i with $i < j$: $2 \cdot 2^{i-1} = 2^i$;



After the selection of $v_k, v_{k-1}, \cdots, v_j$, the "residue" degree:

- v_i with $i < j$: $2 \cdot 2^{i-1} = 2^i$;
- u_1 and u_2: $\sum_{i=1}^{j-1} 2^{i-1} = 2^{j-1} - 1$;

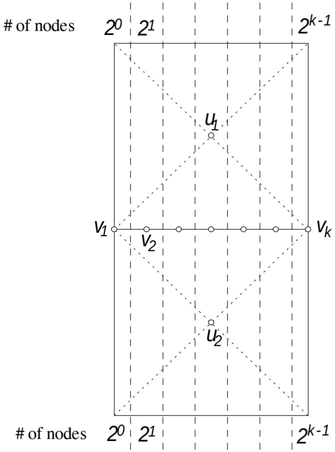
Greedy Solution



After the selection of $v_k, v_{k-1}, \cdots, v_j$, the "residue" degree:

- v_i with $i < j$: $2 \cdot 2^{i-1} = 2^i$;
- u_1 and u_2: $\sum_{i=1}^{j-1} 2^{i-1} = 2^{j-1} - 1$;
- others: $\sum_{i=1}^{j-1} 2^{i-1} - 1 = 2^{j-1} - 2$.

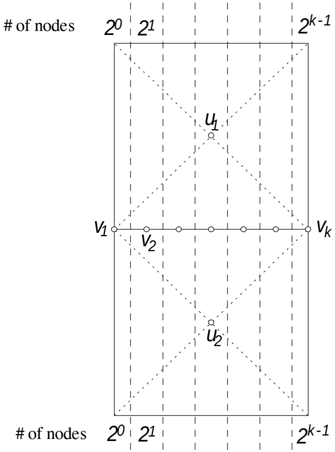
Greedy Solution



After the selection of $v_k, v_{k-1}, \cdots, v_j$, the "residue" degree:

- v_i with $i < j$: $2 \cdot 2^{i-1} = 2^i$;
- u_1 and u_2: $\sum_{i=1}^{j-1} 2^{i-1} = 2^{j-1} - 1$;
- others: $\sum_{i=1}^{j-1} 2^{i-1} - 1 = 2^{j-1} - 2$.

Greedy Solution



After the selection of $v_k, v_{k-1}, \cdots, v_j$, the "residue" degree:

- v_i with $i < j$: $2 \cdot 2^{i-1} = 2^i$;
- u_1 and u_2: $\sum_{i=1}^{j-1} 2^{i-1} = 2^{j-1} - 1$;
- others: $\sum_{i=1}^{j-1} 2^{i-1} - 1 = 2^{j-1} - 2$.

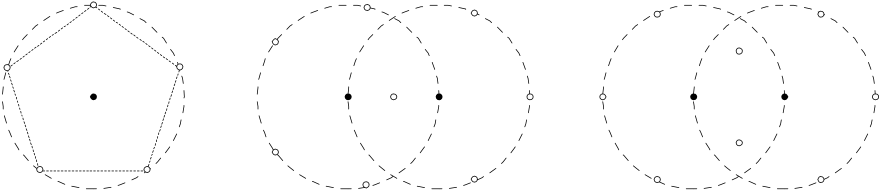
So v_{j-1} is selected as a dominator.

Independence Number vs. Connected Domination Number

α: independence number of G

γ_c: connected domination number of G

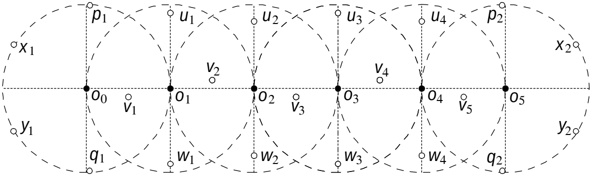
$$\alpha \leq \begin{cases} 5 & \text{if } \gamma_c = 1; \\ 8 & \text{if } \gamma_c = 2; \\ \min\left\{3.4306\gamma_c + 4.8185, 3\frac{2}{3}\gamma_c + 1\right\} & \text{if } \gamma_c \geq 3. \end{cases}$$



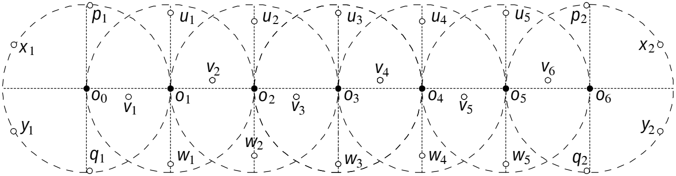
(a) (b) (c)

A Conjecture

Conjecture: if $\gamma_c \geq 3$, then $\alpha \leq 3\gamma_c + 3$.



(a)



(b)

2-Phased Approximation Algorithm

Phase 1: Constructs an MIS with 2-hop separation property: any pair of its nonempty complementary subsets are separated by *exactly two* hops.

Phase 2: Augment with the MIS with "connectors" to form a CDS

Algorithm for Phase 1

1. Construct be an *arbitrary* rooted spanning tree T

Algorithm for Phase 1

1. Construct be an *arbitrary* rooted spanning tree T
2. Select an MIS I in the first-fit manner in the BFS ordering in T.

Algorithm for Phase 1

1. Construct be an *arbitrary* rooted spanning tree T
2. Select an MIS I in the first-fit manner in the BFS ordering in T.

Algorithm for Phase 1

1. Construct be an *arbitrary* rooted spanning tree T
2. Select an MIS I in the first-fit manner in the BFS ordering in T.

$\langle v_1, v_2, \cdots, v_n \rangle$: BFS ordering of V in T.
Initialization: $I \leftarrow \{v_1\}$.
First-fit selection: For $i = 2$ up to n, add v_i to I if v_i is not adjacent to any node in I.

2-Hop Separation Property

Lemma

Any pair of nonempty complementary subsets of I are separate by exactly two hops.

2-Hop Separation Property

Lemma

Any pair of nonempty complementary subsets of I are separate by exactly two hops.

$u_1, u_2, \cdots, u_k$: sequence of nodes added to I.
H_j for $1 \leq j \leq k$: the graph over $\{u_i : 1 \leq i \leq j\}$ in which a pair of nodes is connected by an edge if and only if their graph distance in G is two.

Claim: H_j is connected (by induction on j)

Algorithm for Phase 2

GC

$C \leftarrow \emptyset$;
While $f(C) > 1$ do
  select $v \in V \setminus (I \cup C)$ with maximum $\partial_v f(C)$;
  $C \leftarrow C \cup \{v\}$;
Output C.

- For any subset $U \subseteq V \setminus I$, $f(U) = \#$ of components in $G[I \cup U]$.
- Gain of a node v w.r.t. U: $\partial_w f(U) = f(U) - f(U \cup \{x\})$.

Lower Bound on Gain

Lemma

If $f(U) > 1$, then at least one node w in $V \setminus (I \cup U)$ has gain at least $\max\{1, \lceil f(U)/\gamma_c \rceil - 1\}$.

Lower Bound on Gain

Lemma

If $f(U) > 1$, *then at least one node* w *in* $V \setminus (I \cup U)$ *has gain at least* $\max\{1, \lceil f(U)/\gamma_c \rceil - 1\}$.

Since the set I has 2-hop separation property, at least one node $w \in V \setminus (I \cup U)$ is adjacent to at least two connected components of $G[I \cup U]$.

Lower Bound on Gain

Lemma

If $f(U) > 1$, then at least one node w in $V \setminus (I \cup U)$ has gain at least $\max\{1, \lceil f(U)/\gamma_c \rceil - 1\}$.

Since the set I has 2-hop separation property, at least one node $w \in V \setminus (I \cup U)$ is adjacent to at least two connected components of $G[I \cup U]$.

Since each component of $G[I \cup U]$ must be adjacent to some node in $OPT \setminus (I \cup U)$, at lease some node $w \in OPT \setminus (I \cup U)$ is adjacent to

$$\left\lceil \frac{f(U)}{|OPT \setminus (I \cup U)|} \right\rceil \geq \left\lceil \frac{f(U)}{\gamma_c} \right\rceil$$

components of $G[I \cup U]$.

Main Theorem

C: sequence of selected connectors

Theorem

$|I \cup C| \leq 6.075\gamma_c + 5.425.$

Main Theorem

C: sequence of selected connectors

Theorem

$|I \cup C| \leq 6.075\gamma_c + 5.425.$

- If $\gamma_c = 1$, then $|I| \leq 5$ and $|C| \leq 1$, hence $|I \cup C| \leq 6$.
- If $|I| \leq 3\gamma_c + 2$, then $|I \cup C| \leq 2|I| - 1 \leq 6\gamma_c + 3$.

Main Theorem

C: sequence of selected connectors

Theorem

$|I \cup C| \leq 6.075\gamma_c + 5.425.$

- If $\gamma_c = 1$, then $|I| \leq 5$ and $|C| \leq 1$, hence $|I \cup C| \leq 6$.
- If $|I| \leq 3\gamma_c + 2$, then $|I \cup C| \leq 2|I| - 1 \leq 6\gamma_c + 3$.

From now on, we assume that $\gamma_c \geq 2$ and $|I| > 3\gamma_c + 2$.

Three Subsequences

Break C into three contiguous (and possibly empty) subsequences C_1, C_2 and C_3 as follows.

- C_1: the shortest prefix of C satisfying that $f(C_1) \leq 3\gamma_c + 2$
- $C_1 \cup C_2$: the shortest prefix of C satisfying that $f(C_1 \cup C_2) \leq 2\gamma_c + 1$.

Three Subsequences

Break C into three contiguous (and possibly empty) subsequences C_1, C_2 and C_3 as follows.

- C_1: the shortest prefix of C satisfying that $f(C_1) \leq 3\gamma_c + 2$
- $C_1 \cup C_2$: the shortest prefix of C satisfying that $f(C_1 \cup C_2) \leq 2\gamma_c + 1$.

We will prove that

$$|C_1| \leq \begin{cases} \frac{|I|}{3} - \gamma_c & \text{if } f(C_1) \leq 3\gamma_c + 1, \\ \frac{|I|-2}{3} - \gamma_c & \text{if } f(C_1) = 3\gamma_c + 2; \end{cases}$$

$$|C_2| \leq \begin{cases} \frac{\gamma_c}{2} & \text{if } f(C_1) \leq 3\gamma_c + 1, \\ \frac{\gamma_c+1}{2} & \text{if } f(C_1) = 3\gamma_c + 2; \end{cases}$$

$$|C_3| \leq 2\gamma_c - 1.$$

Three Subsequences

$$|C_1 \cup C_2| \leq \frac{|I|}{3} - \frac{\gamma_c}{2}$$

and

$$|C| \leq \frac{|I|}{3} - \frac{\gamma_c}{2} + 2\gamma_c - 1 = \frac{|I|}{3} + \frac{3}{2}\gamma_c - 1.$$

Three Subsequences

$$|C_1 \cup C_2| \leq \frac{|I|}{3} - \frac{\gamma_c}{2}$$

and

$$|C| \leq \frac{|I|}{3} - \frac{\gamma_c}{2} + 2\gamma_c - 1 = \frac{|I|}{3} + \frac{3}{2}\gamma_c - 1.$$

So,

$$\begin{aligned} |I \cup C| \leq & \frac{4\,|I|}{3} + \frac{3}{2}\gamma_c - 1 \\ \leq & \frac{4}{3}\left(3.4306\gamma_c + 4.8185\right) + \frac{3}{2}\gamma_c - 1 \\ \leq & 6.075\gamma_c + 5.425. \end{aligned}$$

Upper Bound on $|C_1|$

Trivial if $C_1 = \emptyset$. So we assume that $C_1 \neq \emptyset$ and let u be the last node in C_1. Then,

$$f\left(C_1 \setminus \{u\}\right) \geq 3\gamma_c + 3.$$

Case 1: $f\left(C_1\right) \leq 3\gamma_c + 1$.

$$\begin{aligned} & 3\left(|C_1| - 1\right) \leq |I| - f\left(C_1 \setminus \{u\}\right) \leq |I| - \left(3\gamma_c + 3\right) \\ & \Rightarrow |C_1| \leq \frac{|I|}{3} - \gamma_c. \end{aligned}$$

Case 2: $f\left(C_1\right) = 3\gamma_c + 2$.

$$\begin{aligned} & 3|C_1| \leq |I| - f\left(C_1\right) = |I| - \left(3\gamma_c + 2\right) \\ & \Rightarrow |C_1| \leq \frac{|I| - 2}{3} - \gamma_c. \end{aligned}$$

Upper Bound on $|C_2|$

Trivial if $|C_2| \leq 1$. So, assume $|C_2| \geq 2$ and let v be the last node in C_2. Then,

$$f(C_1 \cup C_2 \setminus \{v\}) \geq 2\gamma_c + 2.$$

Case 1: $f(C_1) \leq 3\gamma_c$.

$$2(|C_2| - 1) \leq f(C_1) - f(C_1 \cup C_2 \setminus \{v\}) \leq 3\gamma_c - (2\gamma_c + 2)$$
$$\Rightarrow |C_2| \leq \gamma_c/2.$$

Case 2: $f(C_1) = 3\gamma_c + 1$.

$$3 + 2(|C_2| - 2) \leq f(C_1) - f(C_1 \cup C_2 \setminus \{v\}) \leq 3\gamma_c + 1 - (2\gamma_c + 2)$$
$$\Rightarrow |C_2| \leq \gamma_c/2.$$

Case 3: $f(C_1) = 3\gamma_c + 2$.

$$3 + 2(|C_2| - 2) \leq f(C_1) - f(C_1 \cup C_2 \setminus \{v\}) \leq 3\gamma_c + 2 - (2\gamma_c + 2)$$
$$\Rightarrow |C_2| \leq \frac{\gamma_c + 1}{2}.$$

Upper Bound on $|C_3|$

Case 1: $f(C_1 \cup C_2) \le 2\gamma_c$.

$$|C_3| \le f(C_1 \cup C_2) - 1 \le 2\gamma_c - 1.$$

Case 2: $f(C_1 \cup C_2) = 2\gamma_c + 1$.

$$2 + (|C_3| - 1) \le f(C_1 \cup C_2) - 1 = 2\gamma_c + 1 - 1$$
$$\Rightarrow |C_3| \le 2\gamma_c - 1$$