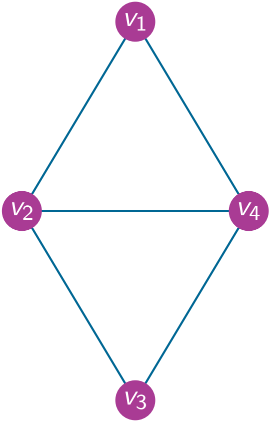
# Facility Location on Planar Graphs with Unreliable Links

N. S. Narayanaswamy
Meghana Nasre
**R. Vijayaragunathan**

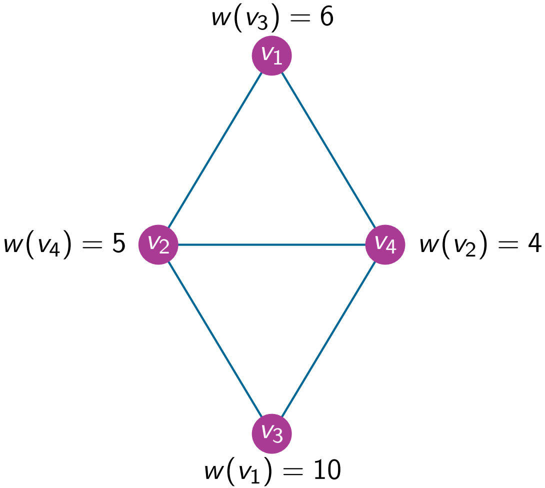
Computer Science and Engineering
Indian Institute of Technology Madras

Sunday 10$^{th}$ June, 2018
The 13$^{th}$ International Computer Science Symposium in Russia

# Facility Networks

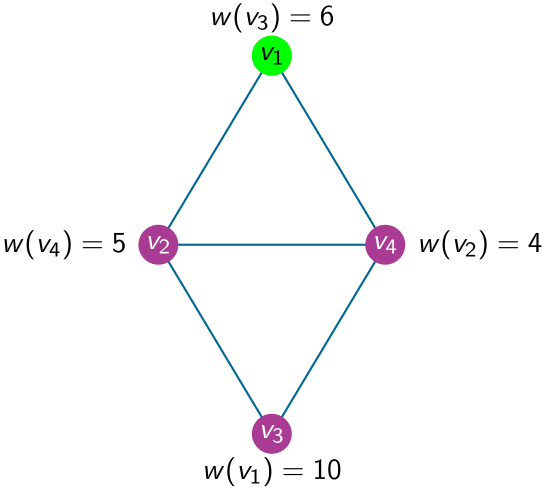


# Facility Networks



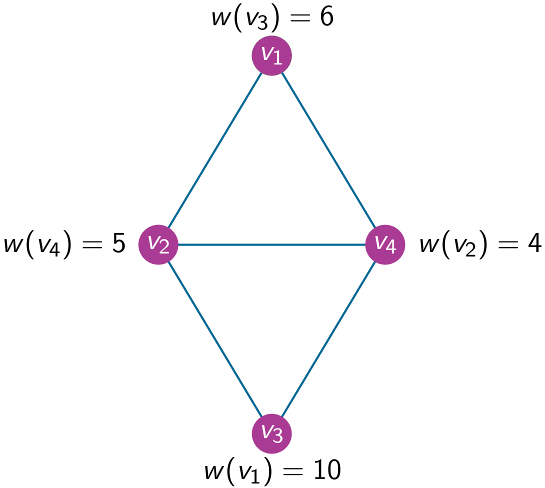

$$w : V \to \mathbb{R}^+$$

# Facility Networks




$$w : V \to \mathbb{R}^+$$

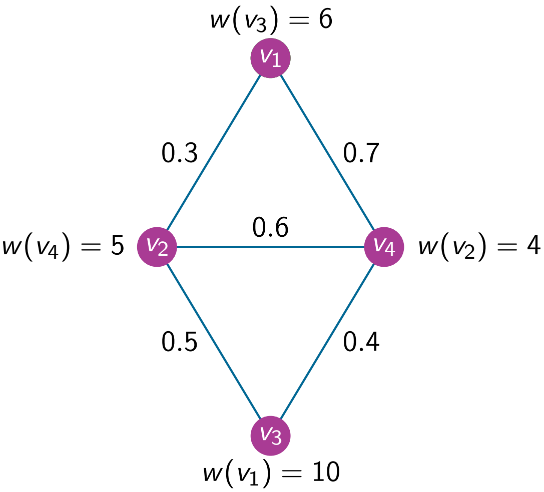
# Facility Networks




$$w : V \to \mathbb{R}^+$$

Budget $\mathcal{B}$

# Facility Networks

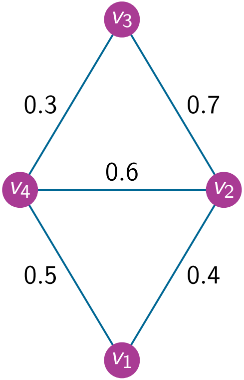



$w : V \rightarrow \mathbb{R}^{+}$

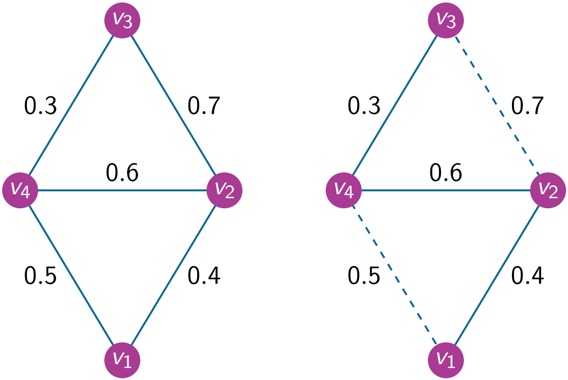
Budget $\mathcal{B}$

$p : E \rightarrow [0, 1]$

# Sub-graph Realizations

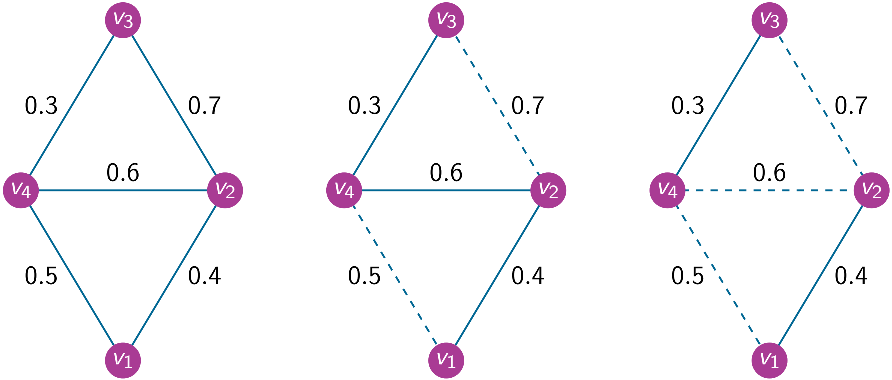


# Sub-graph Realizations




$Q_1$, $P(Q_1)$

# Sub-graph Realizations




$Q_1,\ P(Q_1)$ $Q_2,\ P(Q_2)$

# The MAX-EXP-COVER Problem

# The Max-Exp-Cover Problem

Input: A graph $G = (V, E)$

Demand $w : V \to \mathbb{R}^+$

Survival probability $p : E \to [0, 1]$

Budget $\mathcal{B}$

# The Max-Exp-Cover Problem

Input: A graph $G = (V, E)$

Demand $w : V \to \mathbb{R}^+$

Survival probability $p : E \to [0, 1]$

Budget $\mathcal{B}$

Assume: $\mathcal{Q} \subseteq 2^E$ with $P : \mathcal{Q} \to [0, 1]$

# The MAX-EXP-COVER Problem

Input: A graph $G = (V, E)$

Demand $w : V \to \mathbb{R}^+$

Survival probability $p : E \to [0, 1]$

Budget $\mathcal{B}$

Assume: $\mathcal{Q} \subseteq 2^E$ with $P : \mathcal{Q} \to [0, 1]$

Compute: $$\max_{F \subseteq V, |F| \leq \mathcal{B}} \sum_{Q \in \mathcal{Q}} P(Q) \sum_{v \in V} w(v) \cdot I(Q, F, v)$$

# The Max-Exp-Cover Problem

Input: A graph $G = (V, E)$

Demand $w : V \to \mathbb{R}^+$

Survival probability $p : E \to [0, 1]$

Budget $\mathcal{B}$

Assume: $\mathcal{Q} \subseteq 2^E$ with $P : \mathcal{Q} \to [0, 1]$

Compute:

$$\max_{F \subseteq V, |F| \leq \mathcal{B}} \sum_{Q \in \mathcal{Q}} P(Q) \sum_{v \in V} w(v) \cdot I(Q, F, v)$$

$$I(Q, F, v) = \begin{cases} 1 & \text{if } v \in N_Q[F] \\ 0 & \text{otherwise} \end{cases}$$

# Coverage Function

# Coverage Function

## The coverage function $\mathcal{C}$

Given a set $F \subseteq V$ and a vertex $v \in V$, the function $\mathcal{C}(v, F)$ is the expected coverage of $v$ by $F$.

$$\mathcal{C}(v, F) = w(v) \cdot \sum_{Q \in \mathcal{Q}} P(Q) \cdot I(Q, F, v)$$

# Coverage Function

## The coverage function $\mathcal{C}$

Given a set $F \subseteq V$ and a vertex $v \in V$, the function $\mathcal{C}(v, F)$ is the expected coverage of $v$ by $F$.

$$\mathcal{C}(v, F) = w(v) \cdot \sum_{Q \in \mathcal{Q}} P(Q) \cdot I(Q, F, v)$$

$$\sum_{Q \in \mathcal{Q}} P(Q) \sum_{v \in V} w(v) \cdot I(Q, F, v) \quad = \quad \sum_{v \in V} w(v) \sum_{Q \in \mathcal{Q}} P(Q) \cdot I(Q, F, v)$$

# Coverage Function

## The coverage function $\mathcal{C}$

Given a set $F \subseteq V$ and a vertex $v \in V$, the function $\mathcal{C}(v, F)$ is the expected coverage of $v$ by $F$.

$$\mathcal{C}(v, F) = w(v) \cdot \sum_{Q \in \mathcal{Q}} P(Q) \cdot I(Q, F, v)$$

$$\begin{aligned} \sum_{Q \in \mathcal{Q}} P(Q) \sum_{v \in V} w(v) \cdot I(Q, F, v) &= \sum_{v \in V} w(v) \sum_{Q \in \mathcal{Q}} P(Q) \cdot I(Q, F, v) \\ &= \sum_{v \in V} w(v) \sum_{Q \in \mathcal{Q} : v \in N_Q[F]} P(Q) \end{aligned}$$

# Coverage Function

## The coverage function $\mathcal{C}$

Given a set $F \subseteq V$ and a vertex $v \in V$, the function $\mathcal{C}(v, F)$ is the expected coverage of $v$ by $F$.

$$\mathcal{C}(v, F) = w(v) \cdot \sum_{Q \in \mathcal{Q}} P(Q) \cdot I(Q, F, v)$$

$$\begin{aligned}
\sum_{Q \in \mathcal{Q}} P(Q) \sum_{v \in V} w(v) \cdot I(Q, F, v) &= \sum_{v \in V} w(v) \sum_{Q \in \mathcal{Q}} P(Q) \cdot I(Q, F, v) \\
&= \sum_{v \in V} w(v) \sum_{Q \in \mathcal{Q} : v \in N_Q[F]} P(Q) \\
&= \sum_{v \in V} \mathcal{C}(v, F) = \mathcal{C}(V, F)
\end{aligned}$$

# LRO Model

# LRO Model

**Vulnerability-Based Dependency [Hassin et al., 2009]**

- Given $e_i$ and $e_j$ such that $p(e_i) > p(e_j)$
- $\Pr[e_j \text{ fails} \mid e_i \text{ fails}] = 1$

If an edge $e_i$ fails then the weaker edges than $e_i$ surely fails.

# LRO Model

## Vulnerability-Based Dependency [Hassin et al., 2009]

- Given $e_i$ and $e_j$ such that $p(e_i) > p(e_j)$
- $\Pr[e_j \text{ fails} \mid e_i \text{ fails}] = 1$

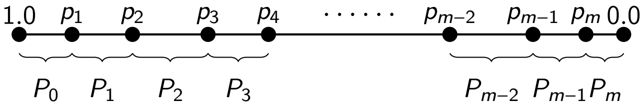
If an edge $e_i$ fails then the weaker edges than $e_i$ surely fails.

## Linear Reliable Ordering [Hassin et al., 2017]

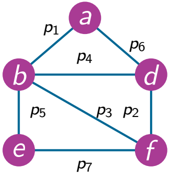
- Every pair of edges are following VB-dependency.
- $m + 1$ realizations are possible.
- Let $G_0, G_1, \ldots, G_m$ be all the possible realizations.

# LRO Model

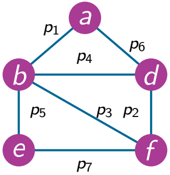
- Order the edges $e_1, e_2, \ldots, e_m$ in descending order of survival probability.
- $G_0$ - Empty graph. When $e_1$ fails.
- $G_i$ occurs when $e_i$ is the weakest link that survives.



# LRO Instance

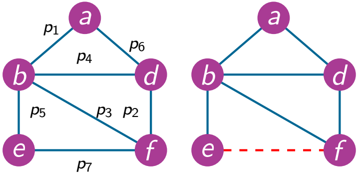


# LRO Instance



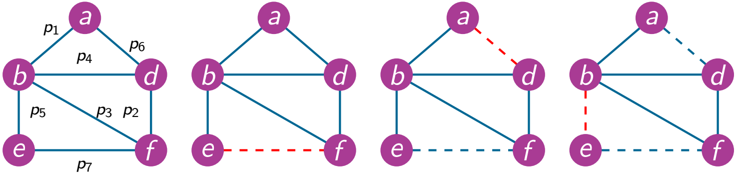

$G, G_7, p_7$

# LRO Instance

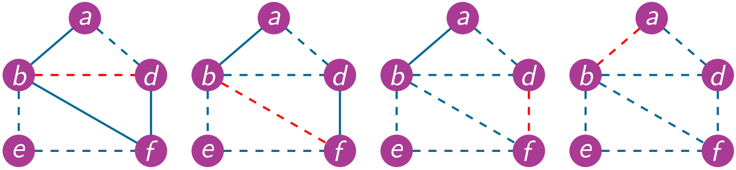



$G$, $G_7$, $p_7$ $\qquad$ $G_6$, $p_6 - p_7$

# LRO Instance




$G, G_7, p_7$ | $G_6, p_6 - p_7$ | $G_5, p_5 - p_6$ | $G_4, p_4 - p_5$




$G_3, p_3 - p_4$ | $G_3, p_2 - p_3$ | $G_1, p_1 - p_2$ | $G_0, 1 - p_1$

# Results

# Results

**The MAX-EXP-COVER problem with LRO Model - Existing**

- When $R = 1$, NP-Hard [Hassin et al., 2009]
- When $R = \infty$, O(m+n) [Hassin et al., 2009]

# Results

**The MAX-EXP-COVER problem with LRO Model - Existing**

- When $R = 1$, NP-Hard [Hassin et al., 2009]
- When $R = \infty$, O(m+n) [Hassin et al., 2009]

**The MAX-EXP-COVER problem with LRO Model - In this work**

- When $R = 1$, FPT on bounded treewidth graph,

# Results

**The MAX-EXP-COVER problem with LRO Model - Existing**

- When $R = 1$, NP-Hard [Hassin et al., 2009]
- When $R = \infty$, O(m+n) [Hassin et al., 2009]

**The MAX-EXP-COVER problem with LRO Model - In this work**

- When $R = 1$, FPT on bounded treewidth graph,
  and PTAS on planar graph

# Results

**The MAX-EXP-COVER problem with LRO Model - Existing**

- When $R = 1$, NP-Hard [Hassin et al., 2009]
- When $R = \infty$, O(m+n) [Hassin et al., 2009]

**The MAX-EXP-COVER problem with LRO Model - In this work**

- When $R = 1$, FPT on bounded treewidth graph, and PTAS on planar graph
- Observed that, the problem has greedy approximation algorithm $(1 - \frac{1}{e})$.

# Tree Decomposition

$H$ - Tree, $\mathcal{X} = \{X_i \subseteq V \mid i \in H\}$ - Bag
A pair $(H, \mathcal{X})$ satisfy the following conditions.

1. $\forall v \in V, \exists i \in H \mid v \in X_i$.
2. $\forall uv \in E, \exists i \in H \mid u, v \in X_i$.
3. $\forall v \in V$, let $T_v = \{i \in H \mid v \in X_i\}$, then $H[T_v]$ is connected.

# Tree Decomposition

$H$ - Tree, $\mathcal{X} = \{X_i \subseteq V \mid i \in H\}$ - Bag
A pair $(H, \mathcal{X})$ satisfy the following conditions.

1. $\forall v \in V, \exists i \in H \mid v \in X_i$.
2. $\forall uv \in E, \exists i \in H \mid u, v \in X_i$.
3. $\forall v \in V$, let $T_v = \{i \in H \mid v \in X_i\}$, then $H[T_v]$ is connected.

- $width = \max_{i \in H} |X_i| - 1$.

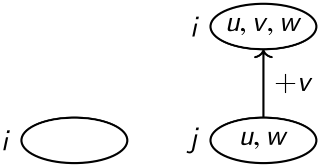
# Nice Tree Decomposition

# Nice Tree Decomposition

- **Leaf:** $i$ has no child and $X_i = \{\}$.

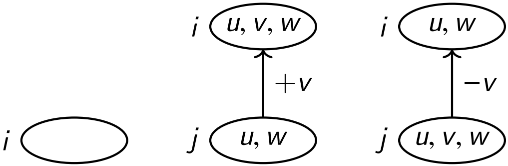
# Nice Tree Decomposition

- **Leaf:** $i$ has no child and $X_i = \{\}$.
- **Introduce:** $i$ has a child $j$ : $X_i = X_j \cup \{v\}$ for some $v \notin X_j$.



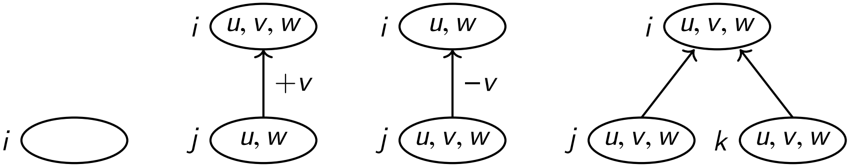
# Nice Tree Decomposition

- **Leaf:** $i$ has no child and $X_i = \{\}$.
- **Introduce:** $i$ has a child $j$ : $X_i = X_j \cup \{v\}$ for some $v \notin X_j$.
- **Forget:** $i$ has a child $j$ : $X_i = X_j \setminus \{v\}$ for some $v \in X_j$.



# Nice Tree Decomposition

- **Leaf:** $i$ has no child and $X_i = \{\}$.
- **Introduce:** $i$ has a child $j : X_i = X_j \cup \{v\}$ for some $v \notin X_j$.
- **Forget:** $i$ has a child $j : X_i = X_j \setminus \{v\}$ for some $v \in X_j$.
- **Join:** $i$ has two children $j$ and $k : X_i = X_j = X_k$.



# Best Neighbour

# Best Neighbour

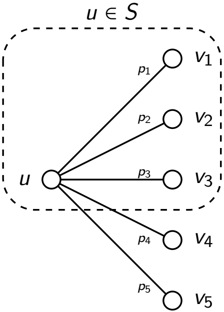
Given a vertex $u \in V$ and a set $S \subseteq V$,

$$bn(u,S) = \begin{cases} u & \text{if } u \in S \\ v = \max_{v' \in N(v) \cap S} p(uv') & \text{if } u \notin S \wedge N(u) \cap S \neq \emptyset \\ \text{undefined} & \text{otherwise} \end{cases}$$

# Best Neighbour

Given a vertex $u \in V$ and a set $S \subseteq V$,

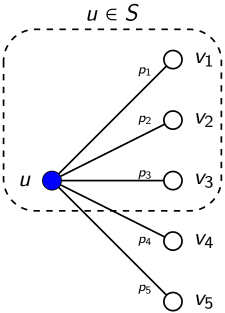
$$bn(u,S) = \begin{cases} u & \text{if } u \in S \\ v = \max_{v' \in N(v) \cap S} p(uv') & \text{if } u \notin S \wedge N(u) \cap S \neq \emptyset \\ \text{undefined} & \text{otherwise} \end{cases}$$



# Best Neighbour

Given a vertex $u \in V$ and a set $S \subseteq V$,

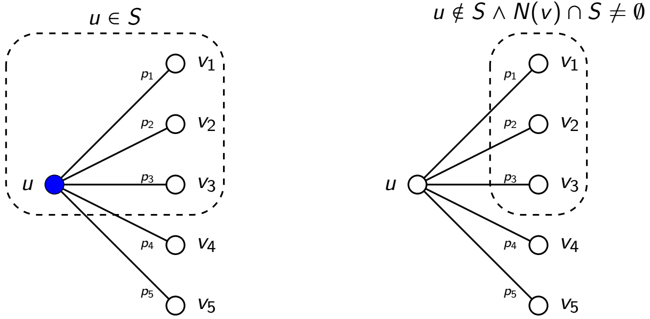
$$bn(u,S) = \begin{cases} u & \text{if } u \in S \\ v = \max_{v' \in N(v) \cap S} p(uv') & \text{if } u \notin S \wedge N(u) \cap S \neq \emptyset \\ \text{undefined} & \text{otherwise} \end{cases}$$



# Best Neighbour

Given a vertex $u \in V$ and a set $S \subseteq V$,

$$bn(u, S) = \begin{cases} u & \text{if } u \in S \\ v = \max_{v' \in N(v) \cap S} p(uv') & \text{if } u \notin S \wedge N(u) \cap S \neq \emptyset \\ \text{undefined} & \text{otherwise} \end{cases}$$

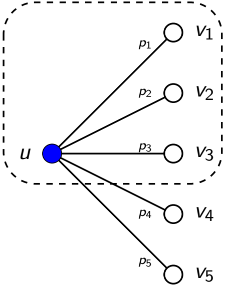


# Best Neighbour

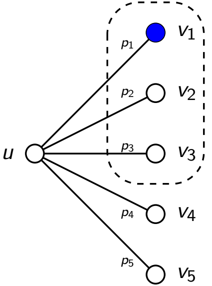
Given a vertex $u \in V$ and a set $S \subseteq V$,

$$bn(u,S) = \begin{cases} u & \text{if } u \in S \\ v = \max_{v' \in N(v) \cap S} p(uv') & \text{if } u \notin S \wedge N(u) \cap S \neq \emptyset \\ \text{undefined} & \text{otherwise} \end{cases}$$

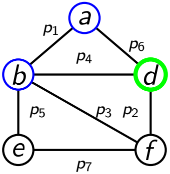
$u \in S$



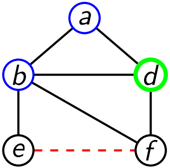

$u \notin S \wedge N(v) \cap S \neq \emptyset$



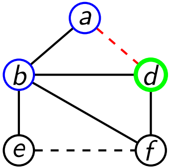
# Best Neighbour - In LRO



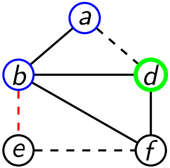

$G$, $G_7$, $p_7$



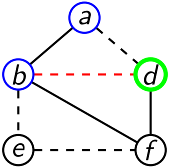

$G_6$, $p_6 - p_7$



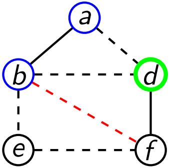

$G_5$, $p_5 - p_6$



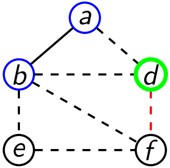

$G_4$, $p_4 - p_5$



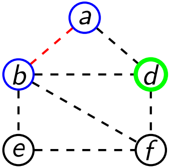

$G_3$, $p_3 - p_4$




$G_3$, $p_2 - p_3$

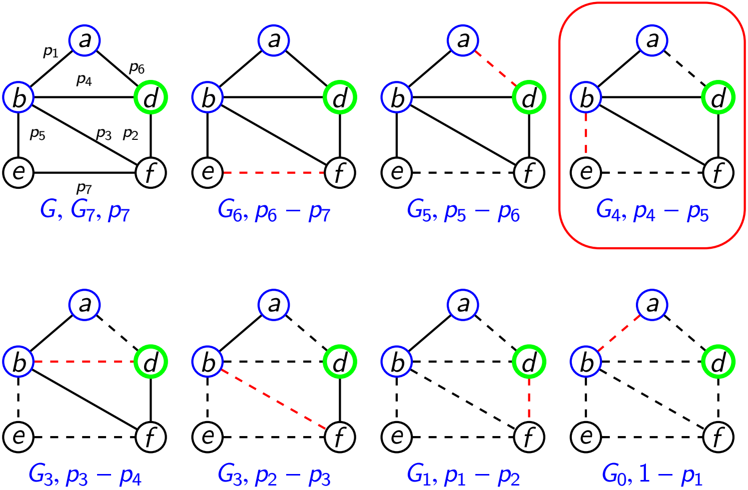



$G_1$, $p_1 - p_2$




$G_0$, $1 - p_1$

# Best Neighbour - In LRO



# Coverage using Best Neighbour

**Lemma**

*Let $u \in V$ be a vertex and $S \subseteq V$ be a set. If the coverage $\mathcal{C}(u, S) > 0$, then there is a vertex $v \in S$ such that $\mathcal{C}(u, S) = \mathcal{C}(u, v)$.*

# Coverage using Best Neighbour

## Lemma

*Let $u \in V$ be a vertex and $S \subseteq V$ be a set. If the coverage $\mathcal{C}(u, S) > 0$, then there is a vertex $v \in S$ such that $\mathcal{C}(u, S) = \mathcal{C}(u, v)$.*

## Proof.

- Since $\mathcal{C}(u, S) > 0$, $N[v] \cap S \neq \emptyset$. Then $S' = N[v] \cap S$.

# Coverage using Best Neighbour

## Lemma

*Let $u \in V$ be a vertex and $S \subseteq V$ be a set. If the coverage $\mathcal{C}(u, S) > 0$, then there is a vertex $v \in S$ such that $\mathcal{C}(u, S) = \mathcal{C}(u, v)$.*

## Proof.

- Since $\mathcal{C}(u, S) > 0$, $N[v] \cap S \neq \emptyset$. Then $S' = N[v] \cap S$.
- When $u \in S$, then $\mathcal{C}(u, S) = \mathcal{C}(u, u) = w(u)$.

# Coverage using Best Neighbour

**Lemma**

*Let $u \in V$ be a vertex and $S \subseteq V$ be a set. If the coverage $\mathcal{C}(u, S) > 0$, then there is a vertex $v \in S$ such that $\mathcal{C}(u, S) = \mathcal{C}(u, v)$.*

**Proof.**

- Since $\mathcal{C}(u, S) > 0$, $N[v] \cap S \neq \emptyset$. Then $S' = N[v] \cap S$.
- When $u \in S$, then $\mathcal{C}(u, S) = \mathcal{C}(u, u) = w(u)$.
- Suppose $u \notin S$, then $S' = \{v_1, v_2, \ldots, v_\ell\}$ for some $0 < \ell \leq d_u$.

# Coverage using Best Neighbour

## Lemma

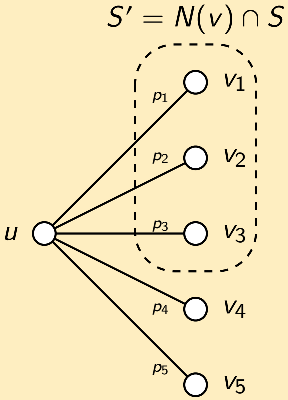
*Let $u \in V$ be a vertex and $S \subseteq V$ be a set. If the coverage $\mathcal{C}(u, S) > 0$, then there is a vertex $v \in S$ such that $\mathcal{C}(u, S) = \mathcal{C}(u, v)$.*

## Proof.

- Since $\mathcal{C}(u, S) > 0$, $N[v] \cap S \neq \emptyset$. Then $S' = N[v] \cap S$.
- When $u \in S$, then $\mathcal{C}(u, S) = \mathcal{C}(u, u) = w(u)$.
- Suppose $u \notin S$, then $S' = \{v_1, v_2, \ldots, v_\ell\}$ for some $0 < \ell \leq d_u$.
- Assume $p(uv_1) > p(uv_2) > \cdots > p(uv_\ell)$.

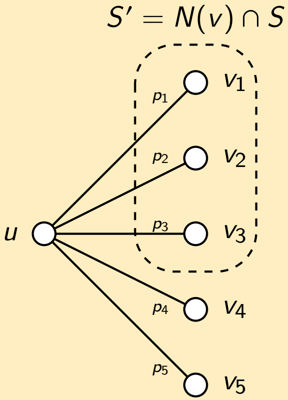
## Proof. (Cont)

$$\mathcal{C}(u, S) \;=\; \mathcal{C}(u, S')$$



## Proof. (Cont)

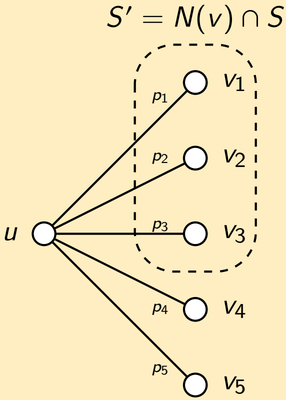
$$\begin{aligned} \mathcal{C}(u, S) &= \mathcal{C}(u, S') \\ &= w(u) \cdot \sum_{G_i \mid u \in N_{G_i}[S']} p(G_i) \end{aligned}$$



## Proof. (Cont)

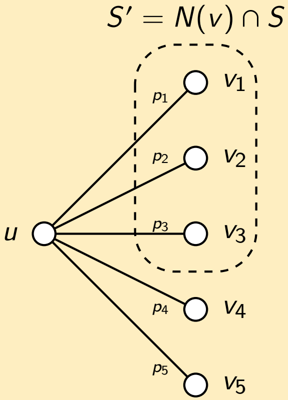
$$\begin{aligned} \mathcal{C}(u,S) &= \mathcal{C}(u,S') \\ &= w(u) \cdot \sum_{G_i \mid u \in N_{G_i}[S']} p(G_i) \\ &= w(u) \cdot \sum_{i=j}^{m} p(G_i) \end{aligned}$$

$G_j \mid (u, v_1)$ survives



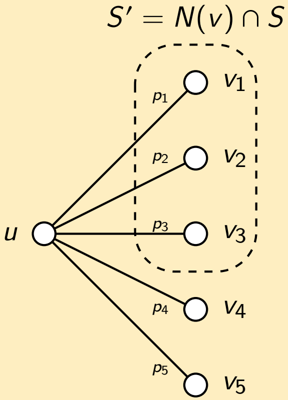
## Proof. (Cont)

$$
\begin{aligned}
\mathcal{C}(u,S) &= \mathcal{C}(u,S') \\
&= w(u)\cdot \sum_{G_i \mid u\in N_{G_i}[S']} p(G_i) \\
&= w(u)\cdot \sum_{i=j}^{m} p(G_i) \\
&\quad G_j \mid (u,v_1) \text{ survives} \\
&= w(u)\cdot p(uv_1)
\end{aligned}
$$



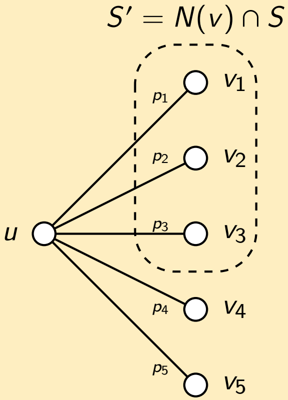
## Proof. (Cont)

$$
\begin{aligned}
C(u,S) &= C(u,S') \\
&= w(u) \cdot \sum_{G_i \mid u \in N_{G_i}[S']} p(G_i) \\
&= w(u) \cdot \sum_{i=j}^{m} p(G_i) \\
& \quad G_j \mid (u, v_1) \text{ survives} \\
&= w(u) \cdot p(uv_1) \\
&= C(u, v_1)
\end{aligned}
$$



## Proof. (Cont)

$$
\begin{aligned}
\mathcal{C}(u,S) &= \mathcal{C}(u,S') \\
&= w(u)\cdot \sum_{G_i \mid u\in N_{G_i}[S']} p(G_i) \\
&= w(u)\cdot \sum_{i=j}^{m} p(G_i) \\
& \quad G_j \mid (u,v_1) \text{ survives} \\
&= w(u)\cdot p(uv_1) \\
&= \mathcal{C}(u,v_1) \\
&= \mathcal{C}(u, bn(u,S))
\end{aligned}
$$




Hence the proof. □

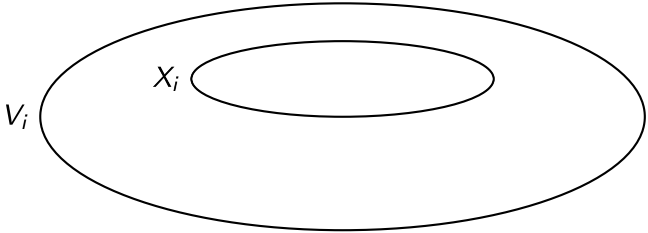
# Structure of a Solution

# Structure of a Solution

- Let $i$ be a node in $H$ with bag $X_i$ and vertex set $V_i$.

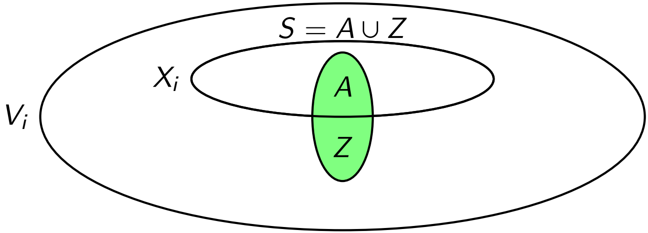
# Structure of a Solution

- Let $i$ be a node in $H$ with bag $X_i$ and vertex set $V_i$.
- Let $S = A \cup Z$ be a solution such that $A = S \cap X_i$ and $Z = S \setminus A$.



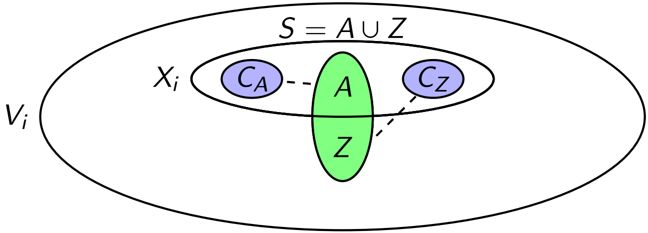
# Structure of a Solution

- Let $i$ be a node in $H$ with bag $X_i$ and vertex set $V_i$.
- Let $S = A \cup Z$ be a solution such that $A = S \cap X_i$ and $Z = S \setminus A$.
- $C_A = \{u \in X_i \mid bn(u, S) \in A\}$ and $C_Z = \{u \in X_i \mid bn(u, S) \in Z\}$.



# Structure of a Solution

- Let $i$ be a node in $H$ with bag $X_i$ and vertex set $V_i$.
- Let $S = A \cup Z$ be a solution such that $A = S \cap X_i$ and $Z = S \setminus A$.
- $C_A = \{u \in X_i \mid bn(u, S) \in A\}$ and $C_Z = \{u \in X_i \mid bn(u, S) \in Z\}$.
- $U = X_i \setminus (A \cup C_A \cup C_Z)$.



# Solution Structure

$$\mathcal{C}(V_i, S) \quad = \quad \mathcal{C}((V_i \setminus X_i) \cup X_i, S)$$

# Solution Structure

$$\begin{aligned} \mathcal{C}(V_i, S) &= \mathcal{C}((V_i \setminus X_i) \cup X_i, S) \\ &= \mathcal{C}(V_i \setminus X_i, S) + \mathcal{C}(C_A, S) + \mathcal{C}(C_Z, S) + \mathcal{C}(A, S) \end{aligned}$$

# Solution Structure

$$\begin{aligned}
\mathcal{C}(V_i, S) &= \mathcal{C}((V_i \setminus X_i) \cup X_i, S) \\
&= \mathcal{C}(V_i \setminus X_i, S) + \mathcal{C}(C_A, S) + \mathcal{C}(C_Z, S) + \mathcal{C}(A, S) \\
&= \mathcal{C}(V_i \setminus X_i, S) + \mathcal{C}(C_A, {\color{red}A}) + \mathcal{C}(C_Z, {\color{red}Z}) + \mathcal{C}(A, {\color{red}A})
\end{aligned}$$

# Solution Structure

$$
\begin{aligned}
\mathcal{C}(V_i, S) &= \mathcal{C}((V_i \setminus X_i) \cup X_i, S) \\
&= \mathcal{C}(V_i \setminus X_i, S) + \mathcal{C}(C_A, S) + \mathcal{C}(C_Z, S) + \mathcal{C}(A, S) \\
&= \mathcal{C}(V_i \setminus X_i, S) + \mathcal{C}(C_A, \color{red}{A}) + \mathcal{C}(C_Z, \color{red}{Z}) + \mathcal{C}(A, \color{red}{A}) \\
&= \mathcal{C}(V_i \setminus X_i, S) + \mathcal{C}(\color{red}{A \cup C_A}, A) + \mathcal{C}(C_Z, Z)
\end{aligned}
$$

# Solution Structure

$$
\begin{aligned}
\mathcal{C}(V_i, S) &= \mathcal{C}((V_i \setminus X_i) \cup X_i, S) \\
&= \mathcal{C}(V_i \setminus X_i, S) + \mathcal{C}(C_A, S) + \mathcal{C}(C_Z, S) + \mathcal{C}(A, S) \\
&= \mathcal{C}(V_i \setminus X_i, S) + \mathcal{C}(C_A, {\color{red}A}) + \mathcal{C}(C_Z, {\color{red}Z}) + \mathcal{C}(A, {\color{red}A}) \\
&= \mathcal{C}(V_i \setminus X_i, S) + \mathcal{C}({\color{red}A \cup C_A}, A) + \mathcal{C}(C_Z, Z)
\end{aligned}
$$

## Partition

For every feasible solution $S \subseteq V_i$, there is a four-way partition $\mathtt{P} = (A, C_A, C_Z, U)$ of $X_i$ such that

$$\mathcal{C}(V_i, S) = \mathcal{C}(V_i \setminus X_i, S) + \mathcal{C}(A \cup C_A, A) + \mathcal{C}(C_Z, Z)$$

# Dynamic Programming

- For each feasible solution, there exists an unique partition P such that the coverage of $S$ can be expressed using P.

# Dynamic Programming

- For each feasible solution, there exists an unique partition $\texttt{P}$ such that the coverage of $S$ can be expressed using $\texttt{P}$.
- We explore all four way partitions, and find the optimal solution *w.r.t* the partition.

# Dynamic Programming

- For each feasible solution, there exists an unique partition P such that the coverage of $S$ can be expressed using P.
- We explore all four way partitions, and find the optimal solution *w.r.t* the partition.
- Maintains a table of size $(\mathcal{B}+1)4^{|X_i|}$.

# Dynamic Programming

- For each feasible solution, there exists an unique partition $\texttt{P}$ such that the coverage of $S$ can be expressed using $\texttt{P}$.
- We explore all four way partitions, and find the optimal solution *w.r.t* the partition.
- Maintains a table of size $(\mathcal{B}+1)4^{|X_i|}$.
- For each $0 \leq b \leq \mathcal{B}$ and each four-way partition $\texttt{P} = (A, C_A, C_Z, U)$ of $X_i$, $T_i[b, \texttt{P}]$ is a tuple $(\textbf{solution}, \textbf{value})$ such that

# Dynamic Programming

- For each feasible solution, there exists an unique partition $\mathtt{P}$ such that the coverage of $S$ can be expressed using $\mathtt{P}$.
- We explore all four way partitions, and find the optimal solution *w.r.t* the partition.
- Maintains a table of size $(\mathcal{B}+1)4^{|X_i|}$.
- For each $0 \leq b \leq \mathcal{B}$ and each four-way partition $\mathtt{P} = (A, C_A, C_Z, U)$ of $X_i$, $T_i[b, \mathtt{P}]$ is a tuple $(\textbf{solution}, \textbf{value})$ such that
  - $T_i[b, \mathtt{P}].\textbf{solution} = S \subseteq V_i$.

# Dynamic Programming

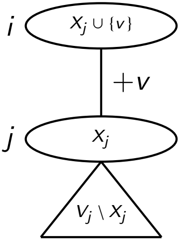
- For each feasible solution, there exists an unique partition $\mathtt{P}$ such that the coverage of $S$ can be expressed using $\mathtt{P}$.
- We explore all four way partitions, and find the optimal solution *w.r.t* the partition.
- Maintains a table of size $(\mathcal{B}+1)4^{|X_i|}$.
- For each $0 \leq b \leq \mathcal{B}$ and each four-way partition $\mathtt{P} = (A, C_A, C_Z, U)$ of $X_i$, $T_i[b, \mathtt{P}]$ is a tuple $(\textbf{solution}, \textbf{value})$ such that
  - $T_i[b, \mathtt{P}].\textbf{solution} = S \subseteq V_i$.
  - $|S| = b$ and $S \cap X_i = A$.

# Dynamic Programming

- For each feasible solution, there exists an unique partition $\mathtt{P}$ such that the coverage of $S$ can be expressed using $\mathtt{P}$.
- We explore all four way partitions, and find the optimal solution *w.r.t* the partition.
- Maintains a table of size $(\mathcal{B}+1)4^{|X_i|}$.
- For each $0 \leq b \leq \mathcal{B}$ and each four-way partition $\mathtt{P} = (A, C_A, C_Z, U)$ of $X_i$, $T_i[b, \mathtt{P}]$ is a tuple $(\textbf{solution}, \textbf{value})$ such that
  - $T_i[b, \mathtt{P}].\textbf{solution} = S \subseteq V_i$.
  - $|S| = b$ and $S \cap X_i = A$.
  - $\mathcal{C}(A \cup C_A, A) + \mathcal{C}(C_Z, Z) + \mathcal{C}(V_i \setminus X_i, S)$ is maximized over all possible such $S \subseteq V_i$.

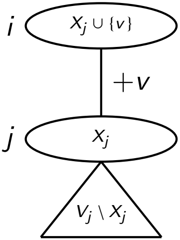
# Introduce Node

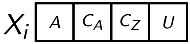
- Let $i$ be an introduce node with child $j$ such that $X_i = X_j \cup \{v\}$ for some $v \notin X_j$.



# Introduce Node

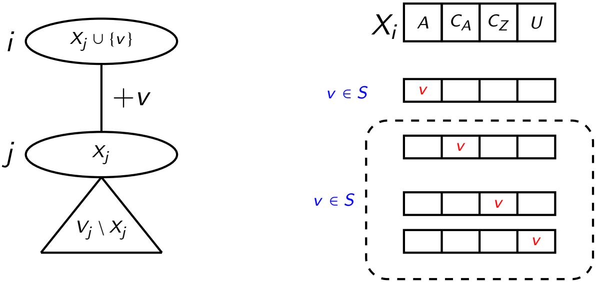
- Let $i$ be an introduce node with child $j$ such that $X_i = X_j \cup \{v\}$ for some $v \notin X_j$.
- Let $0 \leq b \leq \mathcal{B}$ be a budget and $\mathtt{P} = (A, C_A, C_Z, U)$ be a four way partitioning of $X_i$.



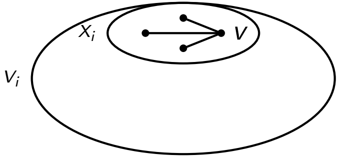



# Introduce Node

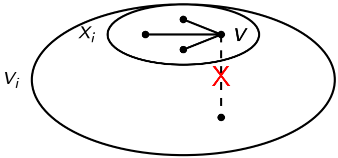
- Let $i$ be an introduce node with child $j$ such that $X_i = X_j \cup \{v\}$ for some $v \notin X_j$.
- Let $0 \le b \le \mathcal{B}$ be a budget and $\mathtt{P} = (A, C_A, C_Z, U)$ be a four way partitioning of $X_i$.
- We consider two cases that $(i) - v \notin A$ and $(ii) - v \in A$.



# Introduce Node (Cont...)

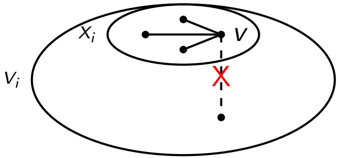


# Introduce Node (Cont...)




## Case: $v \notin A$

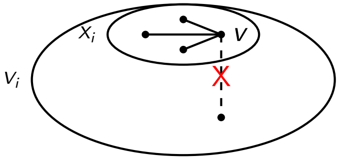
# Introduce Node (Cont...)




## Case: $v \notin A$

$$\mathtt{P} = (A, C_A, C_Z, U) \rightarrow \mathtt{P}_j = (A, C_A \setminus \{v\}, C_Z \setminus \{v\}, U \setminus \{v\})$$

# Introduce Node (Cont...)




**Case: $v \notin A$**

$\mathrm{P} = (A, C_A, C_Z, U) \rightarrow \mathrm{P}_j = (A, C_A \setminus \{v\}, C_Z \setminus \{v\}, U \setminus \{v\})$

$$T_i[b, \mathrm{P}].\textbf{solution} = T_j[b, \mathrm{P}_j].\textbf{solution}$$

$$T_i[b, \mathrm{P}].\textbf{value} = \begin{cases} T_j[b, \mathrm{P}_j].\textbf{value} & \text{if } v \notin C_A \\ T_j[b, \mathrm{P}_j].\textbf{value} + \mathcal{C}(v, A) & \text{if } v \in C_A \end{cases}$$

# Introduce Node (Cont...)

**Case: $v \in A$**

- Let $C_{A_v} = \{u \in C_A \mid bn(u, A) = v\}$.

# Introduce Node (Cont...)

**Case: $v \in A$**

- Let $C_{Av} = \{u \in C_A \mid bn(u, A) = v\}$.
- Let $\mathtt{P}_j = (A \setminus \{v\}, C_A \setminus C_{Av}, C_Z, U \cup C_{Av})$.

# Introduce Node (Cont...)

**Case: $v \in A$**

- Let $C_{Av} = \{u \in C_A \mid bn(u, A) = v\}$.
- Let $\mathtt{P}_j = (A \setminus \{v\}, C_A \setminus C_{Av}, C_Z, U \cup C_{Av})$.

$$\begin{aligned} T_i[b, \mathtt{P}].\textbf{solution} &= T_j[b-1, \mathtt{P}_j].\textbf{solution} \cup \{v\} \\ T_i[b, \mathtt{P}].\textbf{value} &= T_j[b-1, \mathtt{P}_j].\textbf{value} + \mathcal{C}(\{v\} \cup C_{Av}, v) \end{aligned}$$

# Correctness – case $v \notin A$

- Assume $v \in C_A$.
- Let $T_i[b, \mathtt{P}]$.**solution** $= T_j[b, \mathtt{P}']$.**solution** $= A \cup Z$ where $\mathtt{P}' = (A, C_A \setminus \{v\}, C_Z, U)$
- By contradiction, assume $S' = A \cup Z'$ optimal than $S$. That is $T_i[b, \mathtt{P}].\textbf{value} < \mathcal{C}(V_i \setminus X_i, S') + \mathcal{C}(A \cup C_A, A) + \mathcal{C}(C_Z, Z')$.

$$\begin{aligned} T_j[b, \mathtt{P}'].\textbf{value} &= T_i[b, \mathtt{P}].\textbf{value} - \mathcal{C}(v, A) \\ &< \mathcal{C}(V_i \setminus X_i, S') + \mathcal{C}(A \cup C_A, A) + \mathcal{C}(C_Z, Z') - \mathcal{C}(v, A) \\ &< \mathcal{C}(V_j \setminus X_j, S') + \mathcal{C}(A \cup C_A \setminus \{v\}, A) + \mathcal{C}(C_Z, Z') \end{aligned}$$

- Contradicts optimality of $T_j[b, \mathtt{P}']$ by $S'$.

# Correctness – case $v \in A$

- Let $C_{Av} = \{u \in C_A \mid bn(u, A) = v\}$.
- Let $T_i[b, \mathtt{P}]$.**solution** $= T_j[b-1, \mathtt{P}']$.**solution** $\cup \{v\} = A \cup Z$ where $\mathtt{P}' = (A \setminus \{v\}, C_A \setminus C_{Av}, C_Z, U \cup C_{Av})$
- By contradiction, assume $S' = A \cup Z'$ optimal than $S$. That is $T_i[b, \mathtt{P}]$.**value** $< \mathcal{C}(V_i \setminus X_i, S') + \mathcal{C}(A \cup C_A, A) + \mathcal{C}(C_Z, Z')$.

$$
\begin{aligned}
T_j[b, \mathtt{P}'].\textbf{value} &= T_i[b, \mathtt{P}].\textbf{value} - \mathcal{C}(\{v\} \cup C_{Av}, v) \\
&< \mathcal{C}(V_i \setminus X_i, S') + \mathcal{C}(A \cup C_A, A) + \mathcal{C}(C_Z, Z') \\
&\quad - \mathcal{C}(\{v\} \cup C_{Av}, v) \\
&< \mathcal{C}(V_j \setminus X_j, S') + \mathcal{C}((A \cup C_A) \setminus (\{v\}), A \setminus \{v\}) \\
&\quad + \mathcal{C}(C_Z, Z')
\end{aligned}
$$

- Contradicts optimality of $T_j[b, \mathtt{P}']$ by $S'$.

# Forget Node

- Let $0 \leq b \leq \mathcal{B}$ be a budget and $\texttt{P} = (A, C_A, C_Z, U)$ be a four way partitioning of $X_i$.

# Forget Node

- Let $0 \leq b \leq \mathcal{B}$ be a budget and $\mathtt{P} = (A, C_A, C_Z, U)$ be a four way partitioning of $X_i$.
- $\mathtt{P}_1 = (A \cup \{v\}, C_A, C_Z, U)$ $\quad$ $\mathtt{P}_2 = (A, C_A \cup \{v\}, C_Z, U)$
  $\mathtt{P}_3 = (A, C_A, C_Z \cup \{v\}, U)$ $\quad$ $\mathtt{P}_4 = (A, C_A, C_Z, U \cup \{v\})$

# Forget Node

- Let $0 \leq b \leq \mathcal{B}$ be a budget and $\mathtt{P} = (A, C_A, C_Z, U)$ be a four way partitioning of $X_i$.
- $\mathtt{P}_1 = (A \cup \{v\}, C_A, C_Z, U)$ $\quad$ $\mathtt{P}_2 = (A, C_A \cup \{v\}, C_Z, U)$
  $\mathtt{P}_3 = (A, C_A, C_Z \cup \{v\}, U)$ $\quad$ $\mathtt{P}_4 = (A, C_A, C_Z, U \cup \{v\})$
- Let $\mathtt{P}_j = \max_{\mathtt{P}' \in \{\mathtt{P}_1, \mathtt{P}_2, \mathtt{P}_3, \mathtt{P}_4\}} T_j[b, \mathtt{P}'].\textbf{value}$

# Forget Node

- Let $0 \leq b \leq \mathcal{B}$ be a budget and $\mathtt{P} = (A, C_A, C_Z, U)$ be a four way partitioning of $X_i$.
- $\mathtt{P}_1 = (A \cup \{v\}, C_A, C_Z, U)$ $\quad$ $\mathtt{P}_2 = (A, C_A \cup \{v\}, C_Z, U)$
  $\mathtt{P}_3 = (A, C_A, C_Z \cup \{v\}, U)$ $\quad$ $\mathtt{P}_4 = (A, C_A, C_Z, U \cup \{v\})$
- Let $\mathtt{P}_j = \max_{\mathtt{P}' \in \{\mathtt{P}_1, \mathtt{P}_2, \mathtt{P}_3, \mathtt{P}_4\}} T_j[b, \mathtt{P}'].\textbf{value}$

$$T_i[b, \mathtt{P}].\textbf{solution} = T_j[b, \mathtt{P}_j].\textbf{solution}$$
$$T_i[b, \mathtt{P}].\textbf{value} = T_j[b, \mathtt{P}_j].\textbf{value}$$

# Join Node

- Let $C_{Z_j} = \{u \in C_Z \mid bn(u, S) \in Z_j\}$

# Join Node

- Let $C_{Z_j} = \{u \in C_Z \mid bn(u, S) \in Z_j\}$
- Let $C_{Z_k} = \{u \in C_Z \mid bn(u, S) \in Z_k\}$

# Join Node

- Let $C_{Z\,j} = \{u \in C_Z \mid bn(u, S) \in Z_j\}$
- Let $C_{Z\,k} = \{u \in C_Z \mid bn(u, S) \in Z_k\}$
- Let $b_j = |Z_j|$ and $b_k = |Z_k|$.

$$(b', C_{Z\,j}, C_{Z\,k}) = \max_{\substack{0 \le b_1 \le b - |A|, \\ C_{Z\,1} \cup C_{Z\,2} = C_Z}} T_j[b_1 + |A|, \mathtt{P}'_j].\textbf{value} + T_k[b - b_1, \mathtt{P}'_k].\textbf{value}$$

# Join Node

- Let $C_{Zj} = \{u \in C_Z \mid bn(u, S) \in Z_j\}$
- Let $C_{Zk} = \{u \in C_Z \mid bn(u, S) \in Z_k\}$
- Let $b_j = |Z_j|$ and $b_k = |Z_k|$.

$$(b', C_{Zj}, C_{Zk}) = \max_{\substack{0 \leq b_1 \leq b - |A|, \\ C_{Z1} \cup C_{Z2} = C_Z}} T_j[b_1 + |A|, \mathtt{P}'_j].\mathbf{value} + T_k[b - b_1, \mathtt{P}'_k].\mathbf{value}$$

where $\mathtt{P}'_j = (A, C_A, C_{Z1}, U \cup C_{Z2})$ and $\mathtt{P}'_j = (A, C_A, C_{Z2}, U \cup C_{Z1})$

# Join Node

- Let $C_{Z\,j} = \{u \in C_Z \mid bn(u, S) \in Z_j\}$
- Let $C_{Z\,k} = \{u \in C_Z \mid bn(u, S) \in Z_k\}$
- Let $b_j = |Z_j|$ and $b_k = |Z_k|$.

$$(b', C_{Z\,j}, C_{Z\,k}) = \max_{\substack{0 \le b_1 \le b - |A|, \\ C_{Z\,1} \cup C_{Z\,2} = C_Z}} T_j[b_1 + |A|, \mathtt{P}'_j].\textbf{value} + T_k[b - b_1, \mathtt{P}'_k].\textbf{value}$$

where $\mathtt{P}'_j = (A, C_A, C_{Z\,1}, U \cup C_{Z\,2})$ and $\mathtt{P}'_j = (A, C_A, C_{Z\,2}, U \cup C_{Z\,1})$

$$\begin{aligned} T_i[b, \mathtt{P}].\textbf{solution} &= T_j[b' + |A|, \mathtt{P}_j].\textbf{solution} \cup T_k[b - b', \mathtt{P}_k].\textbf{solution} \\ T_i[b, \mathtt{P}].\textbf{value} &= T_j[b' + |A|, \mathtt{P}_j].\textbf{value} + T_k[b - b', \mathtt{P}_k].\textbf{value} \\ & \quad - \mathcal{C}(A \cup C_A, A) \end{aligned}$$

# PTAS in planar graphs

- We are given with an instance of $\textsc{Max-Exp-Cover-1}$ problem and an $\epsilon > 0$.

# PTAS in planar graphs

- We are given with an instance of MAX-EXP-COVER-1 problem and an $\epsilon > 0$.
- Compute an $l$-outerplanar embedding of $G$, for a minimum $l$. Level ordering such that $V = \cup_{i=1}^{l} L_i$.

# PTAS in planar graphs

- We are given with an instance of MAX-EXP-COVER-1 problem and an $\epsilon > 0$.
- Compute an $l$-outerplanar embedding of $G$, for a minimum $l$. Level ordering such that $V = \cup_{i=1}^{l} L_i$.
- Let $k = \frac{1}{\epsilon}$.

# PTAS in planar graphs

- We are given with an instance of MAX-EXP-COVER-1 problem and an $\epsilon > 0$.
- Compute an $l$-outerplanar embedding of $G$, for a minimum $l$. Level ordering such that $V = \cup_{i=1}^{l} L_i$.
- Let $k = \frac{1}{\epsilon}$.
- For $i = 1$ to $k$:

# PTAS in planar graphs

- We are given with an instance of MAX-EXP-COVER-1 problem and an $\epsilon > 0$.
- Compute an $l$-outerplanar embedding of $G$, for a minimum $l$. Level ordering such that $V = \cup_{i=1}^{l} L_i$.
- Let $k = \frac{1}{\epsilon}$.
- For $i = 1$ to $k$:
  - Let $G_i = (V, E_i)$ where $E_i = E \setminus \{(u, v) \in E \mid level(u) \equiv i \mod k, level(v) \equiv (i+1) \mod k\}$

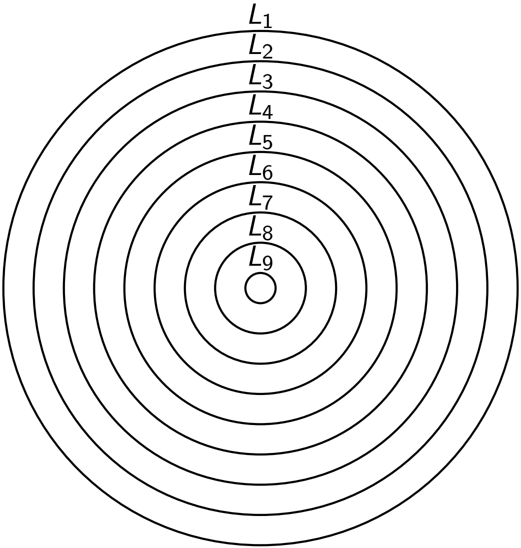
# PTAS in planar graphs

- We are given with an instance of MAX-EXP-COVER-1 problem and an $\epsilon > 0$.
- Compute an $l$-outerplanar embedding of $G$, for a minimum $l$. Level ordering such that $V = \cup_{i=1}^{l} L_i$.
- Let $k = \frac{1}{\epsilon}$.
- For $i = 1$ to $k$:
  - Let $G_i = (V, E_i)$ where
    $E_i = E \setminus \{(u, v) \in E \mid level(u) \equiv i \mod k, level(v) \equiv (i+1) \mod k\}$
- $G_i$ is a collection of $k$-outerplanar graphs.

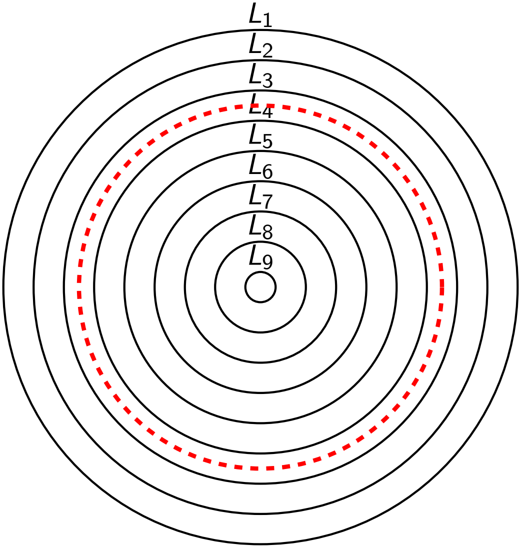
# PTAS in planar graphs

- We are given with an instance of MAX-EXP-COVER-1 problem and an $\epsilon > 0$.
- Compute an $l$-outerplanar embedding of $G$, for a minimum $l$. Level ordering such that $V = \cup_{i=1}^{l} L_i$.
- Let $k = \frac{1}{\epsilon}$.
- For $i = 1$ to $k$:
  - Let $G_i = (V, E_i)$ where $E_i = E \setminus \{(u, v) \in E \mid level(u) \equiv i \mod k, level(v) \equiv (i+1) \mod k\}$
- $G_i$ is a collection of $k$-outerplanar graphs.
- For a $k$-outerplanar graph $G$, a tree decomposition with width at most $3k + 1$ can be computed in linear time using [Shmoys and Williamson, 2011].

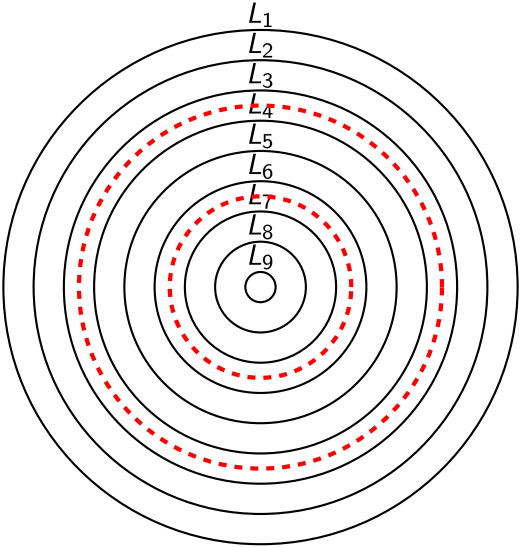
# Subgraph construction



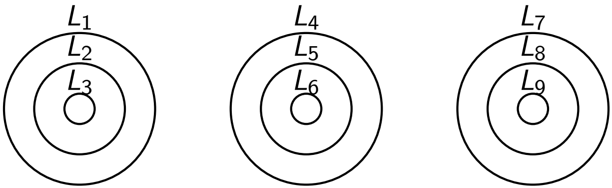
# Subgraph construction



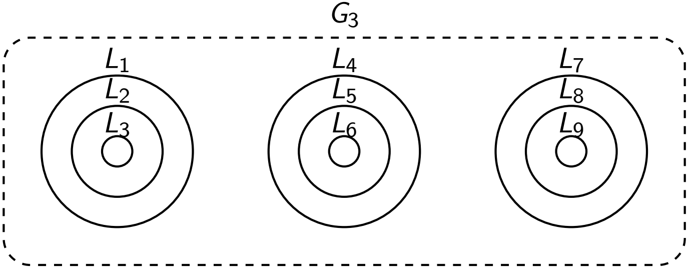
# Subgraph construction



# Subgraph construction



# Subgraph construction



# Solution Construction

- We have $k$ many $k$-outerplanar graphs $(G_1, G_2, \ldots . G_k)$.

# Solution Construction

- We have $k$ many $k$-outerplanar graphs $(G_1, G_2, \ldots .G_k)$.
- Compute the optimal solution $S_i$ in each graph $G_i$.

# Solution Construction

- We have $k$ many $k$-outerplanar graphs $(G_1, G_2, \ldots .G_k)$.
- Compute the optimal solution $S_i$ in each graph $G_i$.
- Let $S$ be the set achieving maximum expected coverage.

# Solution Construction

- We have $k$ many $k$-outerplanar graphs $(G_1, G_2, \ldots. G_k)$.
- Compute the optimal solution $S_i$ in each graph $G_i$.
- Let $S$ be the set achieving maximum expected coverage.

$$S = \max_{S' \in \{S_1, S_2, \ldots, S_k\}} \mathcal{C}(V, S')$$

- Output $S$ such that $\mathcal{C}(V, S)$ is at least $1 - \frac{1}{\epsilon}$ times of optimum expected coverage.

# Approximation Analysis

**Theorem**

*The set $S$ is $(1 - \frac{1}{k})$-approximate solution for the* $\textsc{Max-Exp-Cover-1}$ *problem.*

# Approximation Analysis

## Theorem

*The set $S$ is $(1 - \frac{1}{k})$-approximate solution for the* $\textsc{Max-Exp-Cover-1}$ *problem.*

## Proof.

- Let $OPT = \{v_1, v_2, \ldots, v_{\mathcal{B}}\}$.

# Approximation Analysis

**Theorem**

*The set $S$ is $(1 - \frac{1}{k})$-approximate solution for the* Max-Exp-Cover-1 *problem.*

**Proof.**

- Let $OPT = \{v_1, v_2, \ldots, v_{\mathcal{B}}\}$.
- Let $L_i = \{v \in V \mid level(v) \equiv i \mod k\}$.

# Approximation Analysis

## Theorem

*The set $S$ is $(1 - \frac{1}{k})$-approximate solution for the* MAX-EXP-COVER-1 *problem.*

## Proof.

- Let $OPT = \{v_1, v_2, \ldots, v_{\mathcal{B}}\}$.
- Let $L_i = \{v \in V \mid level(v) \equiv i \mod k\}$.
- Also, $L_i$ is the collection of level-$k$ vertices of $G_i$.

# Approximation Analysis

## Theorem

*The set $S$ is $(1 - \frac{1}{k})$-approximate solution for the* MAX-EXP-COVER-1 *problem.*

## Proof.

- Let $OPT = \{v_1, v_2, \ldots, v_{\mathcal{B}}\}$.
- Let $L_i = \{v \in V \mid level(v) \equiv i \mod k\}$.
- Also, $L_i$ is the collection of level-$k$ vertices of $G_i$.
- Since $L_1, L_2, \ldots L_k$ is a partition of $V$,

# Approximation Analysis

## Theorem

*The set $S$ is $(1 - \frac{1}{k})$-approximate solution for the* MAX-EXP-COVER-1 *problem.*

## Proof.

- Let $OPT = \{v_1, v_2, \ldots, v_{\mathcal{B}}\}$.
- Let $L_i = \{v \in V \mid level(v) \equiv i \mod k\}$.
- Also, $L_i$ is the collection of level-$k$ vertices of $G_i$.
- Since $L_1, L_2, \ldots L_k$ is a partition of $V$,

$$\mathcal{C}(V, OPT) = \sum_{i=1}^{k} \mathcal{C}(L_i, OPT)$$

# Approximation Analysis

## Proof.

- $\exists\ j$ such that $\mathcal{C}(L_j, OPT) \leq \frac{1}{k} \cdot \mathcal{C}(V, OPT)$.

# Approximation Analysis

Proof.

- $\exists\ j$ such that $\mathcal{C}(L_j, OPT) \leq \frac{1}{k} \cdot \mathcal{C}(V, OPT)$.
- Then, $\mathcal{C}(V \setminus L_j, OPT) \geq (1 - \frac{1}{k}) \cdot \mathcal{C}(V, OPT)$.

# Approximation Analysis

## Proof.

- $\exists\ j$ such that $\mathcal{C}(L_j, OPT) \leq \frac{1}{k} \cdot \mathcal{C}(V, OPT)$.
- Then, $\mathcal{C}(V \setminus L_j, OPT) \geq (1 - \frac{1}{k}) \cdot \mathcal{C}(V, OPT)$.
- Let $G_j$ be the graph corresponding to $j$.

# Approximation Analysis

Proof.

- $\exists\ j$ such that $\mathcal{C}(L_j, OPT) \leq \frac{1}{k} \cdot \mathcal{C}(V, OPT)$.
- Then, $\mathcal{C}(V \setminus L_j, OPT) \geq (1 - \frac{1}{k}) \cdot \mathcal{C}(V, OPT)$.
- Let $G_j$ be the graph corresponding to $j$.
- Since $\mathcal{C}(V, S) \geq \mathcal{C}(V, S_j)$.

# Approximation Analysis

## Proof.

- $\exists\ j$ such that $\mathcal{C}(L_j, OPT) \leq \frac{1}{k} \cdot \mathcal{C}(V, OPT)$.
- Then, $\mathcal{C}(V \setminus L_j, OPT) \geq (1 - \frac{1}{k}) \cdot \mathcal{C}(V, OPT)$.
- Let $G_j$ be the graph corresponding to $j$.
- Since $\mathcal{C}(V, S) \geq \mathcal{C}(V, S_j)$.

$$\mathcal{C}(V, S)$$

# Approximation Analysis

## Proof.

- $\exists\ j$ such that $\mathcal{C}(L_j, OPT) \leq \frac{1}{k} \cdot \mathcal{C}(V, OPT)$.
- Then, $\mathcal{C}(V \setminus L_j, OPT) \geq (1 - \frac{1}{k}) \cdot \mathcal{C}(V, OPT)$.
- Let $G_j$ be the graph corresponding to $j$.
- Since $\mathcal{C}(V, S) \geq \mathcal{C}(V, S_j)$.

$$\mathcal{C}(V, S) \quad \geq \quad \mathcal{C}(V, S_j)$$

# Approximation Analysis

## Proof.

- $\exists\, j$ such that $\mathcal{C}(L_j, OPT) \leq \frac{1}{k} \cdot \mathcal{C}(V, OPT)$.
- Then, $\mathcal{C}(V \setminus L_j, OPT) \geq (1 - \frac{1}{k}) \cdot \mathcal{C}(V, OPT)$.
- Let $G_j$ be the graph corresponding to $j$.
- Since $\mathcal{C}(V, S) \geq \mathcal{C}(V, S_j)$.

$$\mathcal{C}(V, S) \quad \geq \quad \mathcal{C}(V, S_j) \geq \mathcal{C}(G_j, V, S_j)$$

# Approximation Analysis

Proof.

- $\exists\ j$ such that $\mathcal{C}(L_j, OPT) \le \frac{1}{k} \cdot \mathcal{C}(V, OPT)$.
- Then, $\mathcal{C}(V \setminus L_j, OPT) \ge (1 - \frac{1}{k}) \cdot \mathcal{C}(V, OPT)$.
- Let $G_j$ be the graph corresponding to $j$.
- Since $\mathcal{C}(V, S) \ge \mathcal{C}(V, S_j)$.

$$\mathcal{C}(V, S) \quad \ge \quad \mathcal{C}(V, S_j) \ge \mathcal{C}(G_j, V, S_j) \ge \mathcal{C}(G_j, V, OPT)$$

# Approximation Analysis

**Proof.**

- $\exists\, j$ such that $\mathcal{C}(L_j, OPT) \leq \frac{1}{k} \cdot \mathcal{C}(V, OPT)$.
- Then, $\mathcal{C}(V \setminus L_j, OPT) \geq (1 - \frac{1}{k}) \cdot \mathcal{C}(V, OPT)$.
- Let $G_j$ be the graph corresponding to $j$.
- Since $\mathcal{C}(V, S) \geq \mathcal{C}(V, S_j)$.

$$
\begin{aligned}
\mathcal{C}(V, S) \quad \geq \quad & \mathcal{C}(V, S_j) \geq \mathcal{C}(G_j, V, S_j) \geq \mathcal{C}(G_j, V, OPT) \\
& \geq \mathcal{C}(G_j, V \setminus L_j, OPT)
\end{aligned}
$$

# Approximation Analysis

## Proof.

- $\exists\ j$ such that $\mathcal{C}(L_j, OPT) \leq \frac{1}{k} \cdot \mathcal{C}(V, OPT)$.
- Then, $\mathcal{C}(V \setminus L_j, OPT) \geq (1 - \frac{1}{k}) \cdot \mathcal{C}(V, OPT)$.
- Let $G_j$ be the graph corresponding to $j$.
- Since $\mathcal{C}(V, S) \geq \mathcal{C}(V, S_j)$.

$$\begin{aligned}\mathcal{C}(V, S) \quad &\geq \quad \mathcal{C}(V, S_j) \geq \mathcal{C}(G_j, V, S_j) \geq \mathcal{C}(G_j, V, OPT) \\ &\geq \mathcal{C}(G_j, V \setminus L_j, OPT) = \mathcal{C}(G, V \setminus L_j, OPT)\end{aligned}$$

# Approximation Analysis

## Proof.

- $\exists\ j$ such that $\mathcal{C}(L_j, OPT) \leq \frac{1}{k} \cdot \mathcal{C}(V, OPT)$.
- Then, $\mathcal{C}(V \setminus L_j, OPT) \geq (1 - \frac{1}{k}) \cdot \mathcal{C}(V, OPT)$.
- Let $G_j$ be the graph corresponding to $j$.
- Since $\mathcal{C}(V, S) \geq \mathcal{C}(V, S_j)$.

$$\begin{aligned}
\mathcal{C}(V, S) \quad \geq \quad & \mathcal{C}(V, S_j) \geq \mathcal{C}(G_j, V, S_j) \geq \mathcal{C}(G_j, V, OPT) \\
& \geq \mathcal{C}(G_j, V \setminus L_j, OPT) = \mathcal{C}(G, V \setminus L_j, OPT) \\
& \leq (1 - \frac{1}{k}) \cdot \mathcal{C}(V, OPT)
\end{aligned}$$

□


THANK
YOU!

Hassin, R., Ravi, R., and Salman, F. S. (2009).
*Tractable Cases of Facility Location on a Network with a Linear Reliability Order of Links*, pages 275–276.
Springer Berlin Heidelberg, Berlin, Heidelberg.

Hassin, R., Ravi, R., and Salman, F. S. (2017).
Multiple facility location on a network with linear reliability order of edges.
*Journal of Combinatorial Optimization*, pages 1–25.

Shmoys, D. B. and Williamson, D. P. (2011).
*The Design of Approximation Algorithms*.
Cambridge University Press, New York, NY, USA, 1st edition.