Approximability of the Unique Coverage Problem

Mohammad R. Salavatipour
Department of Computing Science
University of Alberta

joint with

Erik Demaine
(MIT)

Uriel Feige
(MSR & Weizmann Inst)

MohammadTaghi Hajiaghayi
(MIT)

Unique Coverage Problem

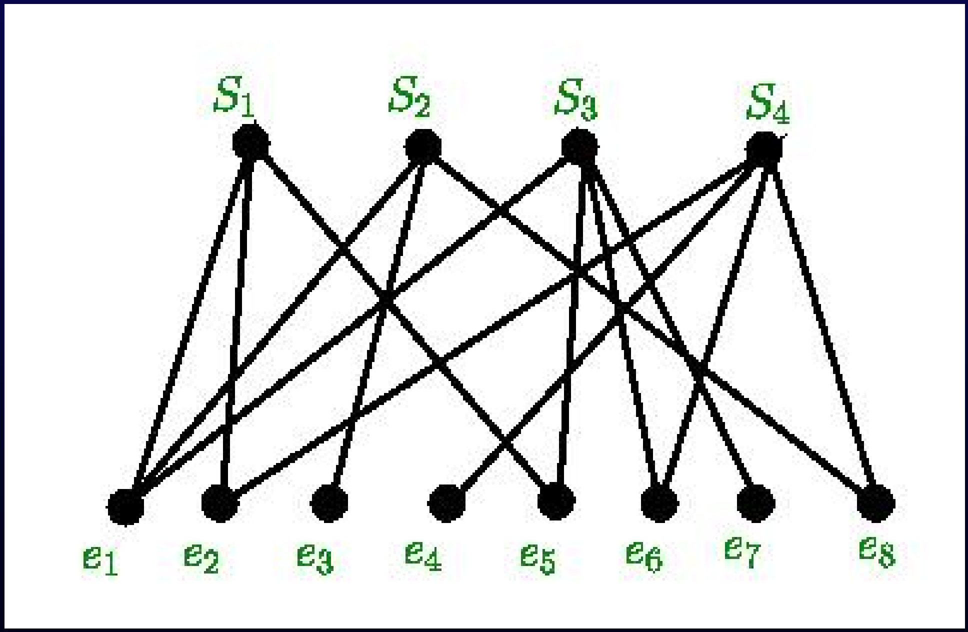
$U = \{e_1, \ldots, e_n\}$ universe of elements

$\mathcal{S} = \{S_1, \ldots, S_m\}$ a collection of subsets of U

Unique Coverage Problem

$U = \{e_1, \ldots, e_n\}$ universe of elements
$\mathcal{S} = \{S_1, \ldots, S_m\}$ a collection of subsets of U

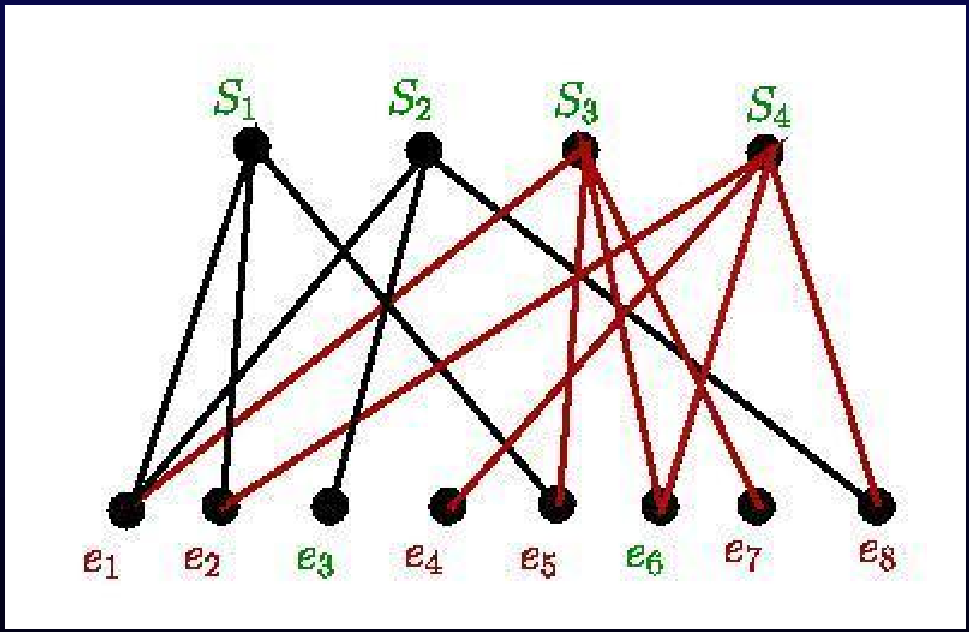
Goal: Find a subcollection $\mathcal{S}' \subseteq S$ to maximize number of elements "uniquely covered" (i.e. appear in exactly one set in $\mathcal{S}'$)



Unique Coverage Problem

$U = \{e_1, \ldots, e_n\}$ universe of elements
$\mathcal{S} = \{S_1, \ldots, S_m\}$ a collection of subsets of U

Goal: Find a subcollection $\mathcal{S}' \subseteq S$ to maximize number of elements "uniquely covered" (i.e. appear in exactly one set in $\mathcal{S}'$)

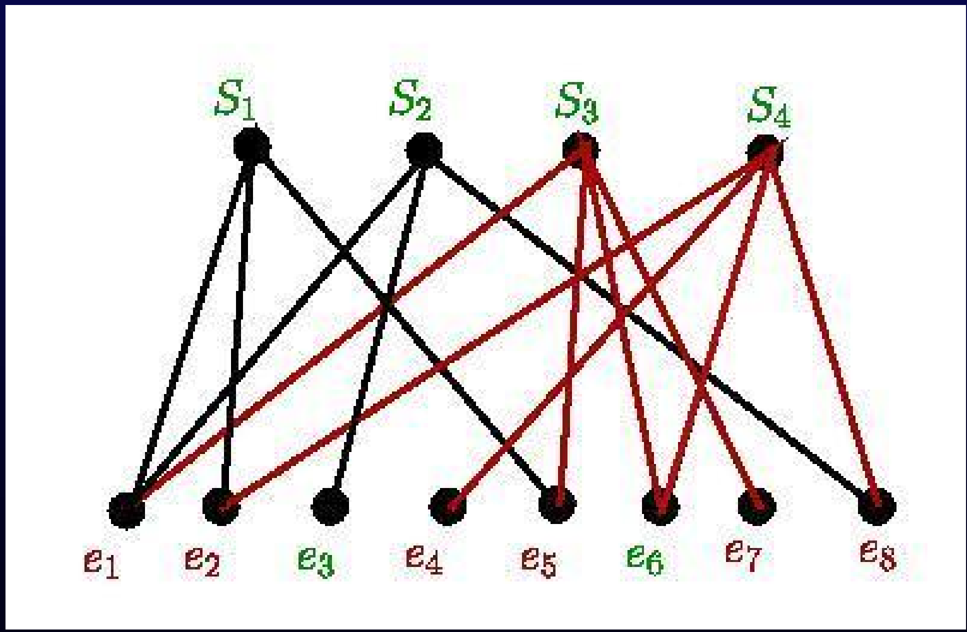


Unique Coverage Problem

$U = \{e_1, \ldots, e_n\}$ universe of elements
$\mathcal{S} = \{S_1, \ldots, S_m\}$ a collection of subsets of U

Goal: Find a subcollection $\mathcal{S}' \subseteq S$ to maximize number of elements "uniquely covered" (i.e. appear in exactly one set in $\mathcal{S}'$)

maximum unique coverage size = 6



Some applications of Unique Coverage

1) Wireless networks:

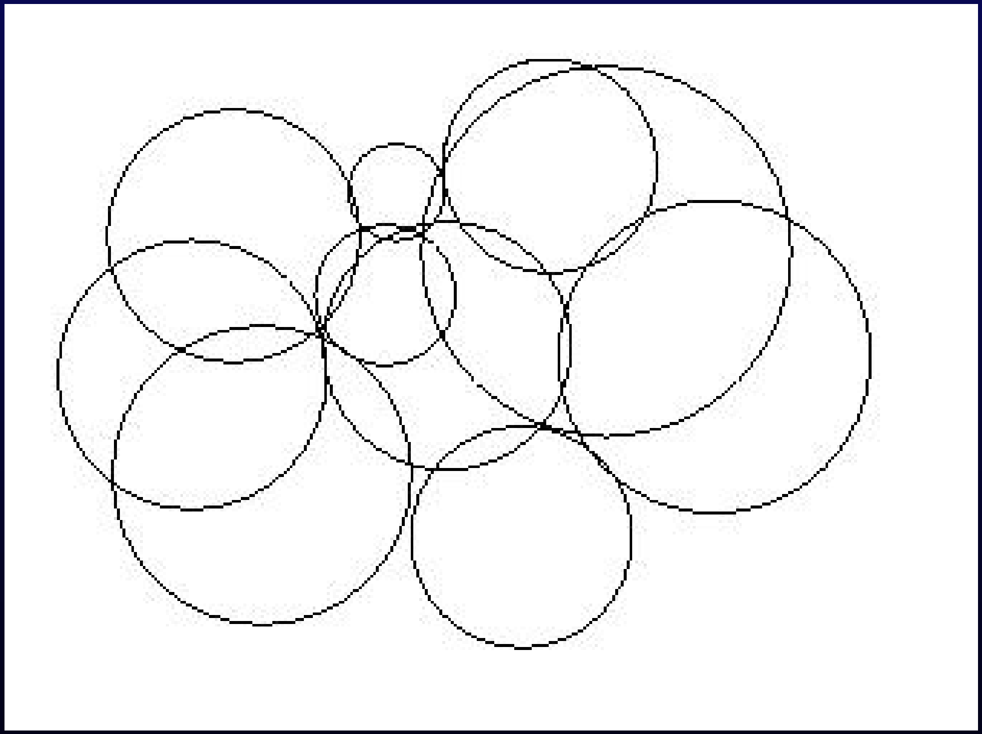
Given a region for mobile clients + collection of candidates for base stations

Some applications of Unique Coverage

1) Wireless networks:

Given a region for mobile clients + collection of candidates for base stations

Each base station will have a certain range of coverage



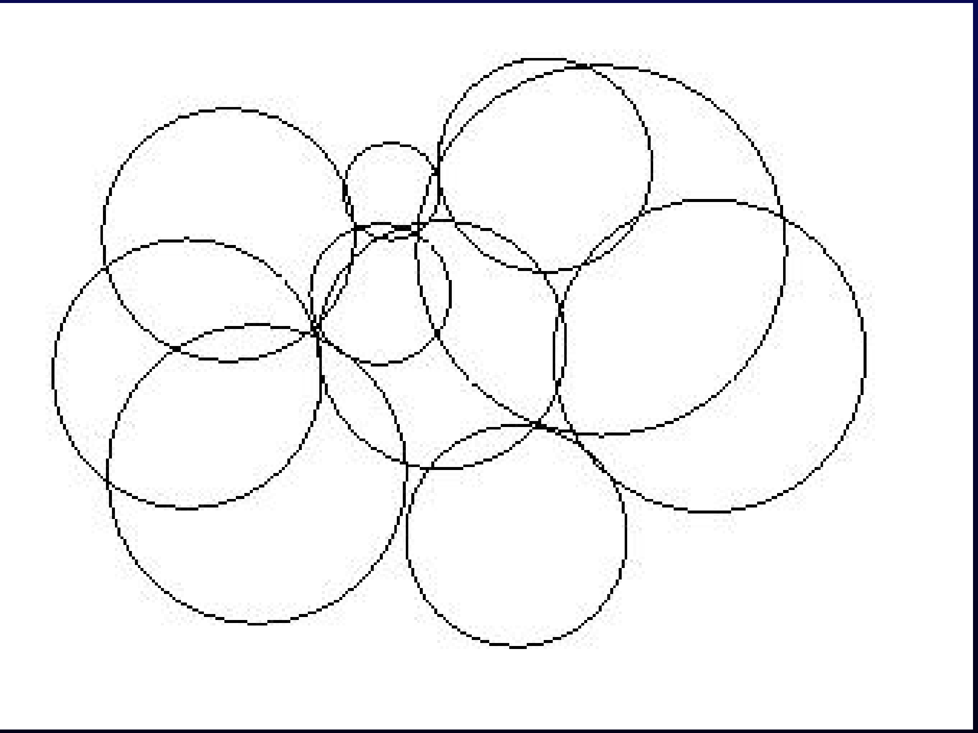
Some applications of Unique Coverage

1) Wireless networks:

Given a region for mobile clients + collection of candidates for base stations

Each base station will have a certain range of coverage

Choose a set of base stations subject to:



Some applications of Unique Coverage

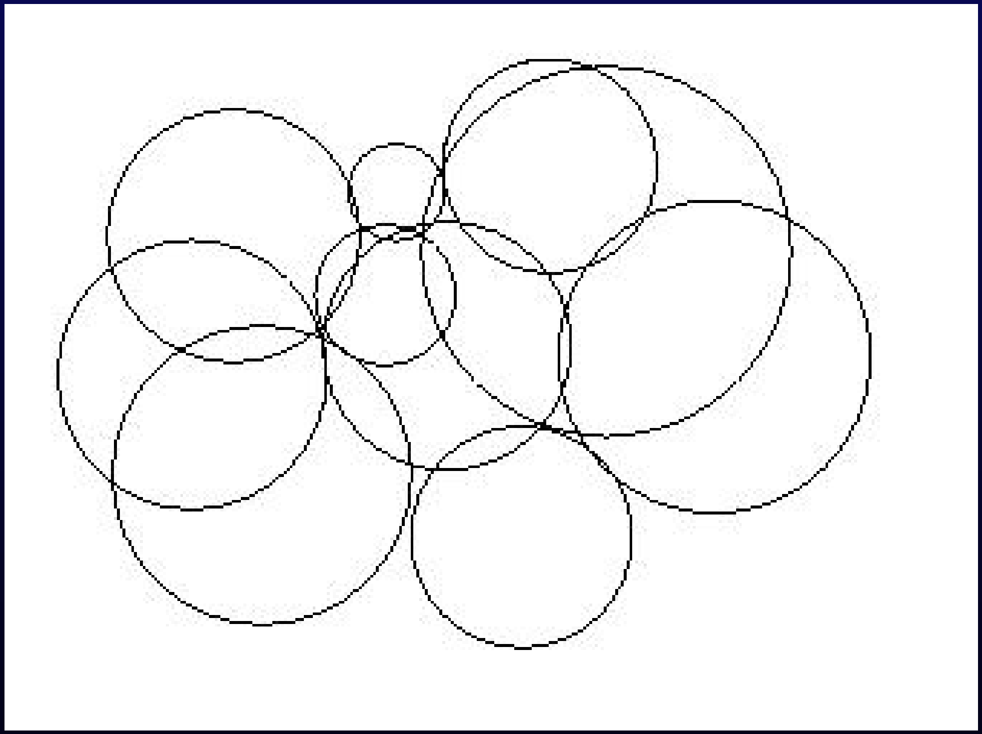
1) Wireless networks:

Given a region for mobile clients + collection of candidates for base stations

Each base station will have a certain range of coverage

Choose a set of base stations subject to:

A client is serviced if it is within range of $i > 0$ stations with small i, best case when $i = 1$;



Some applications of Unique Coverage

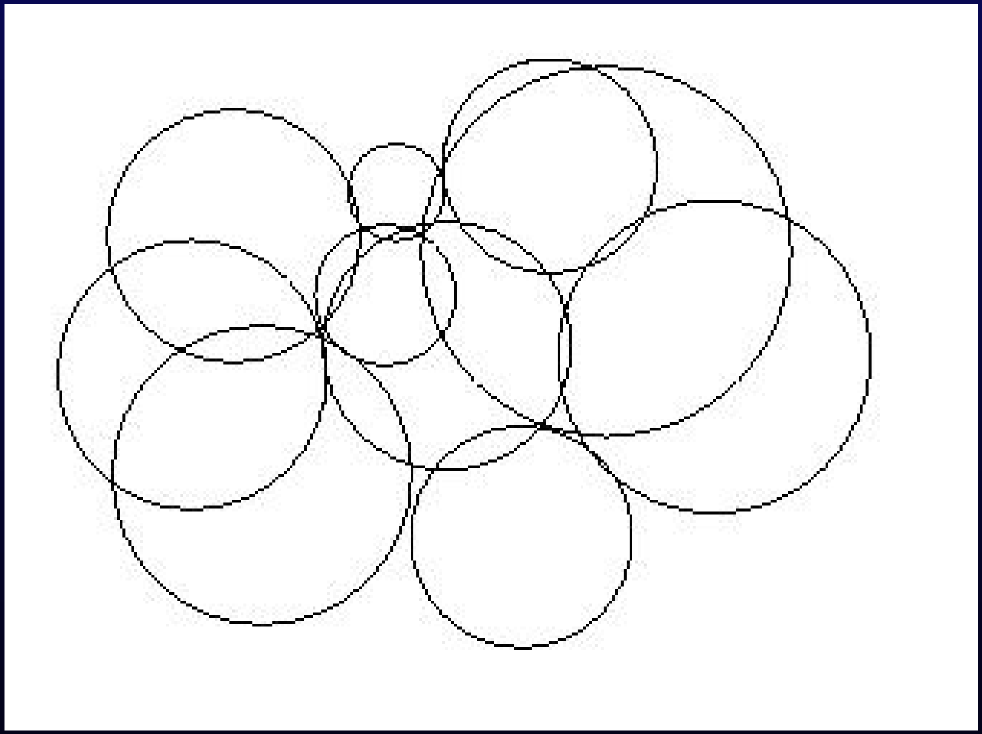
1) Wireless networks:

Given a region for mobile clients + collection of candidates for base stations

Each base station will have a certain range of coverage

Choose a set of base stations subject to:

A client is serviced if it is within range of $i > 0$ stations with small i, best case when $i = 1$; larger $i \longrightarrow$ signal interference



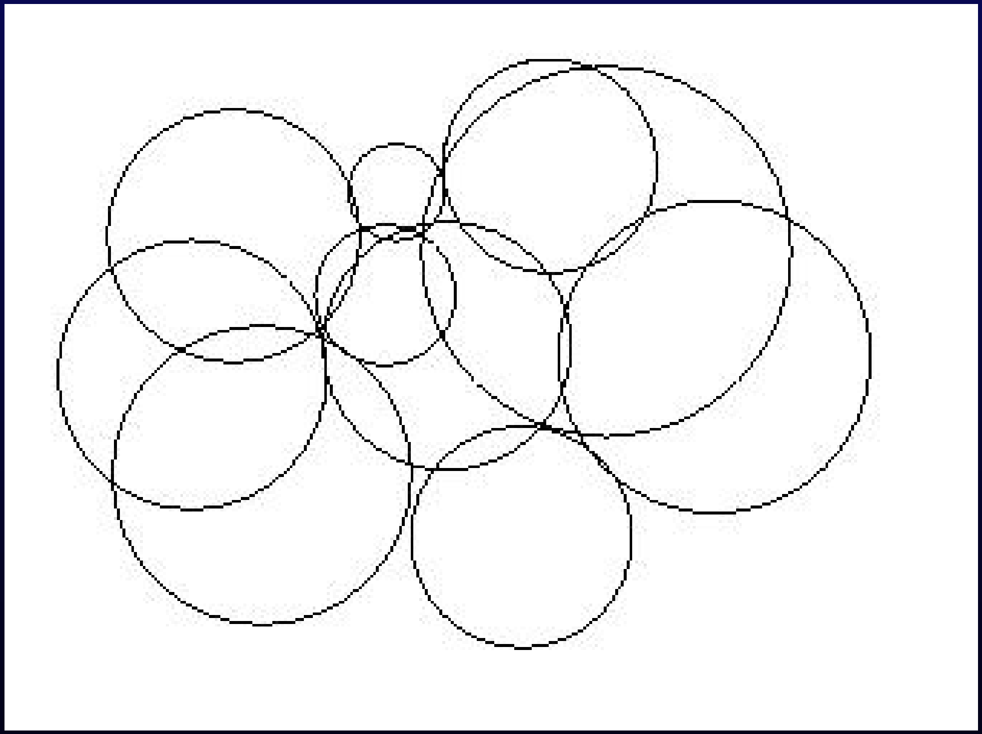
Some applications of Unique Coverage

1) Wireless networks:

Given a region for mobile clients + collection of candidates for base stations

Each base station will have a certain range of coverage

Choose a set of base stations subject to:



A client is serviced if it is within range of $i > 0$ stations with small i, best case when $i = 1$; larger $i \longrightarrow$ signal interference

Goal: maximize the region (number of clients) that are served by "a few" (≈ 1) stations

Some applications of Unique Coverage

1) Wireless networks:

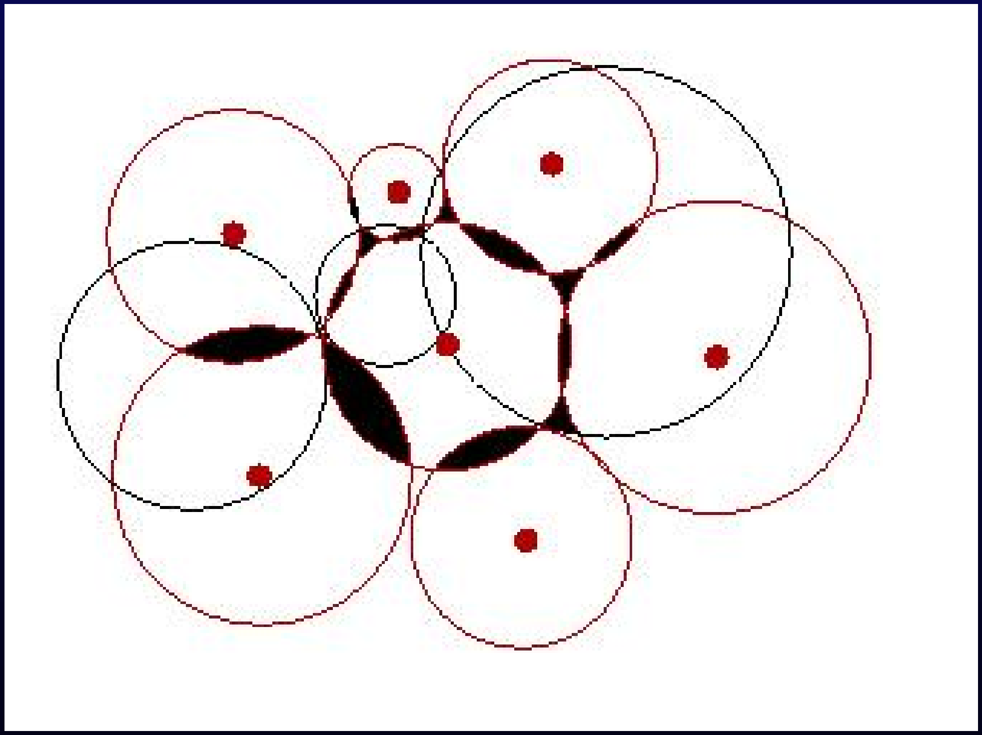
Given a region for mobile clients + collection of candidates for base stations

Each base station will have a certain range of coverage

Choose a set of base stations subject to:

A client is serviced if it is within range of $i > 0$ stations with small i, best case when $i = 1$; larger $i \longrightarrow$ signal interference

Goal: maximize the region (number of clients) that are served by "a few" (≈ 1) stations



General case:

- If a client is within range of i stations $\longrightarrow$ service qualities is s_i, with $s_0 = 0$, $s_1 \geq s_2 \geq \ldots$

General case:

- If a client is within range of i stations $\longrightarrow$ service qualities is s_i, with $s_0 = 0$, $s_1 \geq s_2 \geq \ldots$
- we assume $s_i \neq 0$ only for small values of i

General case:

- If a client is within range of i stations $\longrightarrow$ service qualities is s_i, with $s_0 = 0$, $s_1 \geq s_2 \geq \ldots$
- we assume $s_i \neq 0$ only for small values of i
- A cost $C_i \geq 0$ for building station i

General case:

- If a client is within range of i stations $\longrightarrow$ service qualities is s_i, with $s_0 = 0$, $s_1 \geq s_2 \geq \ldots$
- we assume $s_i \neq 0$ only for small values of i
- A cost $C_i \geq 0$ for building station i
- A total budget B to build stations

General case:

- If a client is within range of i stations $\longrightarrow$ service qualities is s_i, with $s_0 = 0$, $s_1 \geq s_2 \geq \ldots$
- we assume $s_i \neq 0$ only for small values of i
- A cost $C_i \geq 0$ for building station i
- A total budget B to build stations
- maximize total (average) service quality with budget B

2) Envy-free pricing

- There is a single seller to price m different products/items $p_1, \ldots, p_m$

2) Envy-free pricing

- There is a single seller to price m different products/items $p_1, \ldots, p_m$
- Each product/item has a quantity

2) Envy-free pricing

- There is a single seller to price m different products/items $p_1, \ldots, p_m$
- Each product/item has a quantity
- There are n buyers, each wants to by a subset of products (*bundle*)

2) Envy-free pricing

- There is a single seller to price m different products/items $p_1, \ldots, p_m$
- Each product/item has a quantity
- There are n buyers, each wants to by a subset of products (*bundle*)
- Seller knows (more or less) the price each buyer is willing to pay for each bundle (*valuation* of buyer for that bundle)

2) Envy-free pricing

- There is a single seller to price m different products/items $p_1, \ldots, p_m$
- Each product/item has a quantity
- There are n buyers, each wants to by a subset of products (*bundle*)
- Seller knows (more or less) the price each buyer is willing to pay for each bundle (*valuation* of buyer for that bundle)
- Buyer's utility = valuation - price paid (how much you save!)

2) Envy-free pricing

- There is a single seller to price m different products/items $p_1, \ldots, p_m$
- Each product/item has a quantity
- There are n buyers, each wants to by a subset of products (*bundle*)
- Seller knows (more or less) the price each buyer is willing to pay for each bundle (*valuation* of buyer for that bundle)
- Buyer's utility = valuation - price paid (how much you save!)

Goal: set the prices and decide which bundles to be sold s.t. it is *envy-free:* each buyer is sold a bundle with maximum utility for him/her. We want to maximize seller's profit.

2) Envy-free pricing

- There is a single seller to price m different products/items $p_1, \ldots, p_m$
- Each product/item has a quantity
- There are n buyers, each wants to by a subset of products (*bundle*)
- Seller knows (more or less) the price each buyer is willing to pay for each bundle (*valuation* of buyer for that bundle)
- Buyer's utility = valuation - price paid (how much you save!)

Goal: set the prices and decide which bundles to be sold s.t. it is *envy-free:* each buyer is sold a bundle with maximum utility for him/her. We want to maximize seller's profit.

special case: unlimited-supply single-minded buyer
Each buyer considers only one particular bundle; buys it if the cost $\leq$ valuation

2) Envy-free pricing

- There is a single seller to price m different products/items $p_1, \ldots, p_m$
- Each product/item has a quantity
- There are n buyers, each wants to by a subset of products (*bundle*)
- Seller knows (more or less) the price each buyer is willing to pay for each bundle (*valuation* of buyer for that bundle)
- Buyer's utility = valuation - price paid (how much you save!)

Goal: set the prices and decide which bundles to be sold s.t. it is *envy-free:* each buyer is sold a bundle with maximum utility for him/her. We want to maximize seller's profit.

special case: unlimited-supply single-minded buyer
Each buyer considers only one particular bundle; buys it if the cost $\leq$ valuation

This case is APX-hard and has an $O(\log n + \log m)$-approx [GHKKKM'05]

Proposition: U.C. is a special case of envy-free pricing.

Proposition: U.C. is a special case of envy-free pricing.

proof:

- We reduce U.C. to unlimited-supply single-minded (envy-free) pricing

Proposition: U.C. is a special case of envy-free pricing.

proof:

- We reduce U.C. to unlimited-supply single-minded (envy-free) pricing
- For each set S_i we have a product p_i

Proposition: U.C. is a special case of envy-free pricing.

proof:

- We reduce U.C. to unlimited-supply single-minded (envy-free) pricing
- For each set S_i we have a product p_i
- For each element $e_i \in U$ there is a buyer b_i

Proposition: U.C. is a special case of envy-free pricing.

proof:

- We reduce U.C. to unlimited-supply single-minded (envy-free) pricing
- For each set S_i we have a product p_i
- For each element $e_i \in U$ there is a buyer b_i
- Buyer b_i has valuation one for one bundle: $\{p_j | e_i \in S_j\}$

Proposition: U.C. is a special case of envy-free pricing.

proof:

- We reduce U.C. to unlimited-supply single-minded (envy-free) pricing
- For each set S_i we have a product p_i
- For each element $e_i \in U$ there is a buyer b_i
- Buyer b_i has valuation one for one bundle: $\{p_j | e_i \in S_j\}$
- We can assume all prices are either 0 or 1 (at a loss of constant factor in profit)

Proposition: U.C. is a special case of envy-free pricing.

proof:

- We reduce U.C. to unlimited-supply single-minded (envy-free) pricing
- For each set S_i we have a product p_i
- For each element $e_i \in U$ there is a buyer b_i
- Buyer b_i has valuation one for one bundle: $\{p_j | e_i \in S_j\}$
- We can assume all prices are either 0 or 1 (at a loss of constant factor in profit)
- Each buyer will buy his/her bundle if at most one item is priced at 1

Proposition: U.C. is a special case of envy-free pricing.

proof:

- We reduce U.C. to unlimited-supply single-minded (envy-free) pricing
- For each set S_i we have a product p_i
- For each element $e_i \in U$ there is a buyer b_i
- Buyer b_i has valuation one for one bundle: $\{p_j | e_i \in S_j\}$
- We can assume all prices are either 0 or 1 (at a loss of constant factor in profit)
- Each buyer will buy his/her bundle if at most one item is priced at 1
- We have to assign $0/1$ to items (i.e. sets) to maximize the number of bundles (i.e. elements) with exactly one item priced 1.

Proposition: U.C. is a special case of envy-free pricing.

proof:

- We reduce U.C. to unlimited-supply single-minded (envy-free) pricing
- For each set S_i we have a product p_i
- For each element $e_i \in U$ there is a buyer b_i
- Buyer b_i has valuation one for one bundle: $\{p_j | e_i \in S_j\}$
- We can assume all prices are either 0 or 1 (at a loss of constant factor in profit)
- Each buyer will buy his/her bundle if at most one item is priced at 1
- We have to assign $0/1$ to items (i.e. sets) to maximize the number of bundles (i.e. elements) with exactly one item priced 1.
- The profit is exactly the number of bundles sold at price 1

Proposition: U.C. is a special case of envy-free pricing.

proof:

- We reduce U.C. to unlimited-supply single-minded (envy-free) pricing
- For each set S_i we have a product p_i
- For each element $e_i \in U$ there is a buyer b_i
- Buyer b_i has valuation one for one bundle: $\{p_j | e_i \in S_j\}$
- We can assume all prices are either 0 or 1 (at a loss of constant factor in profit)
- Each buyer will buy his/her bundle if at most one item is priced at 1
- We have to assign $0/1$ to items (i.e. sets) to maximize the number of bundles (i.e. elements) with exactly one item priced 1.
- The profit is exactly the number of bundles sold at price 1

Corollary: our hardness results for U.C. imply the same hardness of approximation for envy-free pricing.

3) Max-cut:

- Given $G(V, E)$, find a cut $S, \overline{S}$, $S \subseteq V$, $\overline{S} = V - S$ with maximum number of edges

3) Max-cut:

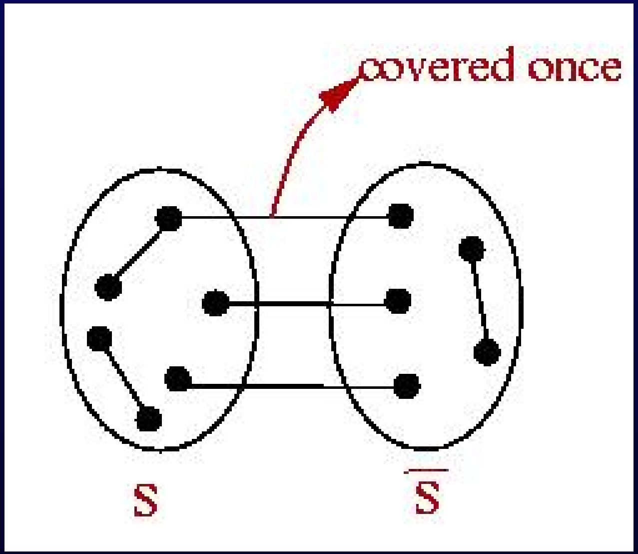
- Given $G(V, E)$, find a cut $S, \overline{S}$, $S \subseteq V$, $\overline{S} = V - S$ with maximum number of edges

- Max-cut is a special case of U.C.:
  sets in U.C. $\longleftrightarrow$ neighborhoods of vertices;
  elements $\longleftrightarrow$ edges

3) Max-cut:

- Given $G(V, E)$, find a cut $S, \overline{S}$, $S \subseteq V$, $\overline{S} = V - S$ with maximum number of edges

- Max-cut is a special case of U.C.:
  sets in U.C. $\longleftrightarrow$ neighborhoods of vertices;
  elements $\longleftrightarrow$ edges

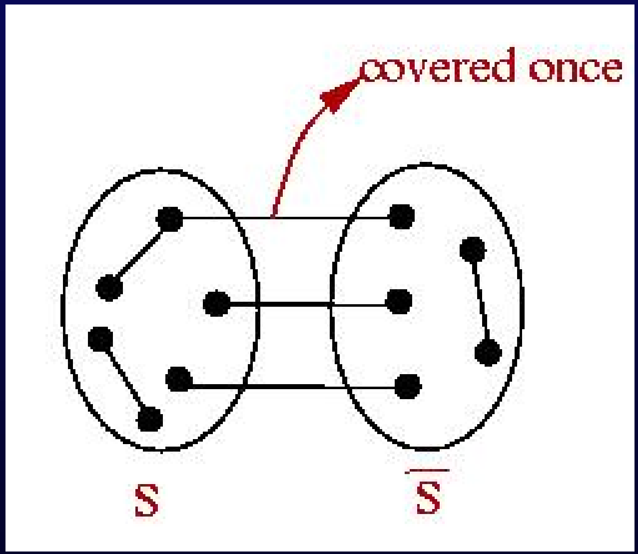


3) Max-cut:

- Given $G(V, E)$, find a cut $S, \overline{S}$, $S \subseteq V$, $\overline{S} = V - S$ with maximum number of edges

- Max-cut is a special case of U.C.:
  sets in U.C. $\longleftrightarrow$ neighborhoods of vertices;
  elements $\longleftrightarrow$ edges

- So U.C. is at least as hard as Max-cut.



Related problems/results

Set Cover: universe $U = \{e_1, \ldots, e_n\}$ and subsets $\mathcal{S} = \{S_1, \ldots, S_m\}$, $S_i \subseteq U$

Goal: Find minimum size $\mathcal{S}' \subseteq \mathcal{S}$ that covers U

Related problems/results

Set Cover: universe $U = \{e_1, \ldots, e_n\}$ and subsets $\mathcal{S} = \{S_1, \ldots, S_m\}$, $S_i \subseteq U$

Goal: Find minimum size $\mathcal{S}' \subseteq S$ that covers U

upper bound: simple greedy $\longrightarrow \ln n$-approx

Related problems/results

Set Cover: universe $U = \{e_1, \ldots, e_n\}$ and subsets $\mathcal{S} = \{S_1, \ldots, S_m\}$, $S_i \subseteq U$

Goal: Find minimum size $\mathcal{S}' \subseteq S$ that covers U

upper bound: simple greedy $\longrightarrow \ln n$-approx

Hardness: no $(1 - \epsilon) \ln n$-approx unless $NP \subseteq DTIME(n^{O(\lg \lg n)})$ [Feige'98]

Related problems/results

Set Cover: universe $U = \{e_1, \ldots, e_n\}$ and subsets $\mathcal{S} = \{S_1, \ldots, S_m\}$, $S_i \subseteq U$

Goal: Find minimum size $\mathcal{S}' \subseteq S$ that covers U

upper bound: simple greedy $\longrightarrow \ln n$-approx

Hardness: no $(1 - \epsilon) \ln n$-approx unless $NP \subseteq DTIME(n^{O(\lg \lg n)})$ [Feige'98]

Maximum Coverage (budgeted): the input is the same as in set cover + each subset S_i has a cost and each element a weight; a total “budget” is also given

Related problems/results

Set Cover: universe $U = \{e_1, \ldots, e_n\}$ and subsets $\mathcal{S} = \{S_1, \ldots, S_m\}$, $S_i \subseteq U$

Goal: Find minimum size $\mathcal{S}' \subseteq S$ that covers U

upper bound: simple greedy $\longrightarrow \ln n$-approx

Hardness: no $(1-\epsilon)\ln n$-approx unless $NP \subseteq DTIME(n^{O(\lg\lg n)})$ [Feige'98]

Maximum Coverage (budgeted): the input is the same as in set cover + each subset S_i has a cost and each element a weight; a total "budget" is also given

Find a collection $\mathcal{S}' \subseteq S$, with total cost $\leq B$ with max weight of covered elements

Related problems/results

Set Cover: universe $U = \{e_1, \ldots, e_n\}$ and subsets $\mathcal{S} = \{S_1, \ldots, S_m\}$, $S_i \subseteq U$

Goal: Find minimum size $\mathcal{S}' \subseteq S$ that covers U

upper bound: simple greedy $\longrightarrow \ln n$-approx

Hardness: no $(1-\epsilon)\ln n$-approx unless $NP \subseteq DTIME(n^{O(\lg \lg n)})$ [Feige'98]

Maximum Coverage (budgeted): the input is the same as in set cover + each subset S_i has a cost and each element a weight; a total "budget" is also given

Find a collection $\mathcal{S}' \subseteq S$, with total cost $\leq B$ with max weight of covered elements

upper bound: it has a simple greedy $(1 - \frac{1}{e})$-approx

Related problems/results

Set Cover: universe $U = \{e_1, \ldots, e_n\}$ and subsets $\mathcal{S} = \{S_1, \ldots, S_m\}$, $S_i \subseteq U$

Goal: Find minimum size $\mathcal{S}' \subseteq S$ that covers U

upper bound: simple greedy $\longrightarrow \ln n$-approx

Hardness: no $(1-\epsilon)\ln n$-approx unless $NP \subseteq DTIME(n^{O(\lg\lg n)})$ [Feige'98]

Maximum Coverage (budgeted): the input is the same as in set cover + each subset S_i has a cost and each element a weight; a total "budget" is also given

Find a collection $\mathcal{S}' \subseteq S$, with total cost $\leq B$ with max weight of covered elements

upper bound: it has a simple greedy $(1-\frac{1}{e})$-approx

Hardness: no $(1-\frac{1}{e}-\epsilon)$-approx unless $NP \subseteq DTIME(n^{O(\lg\lg n)})$

Related problems/results

Set Cover: universe $U = \{e_1, \ldots, e_n\}$ and subsets $\mathcal{S} = \{S_1, \ldots, S_m\}$, $S_i \subseteq U$

Goal: Find minimum size $\mathcal{S}' \subseteq S$ that covers U

upper bound: simple greedy $\longrightarrow \ln n$-approx

Hardness: no $(1-\epsilon)\ln n$-approx unless $NP \subseteq DTIME(n^{O(\lg \lg n)})$ [Feige'98]

Maximum Coverage (budgeted): the input is the same as in set cover + each subset S_i has a cost and each element a weight; a total "budget" is also given

Find a collection $\mathcal{S}' \subseteq S$, with total cost $\leq B$ with max weight of covered elements

upper bound: it has a simple greedy $(1-\frac{1}{e})$-approx

Hardness: no $(1-\frac{1}{e}-\epsilon)$-approx unless $NP \subseteq DTIME(n^{O(\lg \lg n)})$

Unlike set cover and maximum coverage, the greedy doesn't seem to work for U.C. (all immediate algorithms have ratio $\Omega(n)$).

Related problems/results (cont'd)

Radio Broadcast: A classical problem, studied extensively,

- A network of processors that communicate in rounds

Related problems/results (cont'd)

Radio Broadcast: A classical problem, studied extensively,

- A network of processors that communicate in rounds
- initially one node has a message;

Related problems/results (cont'd)

Radio Broadcast: A classical problem, studied extensively,

- A network of processors that communicate in rounds
- initially one node has a message;
- at each round every node that has already received the message can either send it to all its neighbors or do nothing

Related problems/results (cont'd)

Radio Broadcast: A classical problem, studied extensively,

- A network of processors that communicate in rounds
- initially one node has a message;
- at each round every node that has already received the message can either send it to all its neighbors or do nothing
- A node receives a message (in a round) if exactly one of its neighbors transmits in that round

Related problems/results (cont'd)

Radio Broadcast: A classical problem, studied extensively,

- A network of processors that communicate in rounds
- initially one node has a message;
- at each round every node that has already received the message can either send it to all its neighbors or do nothing
- A node receives a message (in a round) if exactly one of its neighbors transmits in that round

Goal: Propagate the message to all the nodes in minimum number of rounds

Related problems/results (cont'd)

Radio Broadcast: A classical problem, studied extensively,

- A network of processors that communicate in rounds
- initially one node has a message;
- at each round every node that has already received the message can either send it to all its neighbors or do nothing
- A node receives a message (in a round) if exactly one of its neighbors transmits in that round

Goal: Propagate the message to all the nodes in minimum number of rounds

The U.C. can be seen as a single round of a greedy alg. for radio broadcast

Related problems/results (cont'd)

Radio Broadcast: A classical problem, studied extensively,

- A network of processors that communicate in rounds
- initially one node has a message;
- at each round every node that has already received the message can either send it to all its neighbors or do nothing
- A node receives a message (in a round) if exactly one of its neighbors transmits in that round

Goal: Propagate the message to all the nodes in minimum number of rounds

The U.C. can be seen as a single round of a greedy alg. for radio broadcast

upper bound: can do in $D + O(\log^2 n)$ rounds (randomly) and $D + O(\log^3 n)$ rounds deterministically, where D is the diameter of network [GPX'05]

Related problems/results (cont'd)

Radio Broadcast: A classical problem, studied extensively,

- A network of processors that communicate in rounds
- initially one node has a message;
- at each round every node that has already received the message can either send it to all its neighbors or do nothing
- A node receives a message (in a round) if exactly one of its neighbors transmits in that round

Goal: Propagate the message to all the nodes in minimum number of rounds

The U.C. can be seen as a single round of a greedy alg. for radio broadcast

upper bound: can do in $D + O(\log^2 n)$ rounds (randomly) and $D + O(\log^3 n)$ rounds deterministically, where D is the diameter of network [GPX'05]

Hardness: Elkin and Kortsarz prove (multiplicative) $\Omega(\log n)$ and additive $\Omega(\log^2 n)$ hardness, assuming $NP \not\subseteq DTIME(n^{O(\lg \lg n)})$

Our results

Theorem 1: There is an $O(\log n)$-approx for U.C. even for the more general case with budget, costs on the sets, and weights on the elements.

Our results

Theorem 1: There is an $O(\log n)$-approx for U.C. even for the more general case with budget, costs on the sets, and weights on the elements.

Theorem 2: Assuming that $NP \not\subseteq BTIME(2^{n^\epsilon})$, U.C. is $\Omega(\log^{\sigma(\epsilon)} n)$-hard, for some $\sigma(\epsilon) > 0$.

Our reduction for Theorem 2 also shows:

Theorem 3: U.C. is $\Omega(1/\log^{1/3-\epsilon} n)$-hard for any $\epsilon > 0$, assuming that refuting random instances of 3SAT is hard on average.

Our results

Theorem 1: There is an $O(\log n)$-approx for U.C. even for the more general case with budget, costs on the sets, and weights on the elements.

Theorem 2: Assuming that $NP \not\subseteq BTIME(2^{n^{\epsilon}})$, U.C. is $\Omega(\log^{\sigma(\epsilon)} n)$-hard, for some $\sigma(\epsilon) > 0$.

Our reduction for Theorem 2 also shows:

Theorem 3: U.C. is $\Omega(1/\log^{1/3-\epsilon} n)$-hard for any $\epsilon > 0$, assuming that refuting random instances of 3SAT is hard on average.

Remark: the known algorithms for Radio broadcast implicitly imply an $O(\log n)$-approx for the (simple) U.C.

Approximation Algorithm

- Consider any minimal set cover solution $\mathcal{S}' \subseteq \mathcal{S}$

Approximation Algorithm

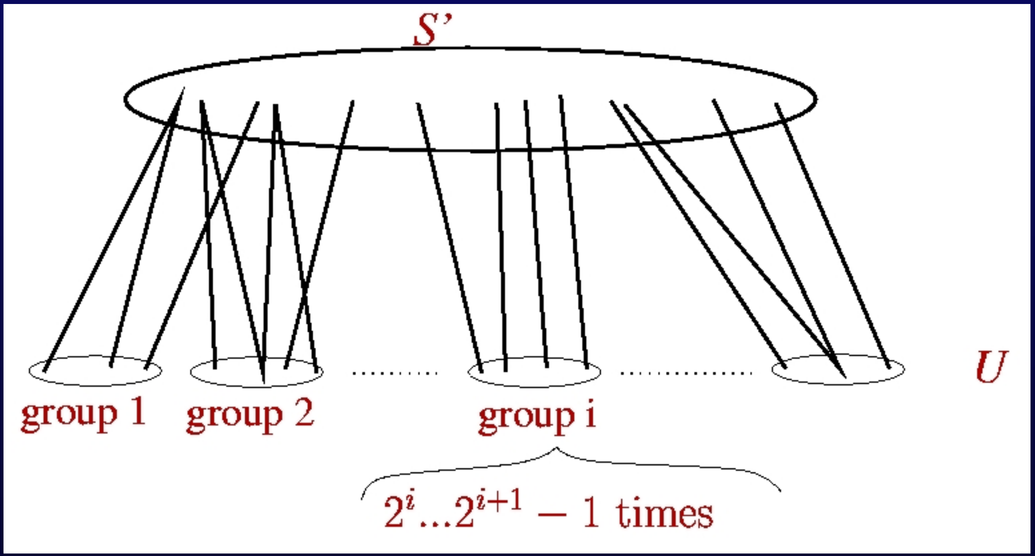
- Consider any minimal set cover solution $\mathcal{S}' \subseteq \mathcal{S}$
- Cluster elements into $\log m$ groups:

Approximation Algorithm

- Consider any minimal set cover solution $\mathcal{S}' \subseteq \mathcal{S}$
- Cluster elements into $\log m$ groups:
- element e is in group $i \longleftrightarrow$ e is covered between 2^i and $2^{i+1} - 1$ times

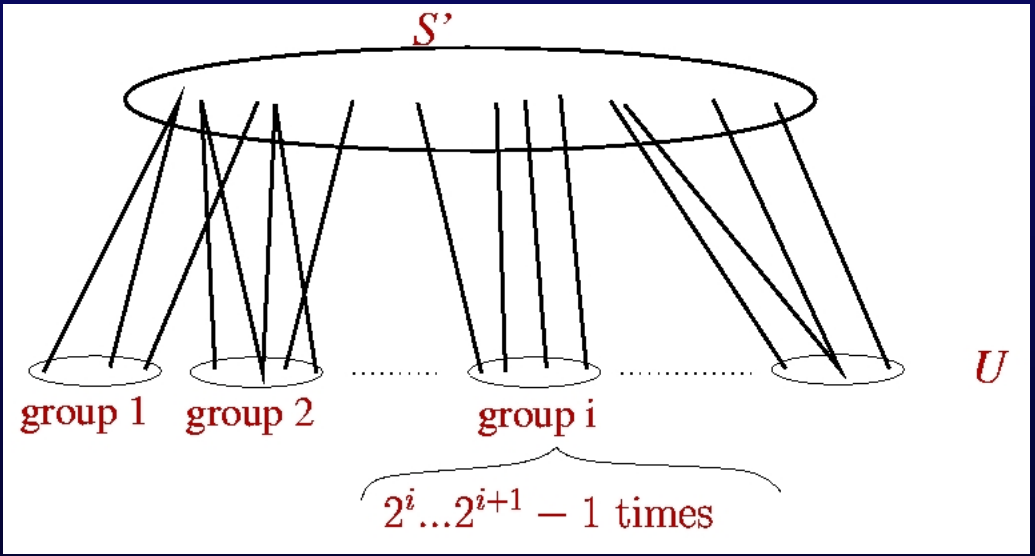
Approximation Algorithm

- Consider any minimal set cover solution $\mathcal{S}' \subseteq \mathcal{S}$
- Cluster elements into $\log m$ groups:
- element e is in group $i \longleftrightarrow$ e is covered between 2^i and $2^{i+1} - 1$ times



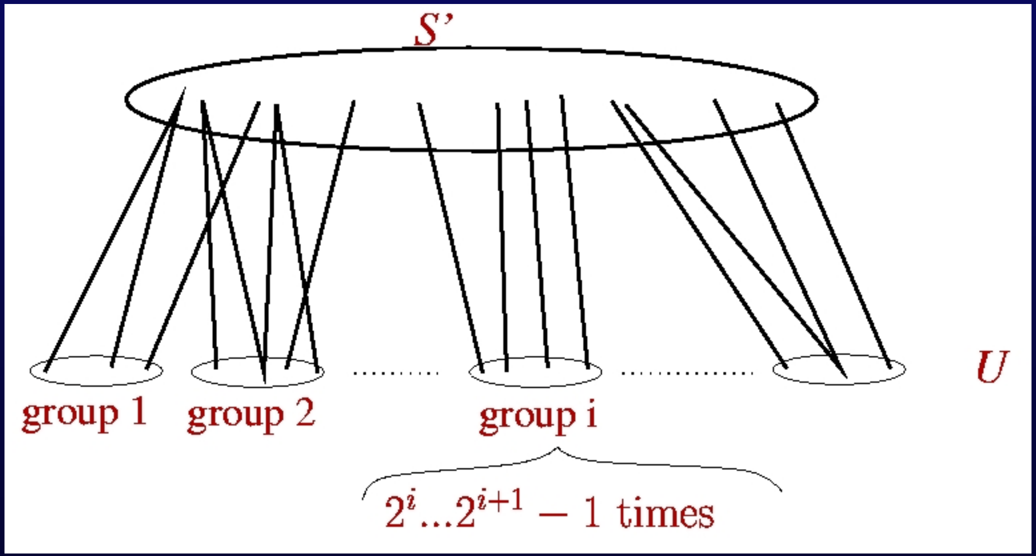
Approximation Algorithm

- Consider any minimal set cover solution $\mathcal{S}' \subseteq \mathcal{S}$
- Cluster elements into $\log m$ groups:
- element e is in group $i \longleftrightarrow$ e is covered between 2^i and $2^{i+1} - 1$ times
- let i be the group with max number of elements



Approximation Algorithm

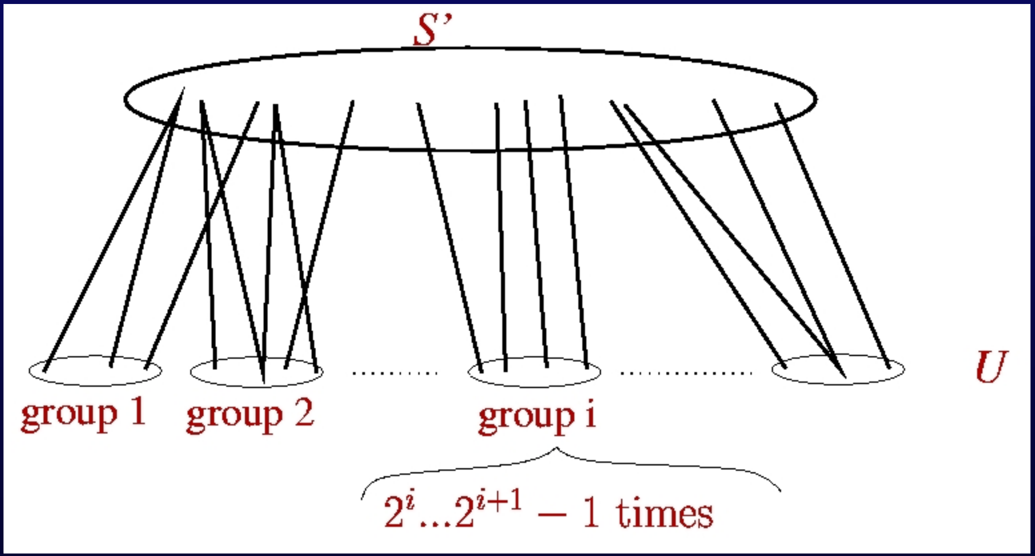
- Consider any minimal set cover solution $\mathcal{S}' \subseteq \mathcal{S}$
- Cluster elements into $\log m$ groups:
- element e is in group $i \longleftrightarrow$ e is covered between 2^i and $2^{i+1} - 1$ times
- let i be the group with max number of elements



- Delete sets from $\mathcal{S}'$ randomly, keep each with prob $\frac{1}{2^i}$

Approximation Algorithm

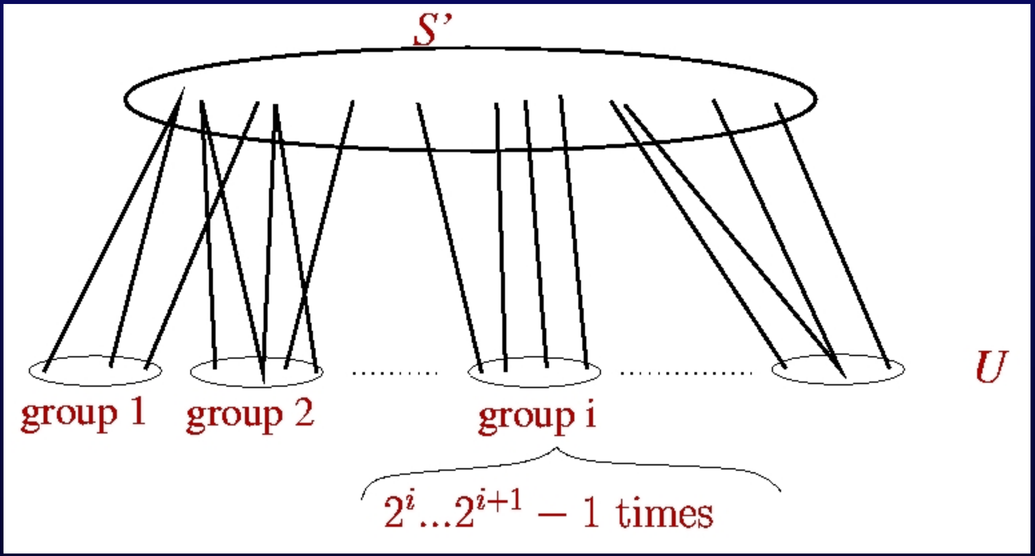
- Consider any minimal set cover solution $\mathcal{S}' \subseteq \mathcal{S}$
- Cluster elements into $\log m$ groups:
- element e is in group $i \longleftrightarrow$ e is covered between 2^i and $2^{i+1} - 1$ times
- let i be the group with max number of elements



- Delete sets from $\mathcal{S}'$ randomly, keep each with prob $\frac{1}{2^i}$
- we expect a constant fraction of the elements in group i be u.c.

Approximation Algorithm

- Consider any minimal set cover solution $\mathcal{S}' \subseteq \mathcal{S}$
- Cluster elements into $\log m$ groups:



- element e is in group $i \longleftrightarrow$ e is covered between 2^i and $2^{i+1}-1$ times
- let i be the group with max number of elements
- Delete sets from $\mathcal{S}'$ randomly, keep each with prob $\frac{1}{2^i}$
- we expect a constant fraction of the elements in group i be u.c.
- So the expected number of elements u.c. is at least $\frac{1}{e^2} \times$ [size of group i] $\in \Omega(\frac{n}{\log n})$

Hardness of Unique Coverage

- Let $G(A \cup B, E)$ be a bip graph with $|A| = |B| = n$

Hardness of Unique Coverage

- Let $G(A \cup B, E)$ be a bip graph with $|A| = |B| = n$
- Finding largest independent set in G is easy

Hardness of Unique Coverage

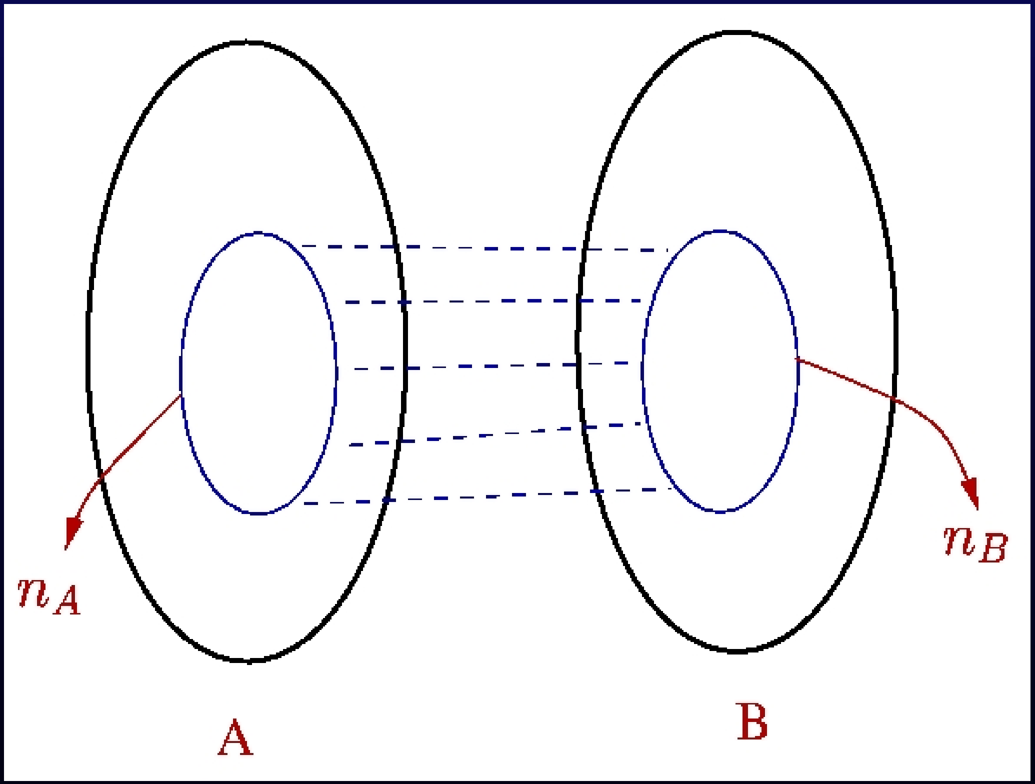
- Let $G(A \cup B, E)$ be a bip graph with $|A| = |B| = n$
- Finding largest independent set in G is easy
- Suppose we are also given n_A and n_B

Hardness of Unique Coverage

- Let $G(A \cup B, E)$ be a bip graph with $|A| = |B| = n$
- Finding largest independent set in G is easy
- Suppose we are also given n_A and n_B
- Want to see if G has an ind. set I with $|A \cap I| \geq n_A$, $|B \cap I| \geq n_B$

Hardness of Unique Coverage

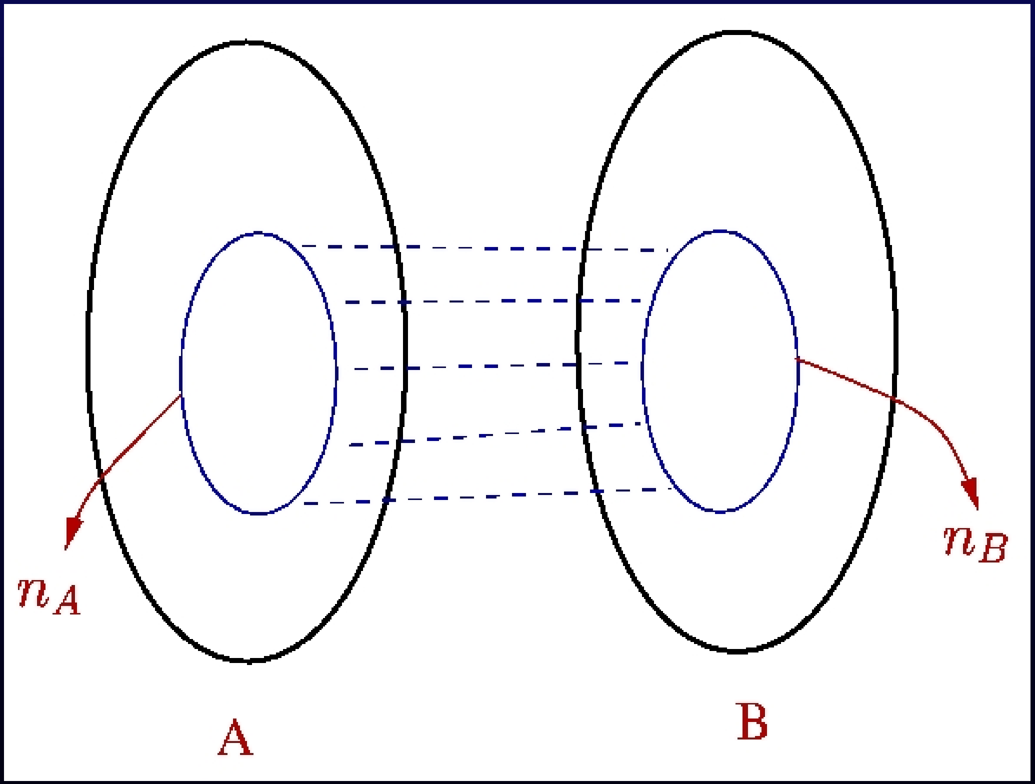
- Let $G(A \cup B, E)$ be a bip graph with $|A| = |B| = n$
- Finding largest independent set in G is easy



- Suppose we are also given n_A and n_B
- Want to see if G has an ind. set I with $|A \cap I| \geq n_A$, $|B \cap I| \geq n_B$

Hardness of Unique Coverage

- Let $G(A \cup B, E)$ be a bip graph with $|A| = |B| = n$
- Finding largest independent set in G is easy



- Suppose we are also given n_A and n_B
- Want to see if G has an ind. set I with $|A \cap I| \geq n_A$, $|B \cap I| \geq n_B$

This question, called *Balanced Bip. Ind. Set* turns out to be difficult.

Hardness result (cont'd)

Definition: Given bip graph $G(A \cup B, E)$ with $|A| = |B| = n$, $0 < \gamma' < \gamma \leq 1$, $0 \leq \delta < \delta' \leq 1$, $BBIS(\gamma, \gamma', \delta, \delta')$, is to decide between:

Hardness result (cont'd)

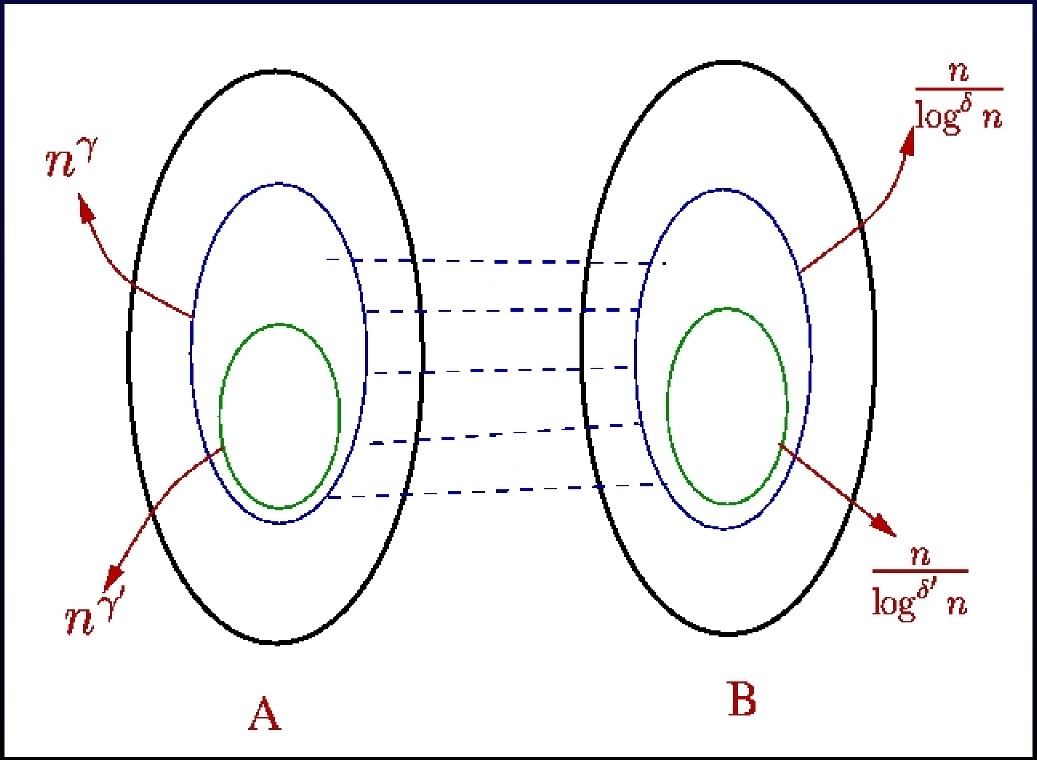
Definition: Given bip graph $G(A \cup B, E)$ with $|A| = |B| = n$, $0 < \gamma' < \gamma \leq 1$, $0 \leq \delta < \delta' \leq 1$, $BBIS(\gamma, \gamma', \delta, \delta')$, is to decide between:

- Yes instance: G has an ind. set with size n^γ on A side and $\frac{n}{\log^\delta n}$ on B side, i.e. a $(n^\gamma, \frac{n}{\log^\delta n})$-BIS

Hardness result (cont'd)

Definition: Given bip graph $G(A \cup B, E)$ with $|A| = |B| = n$, $0 < \gamma' < \gamma \leq 1$, $0 \leq \delta < \delta' \leq 1$, $BBIS(\gamma, \gamma', \delta, \delta')$, is to decide between:

- Yes instance: G has an ind. set with size n^γ on A side and $\frac{n}{\log^\delta n}$ on B side, i.e. a $(n^\gamma, \frac{n}{\log^\delta n})$-BIS
- No instance: G has no $(n^{\gamma'}, \frac{n}{\log^{\delta'} n})$-BIS



Hardness result (cont'd)

Main Theorem: There is a reduction from BBIS to U.C. such that given $G(A \cup B, E)$ and $0 < \gamma' < \gamma \leq 1$, $0 \leq \delta < \delta' \leq 1$, constructs in randomized polytime an instance $H(U \cup S, F)$ of U.C. with $|U| \in \Theta((\gamma - \gamma')n \log n)$ and $|S| = n$, s.t.

Hardness result (cont'd)

Main Theorem: There is a reduction from BBIS to U.C. such that given $G(A \cup B, E)$ and $0 < \gamma' < \gamma \leq 1$, $0 \leq \delta < \delta' \leq 1$, constructs in randomized polytime an instance $H(U \cup S, F)$ of U.C. with $|U| \in \Theta((\gamma - \gamma')n \log n)$ and $|S| = n$, s.t.

- Yes: if G is a Yes instance for $BBIS(\gamma, \gamma', \delta, \delta')$, then H has a solution of size $\Omega((\gamma - \gamma')n \log^{1-\delta} n)$

Hardness result (cont'd)

Main Theorem: There is a reduction from BBIS to U.C. such that given $G(A \cup B, E)$ and $0 < \gamma' < \gamma \leq 1$, $0 \leq \delta < \delta' \leq 1$, constructs in randomized polytime an instance $H(U \cup S, F)$ of U.C. with $|U| \in \Theta((\gamma - \gamma')n \log n)$ and $|S| = n$, s.t.

- Yes: if G is a Yes instance for $BBIS(\gamma, \gamma', \delta, \delta')$, then H has a solution of size $\Omega((\gamma - \gamma')n \log^{1-\delta} n)$

- No: if G is a No instance then every solution of H has size $O((\gamma - \gamma')n \log^{1-\delta'} n)$.

Hardness result (cont'd)

Main Theorem: There is a reduction from BBIS to U.C. such that given $G(A \cup B, E)$ and $0 < \gamma' < \gamma \leq 1$, $0 \leq \delta < \delta' \leq 1$, constructs in randomized polytime an instance $H(U \cup S, F)$ of U.C. with $|U| \in \Theta((\gamma - \gamma')n \log n)$ and $|S| = n$, s.t.

- Yes: if G is a Yes instance for $BBIS(\gamma, \gamma', \delta, \delta')$, then H has a solution of size $\Omega((\gamma - \gamma')n \log^{1-\delta} n)$
- No: if G is a No instance then every solution of H has size $O((\gamma - \gamma')n \log^{1-\delta'} n)$.

Corollary: Assuming that $BBIS(\gamma, \gamma', \delta, \delta')$ is "hard" then U.C. has a hardness of factor $\Omega(\log^{\delta' - \delta} n)$.

Reduction from $BBIS(\gamma, \gamma', \delta, \delta')$

- Given $G(A \cup B, E)$ with $|A| = |B| = n$, let $p = c \cdot \log n$ for some const $c > 0$

Reduction from $BBIS(\gamma, \gamma', \delta, \delta')$

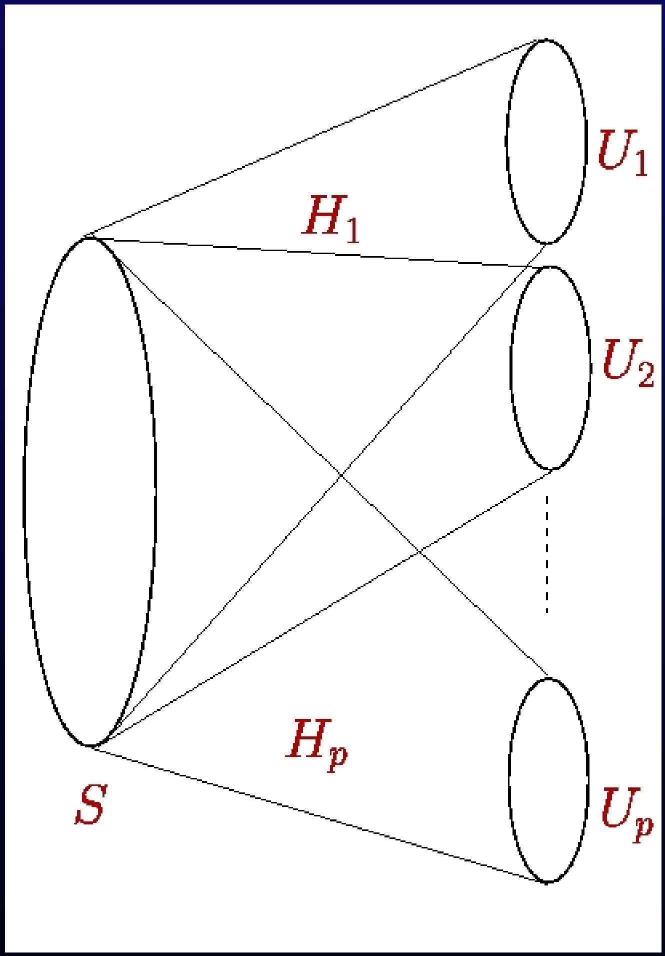
- Given $G(A \cup B, E)$ with $|A| = |B| = n$, let $p = c \cdot \log n$ for some const $c > 0$
- We construct instance $H(U \cup S, F)$ with $|U| = p \cdot n$ and $|S| = n$

Reduction from $BBIS(\gamma, \gamma', \delta, \delta')$

- Given $G(A \cup B, E)$ with $|A| = |B| = n$, let $p = c \cdot \log n$ for some const $c > 0$
- We construct instance $H(U \cup S, F)$ with $|U| = p \cdot n$ and $|S| = n$
- H consists of p smaller graphs: $H = \bigcup_{i=1}^{p} H_i$, $H_i(U_i \cup S, F_i)$,

Reduction from $BBIS(\gamma, \gamma', \delta, \delta')$

- Given $G(A \cup B, E)$ with $|A| = |B| = n$, let $p = c \cdot \log n$ for some const $c > 0$
- We construct instance $H(U \cup S, F)$ with $|U| = p \cdot n$ and $|S| = n$
- H consists of p smaller graphs: $H = \bigcup_{i=1}^{p} H_i$, $H_i(U_i \cup S, F_i)$,



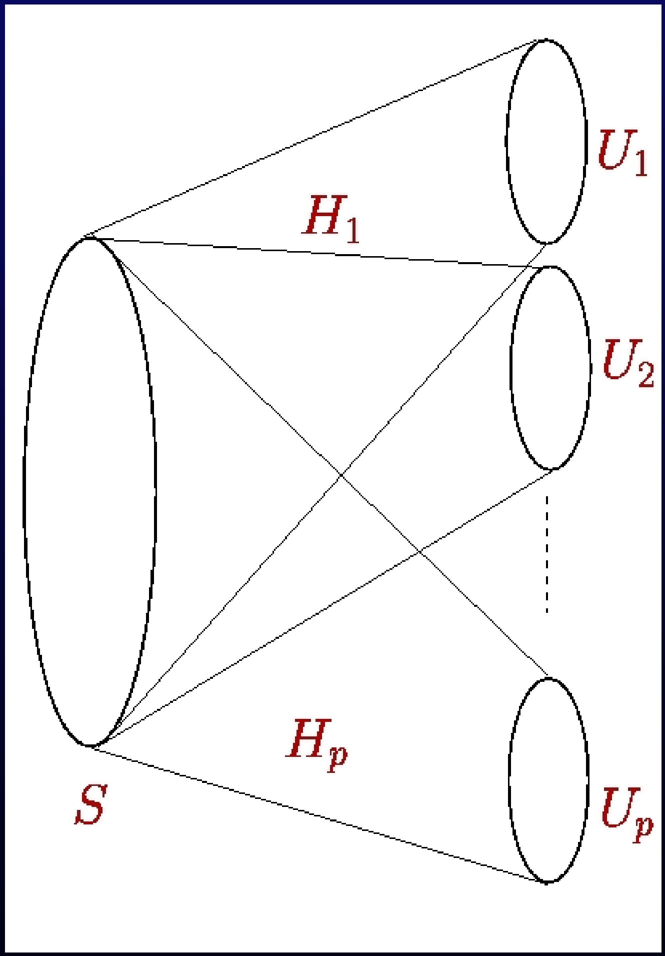
Reduction from $BBIS(\gamma, \gamma', \delta, \delta')$

- Given $G(A \cup B, E)$ with $|A| = |B| = n$, let $p = c \cdot \log n$ for some const $c > 0$

- We construct instance $H(U \cup S, F)$ with $|U| = p \cdot n$ and $|S| = n$

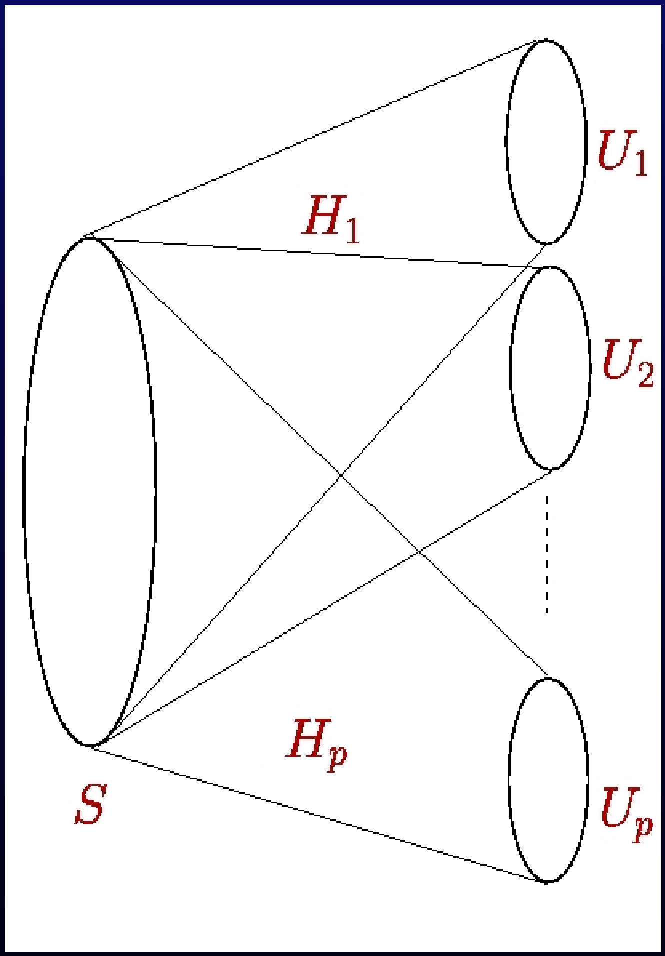
- H consists of p smaller graphs: $H = \bigcup_{i=1}^{p} H_i$, $H_i(U_i \cup S, F_i)$,

- Each H_i will have n in each part, $|U_i| = n$



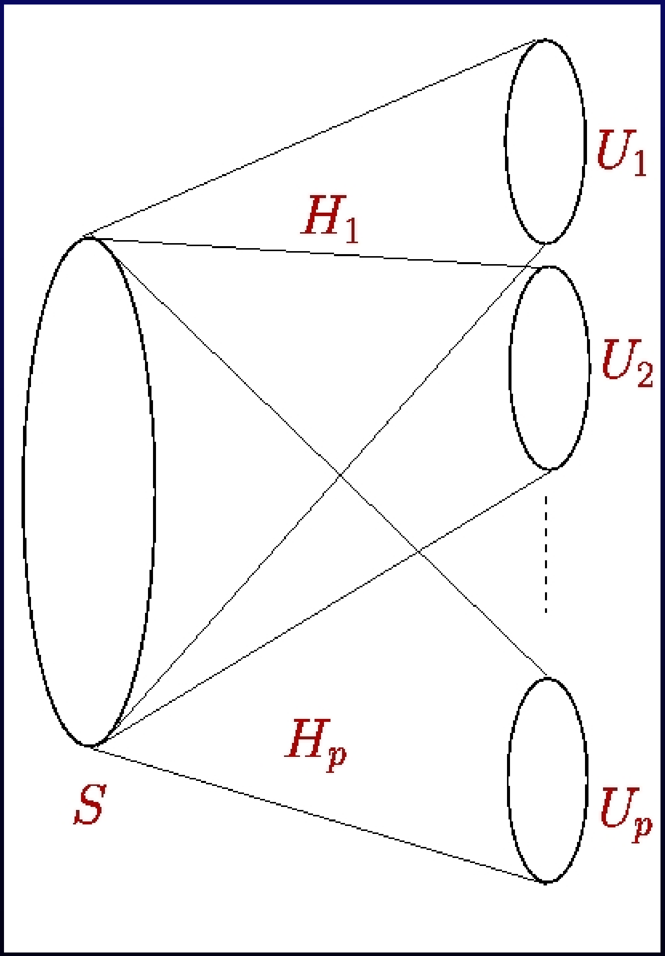
Reduction from $BBIS(\gamma, \gamma', \delta, \delta')$

- Given $G(A \cup B, E)$ with $|A| = |B| = n$, let $p = c \cdot \log n$ for some const $c > 0$
- We construct instance $H(U \cup S, F)$ with $|U| = p \cdot n$ and $|S| = n$
- H consists of p smaller graphs: $H = \bigcup_{i=1}^{p} H_i$, $H_i(U_i \cup S, F_i)$,
- Each H_i will have n in each part, $|U_i| = n$
- initially, $H_1 = G$,



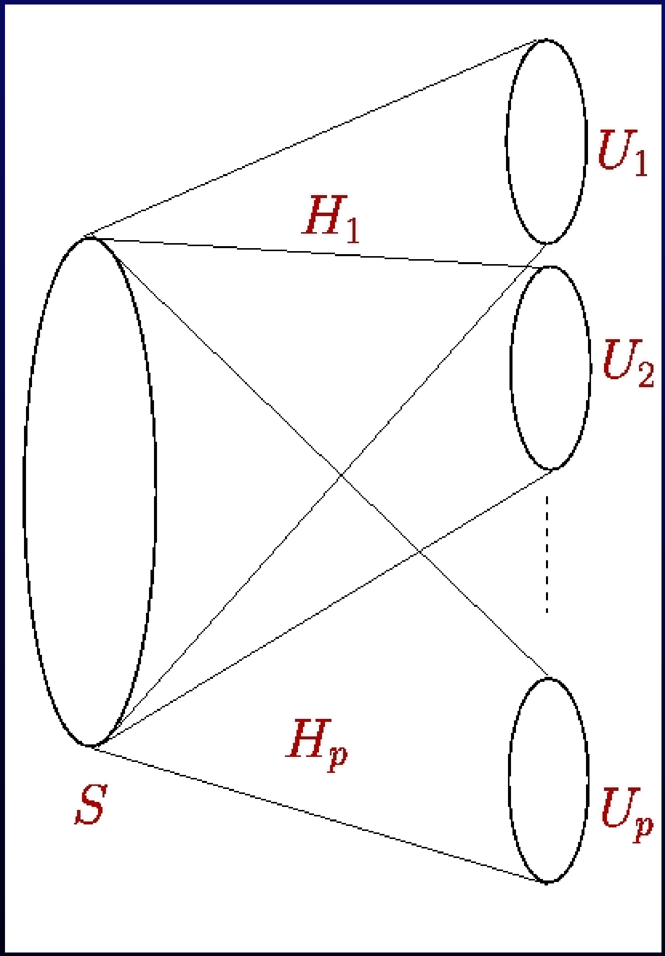
Reduction from $BBIS(\gamma, \gamma', \delta, \delta')$

- Given $G(A \cup B, E)$ with $|A| = |B| = n$, let $p = c \cdot \log n$ for some const $c > 0$
- We construct instance $H(U \cup S, F)$ with $|U| = p \cdot n$ and $|S| = n$
- H consists of p smaller graphs: $H = \bigcup_{i=1}^{p} H_i$, $H_i(U_i \cup S, F_i)$,
- Each H_i will have n in each part, $|U_i| = n$
- initially, $H_1 = G$,
- to go from H_i to H_{i+1}: delete every edge with probability $\frac{1}{2}$



Reduction from $BBIS(\gamma, \gamma', \delta, \delta')$

- Given $G(A \cup B, E)$ with $|A| = |B| = n$, let $p = c \cdot \log n$ for some const $c > 0$
- We construct instance $H(U \cup S, F)$ with $|U| = p \cdot n$ and $|S| = n$
- H consists of p smaller graphs: $H = \bigcup_{i=1}^{p} H_i$, $H_i(U_i \cup S, F_i)$,
- Each H_i will have n in each part, $|U_i| = n$
- initially, $H_1 = G$,
- to go from H_i to H_{i+1}: delete every edge with probability $\frac{1}{2}$



- We will add another set of random edges to each H_i

Reduction from $BBIS(\gamma, \gamma', \delta, \delta')$ (cont'd)

- Let $G'(A' \cup B', E')$ with $|A'| = |B'| = n$ be a random graph with edge prob. $\frac{1}{n^\gamma}$; so expect degree in G' is $n^{1-\gamma}$.

Reduction from $BBIS(\gamma, \gamma', \delta, \delta')$ (cont'd)

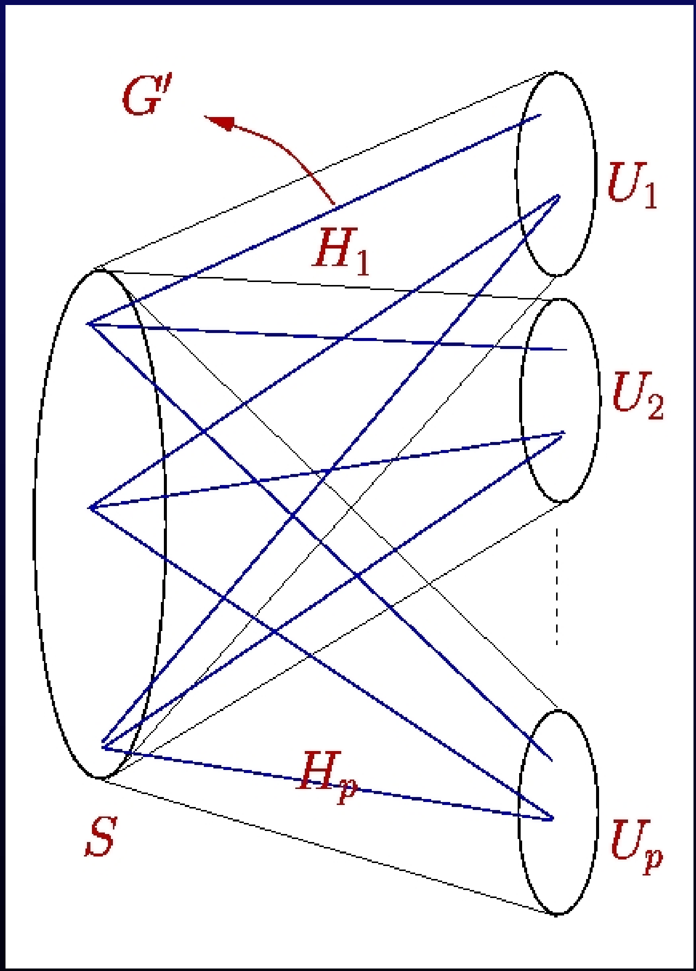
- Let $G'(A' \cup B', E')$ with $|A'| = |B'| = n$ be a random graph with edge prob. $\frac{1}{n^\gamma}$; so expect degree in G' is $n^{1-\gamma}$.

- Add G' on top of each H_i; this will be the graph H_i

Reduction from $BBIS(\gamma, \gamma', \delta, \delta')$ (cont'd)

- Let $G'(A' \cup B', E')$ with $|A'| = |B'| = n$ be a random graph with edge prob. $\frac{1}{n^\gamma}$; so expect degree in G' is $n^{1-\gamma}$.

- Add G' on top of each H_i; this will be the graph H_i



Reduction from $BBIS(\gamma, \gamma', \delta, \delta')$ (cont'd)

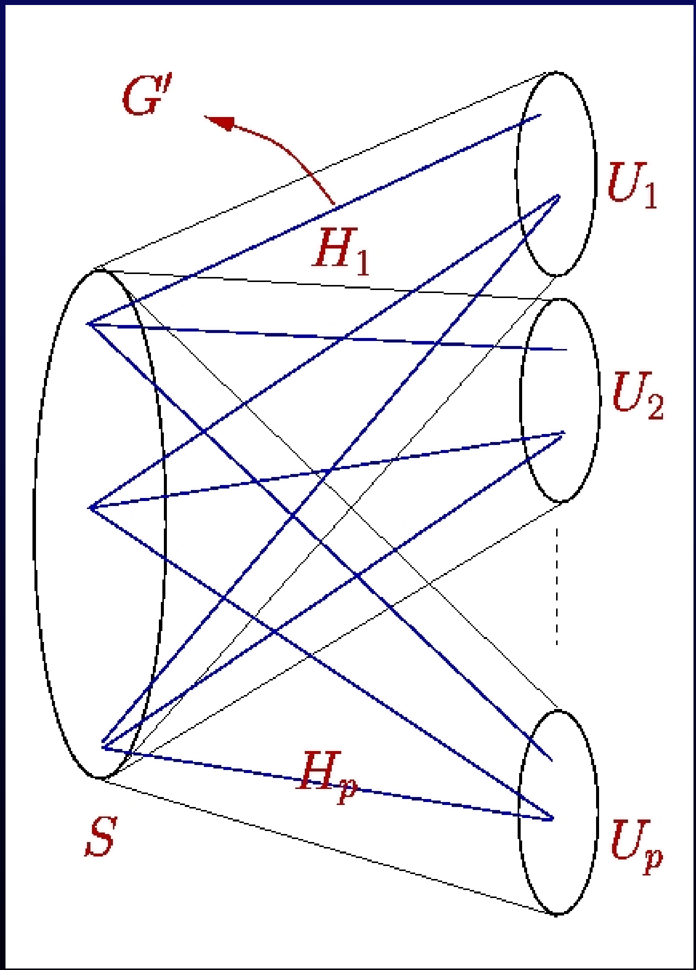
- Let $G'(A' \cup B', E')$ with $|A'| = |B'| = n$ be a random graph with edge prob. $\frac{1}{n^\gamma}$; so expect degree in G' is $n^{1-\gamma}$.

- Add G' on top of each H_i; this will be the graph H_i

- So each H_i has two types of edges:

  Type 1: random edges coming from G'

  Type 2: edges that were originally from G and are left in H_i (with probability $\frac{1}{2^i}$ for H_i).



Reduction from $BBIS(\gamma, \gamma', \delta, \delta')$ (cont'd)

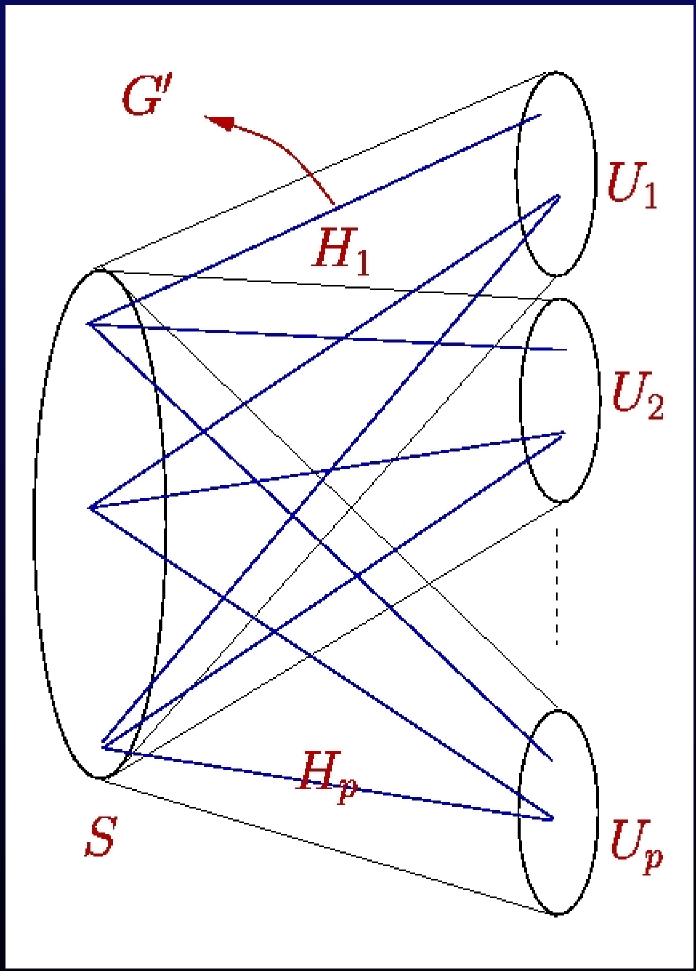
- Let $G'(A' \cup B', E')$ with $|A'| = |B'| = n$ be a random graph with edge prob. $\frac{1}{n^\gamma}$; so expect degree in G' is $n^{1-\gamma}$.

- Add G' on top of each H_i; this will be the graph H_i

- So each H_i has two types of edges:

  Type 1: random edges coming from G'

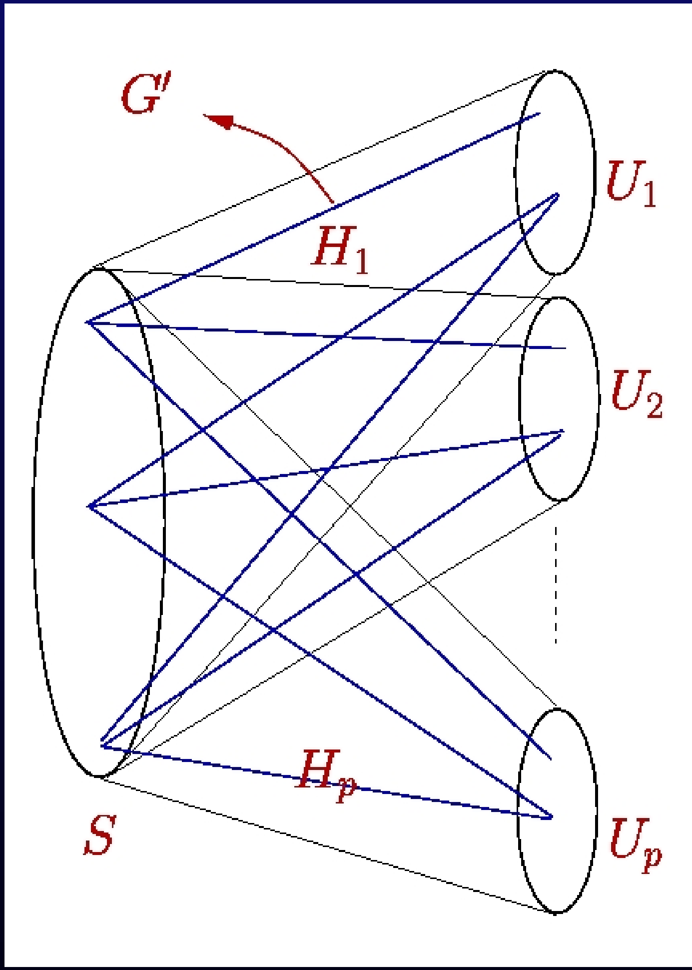
  Type 2: edges that were originally from G and are left in H_i (with probability $\frac{1}{2^i}$ for H_i).



This completes the construction of H from graph G.

Main Steps of the proof

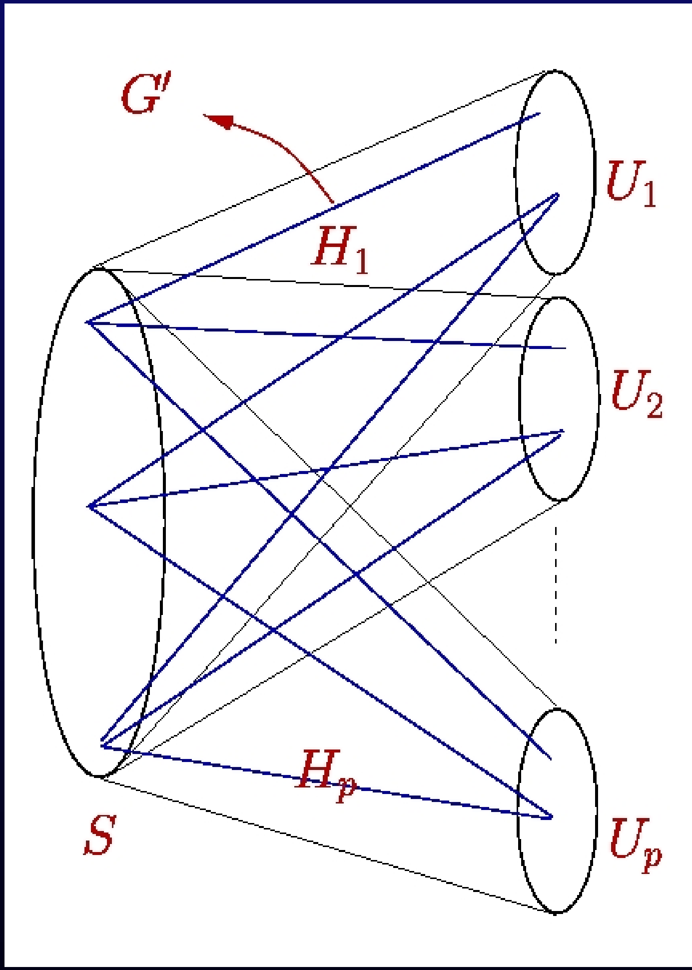
We will show (details to follow):



Main Steps of the proof

We will show (details to follow):

1. Type 2 edges (originally in G) uniquely cover at most $O(n)$ vertices (out of $p \cdot n \approx n \log n$ vertices) of U

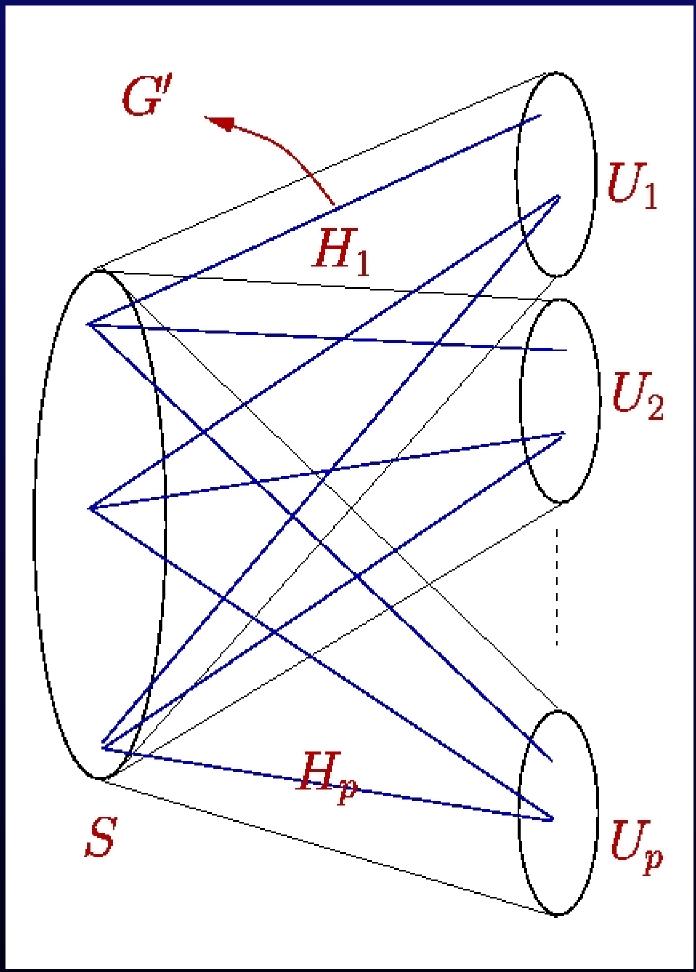


Main Steps of the proof

We will show (details to follow):

1. Type 2 edges (originally in G) uniquely cover at most $O(n)$ vertices (out of $p \cdot n \approx n \log n$ vertices) of U

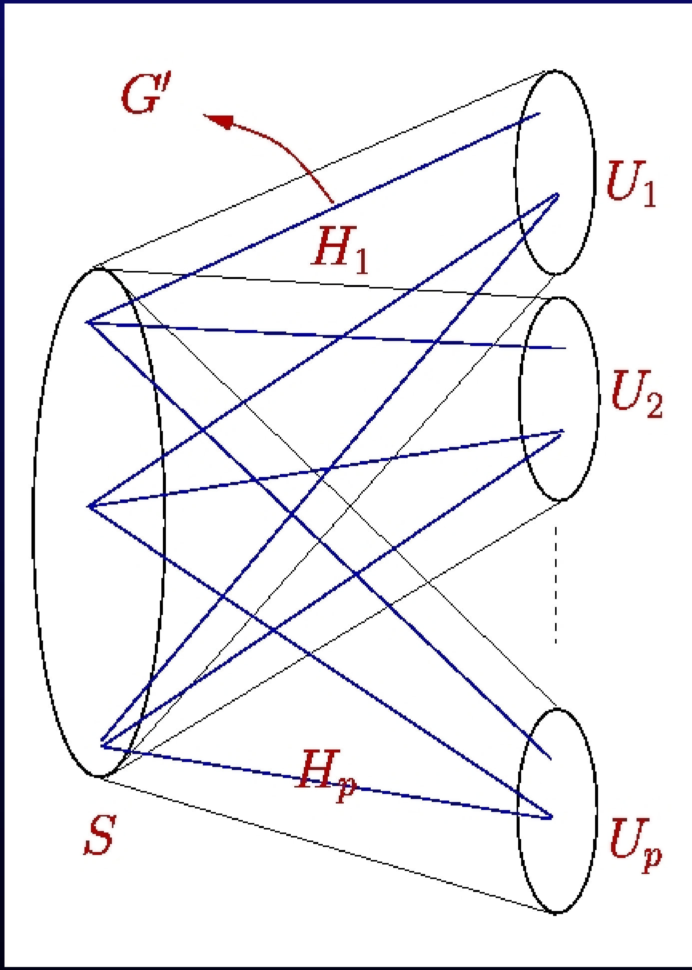
2. If G has a $(n^\gamma, \frac{n}{\log^\delta n})$-BIS (Yes case) then type 1 edges (those coming from G') uniquely cover $\Omega(\frac{n}{\log^\delta n})$ vertices in each $U_i \longrightarrow$ a total of $\Omega(n \log^{1-\delta} n)$ u.c. elements



Main Steps of the proof

We will show (details to follow):

1. Type 2 edges (originally in G) uniquely cover at most $O(n)$ vertices (out of $p \cdot n \approx n \log n$ vertices) of U
2. If G has a $(n^\gamma, \frac{n}{\log^\delta n})$-BIS (Yes case) then type 1 edges (those coming from G') uniquely cover $\Omega(\frac{n}{\log^\delta n})$ vertices in each $U_i \longrightarrow$ a total of $\Omega(n \log^{1-\delta} n)$ u.c. elements
3. If G has no $(n^{\gamma'}, \frac{n}{\log^{\delta'} n})$-BIS (No case) then type 1 edges (from G') uniquely cover at most $O(\frac{n}{\log^{\delta'} n})$ vertices in each $U_i \longrightarrow$ a total of $O(n \log^{1-\delta'} n)$



Details of the proof

Lemma 1: type 2 edges (coming from G) uniquely cover at most $O(n)$ vertices out of $\Omega(p \cdot n)$ vertices.

Details of the proof

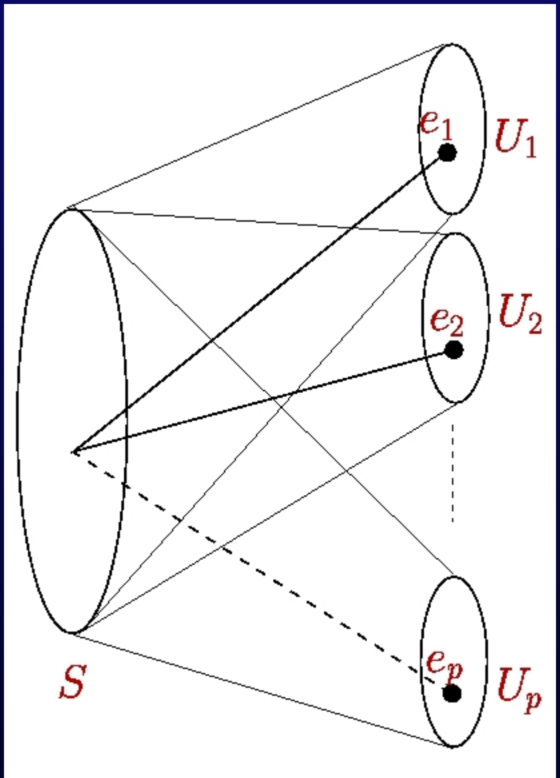
Lemma 1: type 2 edges (coming from G) uniquely cover at most $O(n)$ vertices out of $\Omega(p \cdot n)$ vertices.

Proof:

- Choose $b \in B$ arbitrarily; let $e_1, \ldots, e_p$ be the vertices in $U_1, \ldots, U_p$ corresponding to b

Details of the proof

Lemma 1: type 2 edges (coming from G) uniquely cover at most $O(n)$ vertices out of $\Omega(p \cdot n)$ vertices.

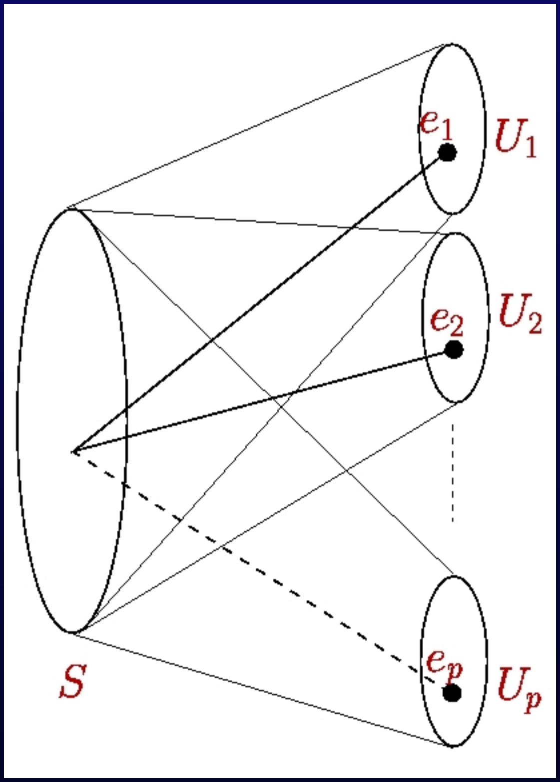


Proof:

- Choose $b \in B$ arbitrarily; let $e_1, \ldots, e_p$ be the vertices in $U_1, \ldots, U_p$ corresponding to b

Details of the proof

Lemma 1: type 2 edges (coming from G) uniquely cover at most $O(n)$ vertices out of $\Omega(p \cdot n)$ vertices.



Proof:

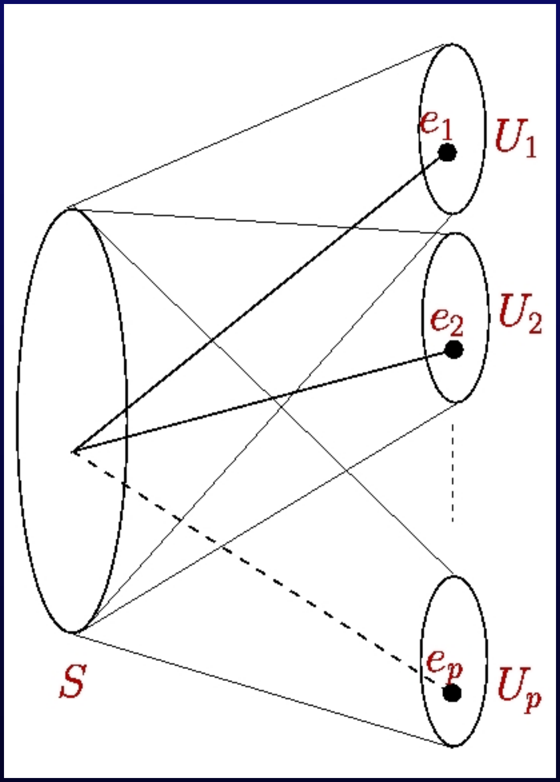
- Choose $b \in B$ arbitrarily; let $e_1, \ldots, e_p$ be the vertices in $U_1, \ldots, U_p$ corresponding to b
- Assuming that e_j is uniquely covered by a type 2 edge, the prob. that $e_j, \ldots, e_{j+i-1}$ are all u.c. by that edge is: $\frac{1}{2^i}$

Details of the proof

Lemma 1: type 2 edges (coming from G) uniquely cover at most $O(n)$ vertices out of $\Omega(p \cdot n)$ vertices.

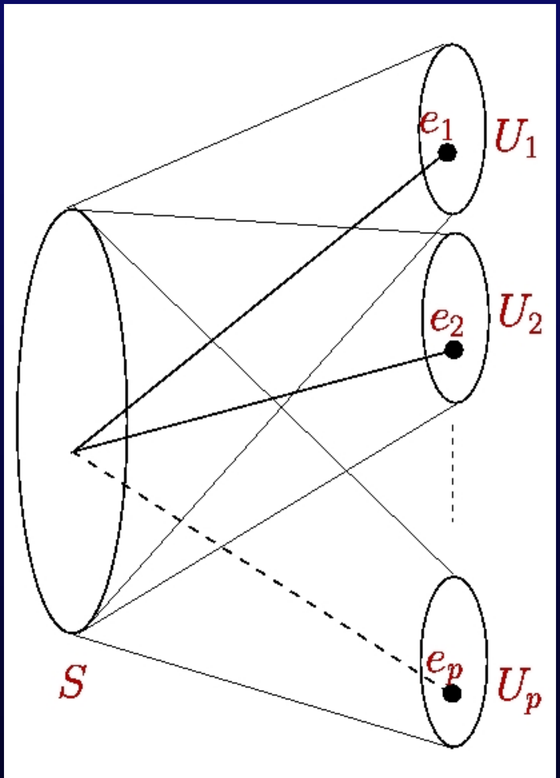
Proof:

- Choose $b \in B$ arbitrarily; let $e_1, \ldots, e_p$ be the vertices in $U_1, \ldots, U_p$ corresponding to b
- Assuming that e_j is uniquely covered by a type 2 edge, the prob. that $e_j, \ldots, e_{j+i-1}$ are all u.c. by that edge is: $\frac{1}{2^i}$
- So we expect only a few copies of $e_1, \ldots, e_p$ be uniquely covered by a type 2 edge



Details of the proof

Lemma 1: type 2 edges (coming from G) uniquely cover at most $O(n)$ vertices out of $\Omega(p \cdot n)$ vertices.



Proof:

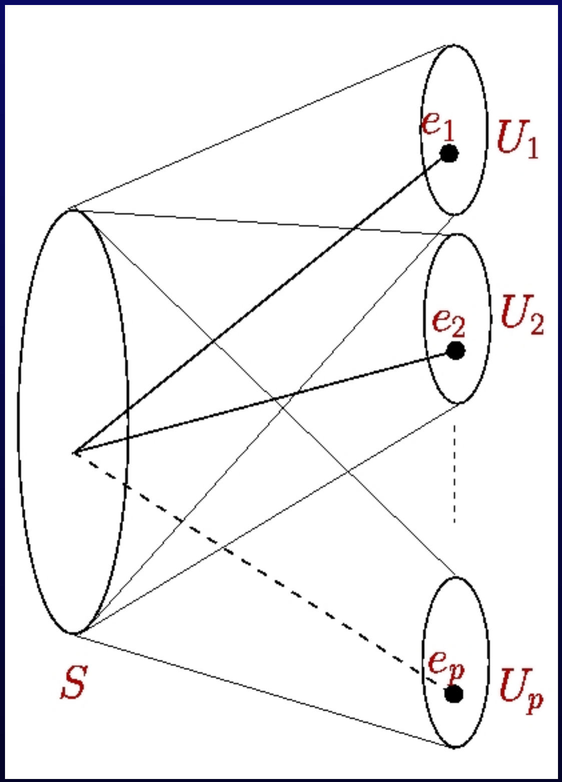
- Choose $b \in B$ arbitrarily; let $e_1, \ldots, e_p$ be the vertices in $U_1, \ldots, U_p$ corresponding to b
- Assuming that e_j is uniquely covered by a type 2 edge, the prob. that $e_j, \ldots, e_{j+i-1}$ are all u.c. by that edge is: $\frac{1}{2^i}$
- So we expect only a few copies of $e_1, \ldots, e_p$ be uniquely covered by a type 2 edge
- Let X_b= number of e_i's that are u.c. by a type 2 edge

Details of the proof

Lemma 1: type 2 edges (coming from G) uniquely cover at most $O(n)$ vertices out of $\Omega(p \cdot n)$ vertices.

Proof:

- Choose $b \in B$ arbitrarily; let $e_1, \ldots, e_p$ be the vertices in $U_1, \ldots, U_p$ corresponding to b
- Assuming that e_j is uniquely covered by a type 2 edge, the prob. that $e_j, \ldots, e_{j+i-1}$ are all u.c. by that edge is: $\frac{1}{2^i}$
- So we expect only a few copies of $e_1, \ldots, e_p$ be uniquely covered by a type 2 edge



- Let X_b= number of e_i's that are u.c. by a type 2 edge

  $X = \sum_{b \in B} X_b$ number of vertices u.c. by type 2 edges

$$\mathrm{E}[X] = n \sum_{i=1}^{p} \frac{i}{2^i} \in O(n)$$

Details of the proof (cont'd)

Lemma 2: If G has a $(n^\gamma, \frac{n}{\log^\delta n})$-BIS (Yes case) then type 1 edges (those coming from G') uniquely cover $\Omega(\frac{n}{\log^\delta n})$ vertices in each U_i (a total of $\Omega(n \log^{1-\delta} n)$).

Details of the proof (cont'd)

Lemma 2: If G has a $(n^\gamma, \frac{n}{\log^\delta n})$-BIS (Yes case) then type 1 edges (those coming from G') uniquely cover $\Omega(\frac{n}{\log^\delta n})$ vertices in each U_i (a total of $\Omega(n \log^{1-\delta} n)$).

Proof:

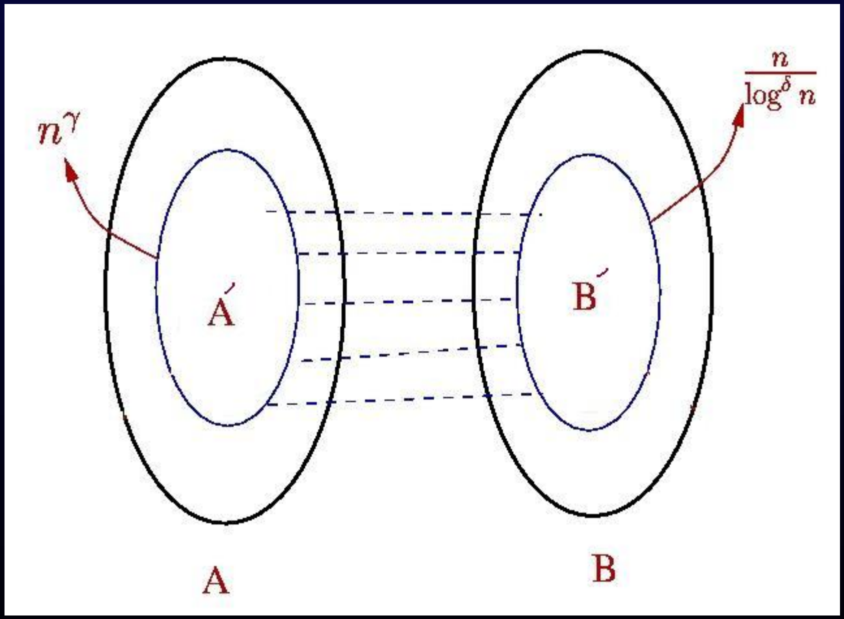
- Assume $A^* \cup B^*$ ($A^* \subseteq A$, $B^* \subseteq B$) is a $(n^\gamma, \frac{n}{\log^\delta n})$-BIS; consider corresponding sets A', B' in G' and S^*, U_i^* in H_i

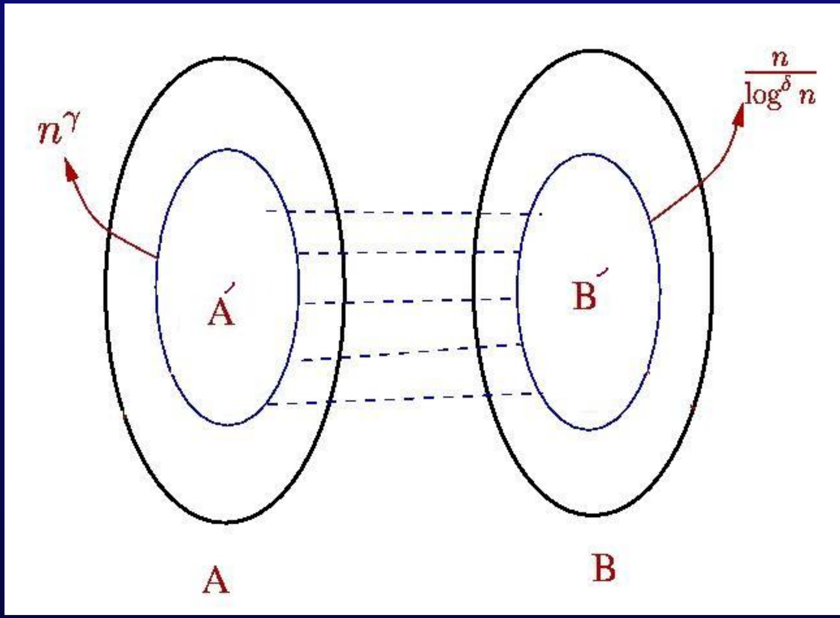
Details of the proof (cont'd)

Lemma 2: If G has a $(n^\gamma, \frac{n}{\log^\delta n})$-BIS (Yes case) then type 1 edges (those coming from G') uniquely cover $\Omega(\frac{n}{\log^\delta n})$ vertices in each U_i (a total of $\Omega(n \log^{1-\delta} n)$).

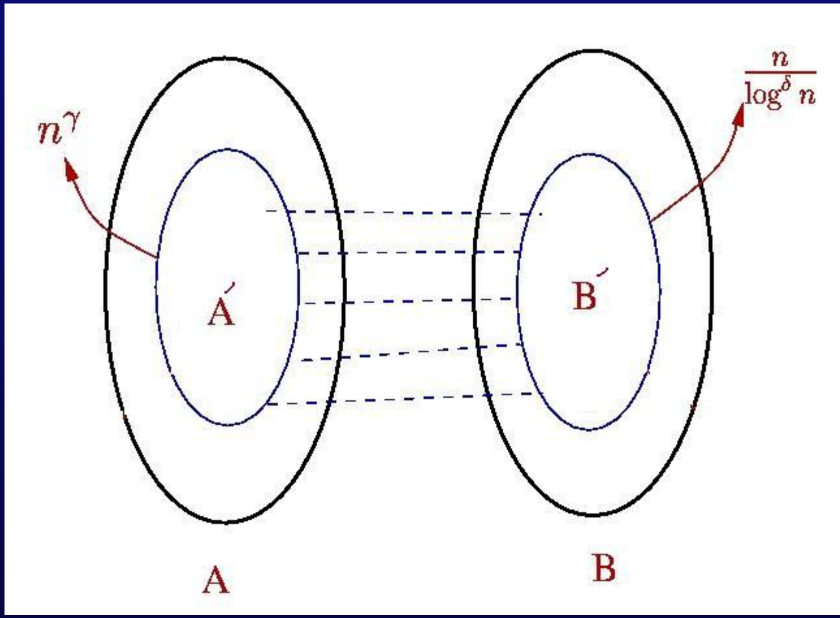
Proof:

- Assume $A^* \cup B^*$ ($A^* \subseteq A$, $B^* \subseteq B$) is a $(n^\gamma, \frac{n}{\log^\delta n})$-BIS; consider corresponding sets A', B' in G' and S^*, U_i^* in H_i

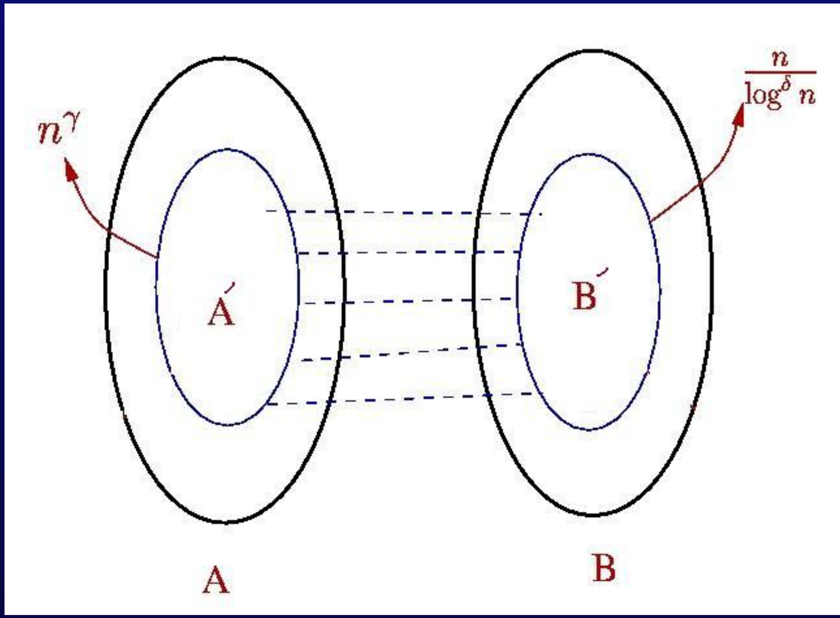




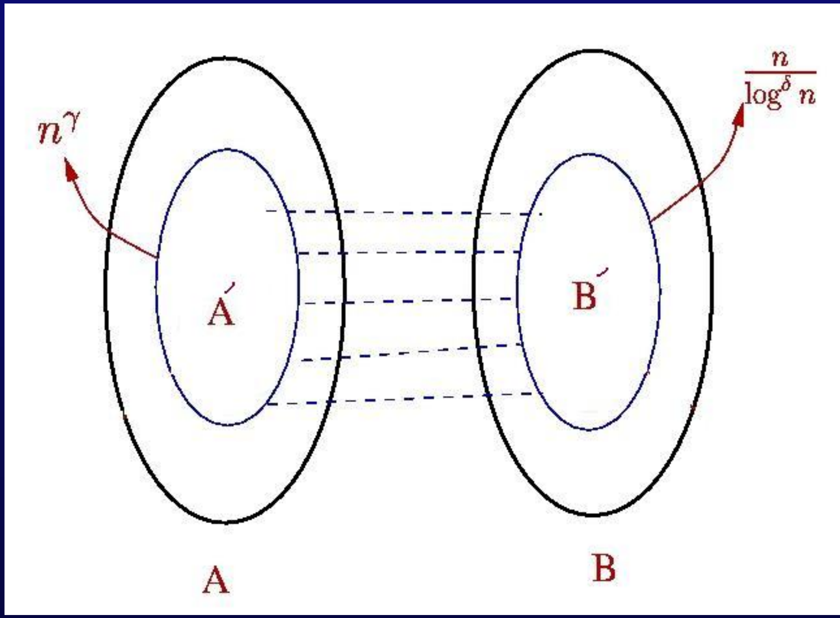
- Note there is no type 2 edges between S^*, U_i^* (for any i)



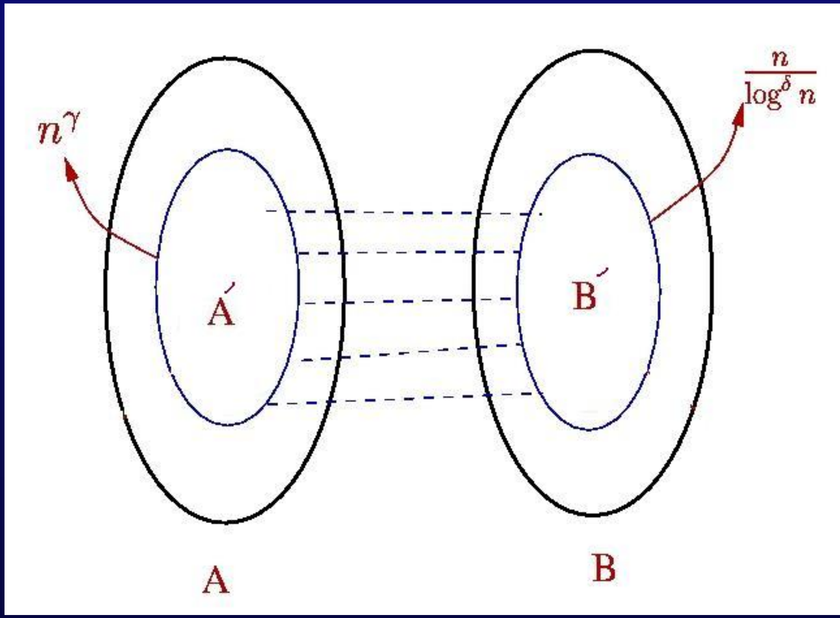
- Note there is no type 2 edges between S^*, U_i^* (for any i)
- Also recall that every edge is present between A', B' with probability $\frac{1}{n^\gamma}$



- Note there is no type 2 edges between S^*, U_i^* (for any i)
- Also recall that every edge is present between A', B' with probability $\frac{1}{n^\gamma}$
- Let's count the number of vertices $b \in B'$ with degree 1 (by a type 1 edge)



- Note there is no type 2 edges between S^*, U_i^* (for any i)
- Also recall that every edge is present between A', B' with probability $\frac{1}{n^\gamma}$
- Let's count the number of vertices $b \in B'$ with degree 1 (by a type 1 edge)
- Say $X_b = 1$ iff b has degree 1; set $X = \sum_b X_b$



- Note there is no type 2 edges between S^*, U_i^* (for any i)
- Also recall that every edge is present between A', B' with probability $\frac{1}{n^\gamma}$
- Let's count the number of vertices $b \in B'$ with degree 1 (by a type 1 edge)
- Say $X_b = 1$ iff b has degree 1; set $X = \sum_b X_b$

$$\mathrm{E}[X] = \sum_{b \in B'} \Pr[X_b = 1] = |B'| \binom{|A'|}{1} \frac{1}{n^\gamma} \left(1 - \frac{1}{n^\gamma}\right)^{|A'|-1} \in \Omega\left(\frac{n}{\log^\delta n}\right)$$

Details of the proof (cont'd)

Lemma 3: If G has no $(n^{\gamma'}, \frac{n}{\log^{\delta'} n})$-BIS then w.h.p. every u.c. solution of H has size $O(n \log^{1-\delta'} n)$.

Details of the proof (cont'd)

Lemma 3: If G has no $(n^{\gamma'}, \frac{n}{\log^{\delta'} n})$-BIS then w.h.p. every u.c. solution of H has size $O(n \log^{1-\delta'} n)$.

Remark 1: By Lemma 1, type 2 edges never u.c. more than $O(n)$ $\longrightarrow$ only need to consider type 1 edges.

Details of the proof (cont'd)

Lemma 3: If G has no $(n^{\gamma'}, \frac{n}{\log^{\delta'} n})$-BIS then w.h.p. every u.c. solution of H has size $O(n \log^{1-\delta'} n)$.

Remark 1: By Lemma 1, type 2 edges never u.c. more than $O(n)$ $\longrightarrow$ only need to consider type 1 edges.

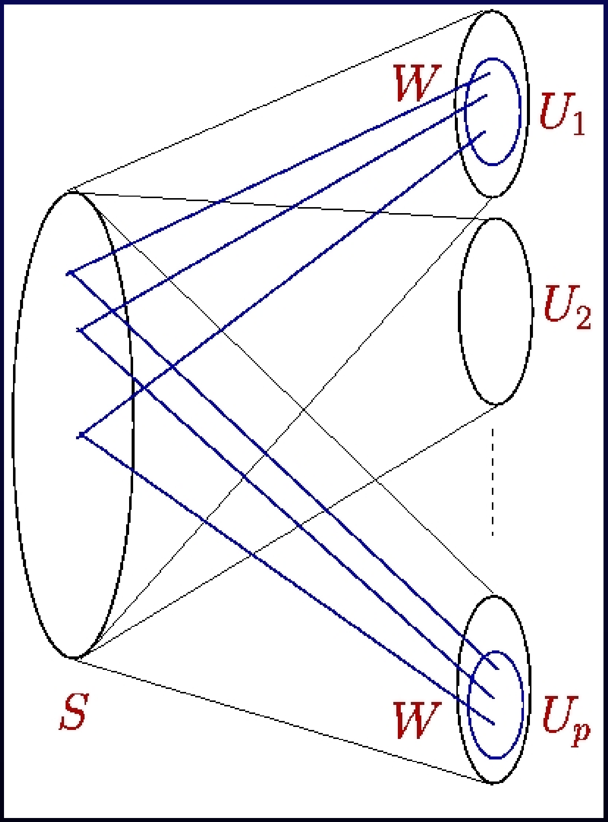
Remark 2: Since edges (of type 2) are delete in $H_i \rightsquigarrow H_{i+1}$, if set $W \subseteq U_i$ is u.c. (by type 1 edges) they are also u.c. in U_{i+1}.

Details of the proof (cont'd)

Lemma 3: If G has no $(n^{\gamma'}, \frac{n}{\log^{\delta'} n})$-BIS then w.h.p. every u.c. solution of H has size $O(n \log^{1-\delta'} n)$.

Remark 1: By Lemma 1, type 2 edges never u.c. more than $O(n) \longrightarrow$ only need to consider type 1 edges.

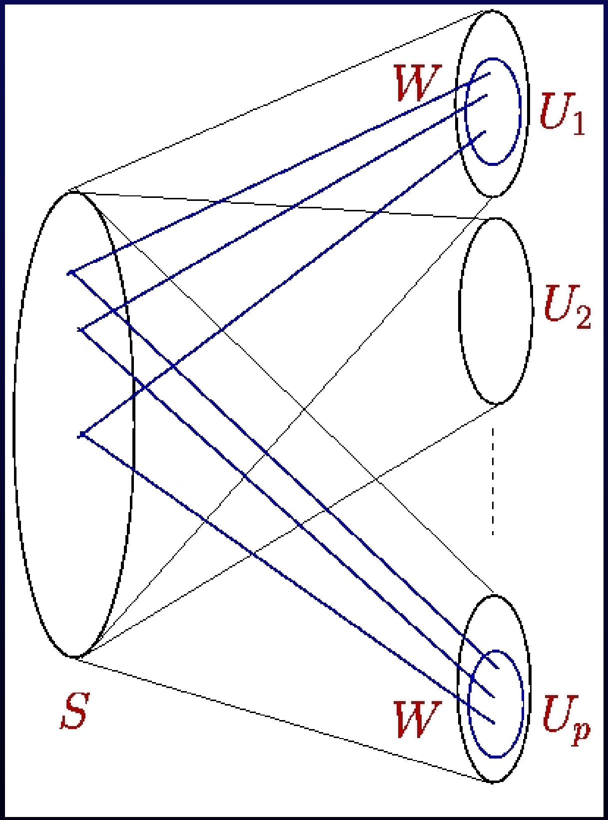
Remark 2: Since edges (of type 2) are delete in $H_i \rightsquigarrow H_{i+1}$, if set $W \subseteq U_i$ is u.c. (by type 1 edges) they are also u.c. in U_{i+1}.



Details of the proof (cont'd)

Lemma 3: If G has no $(n^{\gamma'}, \frac{n}{\log^{\delta'} n})$-BIS then w.h.p. every u.c. solution of H has size $O(n \log^{1-\delta'} n)$.

Remark 1: By Lemma 1, type 2 edges never u.c. more than $O(n)$ $\longrightarrow$ only need to consider type 1 edges.



Remark 2: Since edges (of type 2) are delete in $H_i \rightsquigarrow H_{i+1}$, if set $W \subseteq U_i$ is u.c. (by type 1 edges) they are also u.c. in U_{i+1}.

We bound size of a set like W in U_p

Completing proof of the main theorem

From Lemmas 1, 2, and 3 follows:

Completing proof of the main theorem

From Lemmas 1, 2, and 3 follows:

- If G has $(n^\gamma, \frac{n}{\log^\delta n})$-BIS $\longrightarrow$ Sol of H has size $\in \Omega(\log^{1-\delta} n)$

Completing proof of the main theorem

From Lemmas 1, 2, and 3 follows:

- If G has $(n^\gamma, \frac{n}{\log^\delta n})$-BIS $\longrightarrow$ Sol of H has size $\in \Omega(\log^{1-\delta} n)$
- If G has no $(n^{\gamma'}, \frac{n}{\log^{\delta'} n})$-BIS $\longrightarrow$ Sol of H has size $\in O(\log^{1-\delta'} n)$

Completing proof of the main theorem

From Lemmas 1, 2, and 3 follows:

- If G has $(n^\gamma, \frac{n}{\log^\delta n})$-BIS $\longrightarrow$ Sol of H has size $\in \Omega(\log^{1-\delta} n)$
- If G has no $(n^{\gamma'}, \frac{n}{\log^{\delta'} n})$-BIS $\longrightarrow$ Sol of H has size $\in O(\log^{1-\delta'} n)$

$\longrightarrow$ hardness factor for U.C. $\Omega(\log^{\delta'-\delta} n)$

How to prove hardness for BBIS?

Completing proof of the main theorem

From Lemmas 1, 2, and 3 follows:

- If G has $(n^\gamma, \frac{n}{\log^\delta n})$-BIS $\longrightarrow$ Sol of H has size $\in \Omega(\log^{1-\delta} n)$
- If G has no $(n^{\gamma'}, \frac{n}{\log^{\delta'} n})$-BIS $\longrightarrow$ Sol of H has size $\in O(\log^{1-\delta'} n)$

$\longrightarrow$ hardness factor for U.C. $\Omega(\log^{\delta'-\delta} n)$

How to prove hardness for BBIS?

Theorem [khot'04]: Let $\epsilon > 0$ be some const. and Φ an instance of SAT with n variables. For some constants $\alpha = \alpha(\epsilon) > \beta = \beta(\epsilon)$, and $N = 2^{n^\epsilon}$, we can build a bip graph $G(A \cup B, E)$, $|A| = |B| = N$, s.t.

Completing proof of the main theorem

From Lemmas 1, 2, and 3 follows:

- If G has $(n^\gamma, \frac{n}{\log^\delta n})$-BIS $\longrightarrow$ Sol of H has size $\in \Omega(\log^{1-\delta} n)$
- If G has no $(n^{\gamma'}, \frac{n}{\log^{\delta'} n})$-BIS $\longrightarrow$ Sol of H has size $\in O(\log^{1-\delta'} n)$

$\longrightarrow$ hardness factor for U.C. $\Omega(\log^{\delta'-\delta} n)$

How to prove hardness for BBIS?

Theorem [khot'04]: Let $\epsilon > 0$ be some const. and Φ an instance of SAT with n variables. For some constants $\alpha = \alpha(\epsilon) > \beta = \beta(\epsilon)$, and $N = 2^{n^\epsilon}$, we can build a bip graph $G(A \cup B, E)$, $|A| = |B| = N$, s.t.

- if Φ is SAT $\longrightarrow$ G has a $(\alpha N, \alpha N)$-BIS

Completing proof of the main theorem

From Lemmas 1, 2, and 3 follows:

- If G has $(n^\gamma, \frac{n}{\log^\delta n})$-BIS $\longrightarrow$ Sol of H has size $\in \Omega(\log^{1-\delta} n)$
- If G has no $(n^{\gamma'}, \frac{n}{\log^{\delta'} n})$-BIS $\longrightarrow$ Sol of H has size $\in O(\log^{1-\delta'} n)$

$\longrightarrow$ hardness factor for U.C. $\Omega(\log^{\delta'-\delta} n)$

How to prove hardness for BBIS?

Theorem [khot'04]: Let $\epsilon > 0$ be some const. and Φ an instance of SAT with n variables. For some constants $\alpha = \alpha(\epsilon) > \beta = \beta(\epsilon)$, and $N = 2^{n^\epsilon}$, we can build a bip graph $G(A \cup B, E)$, $|A| = |B| = N$, s.t.

- if Φ is SAT $\longrightarrow$ G has a $(\alpha N, \alpha N)$-BIS
- if Φ is not SAT $\longrightarrow$ G has no $(\beta N, \beta N)$-BIS

But we need much larger gap (here it is only a constant!)

To prove hardness for BBIS

- Use the (standard) technique of graph products to boost the gap

To prove hardness for BBIS

- Use the (standard) technique of graph products to boost the gap
- Given G construct $G^{k_A,k_B}(A' \cup B', E')$ where
  - $\star$ in A', every vertex is a vector of length k_A from vertices of A
  - $\star$ in B', every vertex is a vector of length k_B from vertices of B

To prove hardness for BBIS

- Use the (standard) technique of graph products to boost the gap
- Given G construct $G^{k_A,k_B}(A' \cup B', E')$ where
  - ⋆ in A', every vertex is a vector of length k_A from vertices of A
  - ⋆ in B', every vertex is a vector of length k_B from vertices of B

$(a_1, \ldots, a_{k_A}) \in A', (b_1, \ldots, b_{k_B}) \in B'$ are adjacent $\iff$
$(a_i, b_j) \in E(G)$, for all $1 \leq i \leq k_A$, $1 \leq j \leq k_B$

To prove hardness for BBIS

- Use the (standard) technique of graph products to boost the gap
- Given G construct $G^{k_A,k_B}(A' \cup B', E')$ where
  - ⋆ in A', every vertex is a vector of length k_A from vertices of A
  - ⋆ in B', every vertex is a vector of length k_B from vertices of B

$(a_1, \ldots, a_{k_A}) \in A', (b_1, \ldots, b_{k_B}) \in B'$ are adjacent $\iff$
$(a_i, b_j) \in E(G)$, for all $1 \leq i \leq k_A$, $1 \leq j \leq k_B$

- For suitable k_A, k_B we can show:

Theorem: Unless $NP \subseteq BPTIME(2^{n^\epsilon})$ it is hard to distinguish between:

- ⋆ G^{k_A,k_B} has a $(n^\gamma, \frac{n}{\log^\delta n})$-BIS
- ⋆ G^{k_A,k_B} has no $(n^{\gamma'}, \frac{n}{\log^{\delta'} n})$-BIS

What next (Open problems)

The hardness result is not matching the approximation algorithm ratio ($O(\log n)$ v.s $\Omega(\log^{\delta} n)$) and it requires relatively strong assumption (i.e. $NP \not\subseteq 2^{n^{\epsilon}}$).

What next (Open problems)

The hardness result is not matching the approximation algorithm ratio ($O(\log n)$ v.s $\Omega(\log^{\delta} n)$) and it requires relatively strong assumption (i.e. $NP \not\subseteq 2^{n^{\epsilon}}$).

Hypothesis: Given a bipartite graph $G(A \cup B, E)$, $|A| = |B| = n$, for some $0 < \gamma' < \gamma \leq 1$, it is hard to distinguish between:

- Yes: G has an $(n^{\gamma}, \Omega(n))$-BIS
- No: G has no $(n^{\gamma'}, \frac{n}{\log n})$-BIS

What next (Open problems)

The hardness result is not matching the approximation algorithm ratio ($O(\log n)$ v.s $\Omega(\log^{\delta} n)$) and it requires relatively strong assumption (i.e. $NP \not\subseteq 2^{n^{\epsilon}}$).

Hypothesis: Given a bipartite graph $G(A \cup B, E)$, $|A| = |B| = n$, for some $0 < \gamma' < \gamma \leq 1$, it is hard to distinguish between:

- Yes: G has an $(n^{\gamma}, \Omega(n))$-BIS
- No: G has no $(n^{\gamma'}, \frac{n}{\log n})$-BIS

This would imply an $\Omega(\log n)$-hardness for U.C.

For the Radio Broadcast problem:

- We know [GPX'05] how to find a schedule of size $D + O(\log^2 n)$ (rounds).

For the Radio Broadcast problem:

- We know [GPX'05] how to find a schedule of size $D + O(\log^2 n)$ (rounds).
- For the case that the diameter is constant, e.g. say a bipartite graph plus another node (source) connected to all the vertices in one part, this gives an upper bound of $D + O(\log^2 n)$

For the Radio Broadcast problem:

- We know [GPX'05] how to find a schedule of size $D + O(\log^2 n)$ (rounds).
- For the case that the diameter is constant, e.g. say a bipartite graph plus another node (source) connected to all the vertices in one part, this gives an upper bound of $D + O(\log^2 n)$
- There are such graphs [ABLP'91] that need $\Omega(\log^2 n)$ rounds.

For the Radio Broadcast problem:

- We know [GPX'05] how to find a schedule of size $D + O(\log^2 n)$ (rounds).
- For the case that the diameter is constant, e.g. say a bipartite graph plus another node (source) connected to all the vertices in one part, this gives an upper bound of $D + O(\log^2 n)$
- There are such graphs [ABLP'91] that need $\Omega(\log^2 n)$ rounds.
- On the other hand [EK'04] showed that deciding between the following two is hard:

  ⋆ G has a schedule with $O(\log n)$ rounds
  ⋆ every schedule of G has $\Omega(\log^2 n)$ rounds

  This implies a (multiplicative) $\Omega(\log n)$ and *additive* $\Omega(\log^2 n)$ hardness

For the Radio Broadcast problem:

- We know [GPX'05] how to find a schedule of size $D + O(\log^2 n)$ (rounds).
- For the case that the diameter is constant, e.g. say a bipartite graph plus another node (source) connected to all the vertices in one part, this gives an upper bound of $D + O(\log^2 n)$
- There are such graphs [ABLP'91] that need $\Omega(\log^2 n)$ rounds.
- On the other hand [EK'04] showed that deciding between the following two is hard:

  ⋆ G has a schedule with $O(\log n)$ rounds
  ⋆ every schedule of G has $\Omega(\log^2 n)$ rounds

  This implies a (multiplicative) $\Omega(\log n)$ and *additive* $\Omega(\log^2 n)$ hardness
- Question: Can we prove an $\Omega(\log^2 n)$-hardness?

For the Radio Broadcast problem:

- We know [GPX'05] how to find a schedule of size $D + O(\log^2 n)$ (rounds).
- For the case that the diameter is constant, e.g. say a bipartite graph plus another node (source) connected to all the vertices in one part, this gives an upper bound of $D + O(\log^2 n)$
- There are such graphs [ABLP'91] that need $\Omega(\log^2 n)$ rounds.
- On the other hand [EK'04] showed that deciding between the following two is hard:
  - ⋆ G has a schedule with $O(\log n)$ rounds
  - ⋆ every schedule of G has $\Omega(\log^2 n)$ rounds

  This implies a (multiplicative) $\Omega(\log n)$ and *additive* $\Omega(\log^2 n)$ hardness
- Question: Can we prove an $\Omega(\log^2 n)$-hardness?
- Proposition: An $\Omega(\log^{1+\delta} n)$-hardness for Radio Broadcast implies an $\Omega(\log^{\delta} n)$-hardness for U.C. (easy!).