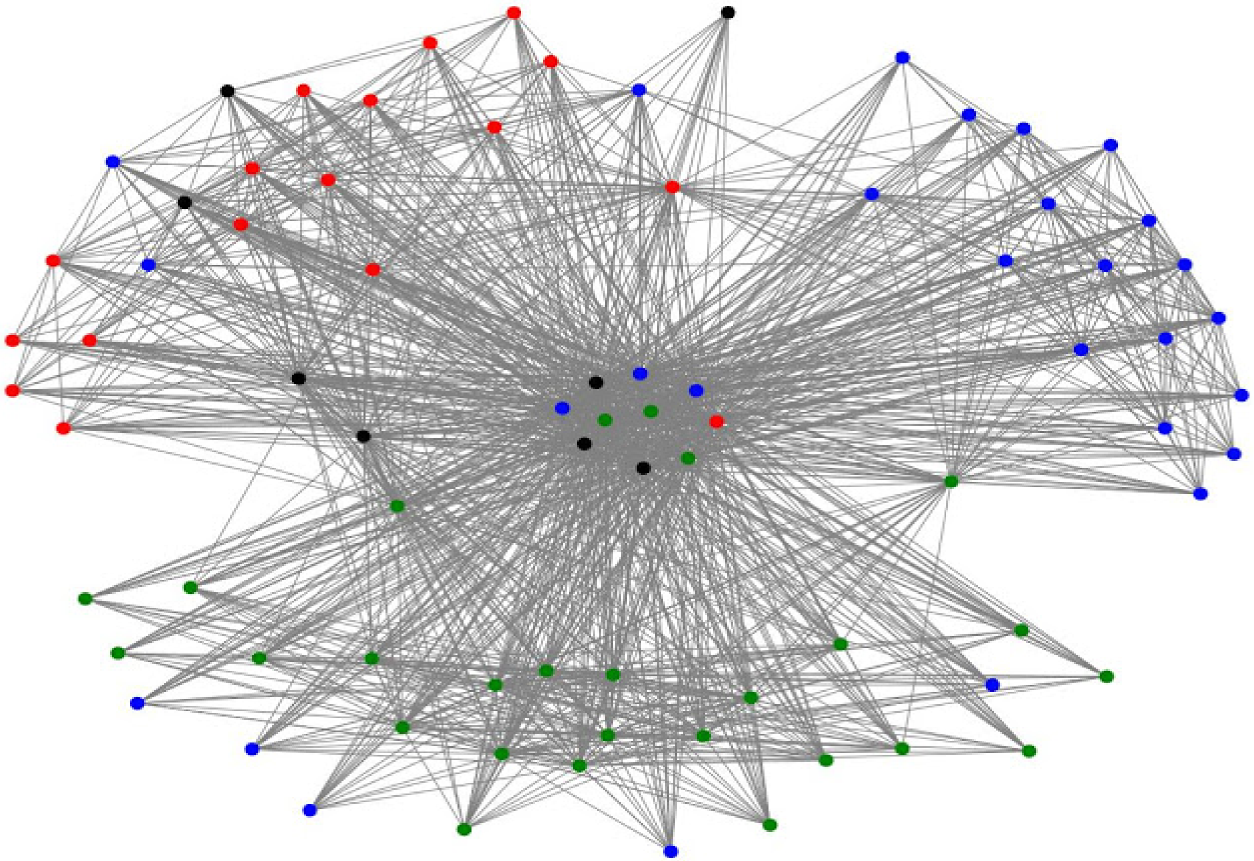
Communication Complexity for Distributed Graphs

Qin Zhang

Indiana University Bloomington

ADGA'16,
September 26, 2016

Graph data



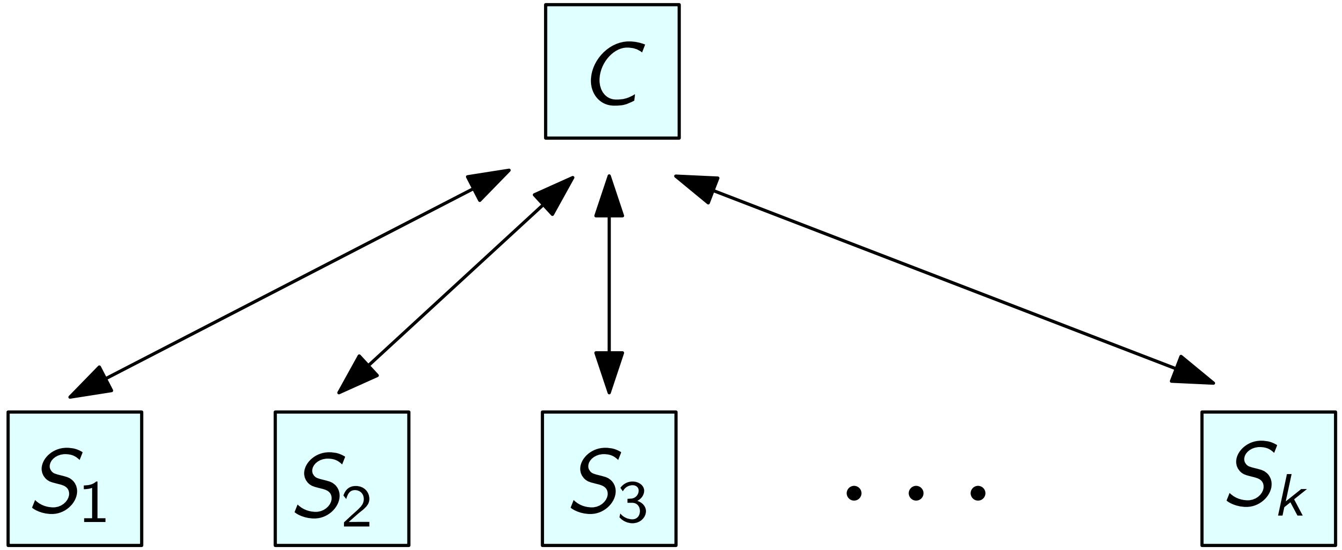
Massive graph, stored in multiple machines

Machines communicate (by message passing) with each other to answer queries

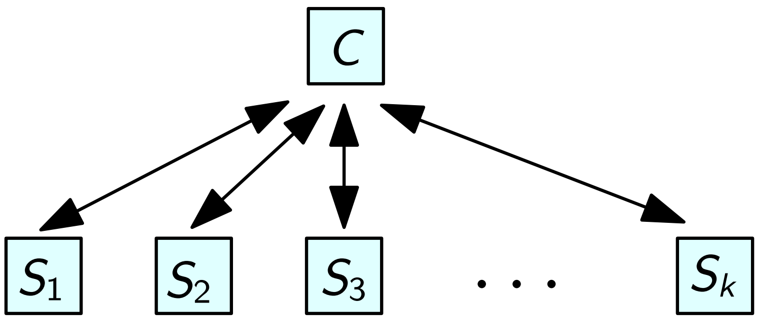
The coordinator model

The coordinator model: We have k machines (sites) and one central server (coordinator).

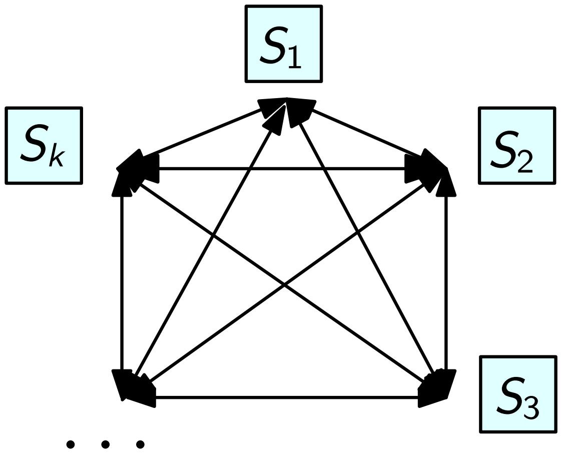
– Each site has a 2-way comm. channel with the coordinator.
– Each site has a piece of data x_i.
– Computation in rounds but no constraint on the message size
– **Task**: compute $f(x_1, \ldots, x_k)$ together via comm., for some f.
– **Goal**: minimize total communication



Coordinator VS k-machine



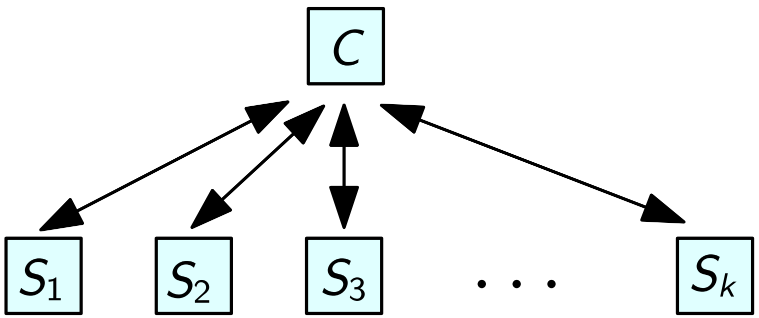
coordinator model



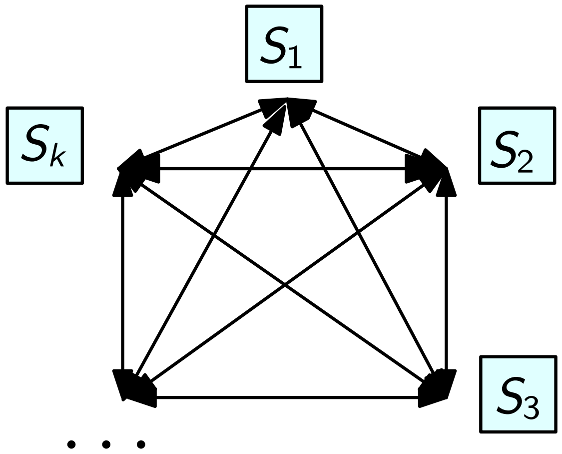
k-machine model

(Klauck, Nanongkai, Pandurangan, Robinson SODA 2015)

Coordinator VS k-machine



coordinator model



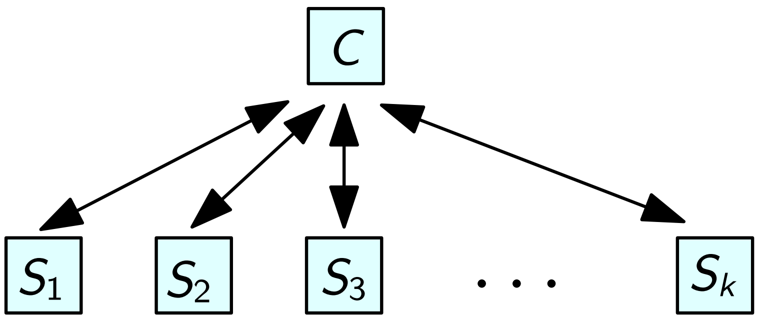
k-machine model

(Klauck, Nanongkai, Pandurangan, Robinson SODA 2015)

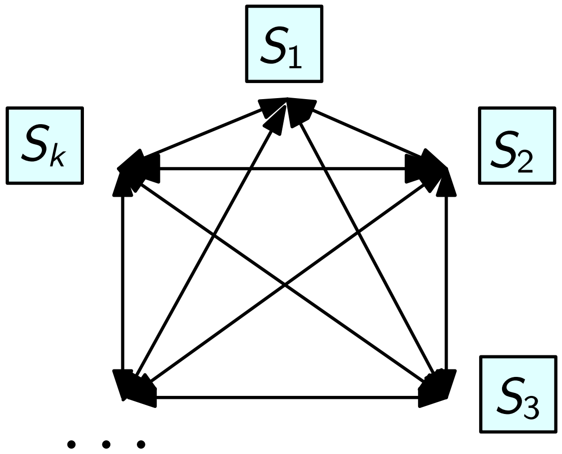
- The coordinator model has star comm. topology, k-machine has clique comm. topology,

  but they differ up to a factor of 2 in terms of total comm.

Coordinator VS k-machine



coordinator model



k-machine model

(Klauck, Nanongkai, Pandurangan, Robinson SODA 2015)

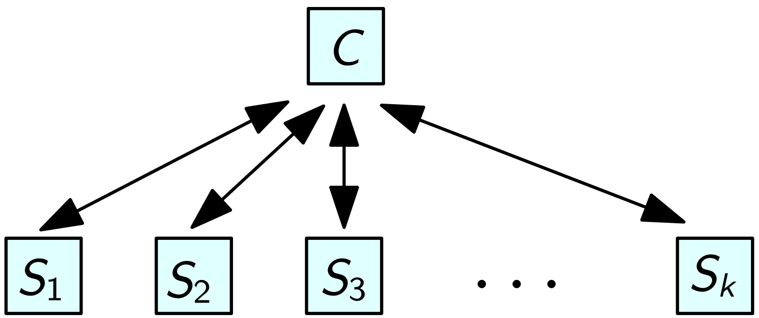
- The coordinator model has star comm. topology,
  k-machine has clique comm. topology,

  but they differ up to a factor of 2 in terms of total comm.

- The coordinator model focuses on the comm. complexity,
  k-machine focuses on the round complexity (or, time), given the bandwidth of each comm. channel B

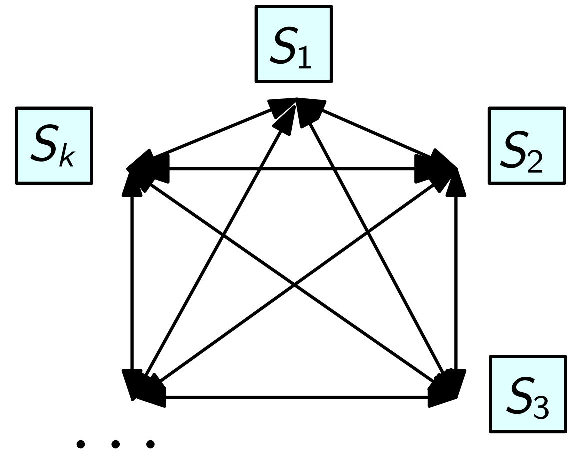
  but an $\Omega(C)$ comm. LB for coordinator also gives $\Omega(C/(k^2 \cdot B))$ round LB for k-machine

Coordinator VS *k*-machine



coordinator model

Today



k-machine model

(Klauck, Nanongkai, Pandurangan, Robinson SODA 2015)

- The coordinator model has star comm. topology,
  k-machine has clique comm. topology,

  but they differ up to a factor of 2 in terms of total comm.

- The coordinator model focuses on the comm. complexity,
  k-machine focuses on the round complexity (or, time), given the bandwidth of each comm. channel B

  but an $\Omega(C)$ comm. LB for coordinator also gives $\Omega(C/(k^2 \cdot B))$ round LB for *k*-machine

Input layout

- Edge partition: Edges are stored among the k sites (may allow duplications).

- Node partition: Nodes (together with all their adjacent edges) are partitioned among the k sites.

  Note: each edge is stored in two sites.

Input layout

- Edge partition: Edges are stored among the k sites (may allow duplications). **Today**

- Node partition: Nodes (together with all their adjacent edges) are partitioned among the k sites.

  Note: each edge is stored in two sites.

Case study 1: Connectivity – the value of input layout

- Connectivity: Test if a graph is connected.

Case study 1: Connectivity – the value of input layout

- Connectivity: Test if a graph is connected.

- For edge partition, we get a LB of $\Omega(kn/\log k)$ bits. (Woodruff and Zhang, DISC 2013) Will show today.

Case study 1: Connectivity – the value of input layout

- Connectivity: Test if a graph is connected.

- For edge partition, we get a LB of $\Omega(kn/\log k)$ bits. (Woodruff and Zhang, DISC 2013) Will show today.

- For node partition, a sketching algorithm by Ahn, Guha, McGregor (SODA 2012) uses $O(n \operatorname{poly}\log n)$ bits.

Case study 2: Diameter – the value of approximation

- Diameter: Compute the diameter of the graph.

Case study 2: Diameter – the value of approximation

- Diameter: Compute the diameter of the graph.

- (Implicitly by Braverman et al. FOCS 2013;
  edge partition with duplications):

  The comm. cost of exact computation is $\Omega(km)$ bits (m : #edges).

Case study 2: Diameter – the value of approximation

- Diameter: Compute the diameter of the graph.

- (Implicitly by Braverman et al. FOCS 2013; edge partition with duplications):

  The comm. cost of exact computation is $\Omega(km)$ bits (m : #edges).

- Exists an algorithm (a distributed implementation of an algo. by Dor, Halperin and Zwick, SICOMP 2000) with comm. $\tilde{O}(kn^{1.5})$ if an approx. of additive 2 is allowed.

Case study 3: Matching – just hard

- Matching: Compute the maximum matching of the graph.

Case study 3: Matching – just hard

- Matching: Compute the maximum matching of the graph.

- For any $\alpha \leq 1$, the comm. needed for computing a α-approximation is $\Omega(\alpha^2 nk)$.
  (Huang, Radunovic, Vojnovic and Zhang, STACS 2015)
  Almost node partition (bipartite graph; left nodes with their adjacent edges are partitioned). Will show today.

Case study 3: Matching – just hard

- Matching: Compute the maximum matching of the graph.

- For any $\alpha \leq 1$, the comm. needed for computing a α-approximation is $\Omega(\alpha^2 nk)$.
(Huang, Radunovic, Vojnovic and Zhang, STACS 2015)
Almost node partition (bipartite graph; left nodes with their adjacent edges are partitioned). Will show today.

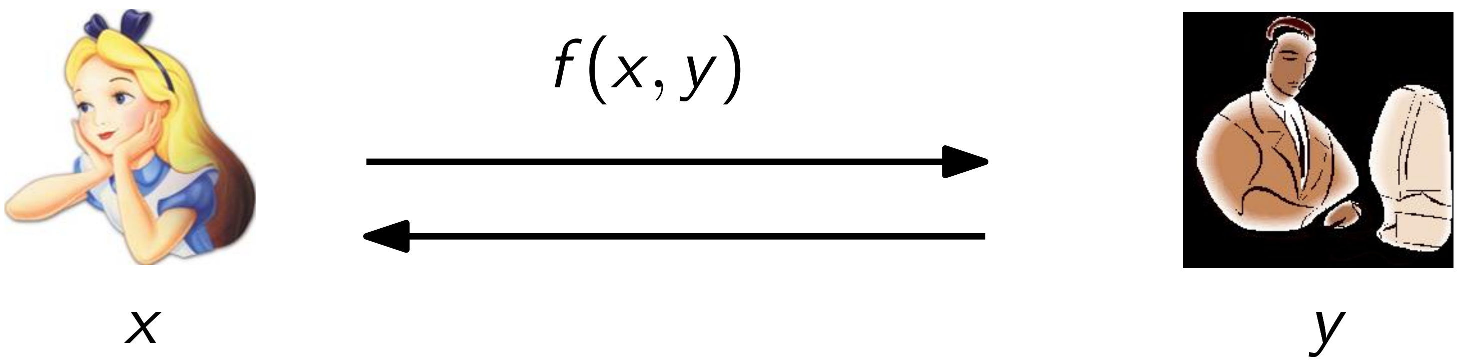
- There is a matching UB for any $\alpha \leq 1/2$

How to prove these LB results?

How to prove these LB results?

Using multi-party communication complexity

Basics on communication complexity



Can be easily extended to multiple players.

- $R^{\delta}(f)$: $\max_{x,y} |\Pi(x, y)|$, $|\Pi(x, y)|$ is the length of the transcript on input x, y. Π is randomized, and $\Pi(x, y) \neq f(x, y)$ w.pr. at most δ for any (x, y).
- $D^{\delta}_{\mu}(f)$: $\max_{x,y} |\Pi(x, y)|$. Π is deterministic, and $\Pi(x, y) \neq f(x, y)$ for at most a δ fraction of (x, y) under distribution μ.
- $ED^{\delta}_{\mu}(f)$: $\mathrm{E}_{(x,y)\sim\mu} |\Pi(x, y)|$. Π is deterministic, and $\Pi(x, y) \neq f(x, y)$ for at most a δ fraction of (x, y) under distribution μ.

Easy direction of Yao's Lemma: $R^{\delta}(f) \geq \max_{\mu} D^{\delta}_{\mu}(f)$.

A direct-sum type theorem in the coordinator model

$f : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$

μ is a distribution over $\mathcal{X} \times \mathcal{Y}$.

$f^k_{\text{OR}} : \mathcal{X}^k \times \mathcal{Y} \to \{0,1\}$ is the problem of computing $f(x_1, y) \vee f(x_2, y) \vee \ldots \vee f(x_k, y)$ in the coordinator model, where P_i has input $x_i \in \mathcal{X}$ for each $i \in [k]$, and the coordinator has $y \in \mathcal{Y}$.

ν is a distribution on $\mathcal{X}^k \times \mathcal{Y}$: First pick $(X_1, Y) \sim \mu$, and then pick $X_2, \ldots, X_k$ from the conditional distribution $\mu \mid Y$.

A direct-sum type theorem in the coordinator model

$f : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$

μ is a distribution over $\mathcal{X} \times \mathcal{Y}$.

$f^k_{\text{OR}} : \mathcal{X}^k \times \mathcal{Y} \to \{0,1\}$ is the problem of computing $f(x_1, y) \vee f(x_2, y) \vee \ldots \vee f(x_k, y)$ in the coordinator model, where P_i has input $x_i \in \mathcal{X}$ for each $i \in [k]$, and the coordinator has $y \in \mathcal{Y}$.

ν is a distribution on $\mathcal{X}^k \times \mathcal{Y}$: First pick $(X_1, Y) \sim \mu$, and then pick $X_2, \ldots, X_k$ from the conditional distribution $\mu \mid Y$.

Theorem (direct-sum). For any $f : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ and any distribution μ on $\mathcal{X} \times \mathcal{Y}$ for which $\mu(f^{-1}(1)) \leq 1/k^2$, we have $D_\nu^{1/k^3}(f^k_{\text{OR}}) = \Omega(k \cdot ED_\mu^{1/(100k^2)}(f))$. (will prove later)

2-DISJ

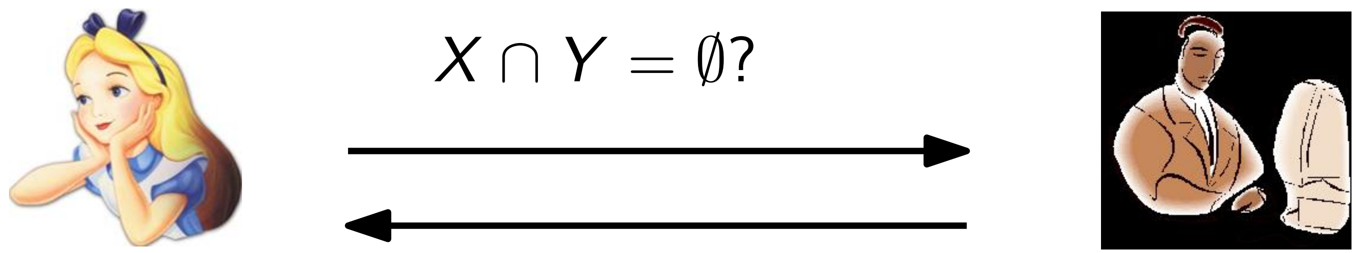
$X \cap Y = \emptyset$?

$X \subseteq \{1, \ldots, n\}$ $Y \subseteq \{1, \ldots, n\}$

Exists a hard distribution τ_β, under which
$|X \cap Y| = 1$ (YES instance) w.p. β and
$|X \cap Y| = 0$ (NO instance) w.p. $1 - \beta$.

Theorem. (Generalization of [Razborov '90, BJKS '04])
$ED_{\tau_\beta}^{\beta/100}(\text{2-DISJ}) = \Omega(n)$

2-DISJ



$X \subseteq \{1, \dots, n\}$ $\quad$ $Y \subseteq \{1, \dots, n\}$

Exists a hard distribution τ_β, under which
$|X \cap Y| = 1$ (YES instance) w.p. β and
$|X \cap Y| = 0$ (NO instance) w.p. $1 - \beta$.

Theorem. (Generalization of [Razborov '90, BJKS '04])
$ED_{\tau_\beta}^{\beta/100}(\text{2-DISJ}) = \Omega(n)$

Let $\mu = \tau_\beta$ with $\beta = 1/k^2$

LB for Connectivity

(Woodruff and Zhang, DISC 2013)

A meta-problem: THRESH

In the THRESH_θ^n problem, site P_i ($i \in [k]$) holds an n-bit vector $x_i = \{x_{i,1}, \ldots, x_{i,n}\}$, and the k sites want to compute

$$\text{THRESH}_\theta^n(x_1, \ldots, x_k) = \begin{cases} 0, & \text{if } \sum_{j \in [n]} (\vee_{i \in [k]} x_{i,j}) \leq \theta, \\ 1, & \text{if } \sum_{j \in [n]} (\vee_{i \in [k]} x_{i,j}) \geq \theta + 1. \end{cases}$$

A meta-problem: THRESH

In the THRESH_θ^n problem, site P_i ($i \in [k]$) holds an n-bit vector $x_i = \{x_{i,1}, \ldots, x_{i,n}\}$, and the k sites want to compute

$$\text{THRESH}_\theta^n(x_1, \ldots, x_k) = \begin{cases} 0, & \text{if } \sum_{j\in[n]}(\vee_{i\in[k]} x_{i,j}) \leq \theta, \\ 1, & \text{if } \sum_{j\in[n]}(\vee_{i\in[k]} x_{i,j}) \geq \theta + 1. \end{cases}$$

Theorem $\exists$ a θ and a distribution ζ, $D_\zeta^{1/k^4}(\text{THRESH}_\theta^n) = \Omega(kn)$.

Corollary $R^{1/3}(\text{THRESH}_\theta^n) = \Omega(kn/\log k)$

A meta-problem: THRESH

In the THRESH_θ^n problem, site P_i ($i \in [k]$) holds an n-bit vector $x_i = \{x_{i,1}, \ldots, x_{i,n}\}$, and the k sites want to compute

$$\text{THRESH}_\theta^n(x_1, \ldots, x_k) = \begin{cases} 0, & \text{if } \sum_{j \in [n]} (\vee_{i \in [k]} x_{i,j}) \le \theta, \\ 1, & \text{if } \sum_{j \in [n]} (\vee_{i \in [k]} x_{i,j}) \ge \theta + 1. \end{cases}$$

Theorem $\exists$ a θ and a distribution ζ, $D_\zeta^{1/k^4}(\text{THRESH}_\theta^n) = \Omega(kn)$.

Corollary $R^{1/3}(\text{THRESH}_\theta^n) = \Omega(kn/\log k)$

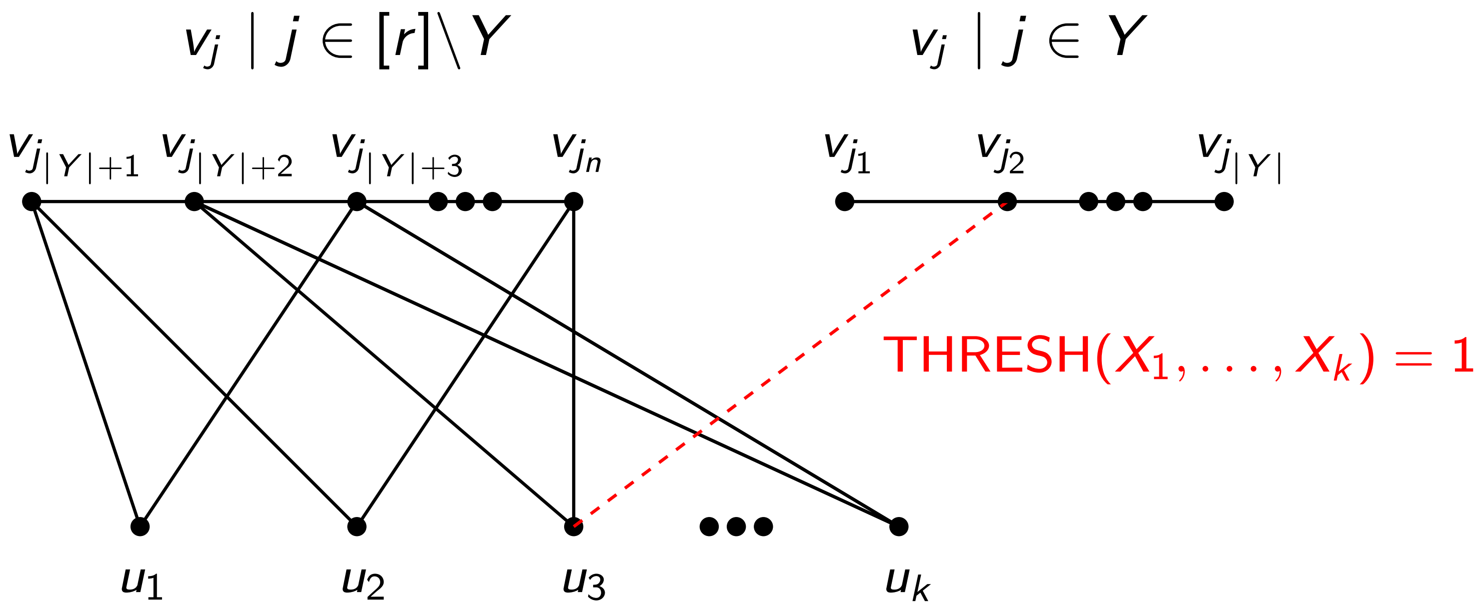
The proof framework:

1. Choose f to be 2-*DISJ* with input distribution μ,
   and denote f_{OR}^k by *OR-DISJ* and its distribution ν
   Apply direct-sum:
   $D_\nu^{1/k^3}(\text{OR-DISJ}) = \Omega(k \cdot ED_\mu^{1/(100k^2)}(\text{2-DISJ})) = \Omega(kn)$.
2. Show for $(X_1, \ldots, X_k, Y) \sim \nu$, whp,
   $\text{OR-DISJ}(X_1, \ldots, X_k, Y) = \text{THRESH}_\theta^n(X_1, \ldots, X_k)$ for some θ.

A reduction from THRESH to Connectivity

Reduction: an input $(X_1, \ldots, X_k)$ for THRESH $\Rightarrow$ a graph.

Each P_i creates an edge (u_i, v_j) for each $X_{i,j} = 1$. In addition, the coordinator reconstructs Y, and then creates a path containing $\{v_j \mid j \in Y\}$ and a path containing $\{v_j \mid j \in [r] \backslash Y\}$.



(u_i, v_j) exists (the graph is connected) if and only if $X_{i,j} = 1$

Other problems

Can prove LBs for a number of problems using similar reductions from THRESH. (Woodruff and Zhang, 2013)

- Cycle-freeness
- Bipartiteness
- Triangle-freeness
- #Connected components
- ...

Proof of the direct-sum theorem

$$D_{\nu}^{1/k^3}(f_{\mathrm{OR}}^k) = \Omega(k \cdot ED_{\mu}^{1/(100k^2)}(f))$$

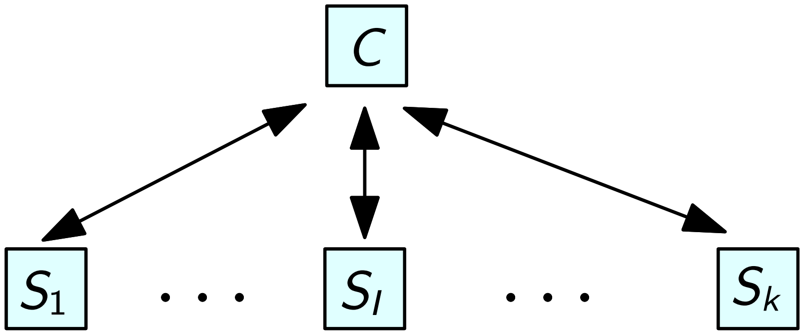
Proof of the direct-sum theorem

Theorem (direct-sum). For any $f : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ and any distribution μ on $\mathcal{X} \times \mathcal{Y}$ for which $\mu(f^{-1}(1)) \leq 1/k^2$, we have $D_{\nu}^{1/k^3}(f_{\mathrm{OR}}^{k}) = \Omega(k \cdot \mathrm{ED}_{\mu}^{1/(100k^2)}(f))$.

Proof of the direct-sum theorem

Theorem (direct-sum). For any $f : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ and any distribution μ on $\mathcal{X} \times \mathcal{Y}$ for which $\mu(f^{-1}(1)) \leq 1/k^2$, we have $D_{\nu}^{1/k^3}(f_{\mathrm{OR}}^k) = \Omega(k \cdot \mathrm{ED}_{\mu}^{1/(100k^2)}(f))$.

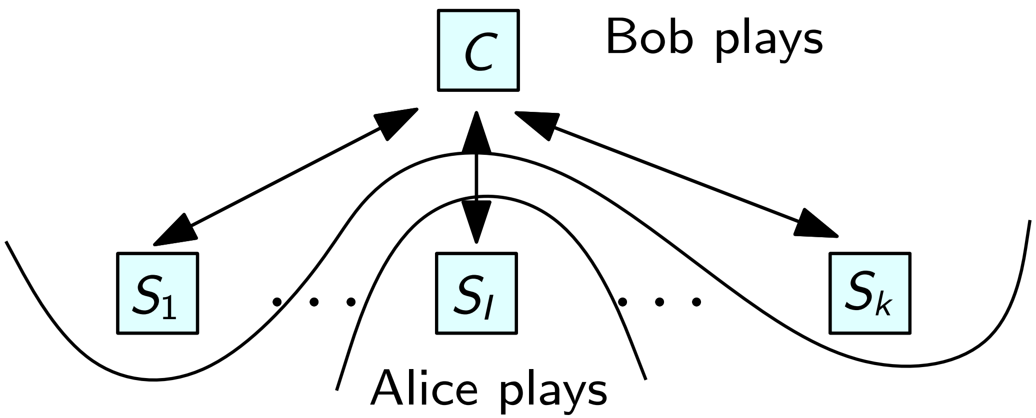
The proof is by a reduction from a 2-player problem to a k-site problem.



Proof of the direct-sum theorem

Theorem (direct-sum). For any $f : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ and any distribution μ on $\mathcal{X} \times \mathcal{Y}$ for which $\mu(f^{-1}(1)) \leq 1/k^2$, we have $D_{\nu}^{1/k^3}(f_{\mathrm{OR}}^k) = \Omega(k \cdot \mathrm{ED}_{\mu}^{1/(100k^2)}(f))$.

The proof is by a reduction from a 2-player problem to a k-site problem.



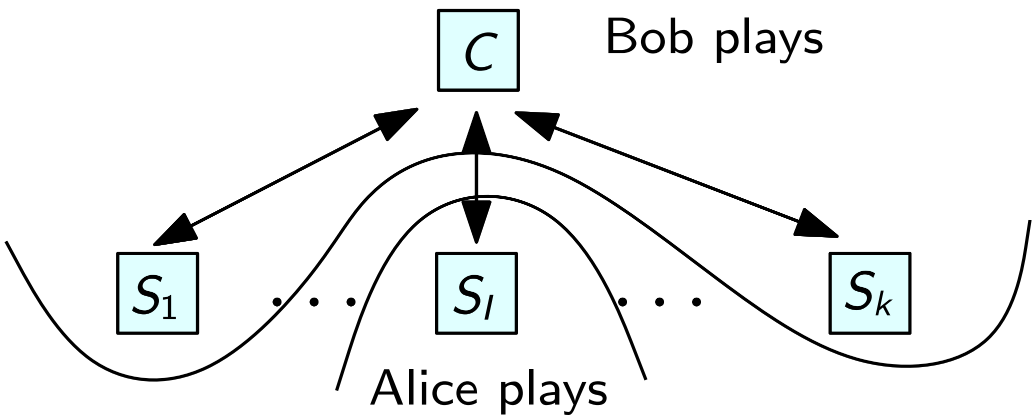
1. Alice and Bob have input $(X, Y) \sim \mu$ $(\mu(f^{-1}(1)) \leq 1/k^2)$

2. **Input reduction**: Alice picks a random site S_I and assigns it with input $X_I = X$. Bob plays the coordinator C and the rest $k-1$ sites. He assigns C with input Y, and S_i $(i \neq I)$ with input $X_i \sim \mu | Y$.

Proof of the direct-sum theorem

Theorem (direct-sum). For any $f : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$ and any distribution μ on $\mathcal{X} \times \mathcal{Y}$ for which $\mu(f^{-1}(1)) \le 1/k^2$, we have $D_\nu^{1/k^3}(f_{\mathrm{OR}}^k) = \Omega(k \cdot \mathrm{ED}_\mu^{1/(100k^2)}(f))$.

The proof is by a reduction from a 2-player problem to a k-site problem.



1. Alice and Bob have input $(X, Y) \sim \mu$ ($\mu(f^{-1}(1)) \le 1/k^2$)

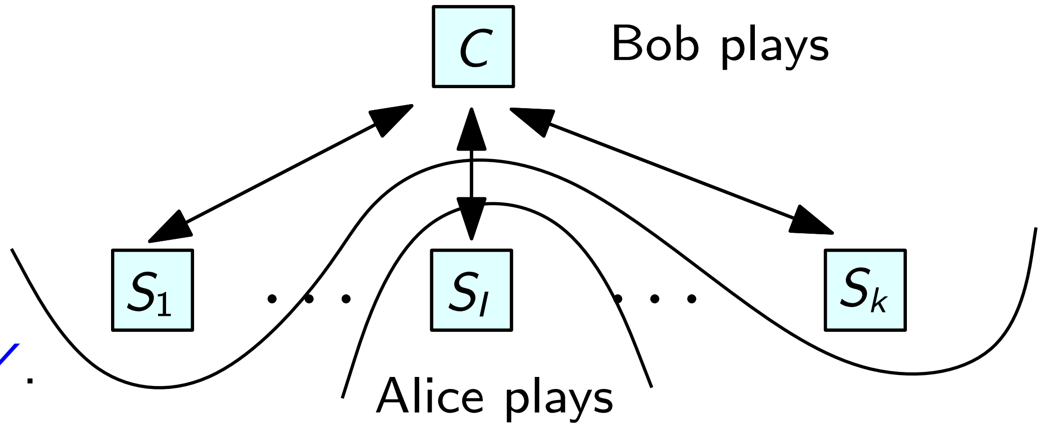
2. **Input reduction**: Alice picks a random site S_I and assigns it with input $X_I = X$. Bob plays the coordinator C and the rest $k - 1$ sites. He assigns C with input Y, and S_i ($i \ne I$) with input $X_i \sim \mu | Y$.

3. They run a protocol for f_{OR}^k, w.pr. $1 - \frac{1}{k}$, $f(X_i, Y) = 0$ for all $i \ne I$, thus $f_{\mathrm{OR}}^k(X_1, \ldots, X_k, Y) = f(X_1, Y) \vee \ldots \vee f(X_k, Y) = f(X_I, Y) = f(X, Y)$.

Proof of the direct-sum theorem (cont.)

1. Alice and Bob have input $(X, Y) \sim \mu$ ($\mu(f^{-1}(1)) \leq 1/k^2$)

2. **Input reduction**: Alice picks a random site S_I and assigns it with input $X_I = X$. Bob plays the coordinator C and the rest $k-1$ sites. He assigns C with input Y, and S_i ($i \neq I$) with input $X_i \sim \mu|Y$.



3. They run a protocol for f_{OR}^k, w.pr. $1 - \frac{1}{k}$, $f(X_i, Y) = 0$ for all $i \neq I$, thus $f_{\mathrm{OR}}^k(X_1, \ldots, X_k, Y) = f(X_1, Y) \vee \ldots \vee f(X_k, Y) = f(X_I, Y) = f(X, Y)$.

4. They repeat the input reduction 3 times (using the same (X, Y)) and run the protocol for f_{OR}^k on each input, the probability that at least in one run (which Bob knows), $f(X_i, Y) = 0$ for all $i \neq I$, is $1 - 1/k^3$.

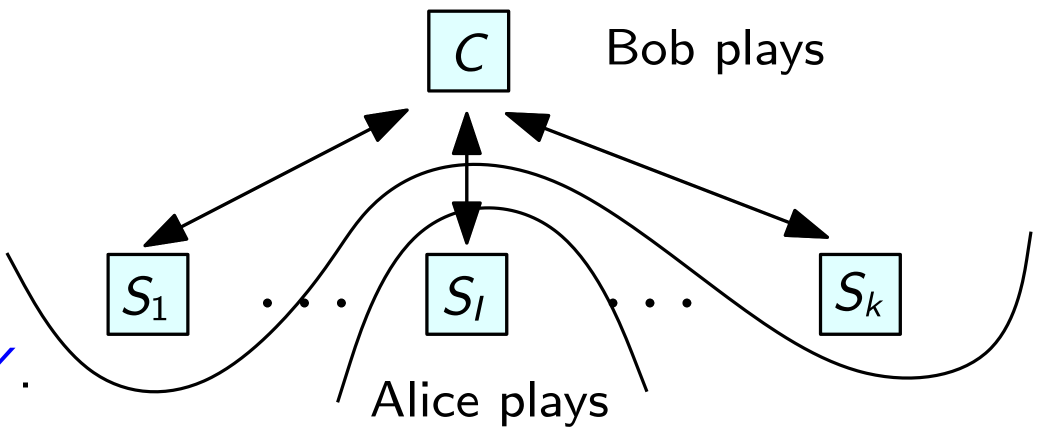
Plus the error prob. of each run is at most $1/k^3$, we get a protocol for f under input dist. μ that succeeds w.pr. $O(1/k^3) \leq 1/(100k^2)$.

Proof of the direct-sum theorem (cont.)

1. Alice and Bob have input $(X, Y) \sim \mu$ $(\mu(f^{-1}(1)) \le 1/k^2)$

$\mathrm{E}[\mathrm{CC}(\text{Alice, Bob})] = \frac{1}{k}\,\mathrm{CC}(k \text{ sites})$

2. **Input reduction**: Alice picks a random site S_I and assigns it with input $X_I = X$. Bob plays the coordinator C and the rest $k-1$ sites. He assigns C with input Y, and S_i $(i \neq I)$ with input $X_i \sim \mu|Y$.



3. They run a protocol for f_{OR}^k, w.pr. $1 - \frac{1}{k}$, $f(X_i, Y) = 0$ for all $i \neq I$, thus $f_{\mathrm{OR}}^k(X_1, \ldots, X_k, Y) = f(X_1, Y) \vee \ldots \vee f(X_k, Y) = f(X_I, Y) = f(X, Y)$.

4. They repeat the input reduction 3 times (using the same (X, Y)) and run the protocol for f_{OR}^k on each input, the probability that at least in one run (which Bob knows), $f(X_i, Y) = 0$ for all $i \neq I$, is $1 - 1/k^3$.

Plus the error prob. of each run is at most $1/k^3$, we get a protocol for f under input dist. μ that succeeds w.pr. $O(1/k^3) \le 1/(100k^2)$.

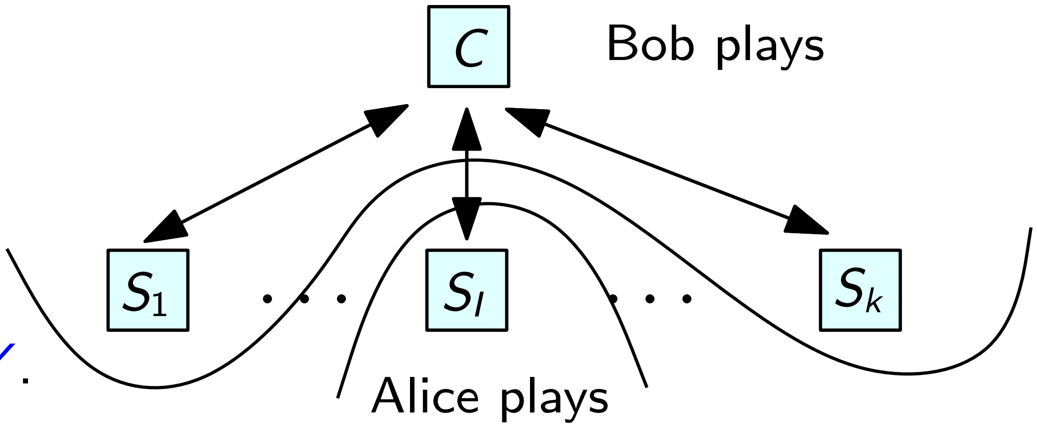
Proof of the direct-sum theorem (cont.)

1. Alice and Bob have input $(X, Y) \sim \mu$ ($\mu(f^{-1}(1)) \leq 1/k^2$)

$E[CC(\text{Alice, Bob})] = \frac{1}{k}$ CC(k sites)

$ED_{\mu}^{1/(100k^2)}(f) = C$

2. **Input reduction**: Alice picks a random site S_I and assigns it with input $X_I = X$. Bob plays the coordinator C and the rest $k-1$ sites. He assigns C with input Y, and S_i ($i \neq I$) with input $X_i \sim \mu | Y$.



3. They run a protocol for f_{OR}^k, w.pr. $1 - \frac{1}{k}$, $f(X_i, Y) = 0$ for all $i \neq I$, thus $f_{\text{OR}}^k(X_1, \ldots, X_k, Y) = f(X_1, Y) \vee \ldots \vee f(X_k, Y) = f(X_I, Y) = f(X, Y)$.

4. They repeat the input reduction 3 times (using the same (X, Y)) and run the protocol for f_{OR}^k on each input, the probability that at least in one run (which Bob knows), $f(X_i, Y) = 0$ for all $i \neq I$, is $1 - 1/k^3$.

Plus the error prob. of each run is at most $1/k^3$, we get a protocol for f under input dist. μ that succeeds w.pr. $O(1/k^3) \leq 1/(100k^2)$.

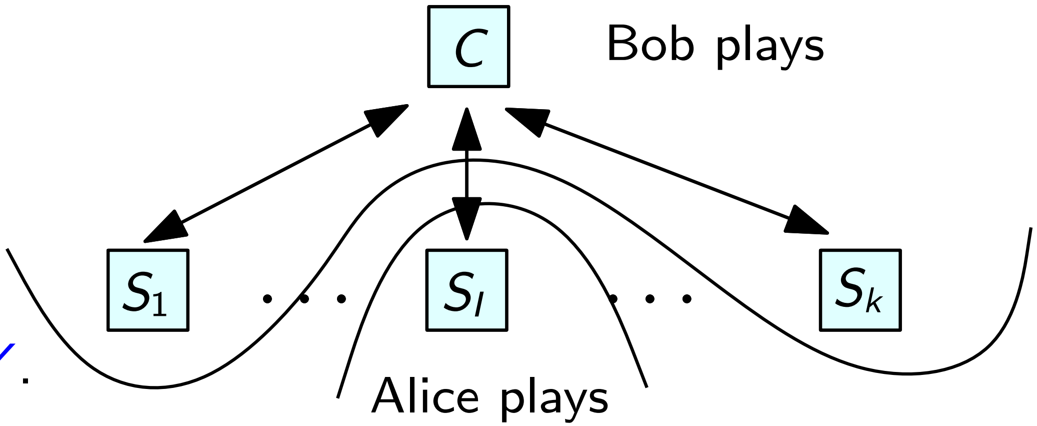
Proof of the direct-sum theorem (cont.)

1. Alice and Bob have input $(X, Y) \sim \mu$ $(\mu(f^{-1}(1)) \leq 1/k^2)$

2. **Input reduction**: Alice picks a random site S_I and assigns it with input $X_I = X$. Bob plays the coordinator C and the rest $k-1$ sites. He assigns C with input Y, and S_i $(i \neq I)$ with input $X_i \sim \mu | Y$.

E[CC(Alice, Bob)] $= \frac{1}{k}$ CC(k sites)

$ED_\mu^{1/(100k^2)}(f) = C \quad D_\nu^{1/k^3}(f_{\mathrm{OR}}^k) = \Omega(kC)$



3. They run a protocol for f_{OR}^k, w.pr. $1 - \frac{1}{k}$, $f(X_i, Y) = 0$ for all $i \neq I$, thus $f_{\mathrm{OR}}^k(X_1, \ldots, X_k, Y) = f(X_1, Y) \vee \ldots \vee f(X_k, Y) = f(X_I, Y) = f(X, Y)$.

4. They repeat the input reduction 3 times (using the same (X, Y)) and run the protocol for f_{OR}^k on each input, the probability that at least in one run (which Bob knows), $f(X_i, Y) = 0$ for all $i \neq I$, is $1 - 1/k^3$.

Plus the error prob. of each run is at most $1/k^3$, we get a protocol for f under input dist. μ that succeeds w.pr. $O(1/k^3) \leq 1/(100k^2)$.

LB for Matching

(Huang, Radunovic, Vojnovic and Zhang, STACS 2015)

Present a "fake" proof to show the main ideas.
Assume the approximation α is a constant.

The hard input graph

How does the hard input graph look like?

- Large set of “noisy” edges,
  but form a small matching

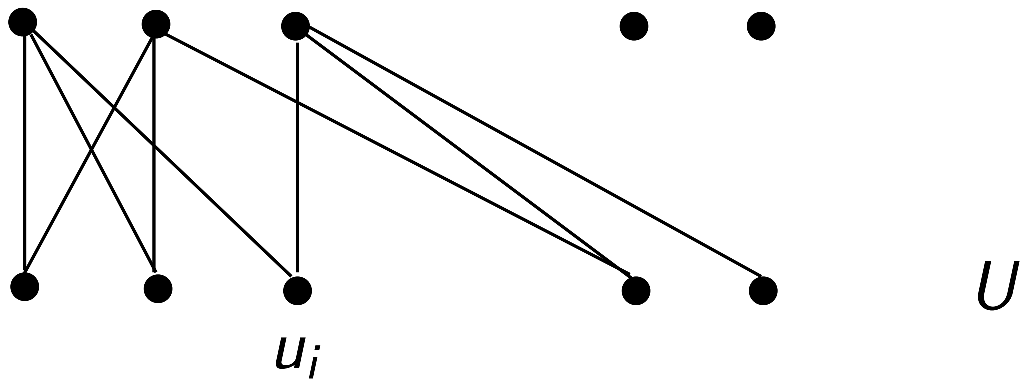
- Small set of “important” edges,
  but form a large matching

The hard input graph (cont.)

- Consider a $2n$-vertex bipartite graph $G = (U, V, E)$
  (assume $k = n$; general case discussed later)

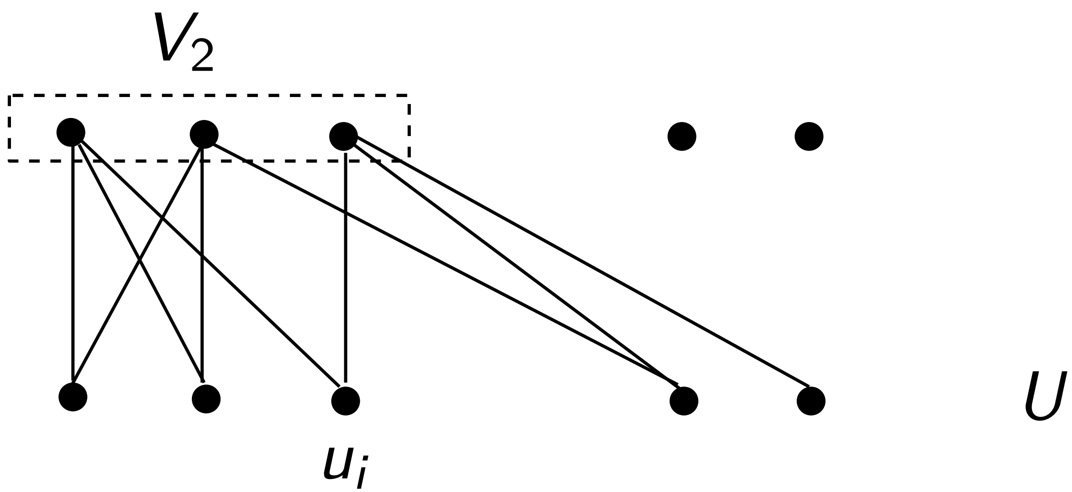
The hard input graph (cont.)

- Consider a $2n$-vertex bipartite graph $G = (U, V, E)$
  (assume $k = n$; general case discussed later)
- Player i gets the edges incident to u_i



The hard input graph (cont.)

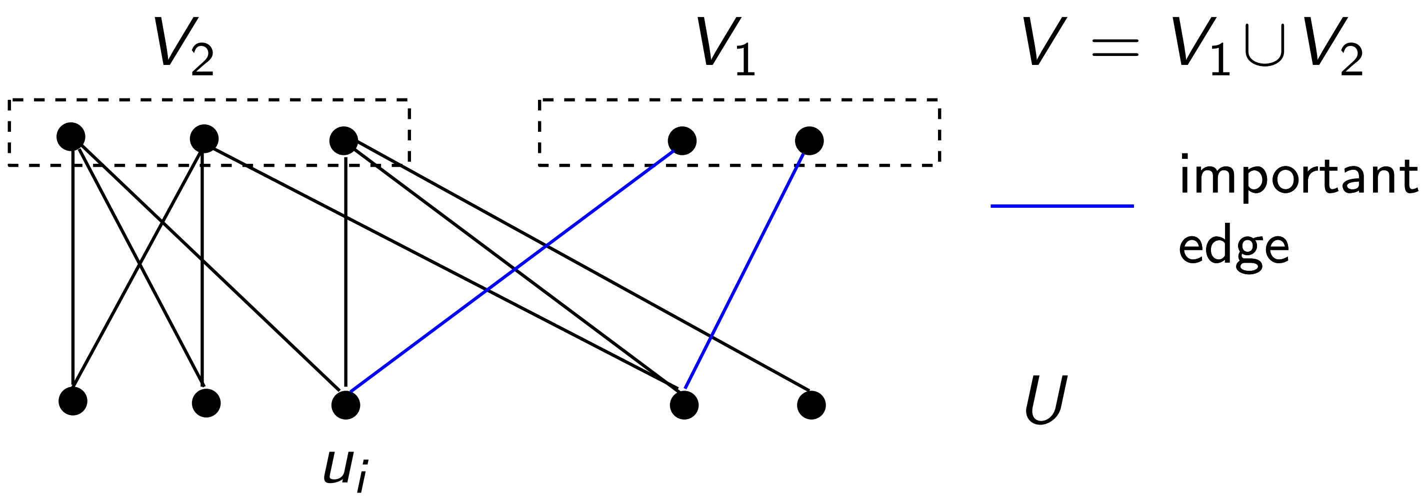
- Consider a $2n$-vertex bipartite graph $G = (U, V, E)$
  (assume $k = n$; general case discussed later)
- Player i gets the edges incident to u_i



Edges between U and V_2 are noisy edges

The hard input graph (cont.)

- Consider a $2n$-vertex bipartite graph $G = (U, V, E)$
  (assume $k = n$; general case discussed later)
- Player i gets the edges incident to u_i

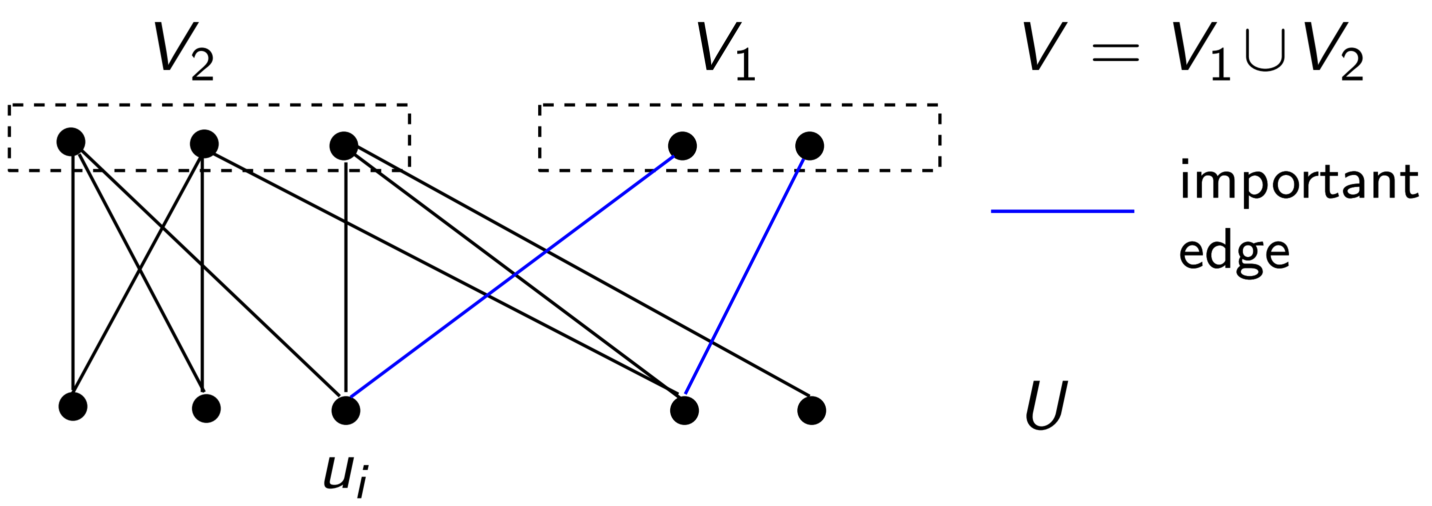


Edges between U and V_2 are noisy edges

Edges between U and V_1 are important edges

The hard input graph (cont.)

- Consider a $2n$-vertex bipartite graph $G = (U, V, E)$
  (assume $k = n$; general case discussed later)
- Player i gets the edges incident to u_i



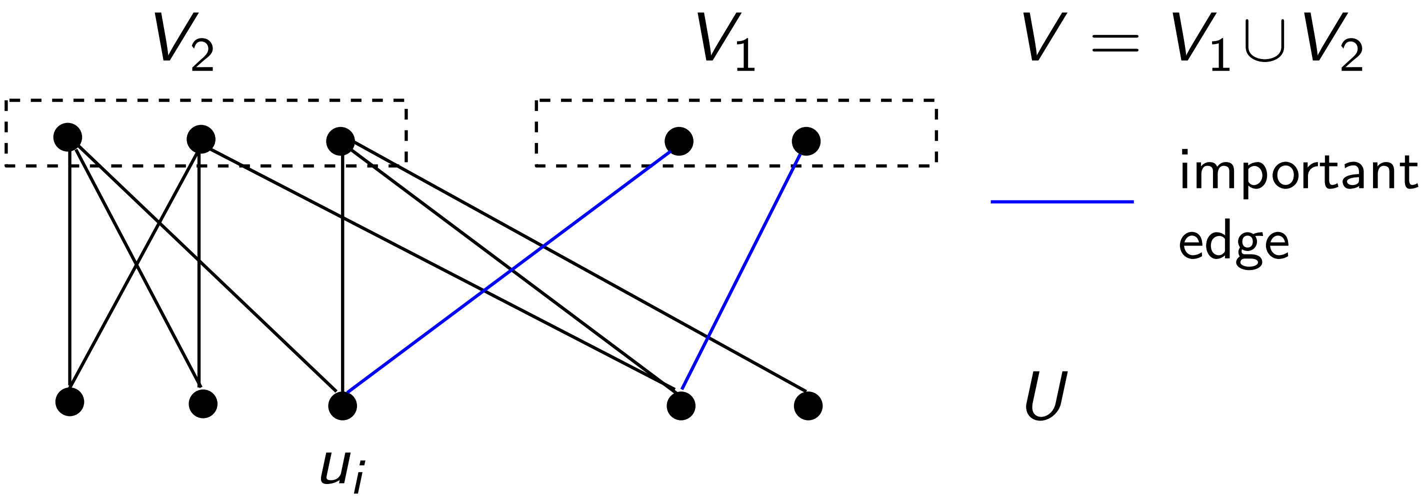
Edges between U and V_2 are noisy edges

Edges between U and V_1 are important edges

Say $|V_1| = 99\,|V_2|$

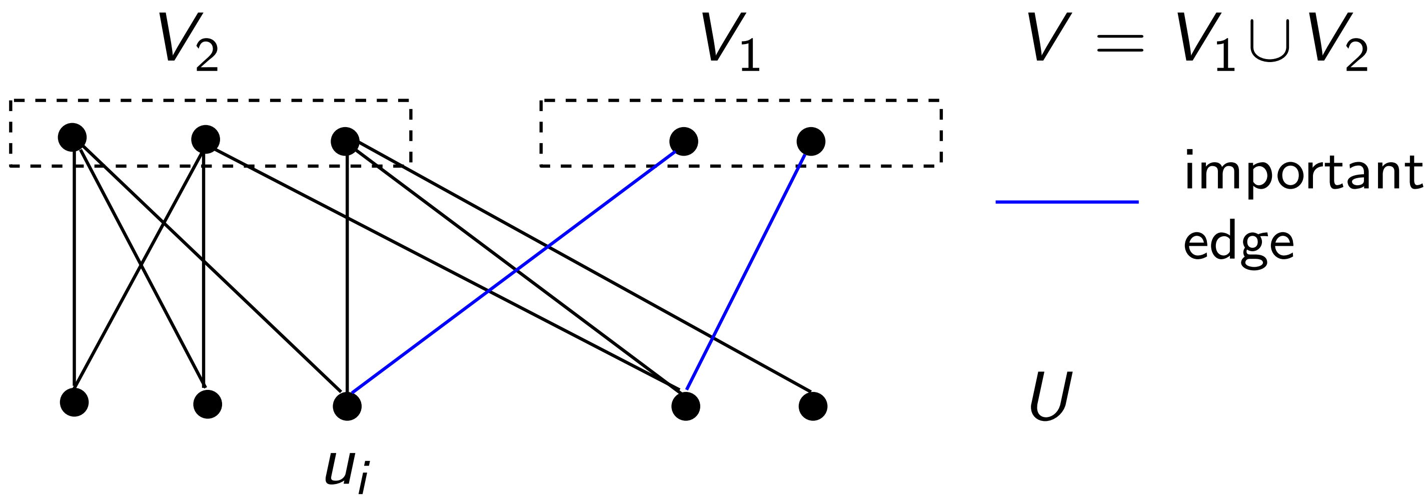
The encoding of the graph

- Use $y \in \{0,1\}^n$ to encode V



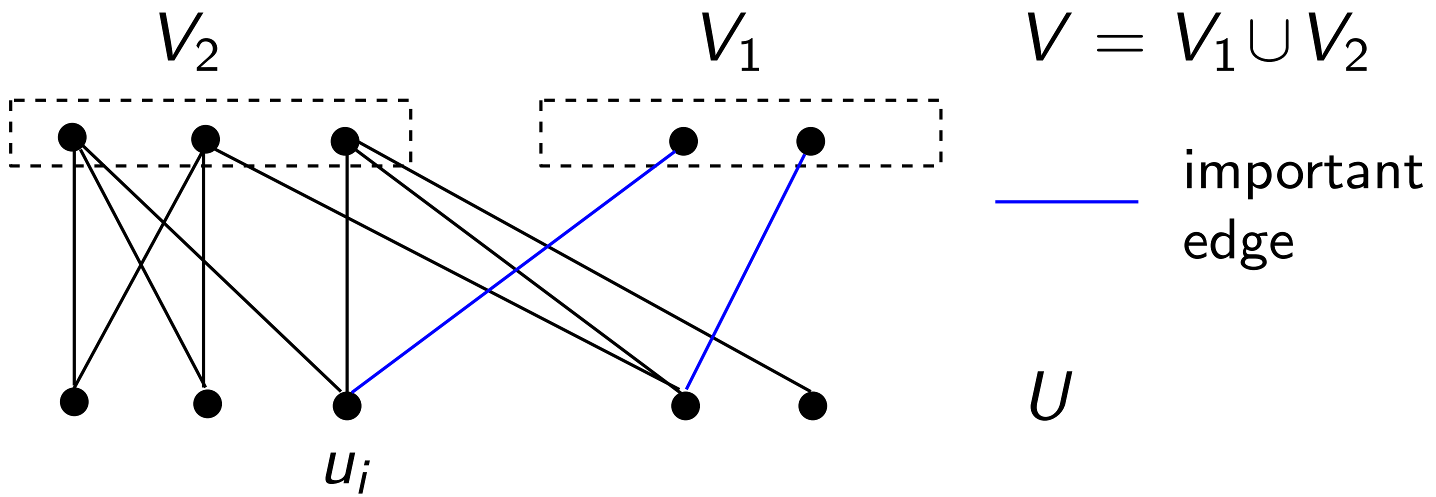
The encoding of the graph

- Use $y \in \{0,1\}^n$ to encode V
- Use $x_i \in \{0,1\}^n$ to encode the neighbors of u_i



The encoding of the graph

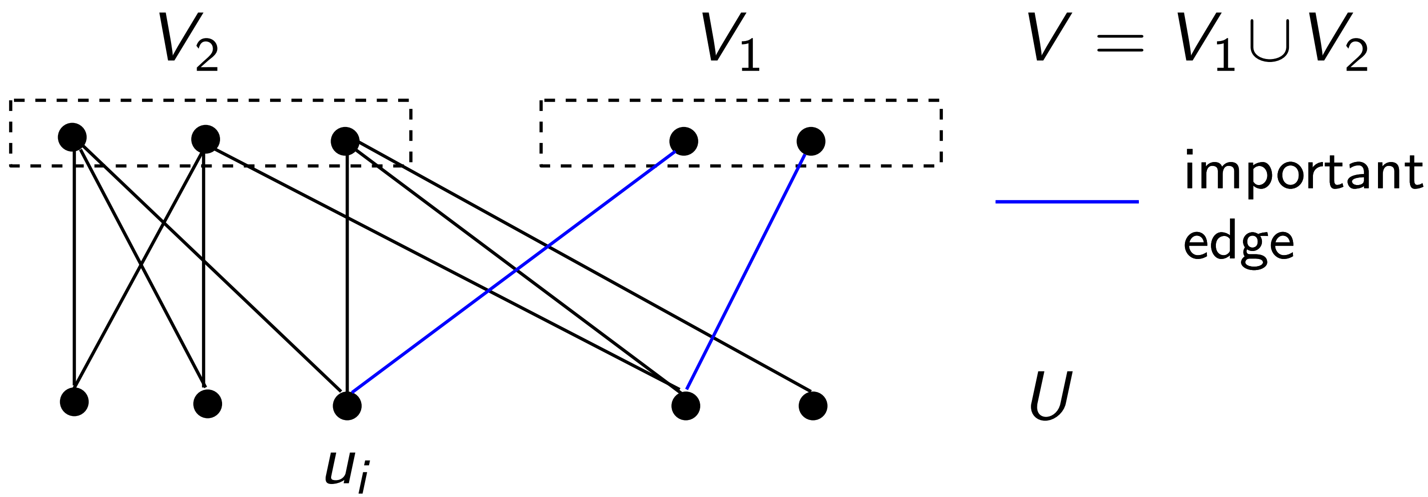
- Use $y \in \{0,1\}^n$ to encode V
- Use $x_i \in \{0,1\}^n$ to encode the neighbors of u_i



Set each $y_j = 0/1$ w.pr. $1/2$. For each i,
if $y_j = 0$ then set $x_{i,j} = 0/1$ w.pr. $1/2$; else if $y_1 = 1$ then set $x_{i,j} = 0$

The encoding of the graph

- Use $y \in \{0,1\}^n$ to encode V
- Use $x_i \in \{0,1\}^n$ to encode the neighbors of u_i

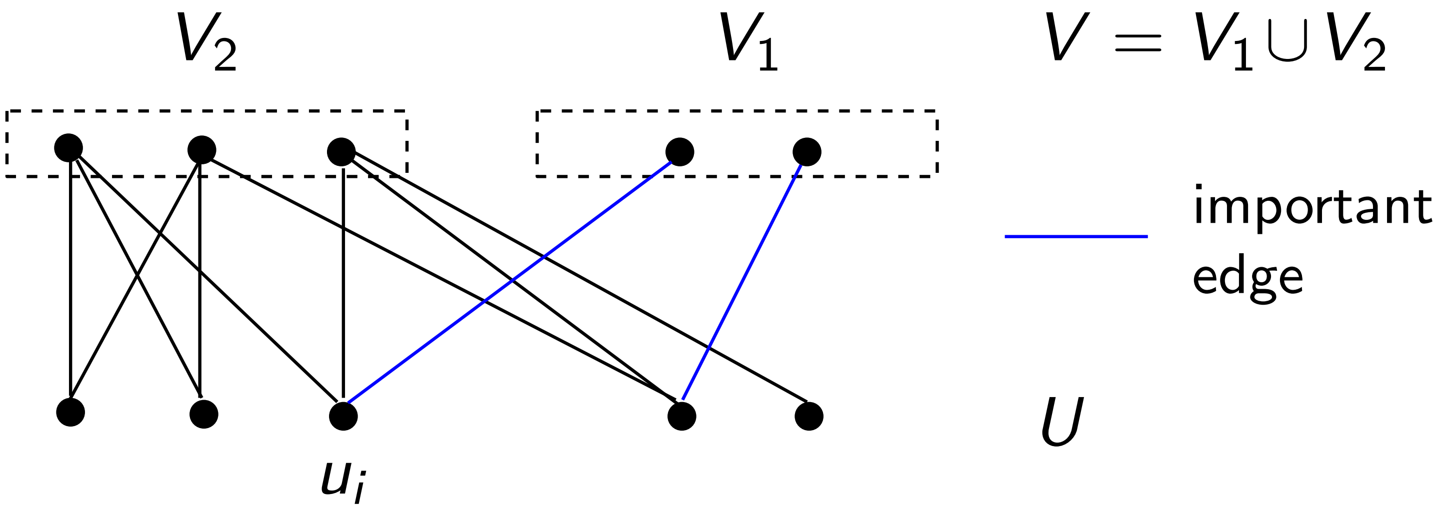


Set each $y_j = 0/1$ w.pr. $1/2$. For each i,
if $y_j = 0$ then set $x_{i,j} = 0/1$ w.pr. $1/2$; else if $y_1 = 1$ then set $x_{i,j} = 0$

For each i, selet a random J s.t. $y_J = 1$, and reset $x_{i,J} = 0/1$ w.pr. $1/2$

The relation to 2-DISJ

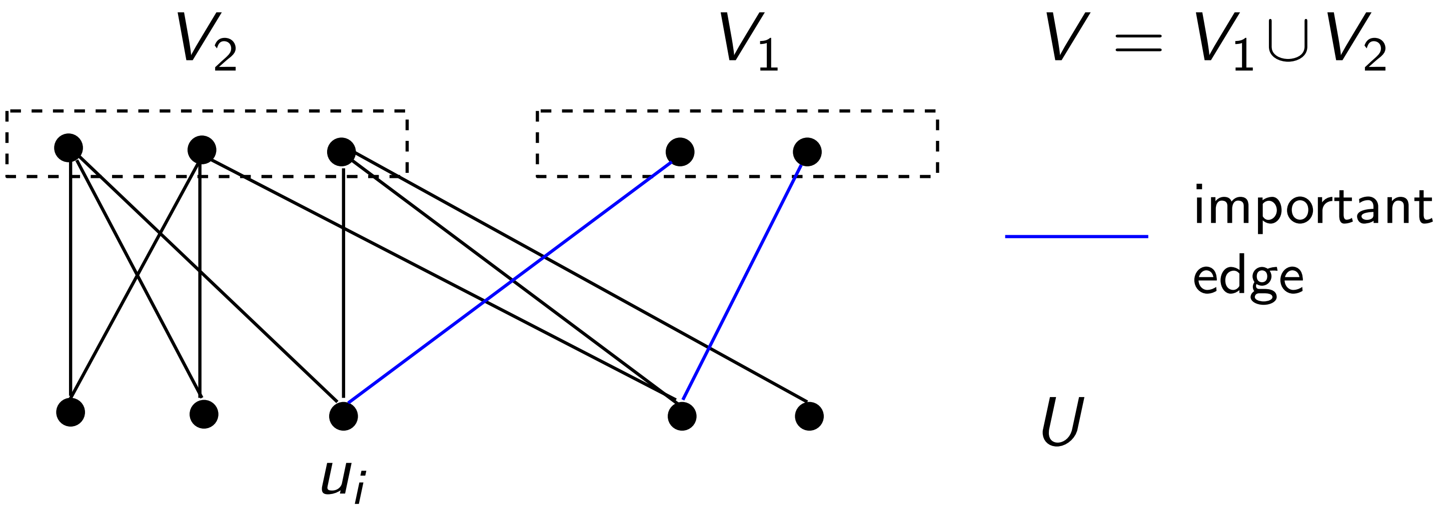
- Consider each pair (y, x_i)



The relation to 2-DISJ

- Consider each pair (y, x_i)

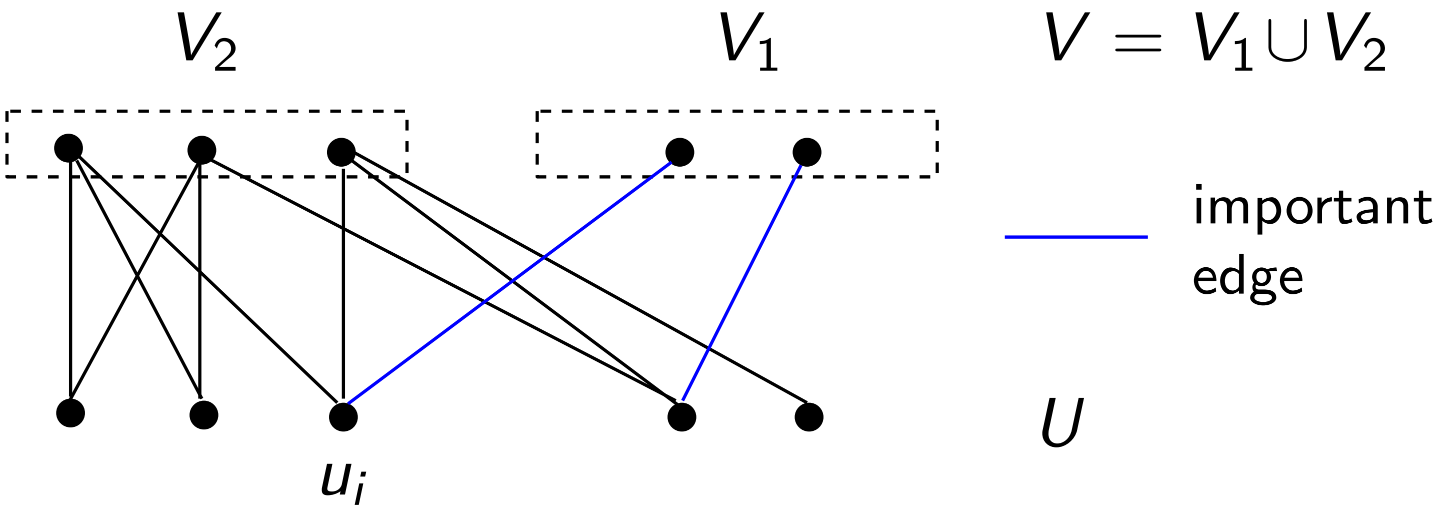
  Form a 2-DISJ instance with a hard input distribution (slightly different from the one used for connectivity)



The relation to 2-DISJ

- Consider each pair (y, x_i)

  Form a 2-DISJ instance with a hard input distribution (slightly different from the one used for connectivity)

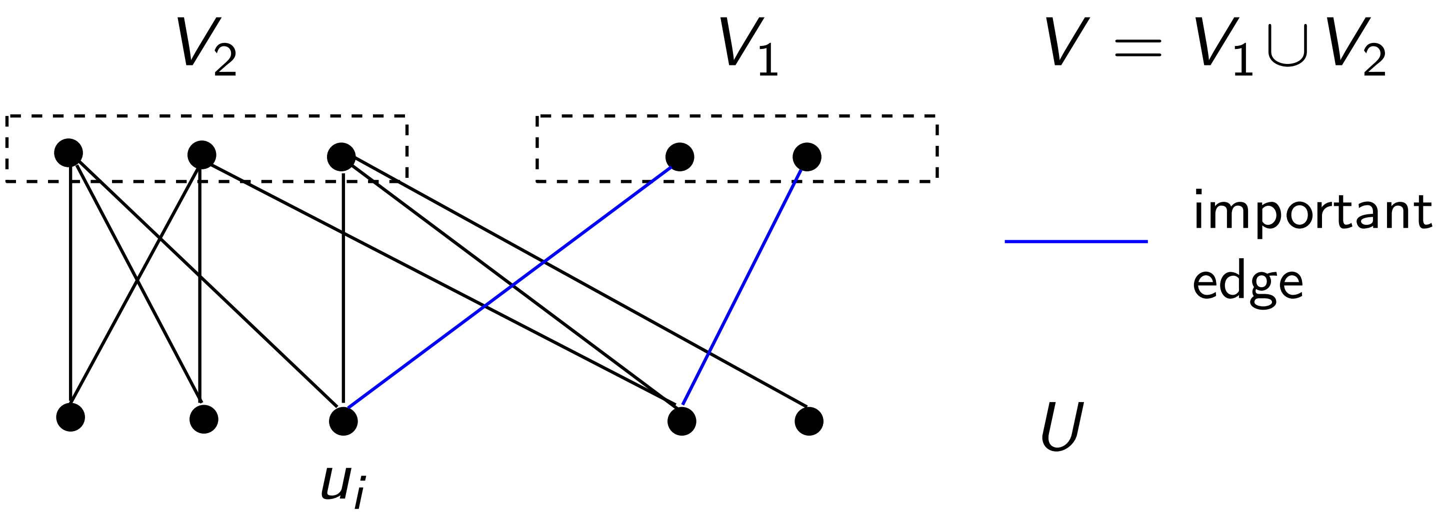


If an important edge of u_i is discovered when computing max matching, then y and x_i have a common element.

The relation to 2-DISJ

- Consider each pair (y, x_i)

  Form a 2-DISJ instance with a hard input distribution (slightly different from the one used for connectivity)



If an important edge of u_i is discovered when computing max matching, then y and x_i have a common element.

Proof ideas: Find a large matching $\rightarrow$ recover $\Omega(n)$ important edges $\rightarrow$ solve $\Omega(n)$ instances of 2-DISJ $\rightarrow$ $\Omega(n^2)$ LB

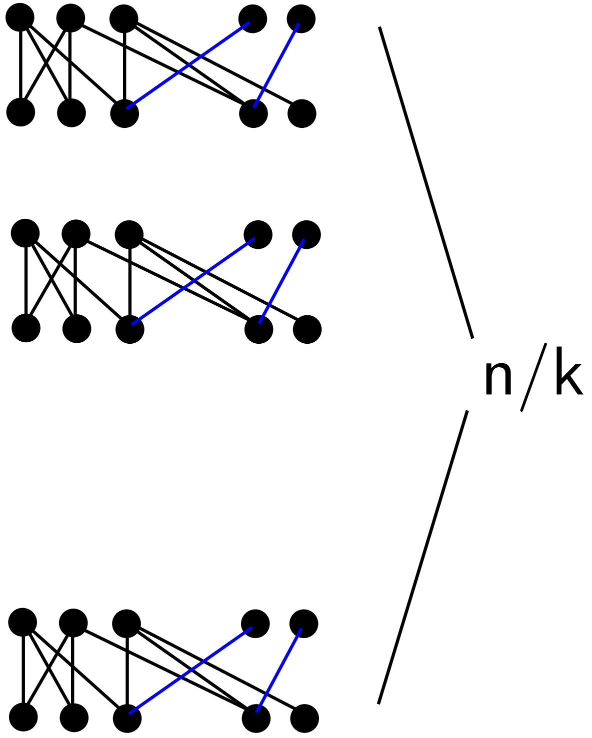
General k

- For general $k \leq n$

General k

- For general $k \leq n$

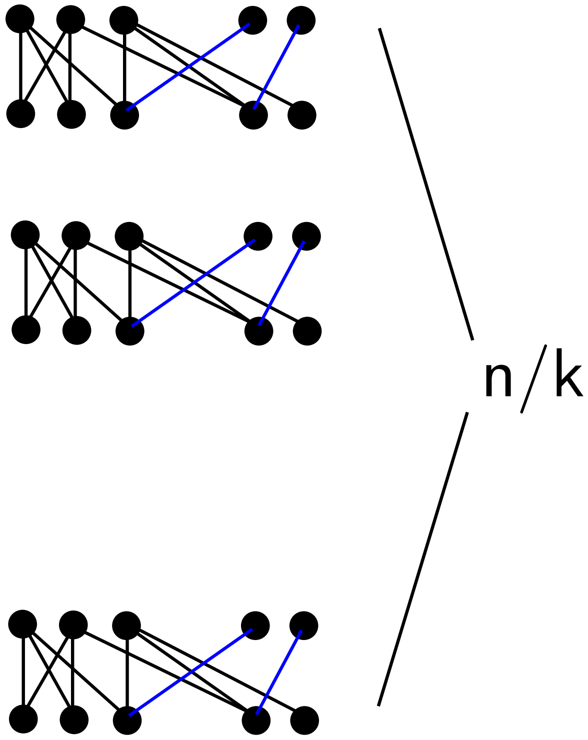
  make n/k independent instances of size k of the previous hard instance



General k

- For general $k \leq n$

  make n/k independent instances of size k of the previous hard instance

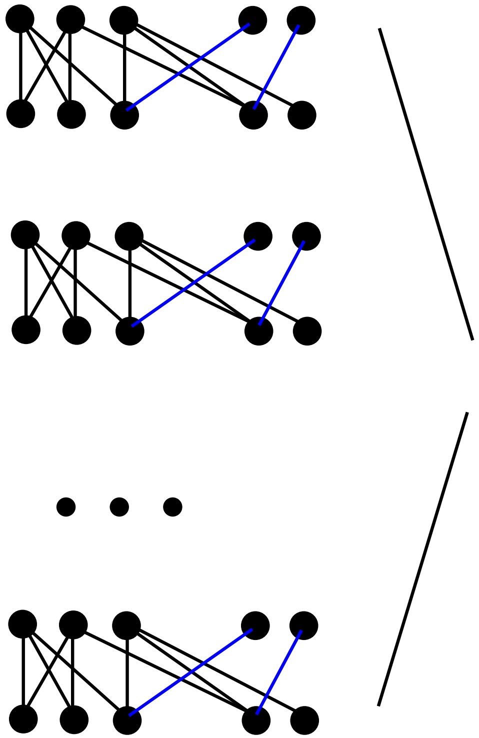


The cost of each instance is $\Omega(k^2)$

General k

- For general $k \leq n$

  make n/k independent instances of size k of the previous hard instance



n/k

The cost of each instance is $\Omega(k^2)$

The total cost is $\Omega(nk)$
(direct-sum using information cost)

Related Work and Future Direction

Related work

- Round LBs for a set of basic graph problems have been proved in the k-machine model (node partition)

  Work for problems with large output size; cannot be used for decision-type problems

  - Distributed Computation of Large-scale Graph Problems
    by Klauck, Nanongkai, Pandurangan and Robinson, SODA 2015
  - Tight Bounds for Distributed Graph Computations
    by Pandurangan, Robinson and Scquizzato, CoRR 2016

Related work

- Round LBs for a set of basic graph problems have been proved in the k-machine model (node partition)

  Work for problems with large output size; cannot be used for decision-type problems

  - Distributed Computation of Large-scale Graph Problems
    by Klauck, Nanongkai, Pandurangan and Robinson, SODA 2015
  - Tight Bounds for Distributed Graph Computations
    by Pandurangan, Robinson and Scquizzato, CoRR 2016

- MultiCC on general comm. topology (not yet for graph problems)

  - Topology Matters in Communication
    by Chattopadhyay, Radhakrishnan, and Rudra, FOCS 2014
  - The Range of Topological Effects on Communication
    by Chattopadhyay and Rudra, ICALP 2015

Future directions

- The complexities of many graph problems are still unknown in the coordinator model.

- For the node-partition model, lower bounds for decision-type problems, e.g., triangle counting, *size* of the max matching, are not known.

  Challenge: input sharing. Each edge is stored in two machines. May need new techniques.

- Techniques for proving round complexities in the k-machine model are still limited.

  Current approaches:
  – (total comm.)/(total network bandwidth)
  – (info. a particular machine needs)/(single link bandwidth)

  Some problems (matching?) may have higher round complexities

Thank you!

Questions?