Distributed Graph Algorithms in the Vertex-Partition Model

Peter Robinson

Augusta University

perobinson@augusta.edu

Joint work with:

Christian Konrad, Gopal Pandurangan, Sriram Pemmaraju, Talal Riaz, Michele Scquizzato, Viktor Zamaraev,

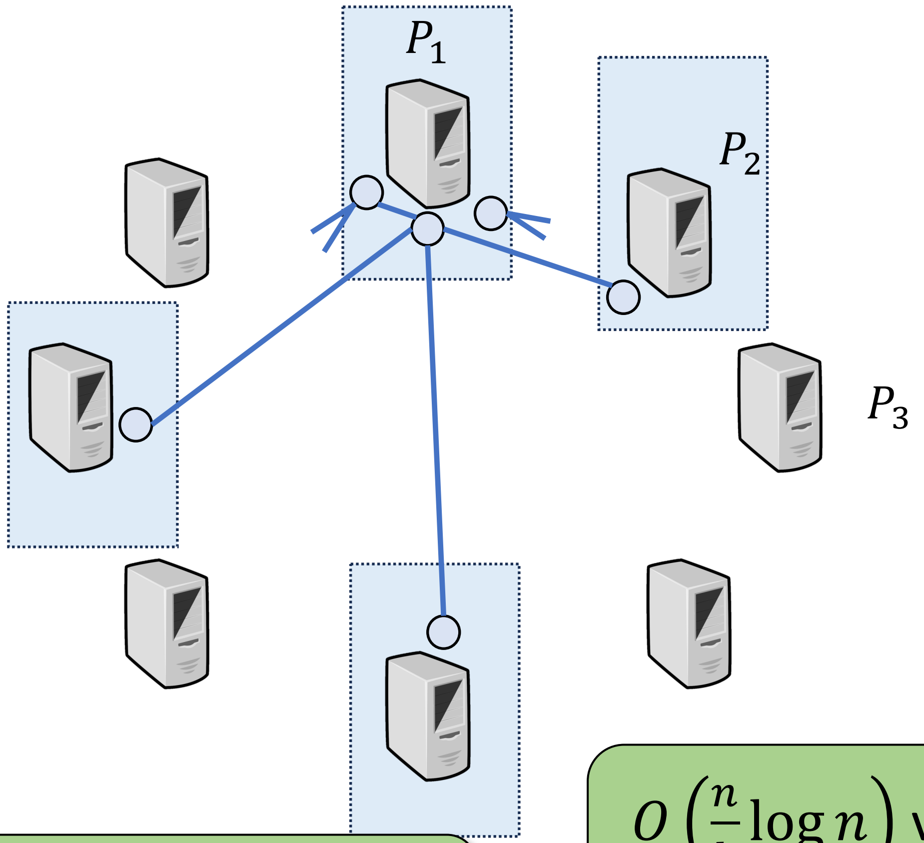
The Vertex Partition Message Passing Model

- k players (or machines) $P_1, \dots, P_k$
- Synchronous clique
- Input:
  - graph G
  - n-nodes, m-edges
- Vertex partitioning
  - random or adversarial (but balanced)
  - KT_1

k-machine model

- Bandwidth of $\beta \log n$ bits per link
- No memory restriction

can send β messages of size $O(\log n)$ over each link each round



$O\left(\frac{n}{k} \log n\right)$ vertices per player (w.h.p.)

Why Vertex Partition Model?

Motivation 1: Iterative Graph Processing Systems

- Google Pregel, Apache Giraph
- vertex centric: “think like a vertex”
- synchronous message passing

Motivation 2: Generalization of the Congested Clique

- Understanding the impact of vertex partitioning on communication

Roadmap

1. Algorithms for Maximal Independent Set
2. Lower Bounds for MIS and other Problems
3. Algorithm for t-Ruling Set
4. Open Problems

Simple MIS Algorithm

(assuming balanced vertex partitioning)

$O\left(\frac{n}{k}\log n\right)$ vertices per player; adversarially assigned

- Players proceed in sequence:
- P_1 computes MIS on $G[V(P_1)]$ and tells everyone about it:
  - Group messages into batches of size $\beta(k-1)$
  - Send 1st batch in 1 round to other $k-1$ players: β msgs to each player
  - Proceed with 2nd batch, etc.
  - Every player broadcasts received messages in sequence.
- Other players update their vertex sets by deactivating nodes accordingly.
- Then proceed with player P_2, and so forth.

How long does one iteration take?

- *"Telling everyone"* requires $O\left(\frac{n}{k}\log n\right)$ messages → takes $O\left(\frac{n}{\beta k^2}\log n\right)$ rounds.
- Each player gets $O\left(\frac{n}{k^2}\log n\right)$ messages. "Broadcasting" takes $O\left(\frac{n}{\beta k^2}\log n\right)$ rounds

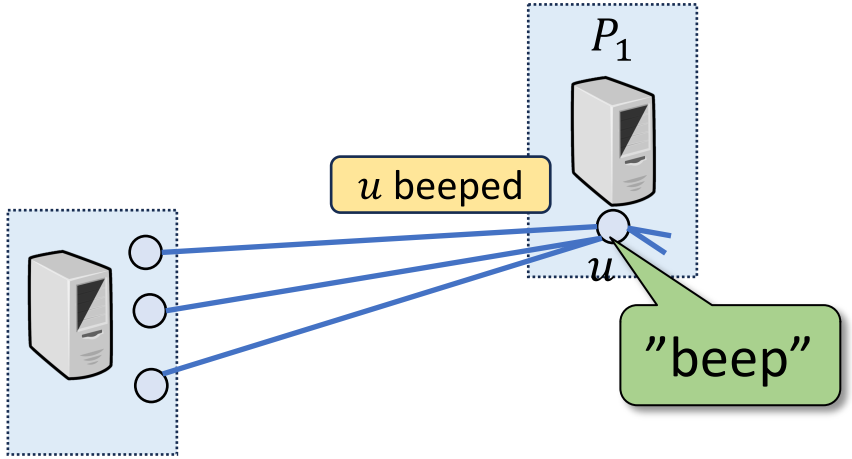
Overall time: $O\left(\frac{n}{\beta k}\log n + k\right) = \boldsymbol{O\left(\frac{n}{\beta k}\log n\right)}$ **rounds**

If $k = O(\sqrt{n/\beta})$

Simulating a Beeping Algorithm
(assuming random vertex partitioning)

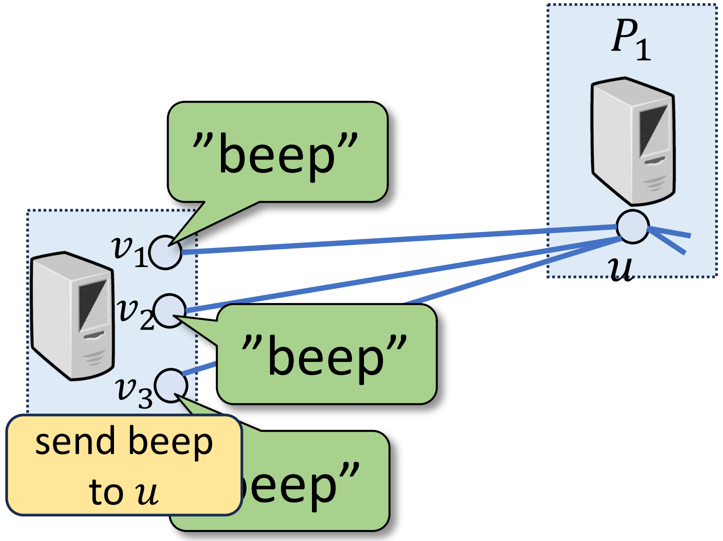
Beeping Model:

- synchronous network
- nodes can broadcast a "beep"
- node u can only distinguish between:
  1. none of its neighbors beeped
  2. ≥ 1 beep among neighbors



Simulation in k-Player Model:

- Each player P simulates beeping algorithm for all hosted nodes.
- Aggregate beeps from common source
- Aggregate beeps to common destination



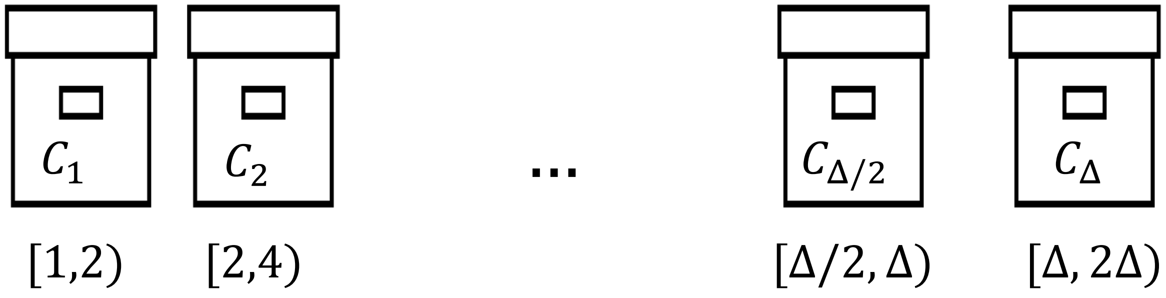
Beeping Simulation

number of edges over which a beep is sent

A beeping algorithm with message complexity M and time complexity T can be simulated in $O\left(\frac{M}{k^2}\log^2 n + T\log^3 n\right)$ rounds, assuming **random vertex partitioning**.

Consider round t:

- Partition beeping nodes into $\log\Delta$ **degree classes**.
- $a_t =$ number of messages sent in round t
- Look at any class C_d:

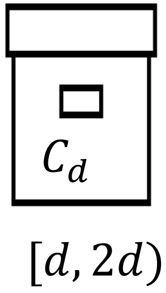


Claim

Each player can send all messages for C_d in $O\left(\frac{a_t}{\beta k^2} + \log n\right)$ rounds.

Claim

Each player can send all messages for C_d in $O\left(\frac{a_t}{\beta k^2} + \log n\right)$ rounds.



Assume $|C_d| = \Omega(k \log n)$:

- $|C_d| \leq \frac{a_t}{d}$
- for every player P_i:

$$\mathbf{E}[C_d \cap V(P_i)] \leq \frac{a_t}{d\,k}$$

- P_i needs to send $O\left(\frac{a_t}{d\,k} \cdot 2d\right) = O\left(\frac{a_t}{k}\right)$ messages in expectation (and w.h.p.)
- Use random hash function $h : IDs \rightarrow [k]$
- Send message intended for v to random player $\mathrm{P}_{h(v)}$, who forwards msg to destination

→ $O\left(\frac{a_t}{k}\right)$ messages uniformly distributed over $k-1$ links.

→ $O\left(\frac{a_t}{\beta k^2}\right)$ rounds.

aggregate messages to same vertex

Claim

Each player can send all messages for C_d in $O\left(\frac{a_t}{\beta k^2} + \log n\right)$ rounds.

- We have $\log \Delta$ degree classes:

$$O\left(\frac{a_t}{\beta k^2} \log n + \log^2 n\right)$$

- Overall:

$$\sum_{t=1}^{T} O\left(\frac{a_t}{\beta k^2} \log n + \log^2 n\right) = O\left(\frac{M}{\beta k^2} \log n + T \log^2 n\right)$$

$$\sum_{i=t}^{T} a_t = M$$

Beeping Simulation

A beeping algorithm with message complexity M and time complexity T can be simulated in $O\left(\frac{M}{\beta k^2}\log n + T\log^2 n\right)$ rounds, assuming random vertex partitioning.

[Jeavons, Scott, Xu 2016]: MIS in $O(\log n)$ rounds in beeping model.

→ MIS in $O\left(\frac{m\log n}{\beta k^2}\log n + \log^3 n\right) = O\left(\frac{m}{\beta k^2}\log^3 n\right)$ rounds.

Theorem

MIS can be solved in $\boldsymbol{O\left(\min\left\{\frac{m}{\beta k^2}\log^3 n, \frac{n}{\beta k}\log n\right\}\right)}$ rounds w.h.p., assuming random vertex partitioning

A Lower Bound for MIS

1. Show lower bound on information complexity for computing an MIS on constant-size graph “gadget” in 2-party model.
2. Simulate k-player algorithm in 2-party model for solving $\Theta(n)$ gadgets.

The Lower Bound Gadget

Gadget $\boldsymbol{H}$:

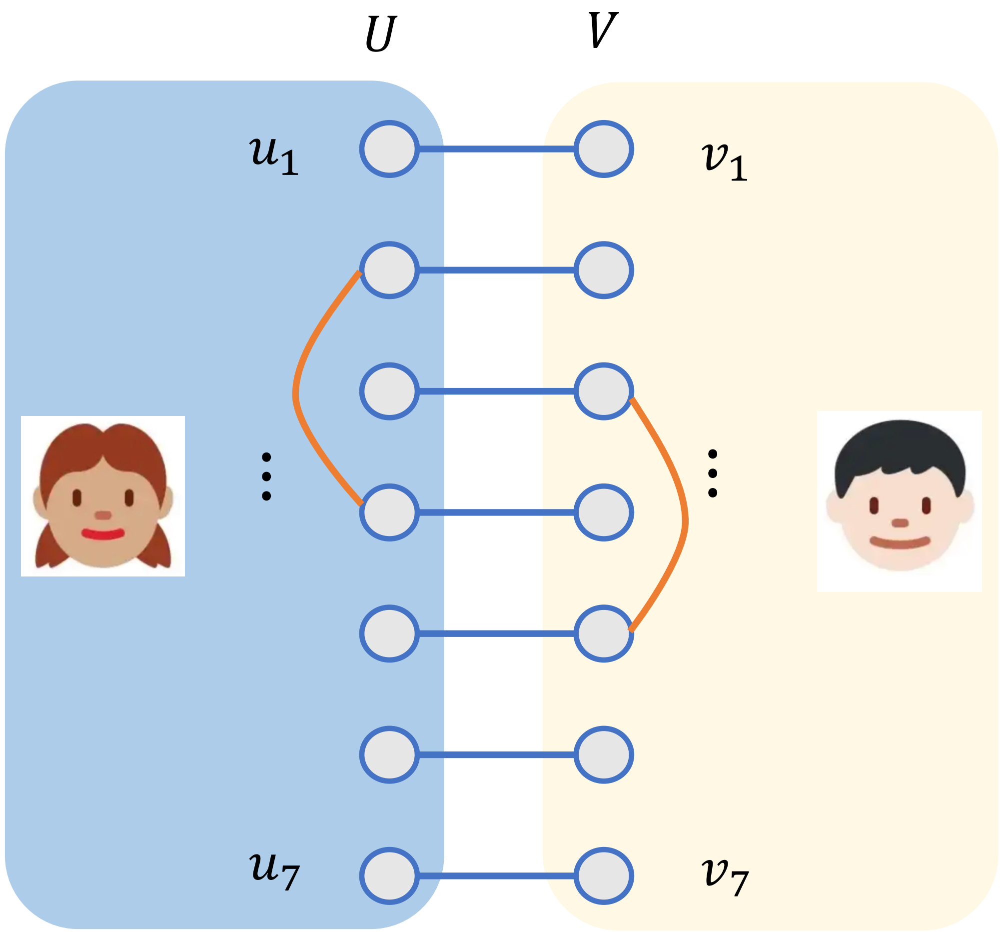
- 14 vertices: $U \cup V$
- fixed perfect matching between U, V.
- 2 random edges e_{Alice}, e_{Bob} on $H[U]$ and $H[V]$

2-party model:

- Alice's input A: U, V, e_{Alice}
- Bob's input B: U, V, e_{Bob}
- Shared randomness 
- **Goal**: Compute MIS S on H

  Alice outputs $S_{\text{Alice}} = S \cap U$

  Bob outputs $S_{Bob} = S \cap V$



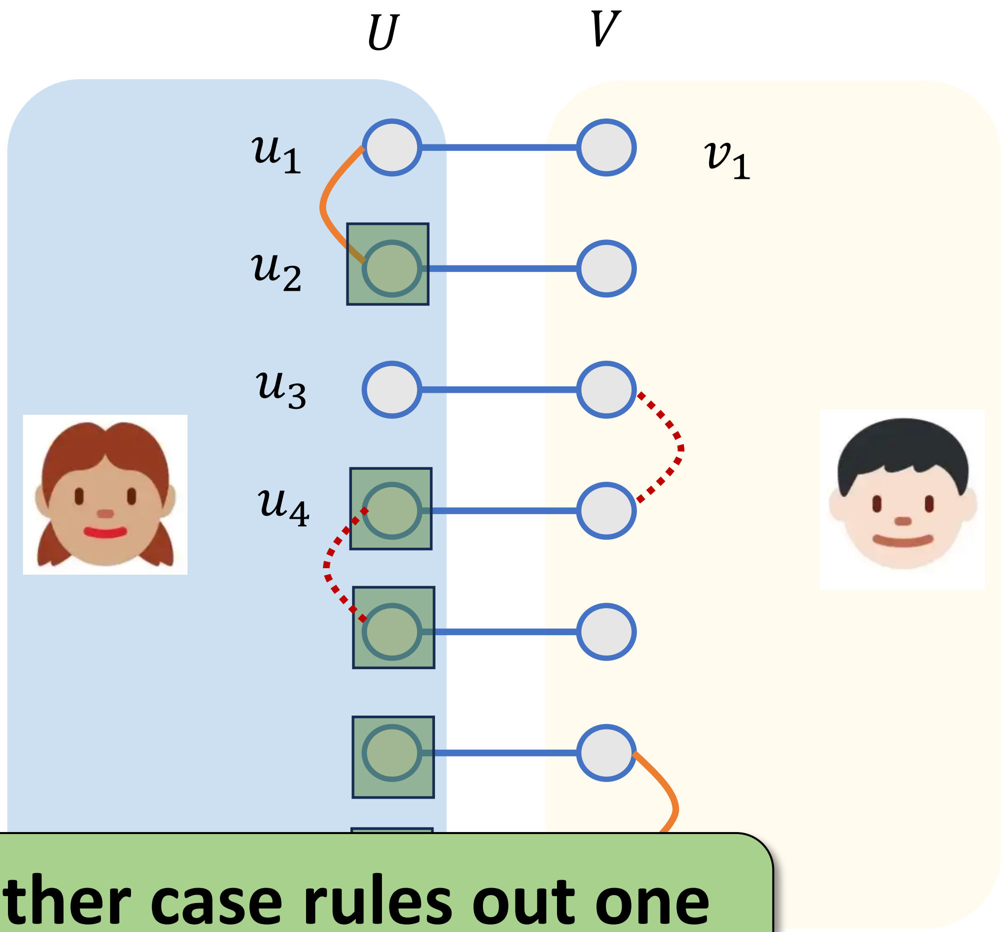
Claim

Let S be any MIS on H. Then $\mathbf{I}[\, S_{Alice} : B \mid A \,] + \mathbf{I}[\, S_{Bob} : A \mid B \,] = \Omega(1)$.

- Assume $e_{Alice} = \{u_1, u_2\}$

1. S contains ≤ 3 nodes from $\{u_3, \dots, u_7\}$:
- Suppose $u_3, u_4 \notin S$
- Not possible that $e_{Bob} = \{v_3, v_4\}$

2. S contains ≥ 4 nodes from $\{u_3, \dots, u_7\}$:
- Suppose $u_4, \dots, u_7 \in S$
- At least 2 nodes in $\{v_4, \dots, v_7\}$ are not incident to e_{Bob}
- wlog: v_4, v_5
- Not possible that $e_{Alice} = \{u_4, u_5\}$



Either case rules out one possible input!

Claim

Let S be any MIS on H. Then $\mathbf{I}[\, S_{Alice} : B \mid A \,] + \mathbf{I}[\, S_{Bob} : A \mid B \,] = \Omega(1)$.

In first case:

- Initially: $\binom{7}{2} = 21$ possibilities each for e_{Bob} and e_{Alice}
- After computing MIS: $\binom{7}{2} = 20$ possibilities left
- $\mathbf{I}[\, S_{Alice} : B \mid A \,] = \mathbf{H}[B \mid A] - \mathbf{H}[B \mid S_{Bob}, A]$

$$= \log_2 21 \;\; - \log_2 20$$

$$= \Omega(1)$$

Similar for 2nd case.

A Lower Bound for MIS

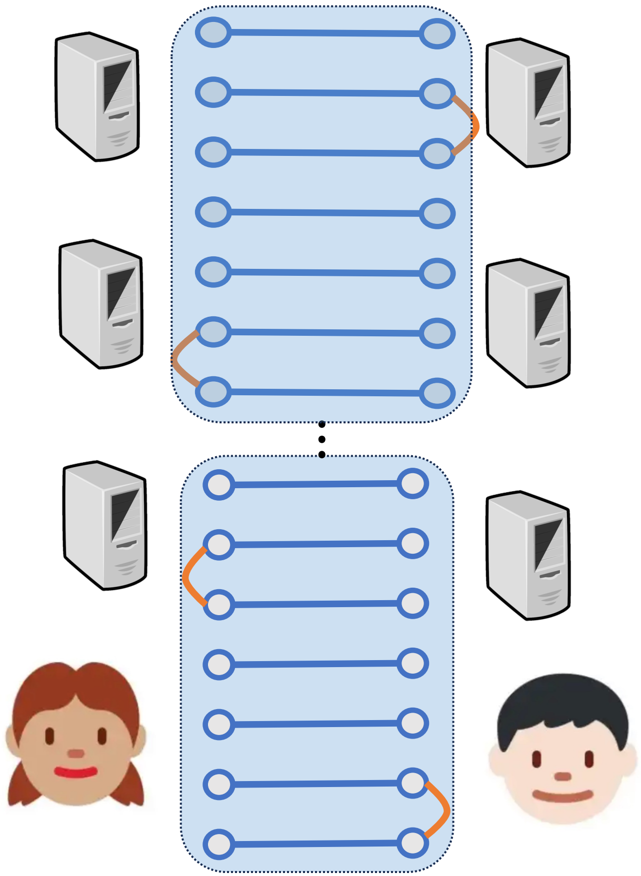
1. Show lower bound on information complexity for computing an MIS on constant-size graph "gadget" in 2-party model.
2. Simulate k-player algorithm in 2-party model for solving $\Theta(n)$ gadgets.

Simulating k-Players in the 2-Party Model

- Suppose G consists of $\boldsymbol{m = n/14}$ **randomly sampled gadgets** $H_1, \ldots, H_{n/14}$
  - Alice's input A^m = "left" side of all gadgets.
  - Bob's input B^m = "right" side of all gadgets.
- Let Q be k-player MIS algorithm.
- Alice and Bob simulate $k/2$ players each.
- Alice assigns n/k vertices to each of her players using some fixed rule.
- Same for Bob.

→ balanced vertex partitioning

- Compute MIS on G using Q.



Claim

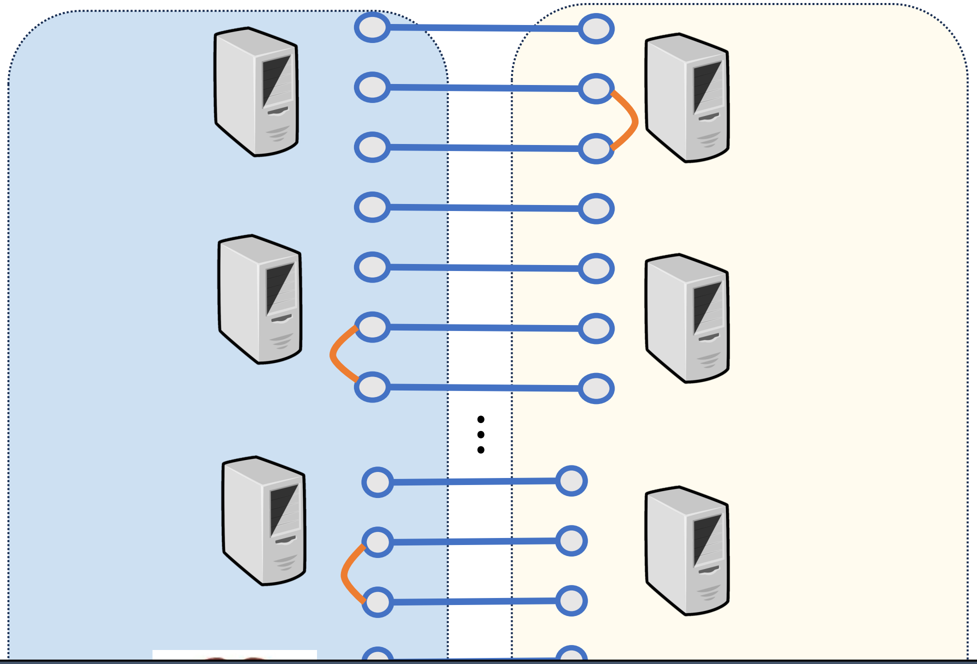
Let S^m be any MIS on $\boldsymbol{G}$. Then $\mathbf{I}[\,S^m_{Alice} : B^m \mid A^m] + \mathbf{I}[\,S^m_{Bob} : A^m \mid B^m] = \boldsymbol{\Omega(n)}$.

- Gadgets are sampled independently:

$$\mathbf{I}[\,S^m_{Alice} : B^m \mid A^m\,] \geq \sum_{i=1}^{n/14} \mathbf{I}[\,S_{Alice} : B \mid A]$$

and

$$\mathbf{I}[S^m_{Bob} : A^m \mid B^m\,] \geq \sum_{i=1}^{n/14} \mathbf{I}[\,S_{Bob} : A \mid B]$$



→ Either Alice's or Bob's pla
$\Omega(n)$ bits in simulation.

Claim

Let S be any MIS on H. Then $\mathbf{I}[\,S_{Alice} : B \mid A\,] + \mathbf{I}[\,S_{Bob} : A \mid B\,] = \Omega(1)$.

Claim ✓

Let S^m be any MIS on $\boldsymbol{G}$. Then $\mathbf{I}[\, S^m_{Alice} : B^m \mid A^m] + \mathbf{I}[\, S^m_{Bob} : A^m \mid B^m] = \boldsymbol{\Omega(n)}$.

- Either Alice or Bob learns $\Omega(n)$ bits.
- Simulation sends $\Omega(n)$ bits from Alice's $\frac{k}{2}$ simulated players to Bob's or vice versa.
- $O(k^2)$ links between Alice and Bob's players.

→ Simulation takes $\Omega\left(\frac{n}{\beta k^2 \log n}\right)$ rounds.

Lower Bound

Computing an MIS with constant error takes $\Omega\left(\frac{n}{\beta k^2 \log n}\right)$ rounds, assuming **balanced vertex partitioning**.

Extension to Random Vertex Partitioning

- Consider **random-input 2-party model**
- Each vertex assigned to Alice or Bob w.p. $\frac{1}{2}$
- Gadget is *good* if Alice gets entire left side

$$\Pr[\text{ Gadget is good }] = \Omega(1)$$

- Constant fraction of gadgets will be good in expectation.
- Previous argument applies to good gadgets.

Lower Bound

Computing an MIS with constant error takes $\Omega\left(\frac{n}{\beta k^2 \log n}\right)$ rounds, assuming random vertex partitioning.

Lower Bound

Computing an MIS with constant error takes $\Omega\left(\frac{n}{\beta k^2 \log n}\right)$ rounds, assuming random vertex partitioning.

Upper Bound

MIS can be solved in $O\left(\min\left\{\frac{m}{\beta k^2}\log^3 n, \frac{n}{\beta k}\log n\right\}\right)$ rounds w.h.p., assuming random vertex partitioning

- *Can we improve upper bound?*
  - Faster algorithm would require lower message complexity.
  - Related open problem: Computing MIS in CONGEST (KT_1) with $o(m)$ messages:
    - Any $o(m)$-algorithm must use node IDs / graph sketching in non-trivial way.

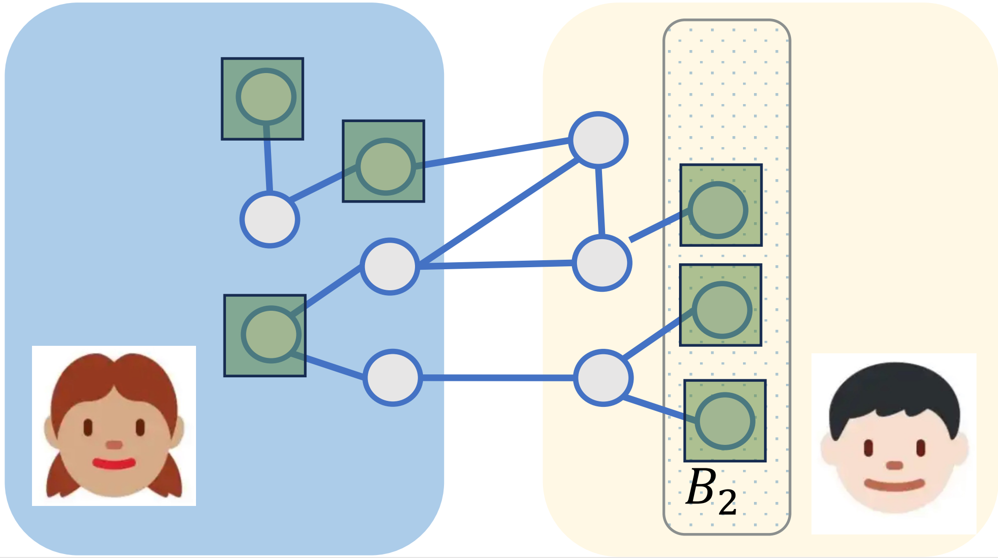
$\Omega(m)$ lower bound for comparison-based algorithms

- *Can we improve lower bound?*
  - multi-party approach seems promising.
  - Currently known techniques may not suffice…

Lower Bounds for other Problems: **Ruling sets?**

- S is ***t*-ruling set** of G if:
  - S is independent set of G
  - Every node in G is $\leq t$ hops from some node in S.

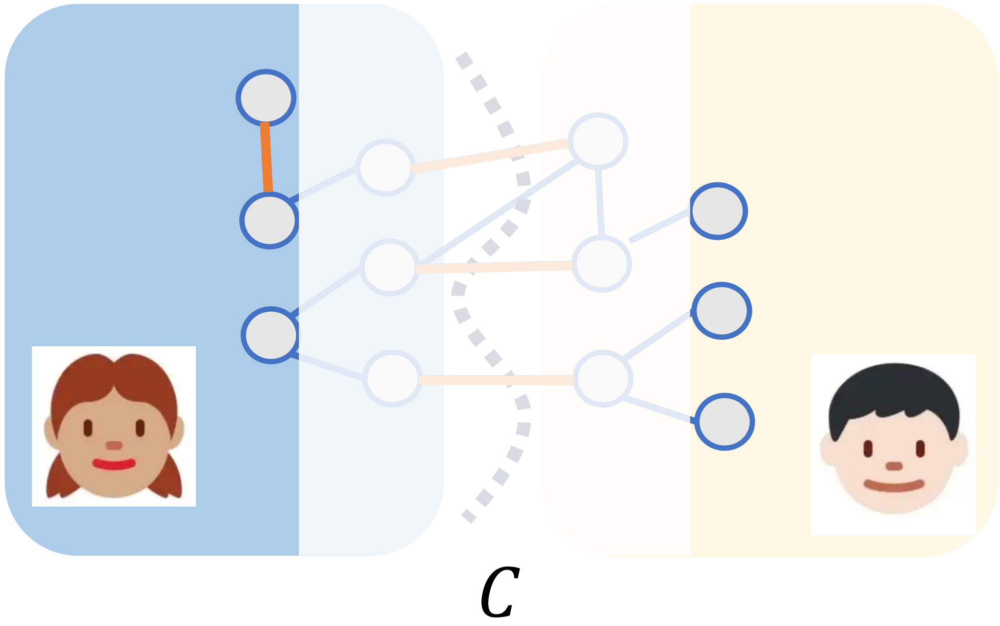
- Consider **any** graph in 2-party model
- Partition vertices in adversarial way
- Alice:
  - Computes MIS on her subgraph
- Bob:
  - Let B_2 contain vertices that have distance ≥ 2 from Alice
  - Compute MIS on B_2



→ **2-ruling set *without* communication!**

Lower Bounds for other Problems: **Maximal Matching?**

- Consider any graph in 2-party model.
- Again, adversarial vertex partition.
- Alice & Bob know all cut edges C.
- Locally compute maximal matching on $G[C]$.
- Both deactivate matched nodes.
- Both compute maximal matching on their residual graph.
- Maximal matching on gadget is union of these matchings.



→ **maximal matching *without* communication!**

Multiparty Approach for t-ruling set

- $k = t + 2$ players
- **Input:** Graph H sampled from all k-vertex graphs
- "congested clique"-like model: each player gets 1 vertex
- Limitation: only works for **"component stable" algorithms**

Claim

For every t-ruling set algorithm $\mathcal{A}$ there is a $(t + 2)$-node graph H where, for some vertex v_i with neighborhood N_i:

$$\mathbf{I}[\,\mathcal{A}(v_i) : \overline{N_i} \mid N_i, R\,] = \Omega(1)$$

$H \setminus N_i$

shared randomness

Claim

For every t-ruling set algorithm $\mathcal{A}$ there is a $(t+2)$-node graph H where, for some vertex v_i with neighborhood N_i:

$$\mathbf{I}[\,\mathcal{A}(v_i) : \overline{N_i} \mid N_i, R\,] = \Omega(1)$$

- Look at deterministic algorithm D.
- Sufficient to show that there is some v_i with N_i:

$$\left|\, \Pr(\, D(v_i) = 1 \mid N_i = \, n_i \,) - \overbrace{\Pr(\, D(v_i) = 1 \mid H)}^{= 0, \text{ because } D \text{ is “component stable”}} \right| = \Omega(1)$$

- Suppose D is silent on every $(t+2)$-node graph H.
- Let S be computed ruling set.
- Vertex u **dominates** v “$v <_D u$” if, for every H: $\{u, v\} \in H$ implies $u \notin S$.

Claim

For every t-ruling set algorithm $\mathcal{A}$ there is a $(t+2)$-node graph H where, for some vertex v_i with neighborhood N_i:

$$\mathbf{I}[\,\mathcal{A}(v_i) : \overline{N_i} \mid N_i, R\,] = \Omega(1)$$

- Vertex u **dominates** v "$v <_D u$" if, for every H: $\{u, v\} \in H$ implies $u \notin S$.
- Can prove that $(<_D)$ is **strict total order** over $v_1, \dots, v_k$.
- W.l.og: $v_1 <_D v_2 <_D \cdots <_D v_k$
- Run algorithm on path $v_1 - v_2 - \cdots - v_k$
- Only v_k can enter S
- Invalid since $k \geq t + 2$.

Use direct sum argument to show $\Omega\left(\frac{n}{\beta k^2\, t}\right)$ rounds for t-ruling set in k-player model (for component stable algorithms).

t-Ruling Set Algorithm

1. $R_1 \leftarrow V(G)$
2. **for** iteration $i \leftarrow 2, \dots, t$ **do:**
3. Every node in R_{i-1} marks itself w.p. $\Theta\left(\frac{\log n}{\Delta^{1-(i-1)/t}}\right)$
4. $M_i \leftarrow$ marked nodes
5. Inform all neighbors of M_i
6. $S_i \leftarrow$ MIS($G[M_i]$)
7. $D_i \leftarrow$ nodes that (a) have a neighbor in M_i, (b) or degree > $\Delta^{1-(i-1)/t}$
8. Deactivate all nodes in D_i
9. $R_i \leftarrow R_{i-1} \setminus (M_i \cup D_i)$
10. Inform all neighbors of nodes in R_t
11. $S \leftarrow$ MIS($G[R_t]$)
12. Return $(S \cup S_1 \cup \cdots \cup S_t)$

Lemma 1: max. deg. of nodes in R_i is $\leq \Delta^{1-(i-1)/t}$

Lemma 2: max. deg. in $G[M_i]$ is $\leq O(\Delta^{1/t}\log n)$ (w.h.p.)

- Because max. deg. in $R_{i-2} \leq \Delta^{1-(i-2)/t}$

Lemma 3: Informing neighbors takes $\tilde{O}(n\Delta^{1/t}/\beta k^2)$ rnds

- By random partitioning: $\tilde{O}\left(\frac{n}{k}\right)$ nodes per player
- Thus: $\tilde{O}\left(\frac{n}{k\Delta^{1-(i-1)/t}}\right)$ marked nodes per player.
- By Lem 1: max.deg. is $\leq \Delta^{1-(i-2)/t}$
- Need to send/rcv $\tilde{O}\left(\frac{n}{k\Delta^{1/t}}\right)$ messages per player.
- Takes $\tilde{O}\left(\frac{n}{\beta k^2\Delta^{1/t}}\right)$ rounds using random routing.

Lemma 4: MIS($G[M_i]$) takes $\tilde{O}(n\,\Delta^{1/t}/\beta k^2)$ rounds

Lemma 5: Algorithm takes $\tilde{O}(n\,t\,\Delta^{1/t}/\beta k^2)$ rounds

Theorem

There is a t-ruling set algorithm that takes $\tilde{O}\left(t\, n\, \Delta^{\frac{1}{t}}/\beta k^2\right)$ rounds w.h.p.

3. Every node in R_{i-1} marks itself w.p. $\Theta\left(\frac{\log n}{\Delta^{1-(i-1)/t}}\right)$

4. $M_i \leftarrow$ marked nodes

5. Inform all neighbors of M_i

6. $S_i \leftarrow$ MIS($G[M_i]$)

7. $D_i \leftarrow$ nodes that (a) have a neighbor in M_i, (b) or degree > $\Delta^{1-(i-1)/t}$

8. Deactivate all nodes in D_i

9. $R_i \leftarrow R_{i-1} \setminus (M_i \cup D_i)$

10. Inform all neighbors of nodes in R_t

11. $S \leftarrow$ MIS($G[R_t]$)

12. Return $X = (S \cup S_1 \cup \cdots \cup S_t)$

Lemma 4: Output is a t-Ruling Set.

- All marked nodes have a neighbor in X
- All nodes in R_t have a neighbor in X
- Consider $u \in D_i$, for some iteration i
- Show u has dist. $\leq i$ to some node in X

1. Case (a):
   - u is 2 hops away from some node in X
2. Case (b):
   - u has large degree
   - No marked neighbor → some neighbor w was deactivated in iteration $j < i$.
   - By induction: w has dist. $\leq j$ to X.

Main Open Problems

- **MIS**: $\tilde{O}(\min\{\frac{n}{\beta k}, \frac{m}{\beta k^2}\})$ or $\widetilde{\Omega}\left(\frac{n}{\beta k^2}\right)$?
- **Maximal Matching**: Can we go below $\tilde{O}\left(\frac{n}{\beta k}\right)$?
- **Lower bounds** for **component-unstable algorithms**?
  - Does every "interesting" problem have an $\widetilde{\Omega}\left(\frac{n}{\beta k^2}\right)$ lower bound?
- **Lower bound** of $\Omega\left(\frac{m}{\beta k^2}\right)$ possible for some problems?

Thanks!