The NOF Communication Complexity of Multi-Party Pointer Jumping

Joshua Brody

Dartmouth College
Hanover, NH USA

IAS Computer Science/Discrete Math Seminar
December 7, 2009

Talk Outline

- Multi-Party Communication Games
- The Multi-Party Pointer Jumping Problem
- Upper Bounds
- Restricted Protocols
- Conclusions

Multi-party Communication Games

Multi-Party Communication Games

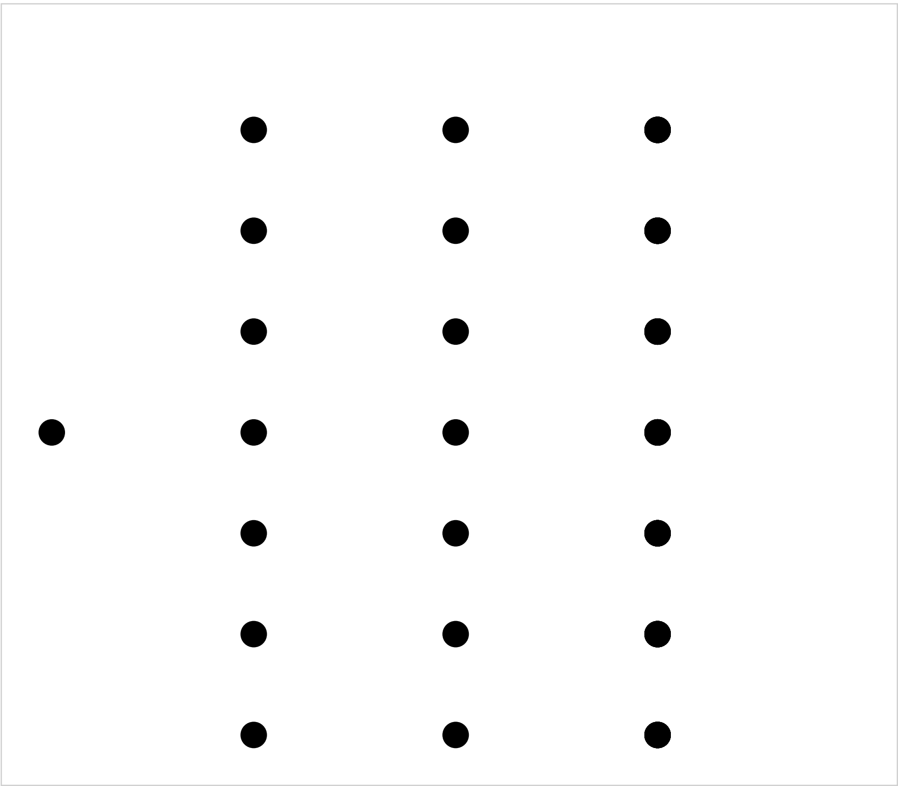
Input $x = (x_1, \ldots, x_k)$ is split between k players.

Goal: minimize communication needed to compute $f(x)$.

Our model of communication:

- Player i sees every input <u>except</u> x_i (NOF model).
- One-way communication: each player speaks once and in order.
- Blackboard communication: all players see every message sent.

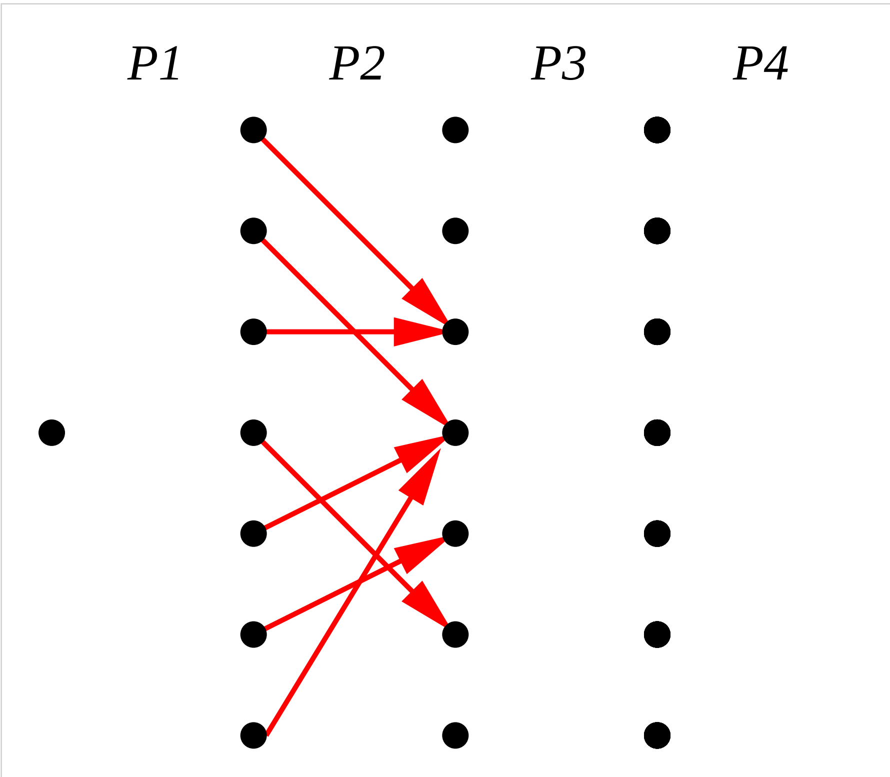
Pointer Jumping



Vertices:

- $k-1$ layers, plus start vertex
- layers have n vertices

Pointer Jumping



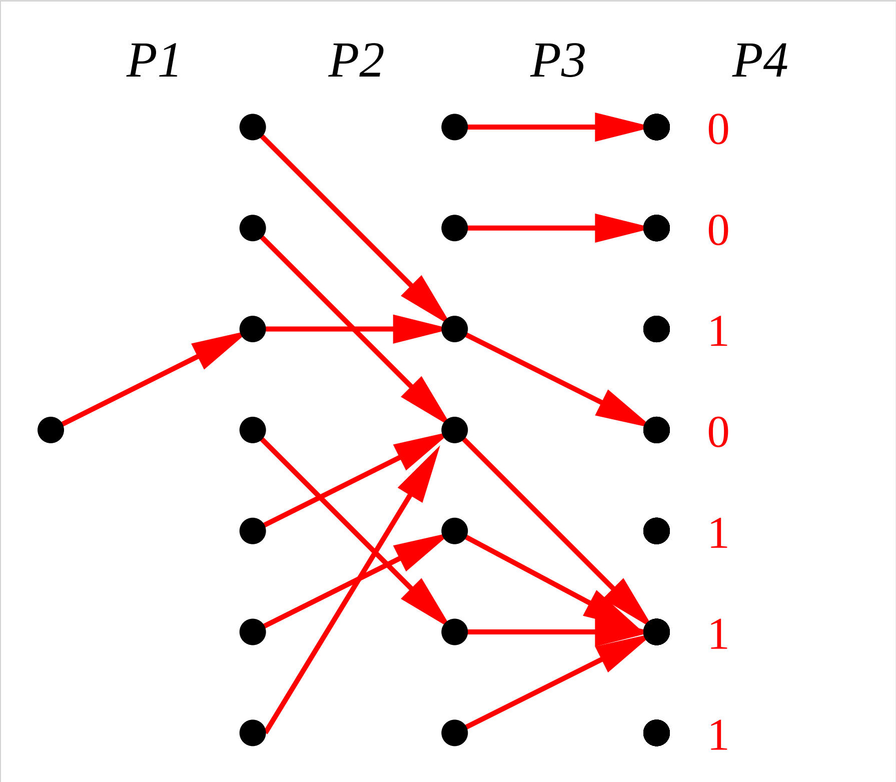
Vertices:

- $k-1$ layers, plus start vertex
- layers have n vertices

Input:

- $k-1$ layers of pointers
- n bit string

Pointer Jumping



Vertices:

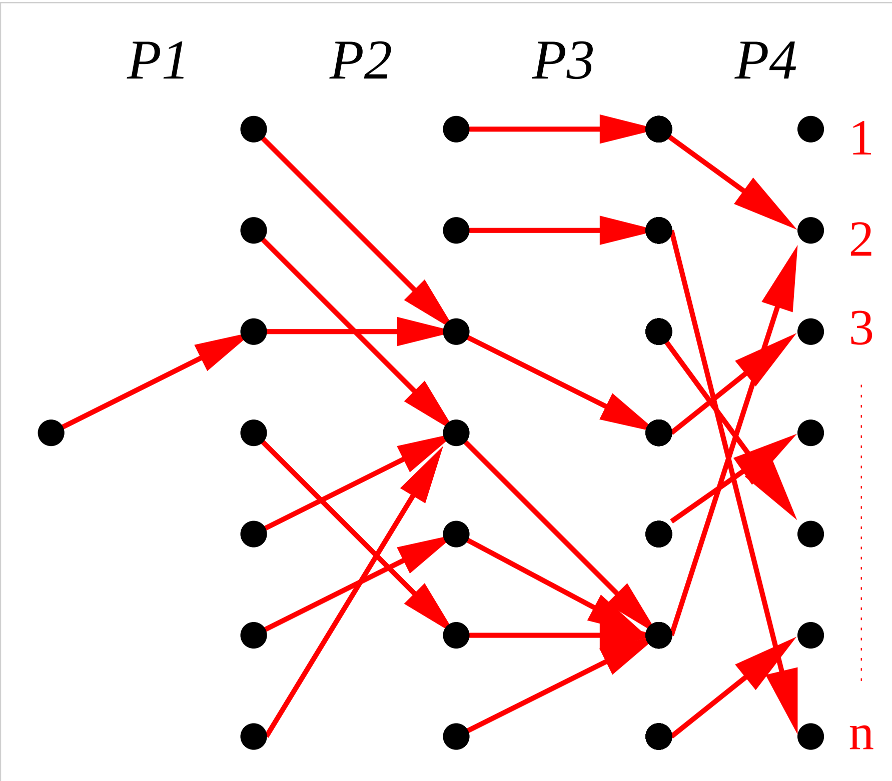
- $k-1$ layers, plus start vertex
- layers have n vertices

Input:

- $k-1$ layers of pointers
- n bit string

Compute MPJ_k = bit reached by following pointers from start vertex.

Pointer Jumping: non-Boolean version



Vertices:

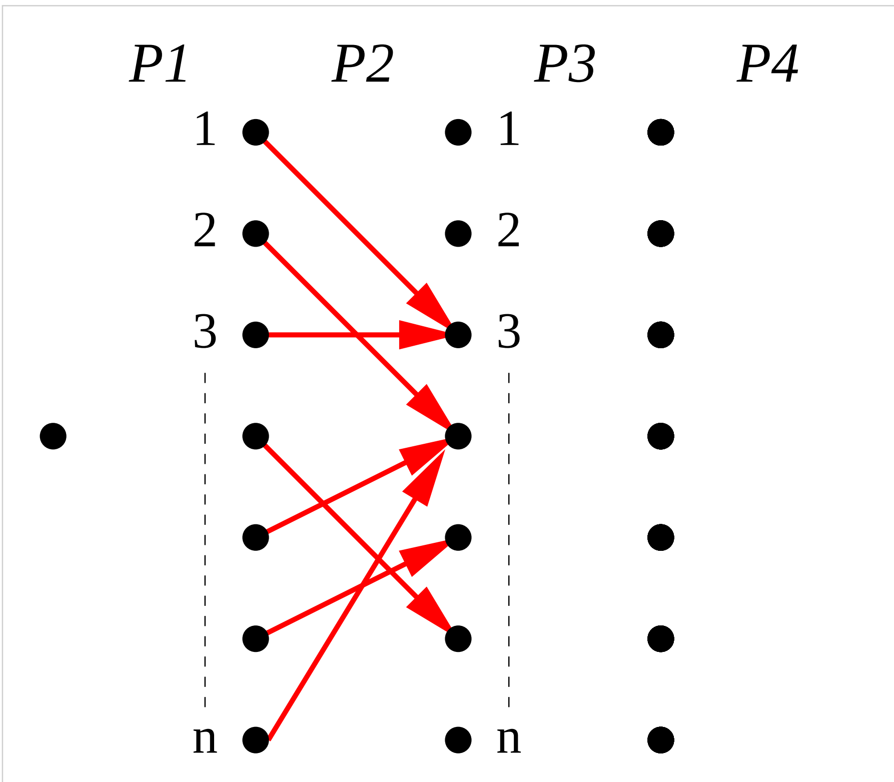
- k layers, plus start vertex
- layers have n vertices

Input:

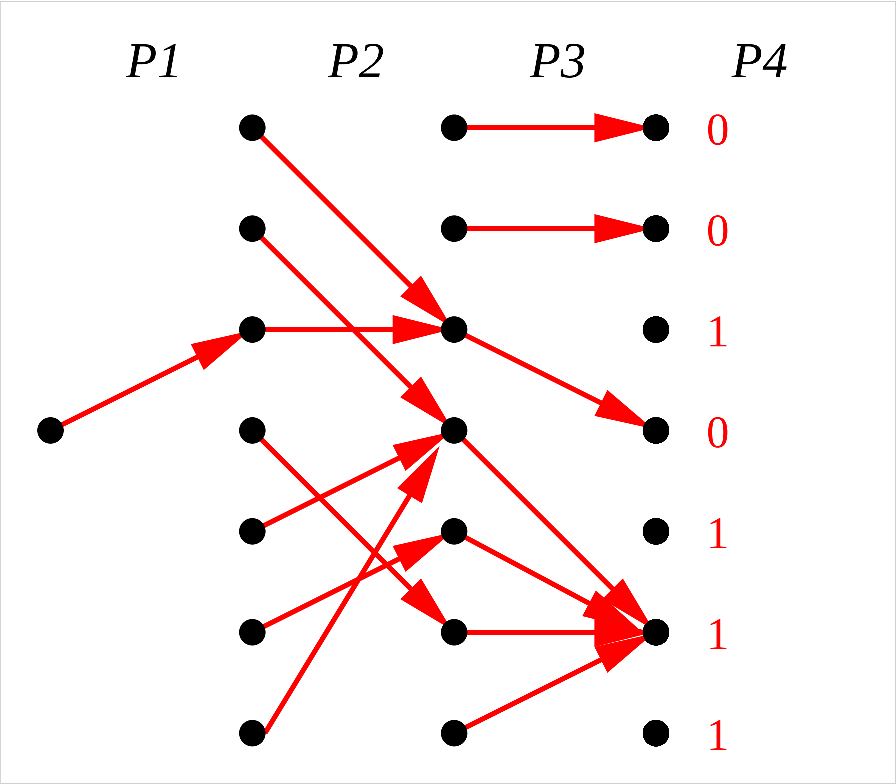
- k layers of pointers

Compute $\widehat{\mathrm{MPJ}}_k$ = vertex reached by following pointers from start vertex.

Layers of Edges are Functions



Layers of Edges are Functions



Formal Definition:

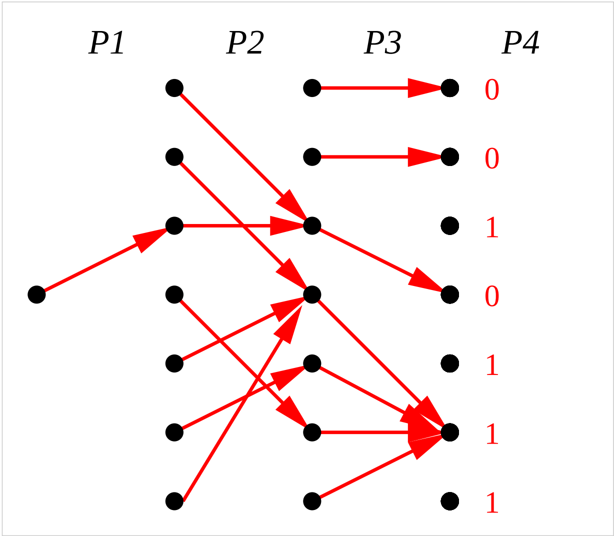
Inputs:

- $i \in [n]$
- $f_2, \ldots, f_{k-1} : [n] \to [n]$
- $x \in \{0,1\}^n$

Output:

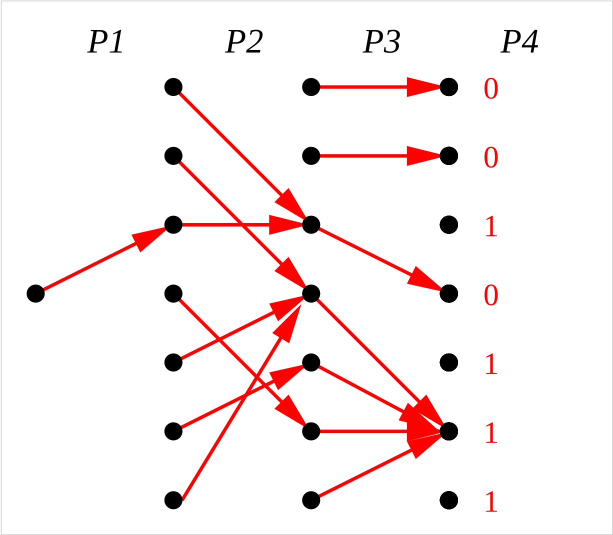
- $\text{MPJ}_k := x[f_{k-1} \circ \cdots \circ f_2(i)]$

Pointer Jumping: Trivial Bounds



- One-way: any order <u>except</u> $P1, P2, \ldots, Pk$: $O(\log n)$
- One way: in the order $P1, P2, \ldots, Pk$: $O(n)$

Pointer Jumping: Trivial Bounds



- One-way: any order except $P1, P2, \ldots, Pk$: $O(\log n)$
- One way: in the order $P1, P2, \ldots, Pk$: $O(n)$
  - Problem seems hard. Maybe $n^{\Omega(1)}$ lower bound?

Motivation

ACC^0 complexity class: AC^0 plus MOD_m gates.

- No function $f \notin \mathsf{ACC}^0$ is known.
- If $f : \{0,1\}^n \to \{0,1\}$ and $f \in \mathsf{ACC}^0$, then f has deterministic NOF protocol with $\mathrm{poly}(\log n)$ communication, for $k = \mathrm{poly}(\log n)$ players.

[Yao'90], [Håstad-Goldmann'91], [Beigel-Tarui'94]

More Motivation

ACC^0 complexity class: AC^0 plus MOD_m gates.

- No function $f \notin \mathsf{ACC}^0$ is known.
- If $f : \{0,1\}^n \to \{0,1\}$ and $f \in \mathsf{ACC}^0$, then f has deterministic NOF protocol with $\text{poly}(\log n)$ communication, for $k = \text{poly}(\log n)$ players.

[Yao'90], [Håstad-Goldmann'91], [Beigel-Tarui'94]

Recently pointer jumping has been used to prove lower bounds in:

- threshold circuits [Razborov-Wigderson'93]
- proof size [Beame-Pitassi-Segerlind'05]
- matroid intersection queries [Harvey'08]
- randomly-ordered data streams [Chakrabarti-Cormode-McGregor'08]

Previous Result Highlights

Far from proving $\text{MPJ}_{\text{poly}(\log n)} \notin \mathsf{ACC}^0$

- $\Omega(\sqrt{n})$ for MPJ_3 [Wigderson'97]
- $\Omega(n^{1/(k-1)}/k^k)$ for MPJ_k [Viola-Wigderson'07]
- lower bounds for restricted protocols

Previous Result Highlights

Far from proving $\text{MPJ}_{\text{poly}(\log n)} \notin \mathsf{ACC}^0$

- $\Omega(\sqrt{n})$ for MPJ_3 [Wigderson'97]
- $\Omega(n^{1/(k-1)}/k^k)$ for MPJ_k [Viola-Wigderson'07]
- lower bounds for restricted protocols
- $O\left(n \log^{(k-1)} n\right)$ for $\widehat{\text{MPJ}}_k$ [Damm-Jukna-Sgall'96]
- $O\left(n \frac{\log\log n}{\log n}\right)$ for MPJ_3 when middle layer is a permutation.

[Pudlák-Rödl-Sgall '97]

Previous Result Highlights

Far from proving $\text{MPJ}_{\text{poly}(\log n)} \notin \text{ACC}^0$

- $\Omega(\sqrt{n})$ for MPJ_3 [Wigderson'97]
- $\Omega(n^{1/(k-1)}/k^k)$ for MPJ_k [Viola-Wigderson'07]
- lower bounds for restricted protocols
- $O\left(n \log^{(k-1)} n\right)$ for $\widehat{\text{MPJ}}_k$ [Damm-Jukna-Sgall'96]
- $O\left(n \frac{\log\log n}{\log n}\right)$ for MPJ_3 when middle layer is a permutation.
  [Pudlák-Rödl-Sgall '97]

Our Results

- $O\left(n \sqrt{\frac{\log\log n}{\log n}}\right)$ for MPJ_3 [B.-Chakrabarti'08]
- bounds for restricted protocols [B.'09]

Previous Result Highlights

Far from proving $\text{MPJ}_{\text{poly}(\log n)} \notin \text{ACC}^0$

- $\Omega(\sqrt{n})$ for MPJ_3 [Wigderson'97]
- $\Omega(n^{1/(k-1)}/k^k)$ for MPJ_k [Viola-Wigderson'07]
- lower bounds for restricted protocols (2nd half of talk)
- $O\left(n \log^{(k-1)} n\right)$ for $\widehat{\text{MPJ}}_k$ [Damm-Jukna-Sgall'96]
- $O\left(n \frac{\log \log n}{\log n}\right)$ for MPJ_3 when middle layer is a permutation. [Pudlák-Rödl-Sgall '97]

Our Results

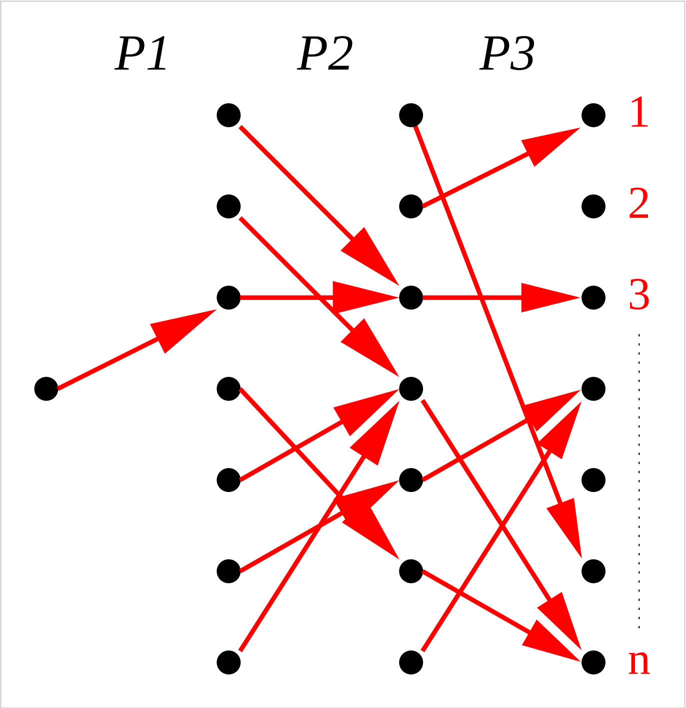
- $O\left(n \sqrt{\frac{\log \log n}{\log n}}\right)$ for MPJ_3 [B.-Chakrabarti'08]
- bounds for restricted protocols (2nd half of talk) [B.'09]

Talk Outline

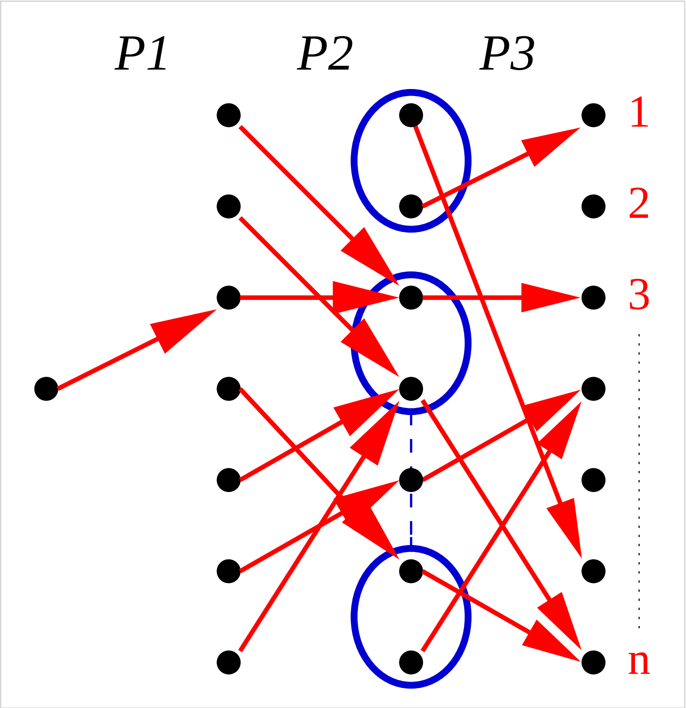
- Multi-Party Communication Games
- The Multi-Party Pointer Jumping Problem
- Upper Bounds
- Restricted Protocols
- Conclusions

The Damm-Jukna-Sgall Protocol

3 players:



The Damm-Jukna-Sgall Protocol



3 players:

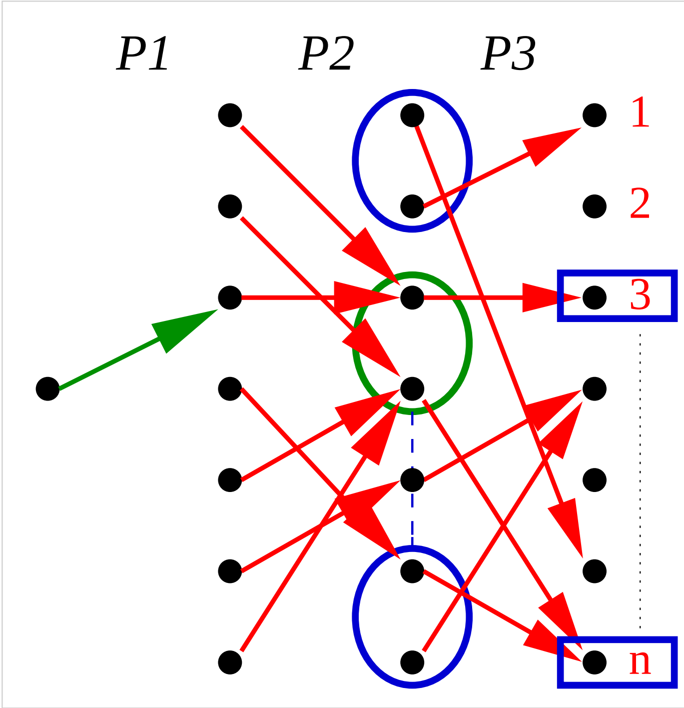
$P1$ sends $\log \log n$ bits of $f_2(i)$ for each i

The Damm-Jukna-Sgall Protocol

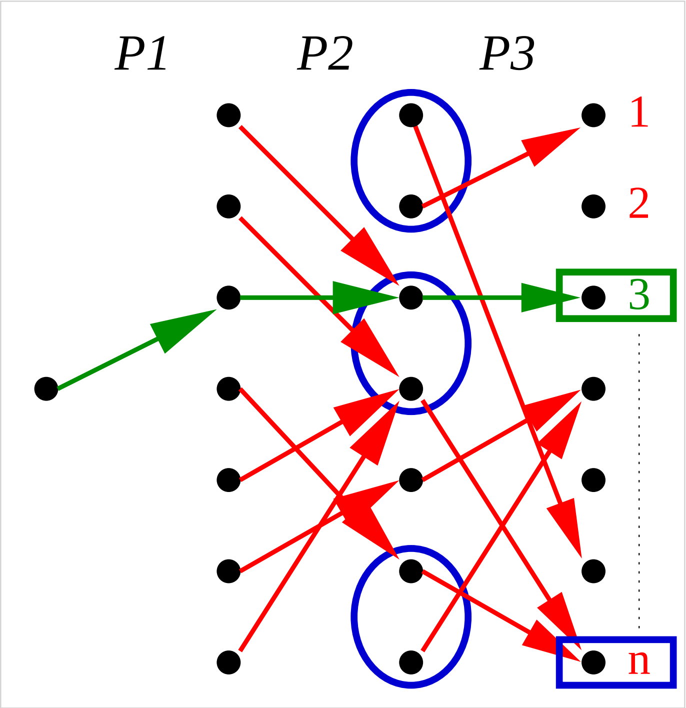
3 players:

$P1$ sends $\log \log n$ bits of $f_2(i)$ for each i

$P2$ sends $f_3(j)$ for each possibile j



The Damm-Jukna-Sgall Protocol



3 players:

$P1$ sends $\log \log n$ bits of $f_2(i)$ for each i

$P2$ sends $f_3(j)$ for each possibile j

$P3$ outputs $f_3(f_2(i))$.

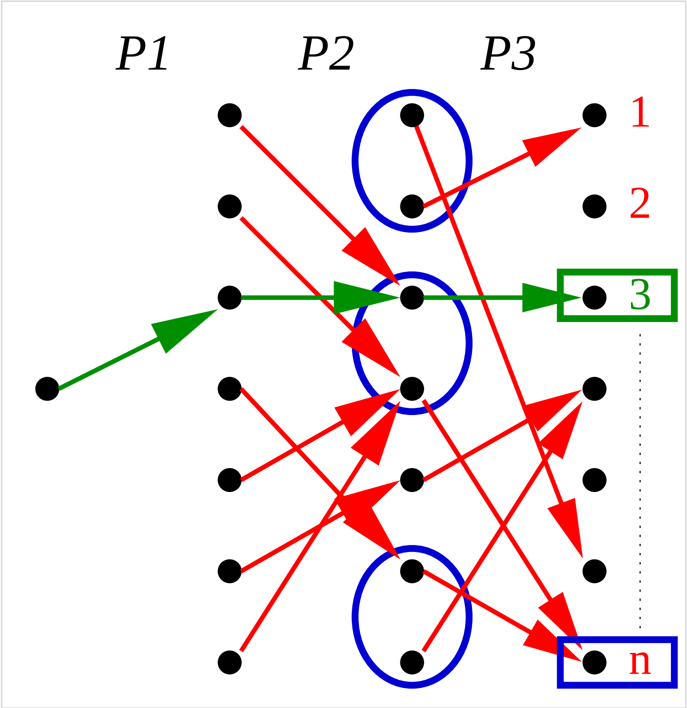
The Damm-Jukna-Sgall Protocol

3 players:

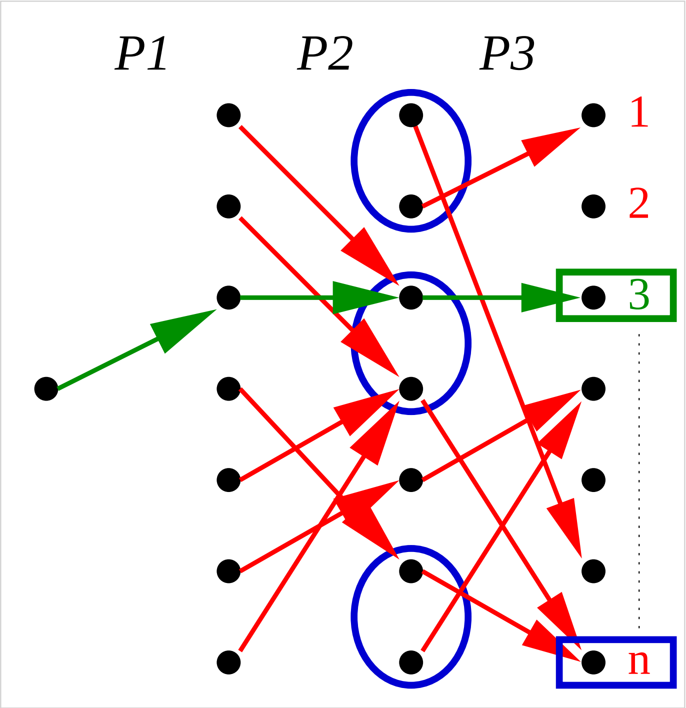
$P1$ sends $\log\log n$ bits of $f_2(i)$ for each i $\Rightarrow n \log\log n$ bits.

$P2$ sends $f_3(j)$ for each possibile j $\Rightarrow \frac{n}{2^{\log\log n}} \log n = n$ bits.

$P3$ outputs $f_3(f_2(i))$.



The Damm-Jukna-Sgall Protocol



3 players:

$P1$ sends $\log\log n$ bits of $f_2(i)$ for each i $\Rightarrow n \log\log n$ bits.

$P2$ sends $f_3(j)$ for each possibile j

$\Rightarrow \frac{n}{2^{\log\log n}} \log n = n$ bits.

$P3$ outputs $f_3(f_2(i))$.

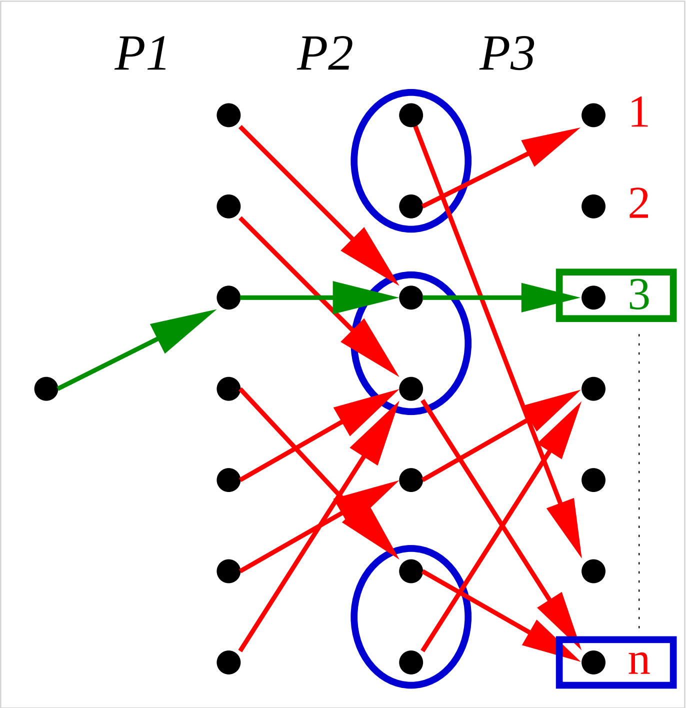
k players:

$P1$ sends $\log^{(k-1)} n$ bits for each pointer.

$P2$ sends $\log^{(k-2)} n$ bits for each of $n/\log^{(k-2)} n$ possible pointers.

...

The Damm-Jukna-Sgall Protocol



3 players:

$P1$ sends $\log\log n$ bits of $f_2(i)$ for each i $\Rightarrow n \log\log n$ bits.

$P2$ sends $f_3(j)$ for each possibile j

$\Rightarrow \frac{n}{2^{\log\log n}} \log n = n$ bits.

$P3$ outputs $f_3(f_2(i))$.

k players:

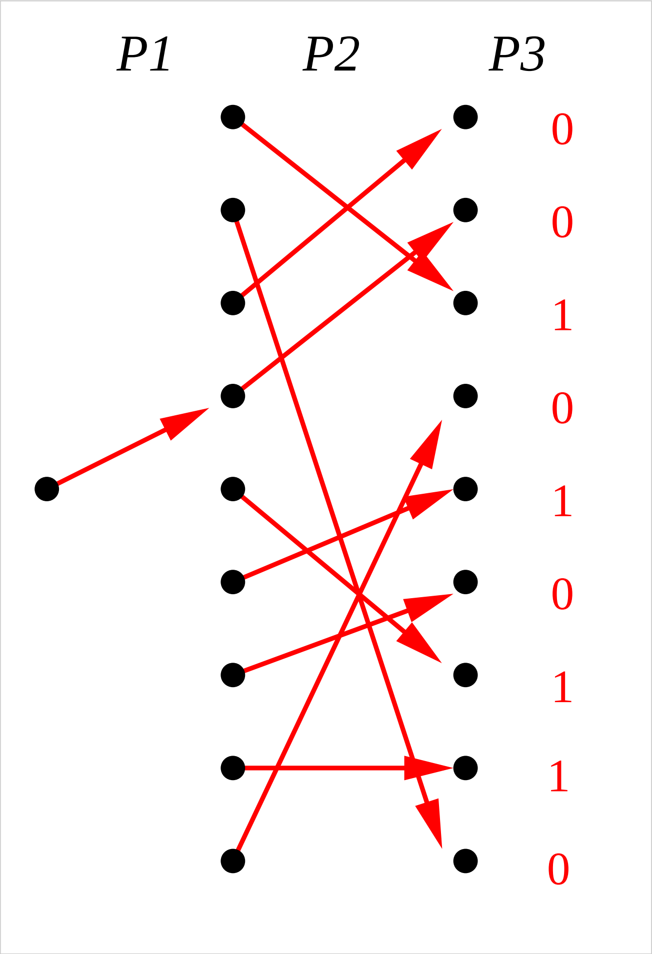
$P1$ sends $\log^{(k-1)} n$ bits for each pointer.

$P2$ sends $\log^{(k-2)} n$ bits for each of $n/\log^{(k-2)} n$ possible pointers.

...

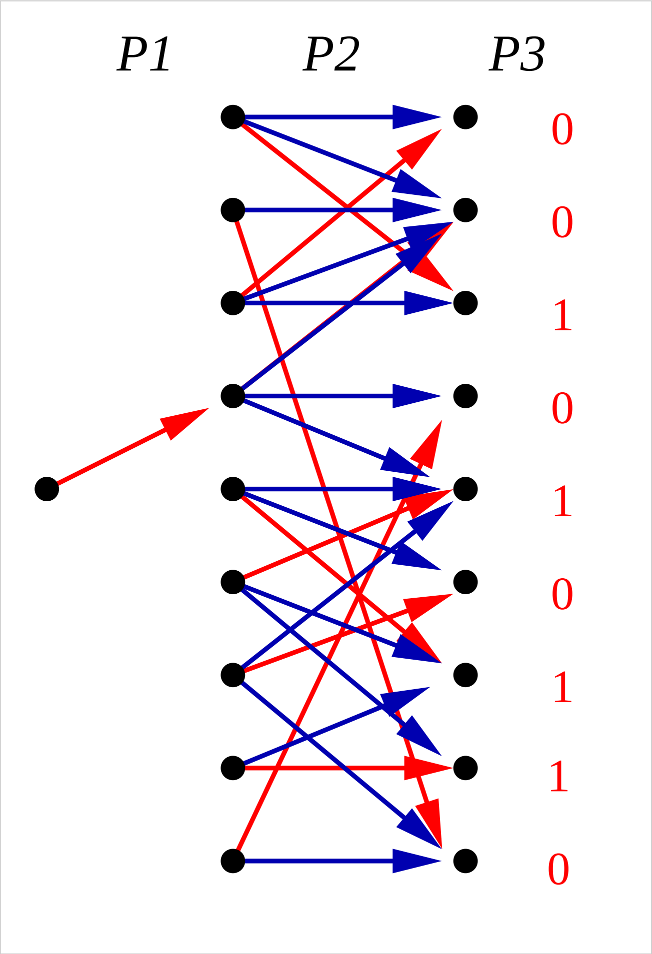
Total communication: $O(n \log^{(k-1)} n)$ bits.

The Pudlák-Rödl-Sgall Protocol



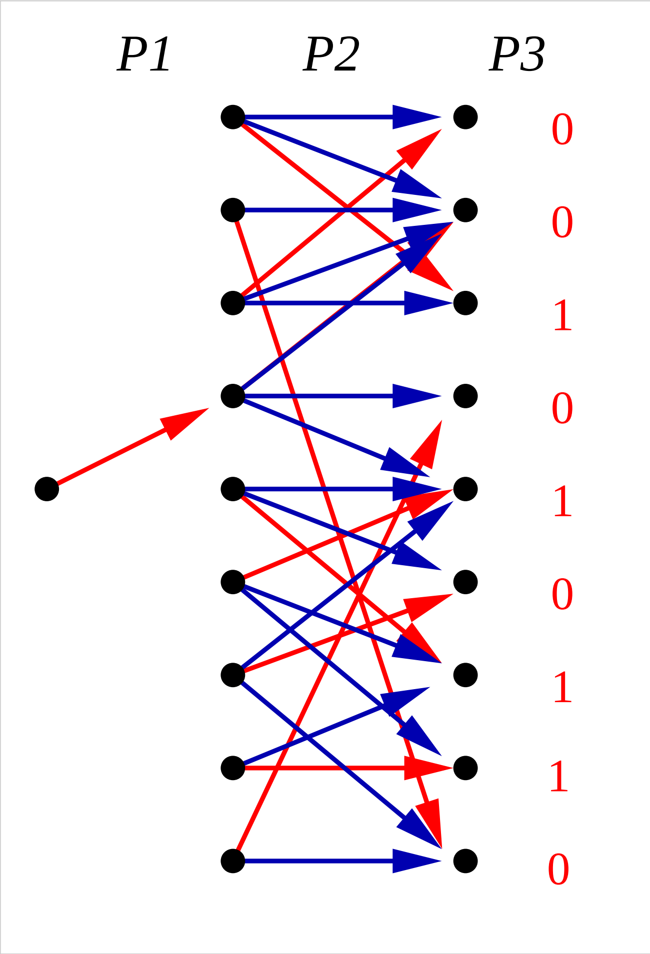
Step 0: Generate random bipartite graph H

The Pudlák-Rödl-Sgall Protocol



Step 0: Generate random bipartite graph H

The Pudlák-Rödl-Sgall Protocol

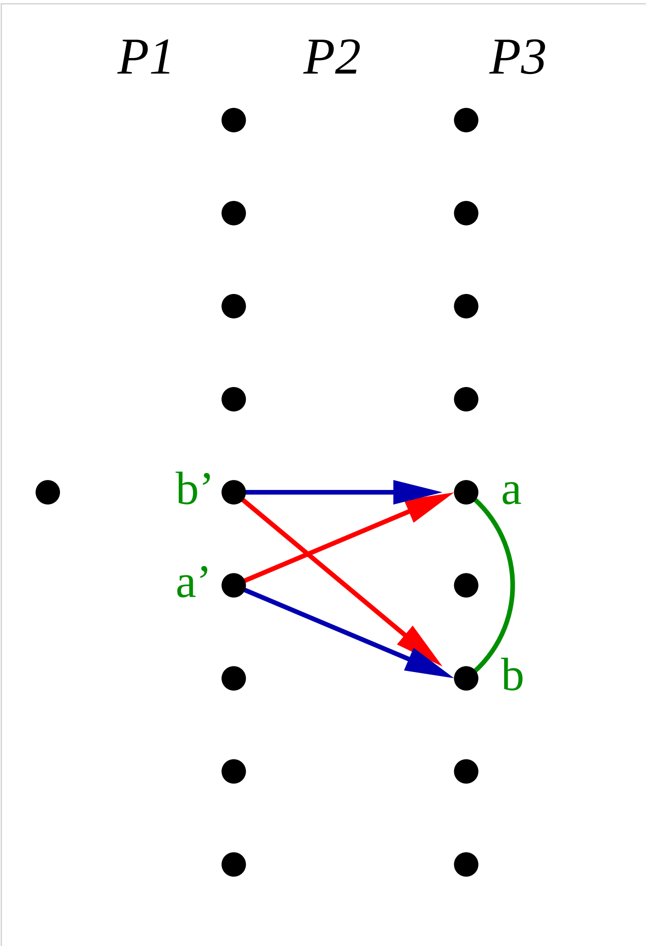


Step 0: Generate random bipartite graph H

Step 1: $P1$ sees π, knows H

- creates graph G_π on vertices in second layer
- $(a, b) \in E$ iff $(y_{\pi^{-1}(a)}, x_b)$ and $(y_{\pi^{-1}(b)}, x_a)$ in H

The Pudlák-Rödl-Sgall Protocol

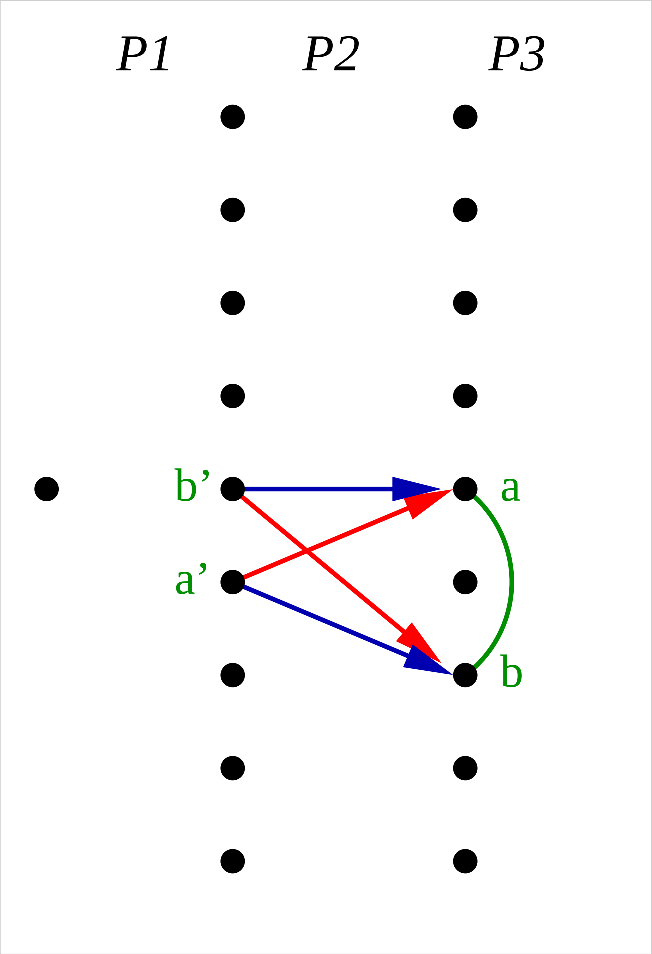


Step 0: Generate random bipartite graph H

Step 1: $P1$ sees π, knows H

- creates graph G_π on vertices in second layer
- $(a, b) \in E$ iff $(y_{\pi^{-1}(a)}, x_b)$ and $(y_{\pi^{-1}(b)}, x_a)$ in H

The Pudlák-Rödl-Sgall Protocol

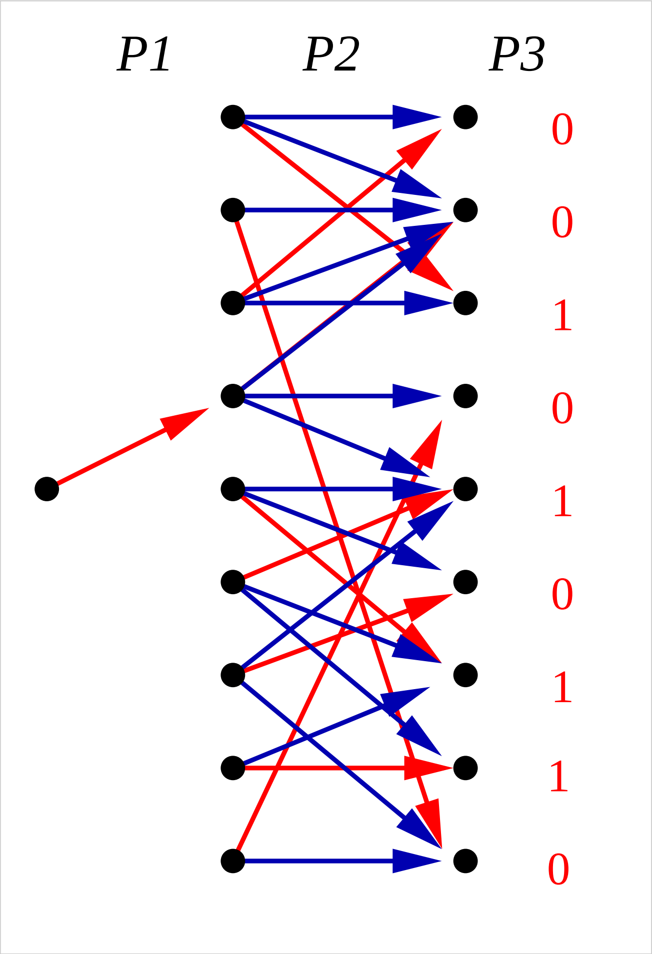


Step 0: Generate random bipartite graph H

Step 1: $P1$ sees π, knows H

- creates graph G_π on vertices in second layer
- $(a, b) \in E$ iff $(y_{\pi^{-1}(a)}, x_b)$ and $(y_{\pi^{-1}(b)}, x_a)$ in H
- Let $C_1, \ldots, C_r$ be a clique cover of G_π
- For each $1 \leq i \leq r$, $P1$ sends parity of bits in C_i

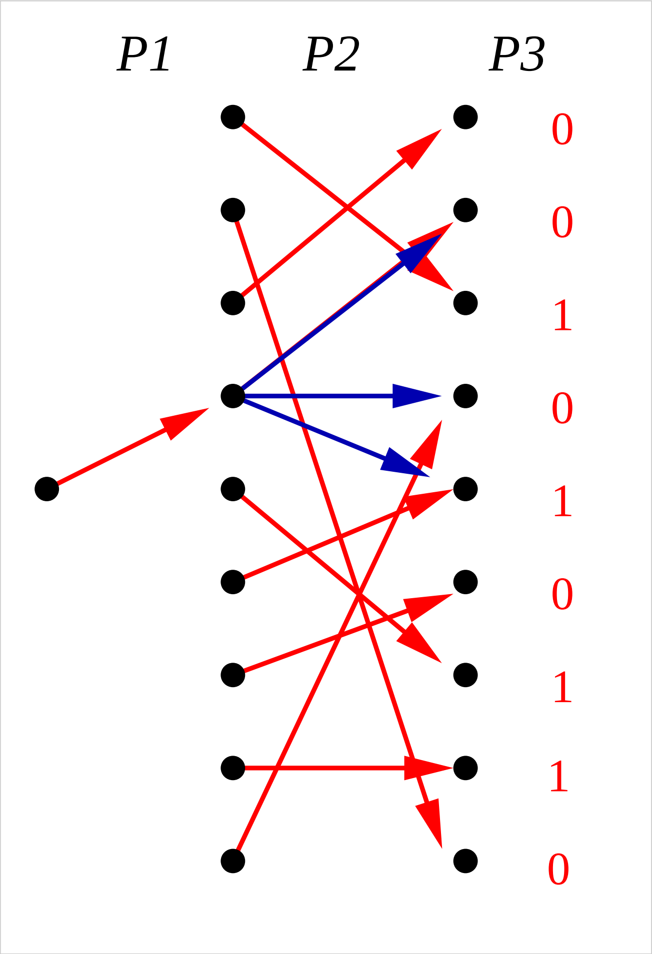
The Pudlák-Rödl-Sgall Protocol



Step 2: $P2$ sees i, knows H

- sends x_j for each $(y_i, x_j) \in H$

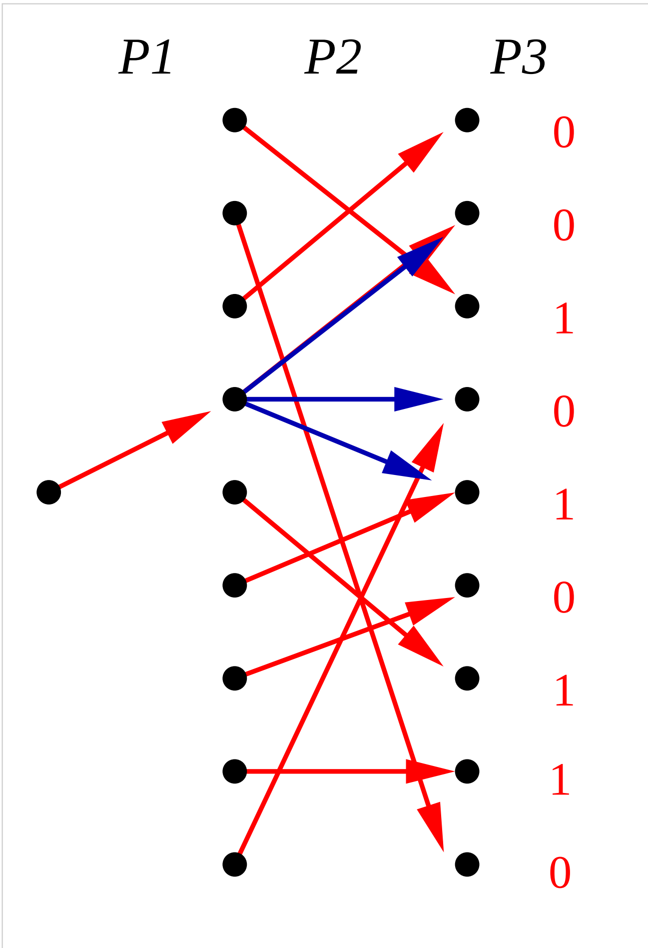
The Pudlák-Rödl-Sgall Protocol



Step 2: $P2$ sees i, knows H

- sends x_j for each $(y_i, x_j) \in H$

The Pudlák-Rödl-Sgall Protocol



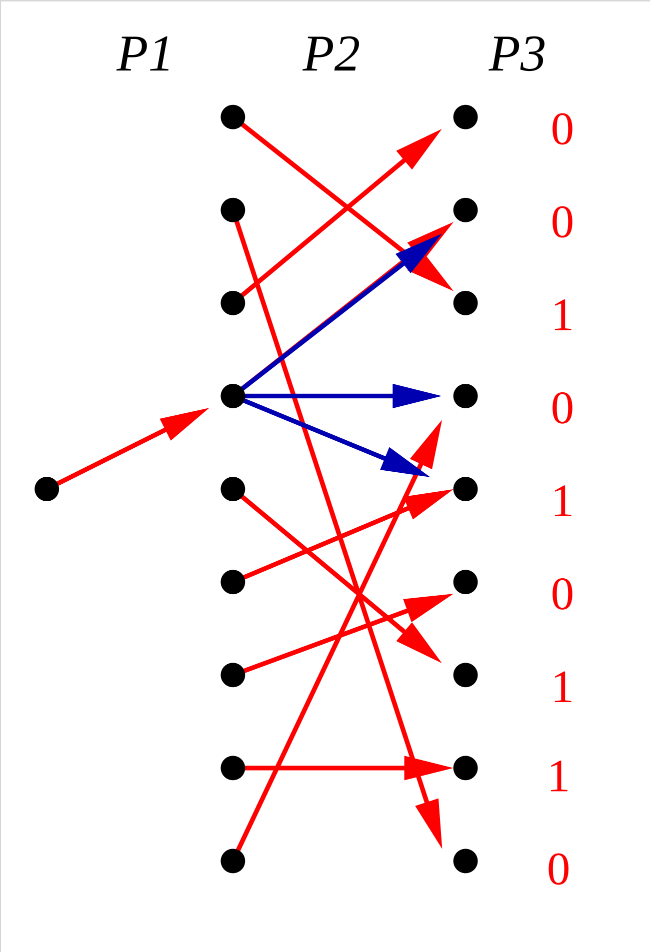
Step 2: $P2$ sees i, knows H

- sends x_j for each $(y_i, x_j) \in H$

Step 3: $P3$ sees i, π, knows H

- $C :=$ clique containing $\pi(i)$

The Pudlák-Rödl-Sgall Protocol



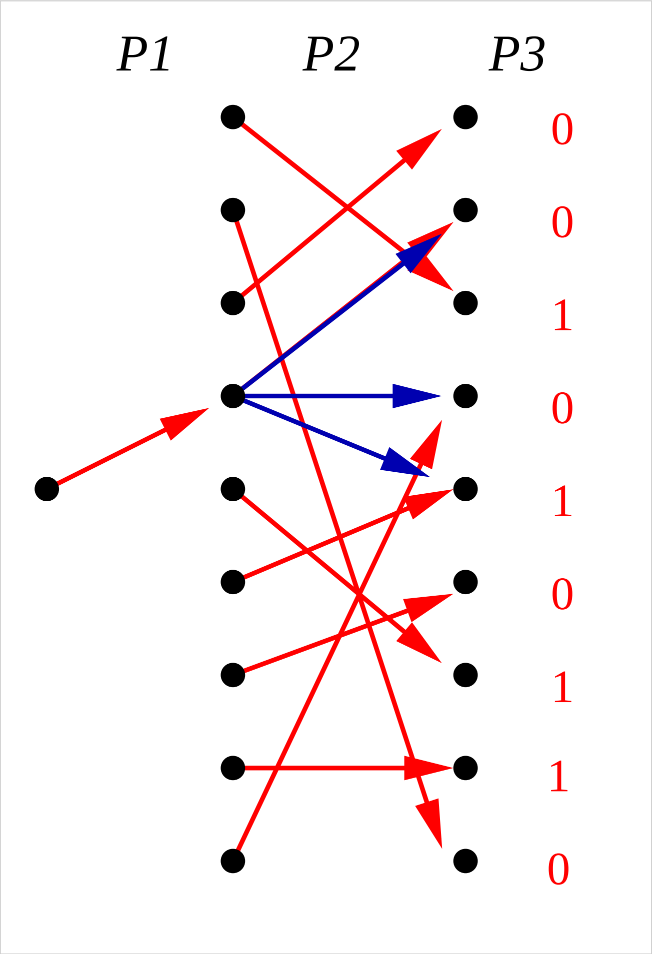
Step 2: $P2$ sees i, knows H

- sends x_j for each $(y_i, x_j) \in H$

Step 3: $P3$ sees i, π, knows H

- $C :=$ clique containing $\pi(i)$
- Note: $j \neq \pi(i) \in C \Rightarrow (j, \pi(i)) \in G_\pi$

The Pudlák-Rödl-Sgall Protocol



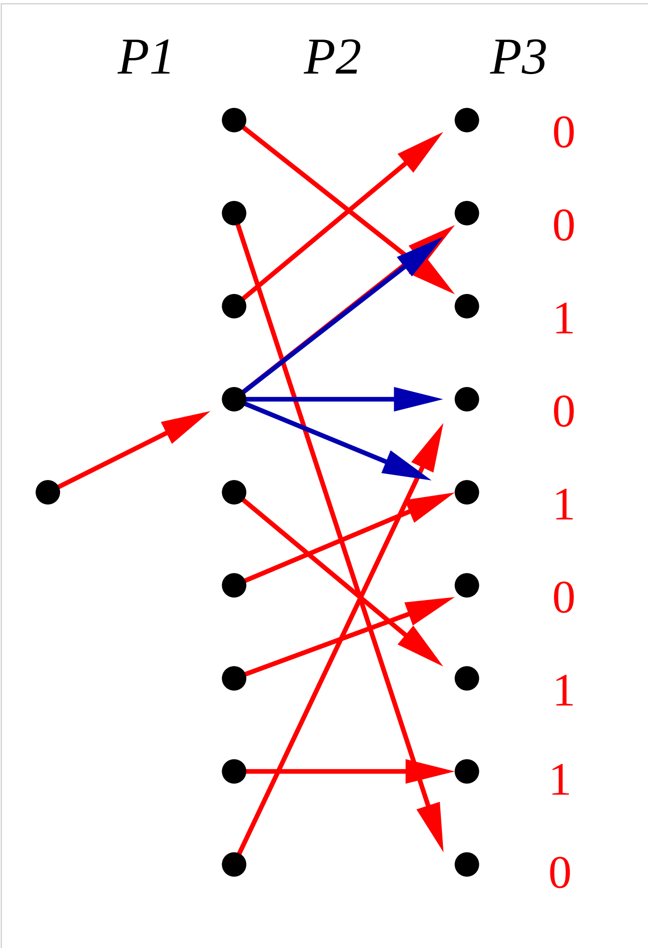
Step 2: $P2$ sees i, knows H

- sends x_j for each $(y_i, x_j) \in H$

Step 3: $P3$ sees i, π, knows H

- $C :=$ clique containing $\pi(i)$
- Note: $j \neq \pi(i) \in C \Rightarrow (j, \pi(i)) \in G_\pi$
- $\therefore (y_i, x_j) \in H \Rightarrow P2$ sent x_j.

The Pudlák-Rödl-Sgall Protocol



Step 2: $P2$ sees i, knows H

- sends x_j for each $(y_i, x_j) \in H$

Step 3: $P3$ sees i, π, knows H

- $C :=$ clique containing $\pi(i)$
- Note: $j \neq \pi(i) \in C \Rightarrow (j, \pi(i)) \in G_\pi$
- $\therefore (y_i, x_j) \in H \Rightarrow P2$ sent x_j.
- $P3$ takes clique bit, XORs out all $x_j \neq x_{\pi(i)}$, recovers $x_{\pi(i)}$.

PRS Protocol Analysis

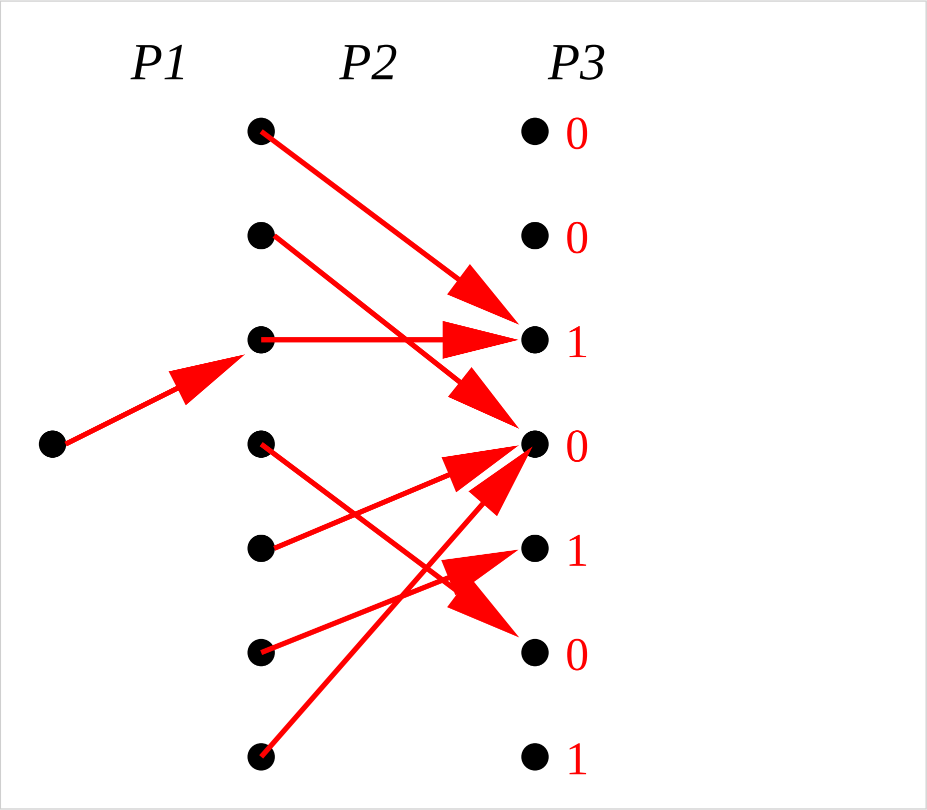
Lemma: [PRS'96], [Bollobás'88]

There exists a bipartite graph H such that for all i, π

1. G_π has $O\left(n\frac{\log\log n}{\log n}\right)$ cliques

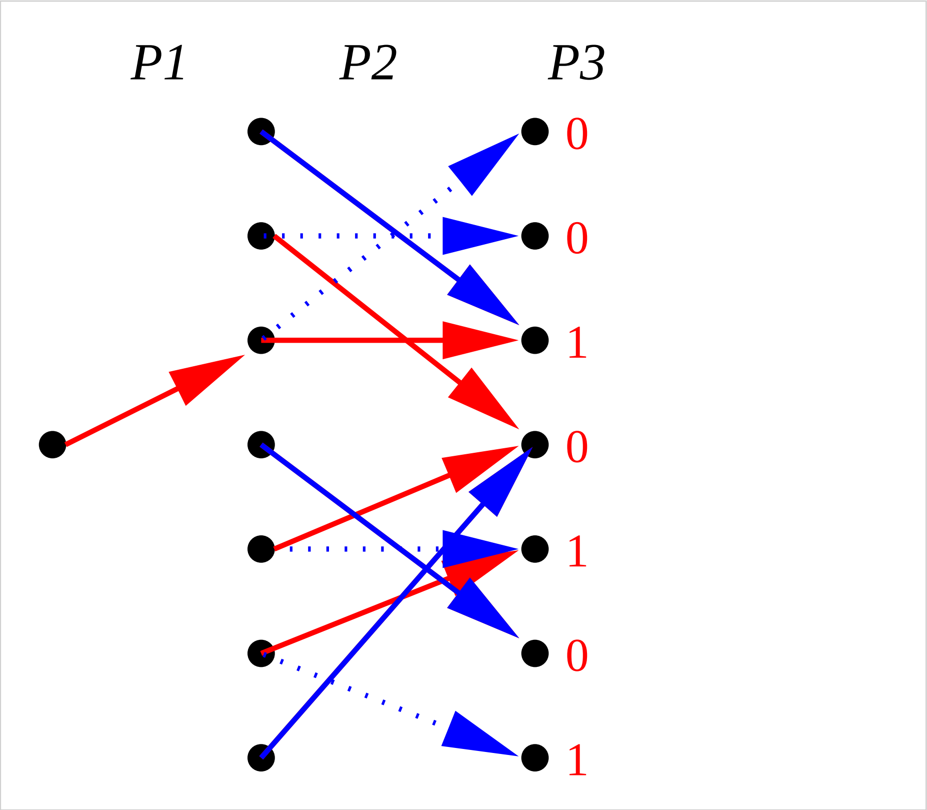
2. y_i has outdegree $O\left(n\frac{\log\log n}{\log n}\right)$

A General Protocol: 3 players



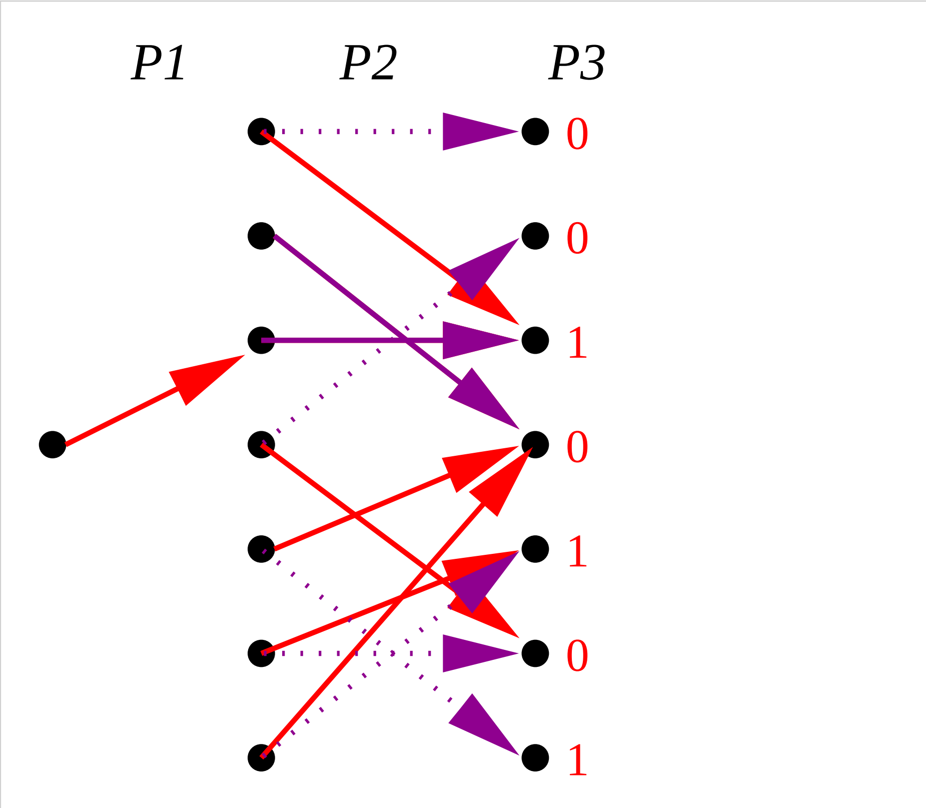
Idea: - Run PRS several times in parallel.

A General Protocol: 3 players



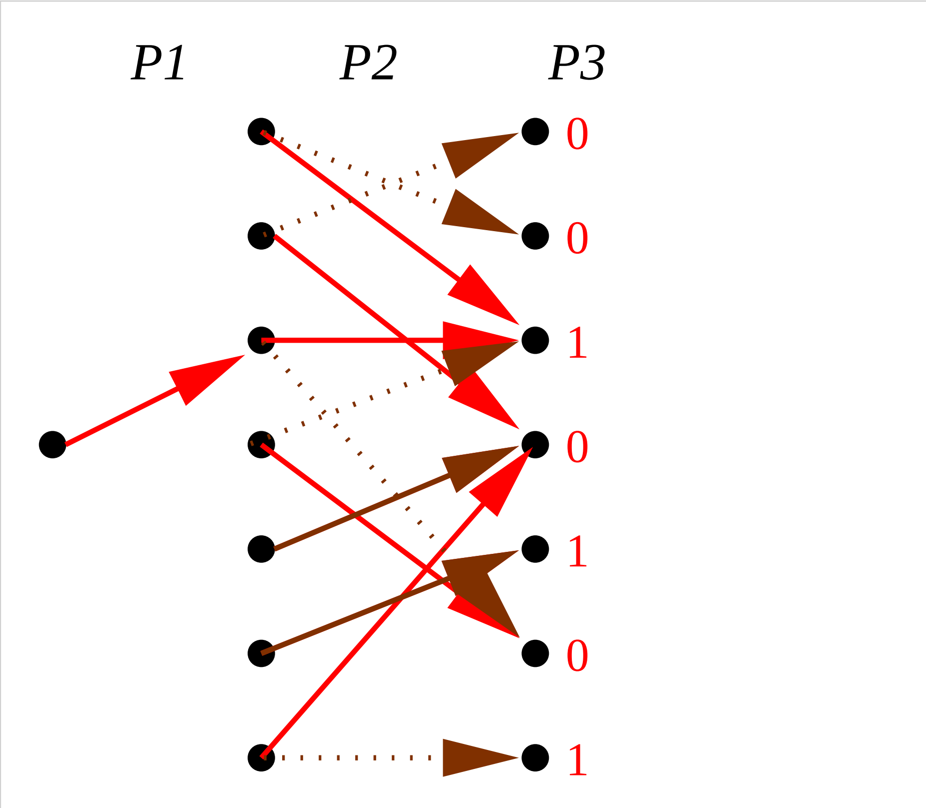
Idea:
- Run PRS several times in parallel.
- Pick permutations $\pi_1, \pi_2, \ldots, \pi_d$ such that $f(i) = \pi_j(i)$ for <u>some</u> permutation.

A General Protocol: 3 players



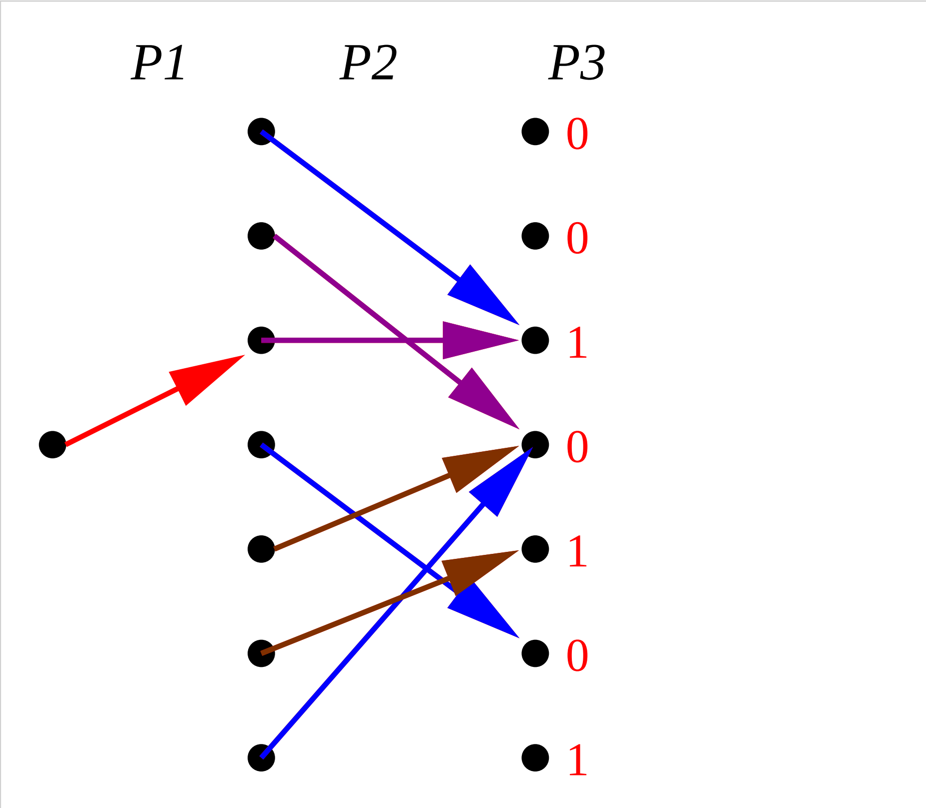
Idea:
- Run PRS several times in parallel.
- Pick permutations $\pi_1, \pi_2, \ldots, \pi_d$ such that $f(i) = \pi_j(i)$ for some permutation.

A General Protocol: 3 players



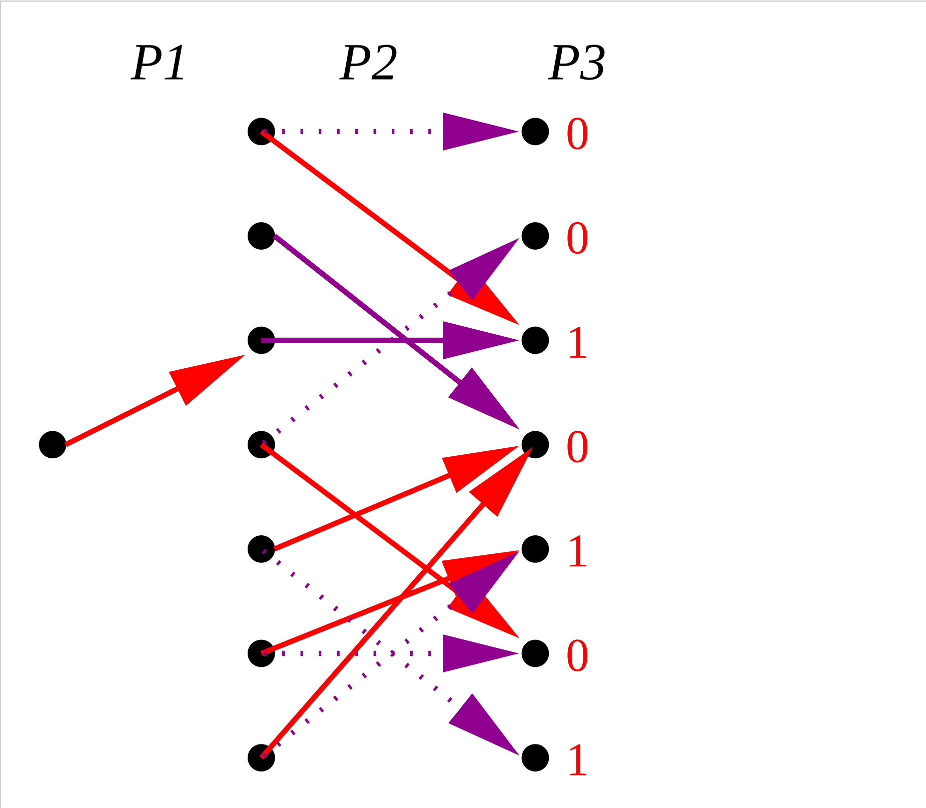
Idea:
- Run PRS several times in parallel.
- Pick permutations $\pi_1, \pi_2, \ldots, \pi_d$ such that $f(i) = \pi_j(i)$ for some permutation.

A General Protocol: 3 players



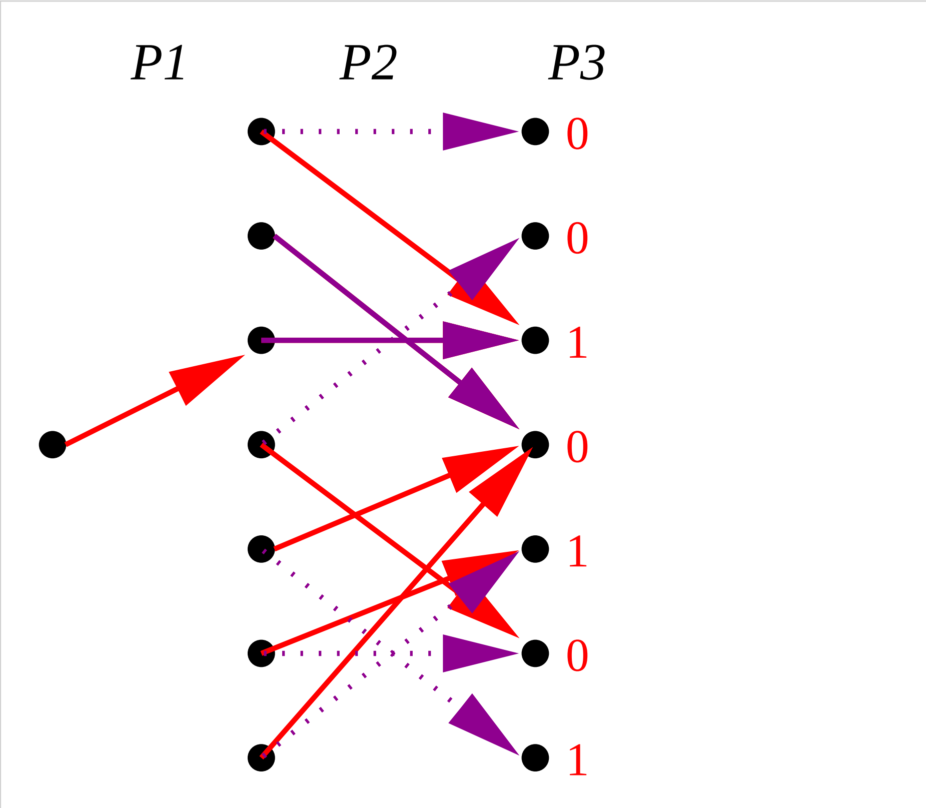
Idea:
- Run PRS several times in parallel.
- Pick permutations $\pi_1, \pi_2, \ldots, \pi_d$ such that $f(i) = \pi_j(i)$ for some permutation.

A General Protocol: 3 players



Idea:
- Run PRS several times in parallel.
- Pick permutations $\pi_1, \pi_2, \ldots, \pi_d$ such that $f(i) = \pi_j(i)$ for some permutation.
- $P3$ determines which permutation matches $f(i)$.

A General Protocol: 3 players



Idea:
- Run PRS several times in parallel.
- Pick permutations $\pi_1, \pi_2, \ldots, \pi_d$ such that $f(i) = \pi_j(i)$ for some permutation.
- $P3$ determines which permutation matches $f(i)$.

It turns out we can't do this efficiently, but we can get close enough.

Technical Details

Defintion: A set of permutations $A \subseteq S_n$ d-covers f if for all $i \in [n]$, one of the following conditions holds:

- There exists $\pi \in A$ such that $\pi(i) = f(i)$.
- $f(i)$ has a large preimage: $|f^{-1}(f(i))| > d$.

Technical Details

Defintion: A set of permutations $A \subseteq S_n$ d-covers f if for all $i \in [n]$, one of the following conditions holds:

- There exists $\pi \in A$ such that $\pi(i) = f(i)$.
- $f(i)$ has a large preimage: $|f^{-1}(f(i))| > d$.

Lemma: We can always find a set of d permutations that d-covers f.

Technical Details

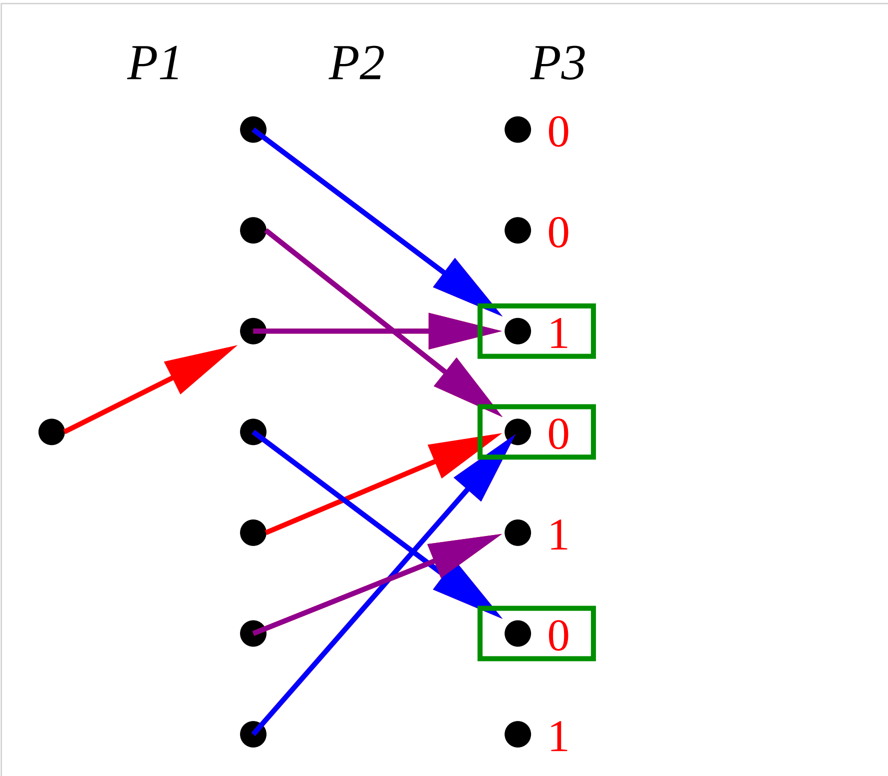
Defintion: A set of permutations $A \subseteq S_n$ d-covers f if for all $i \in [n]$, one of the following conditions holds:

- There exists $\pi \in A$ such that $\pi(i) = f(i)$.
- $f(i)$ has a large preimage: $|f^{-1}(f(i))| > d$.

Lemma: We can always find a set of d permutations that d-covers f.

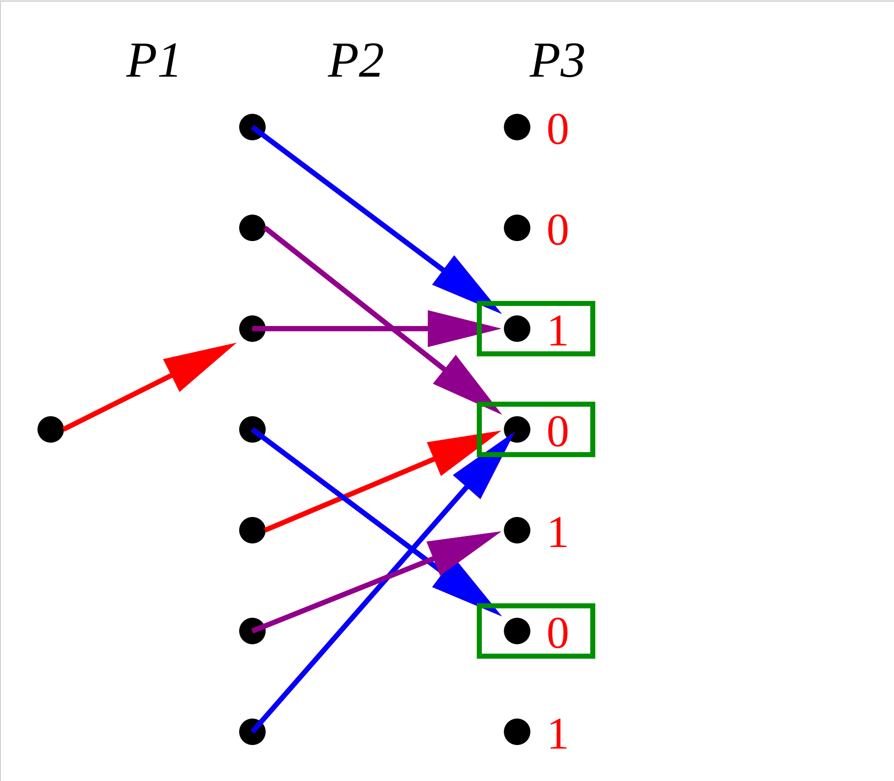
Note: There can be at most n/d points with large preimages.

A General Protocol: 3 players



Players agree on d and a d-covering set $A_d(f)$ for each f.

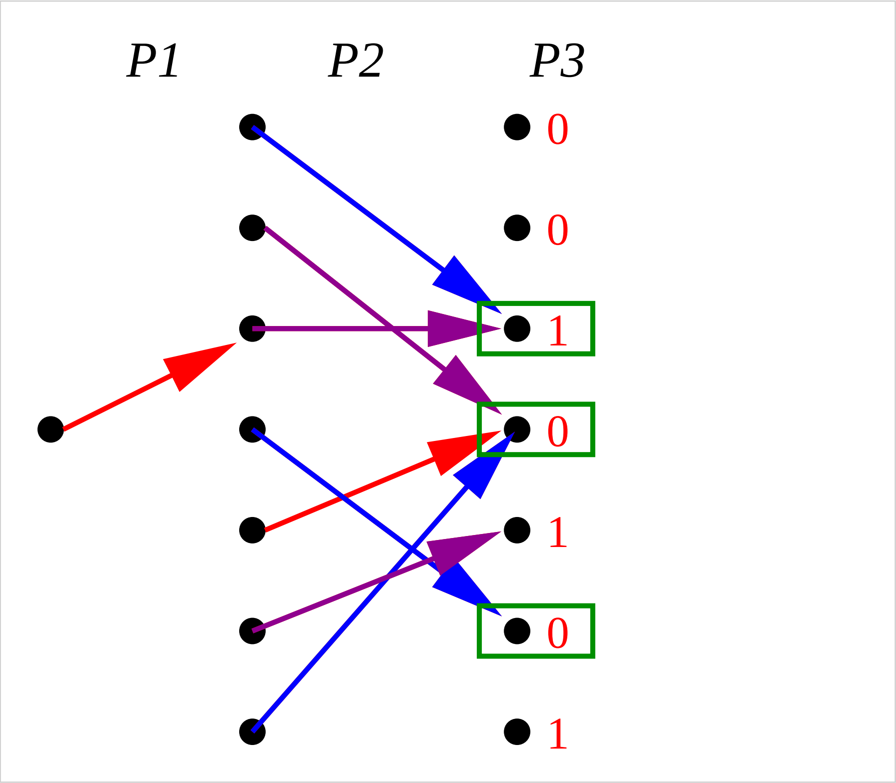
A General Protocol: 3 players



Players agree on d and a d-covering set $A_d(f)$ for each f.

- $P1$ sends $\{\alpha(\pi, x)\}_{\pi \in A_d(f)}$.
- $P1$ also sends $x[j]$ for any j with a large preimage.

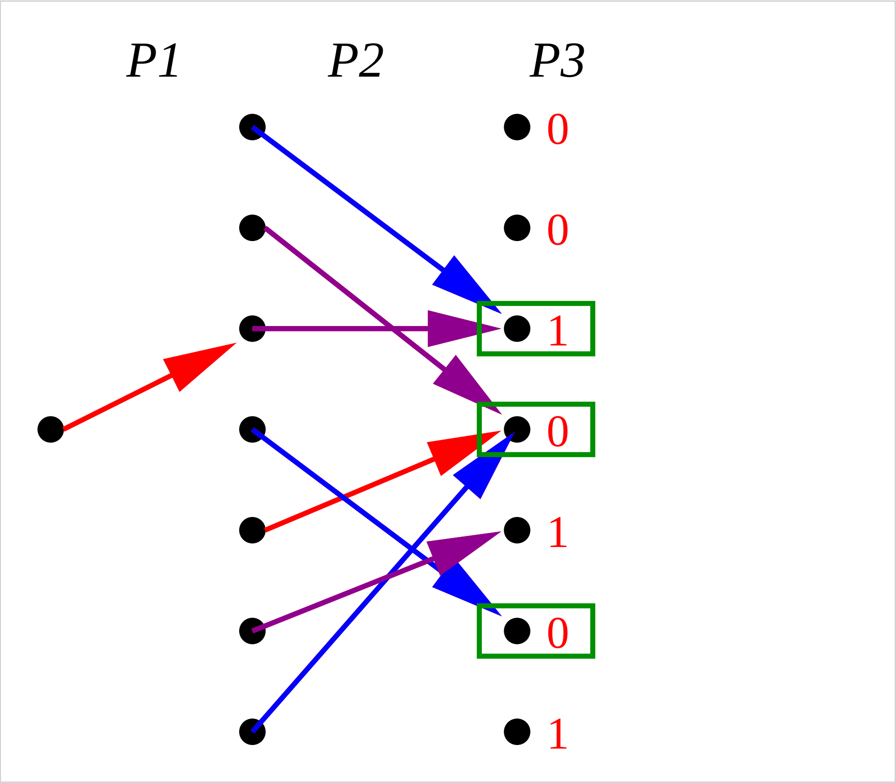
A General Protocol: 3 players



Players agree on d and a d-covering set $A_d(f)$ for each f.

- $P1$ sends $\{\alpha(\pi, x)\}_{\pi \in A_d(f)}$.
- $P1$ also sends $x[j]$ for any j with a large preimage.
- $P2$ sends $\{\beta(i, x, \alpha)\}_\alpha$.

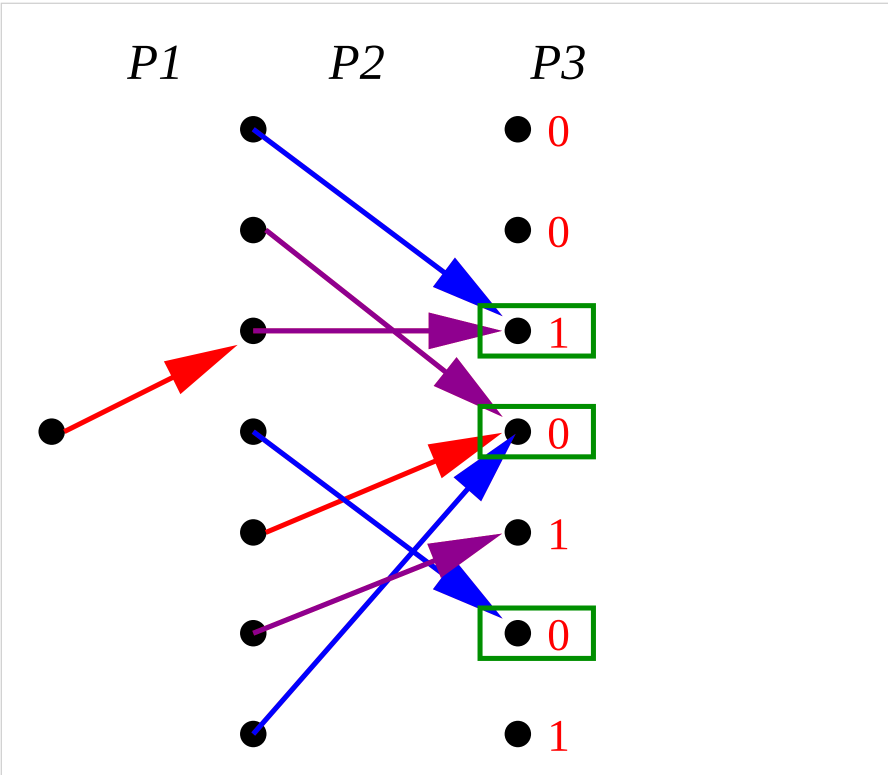
A General Protocol: 3 players



Players agree on d and a d-covering set $A_d(f)$ for each f.

- $P1$ sends $\{\alpha(\pi, x)\}_{\pi \in A_d(f)}$.
- $P1$ also sends $x[j]$ for any j with a large preimage.
- $P2$ sends $\{\beta(i, x, \alpha)\}_\alpha$.
- $P3$ recovers $x[f(i)]$ from PRS or from $P1$'s extra bits.

A General Protocol: 3 players



Players agree on d and a d-covering set $A_d(f)$ for each f.

- $P1$ sends $\{\alpha(\pi, x)\}_{\pi \in A_d(f)}$.
- $P1$ also sends $x[j]$ for any j with a large preimage.
- $P2$ sends $\{\beta(i, x, \alpha)\}_\alpha$.
- $P3$ recovers $x[f(i)]$ from PRS or from $P1$'s extra bits.

With $d = \sqrt{\frac{\log n}{\log \log n}}$, the protocol costs $O\left(n\sqrt{\frac{\log \log n}{\log n}}\right)$.

Talk Outline

- Multi-Party Communication Games
- The Multi-Party Pointer Jumping Problem
- Upper Bounds
- Restricted Protocols
- Conclusions

Restricted Protocols

Partial progress: protocols with more restricted forms of information sharing

- Myopic protocols: Pj only sees layers $1, \ldots, (j-1)$ as well as layer $(j+1)$ of graph. (i.e., limited visibility of layers ahead)

  [Gronemeier'06]

- Conservative protocols: Pj sees layers $(j+1), \ldots, k$ of graph, plus composition of layers $1, \ldots, (j-1)$. Doesn't see individual layers $1, \ldots, (j-1)$ themselves. (i.e., limited visibility of layers behind)

  [Damm-Jukna-Sgall'96]

Restricted Protocols

Partial progress: protocols with more restricted forms of information sharing

- Myopic protocols: Pj only sees layers $1, \ldots, (j-1)$ as well as layer $(j+1)$ of graph. (i.e., limited visibility of layers ahead)

  [Gronemeier'06]

- Conservative protocols: Pj sees layers $(j+1), \ldots, k$ of graph, plus composition of layers $1, \ldots, (j-1)$. Doesn't see individual layers $1, \ldots, (j-1)$ themselves. (i.e., limited visibility of layers behind)

  [Damm-Jukna-Sgall'96]

Note: The DJS protocol for $\widehat{\mathrm{MPJ}}_k$ is both myopic and conservative!

Restricted Protocols

Partial progress: protocols with more restricted forms of information sharing

- Myopic protocols: Pj only sees layers $1, \ldots, (j-1)$ as well as layer $(j+1)$ of graph. (i.e., limited visibility of layers ahead)

  [Gronemeier'06]

- Conservative protocols: Pj sees layers $(j+1), \ldots, k$ of graph, plus composition of layers $1, \ldots, (j-1)$. Doesn't see individual layers $1, \ldots, (j-1)$ themselves. (i.e., limited visibility of layers behind)

  [Damm-Jukna-Sgall'96]

Note: The DJS protocol for $\widehat{\mathrm{MPJ}}_k$ is both myopic and conservative!

[Chakrabarti'07] gave randomized lower bounds for restricted protocols:

- myopic: $\Omega(n/k)$ bits.
- conservative: $\Omega(n/k^2)$ bits.

Restricted Protocols

Partial progress: protocols with more restricted forms of information sharing

- Myopic protocols: Pj only sees layers $1, \ldots, (j-1)$ as well as layer $(j+1)$ of graph. (i.e., limited visibility of layers ahead)

  [Gronemeier'06]

- Conservative protocols: Pj sees layers $(j+1), \ldots, k$ of graph, plus composition of layers $1, \ldots, (j-1)$. Doesn't see individual layers $1, \ldots, (j-1)$ themselves. (i.e., limited visibility of layers behind)

  [Damm-Jukna-Sgall'96]

Note: The DJS protocol for $\widehat{\mathrm{MPJ}}_k$ is both myopic and conservative!

[Chakrabarti'07] gave randomized lower bounds for restricted protocols:

- myopic: $\Omega(n/k)$ bits.
- conservative: $\Omega(n/k^2)$ bits.

For the rest of this talk: all protocols are myopic.

Our Results

Question: Can there be <u>any</u> nontrivial myopic protocol for MPJ_k?

Our Results

Question: Can there be any nontrivial myopic protocol for MPJ_k?

No, but in an interesting way...

Our Results

Question: Can there be any nontrivial myopic protocol for MPJ_k?

No, but in an interesting way...

Theorem: In any myopic protocol for MPJ_k, some player must send at least $n/2$ bits.

Our Results

Question: Can there be any nontrivial myopic protocol for MPJ_k?

No, but in an interesting way...

Theorem: In any myopic protocol for MPJ_k, some player must send at least $n/2$ bits.

Definitions:

- $\mathrm{cost}(\mathcal{P})$:= cost of largest message of $\mathcal{P}$.
- $\mathrm{tcost}(\mathcal{P})$:= total cost of $\mathcal{P}$.
- δn-bit protocol: $\mathrm{cost}(\mathcal{P}) = \delta n$.

Detailed Results

Main Theorem: There exists a decreasing function $\phi : \mathbb{N} \to \mathbb{R}$ with $\lim_{k \to \infty} \phi(k) = \frac{1}{2}$ such that

1. Any deterministic protocol for $\textsc{MPJ}_k$ costs at least $\phi(k)n$ bits.

Detailed Results

Main Theorem: There exists a decreasing function $\phi : \mathbb{N} \to \mathbb{R}$ with $\lim_{k\to\infty} \phi(k) = \frac{1}{2}$ such that

1. Any deterministic protocol for $\textsc{mpj}_k$ costs at least $\phi(k)n$ bits.
2. There exists a protocol $\mathcal{P}$ for $\textsc{mpj}_k$ with $\text{cost}(\mathcal{P}) = \phi(k)n + o(n)$.

Detailed Results

Main Theorem: There exists a decreasing function $\phi : \mathbb{N} \to \mathbb{R}$ with $\lim_{k\to\infty} \phi(k) = \frac{1}{2}$ such that

1. Any deterministic protocol for MPJ_k costs at least $\phi(k)n$ bits.
2. There exists a protocol $\mathcal{P}$ for MPJ_k with $\text{cost}(\mathcal{P}) = \phi(k)n + o(n)$.

Theorem: Any deterministic protocol for MPJ_k has total cost at least n.

Theorem: If $\mathcal{P}$ is a deterministic protocol for $\widehat{\text{MPJ}}_k$ then

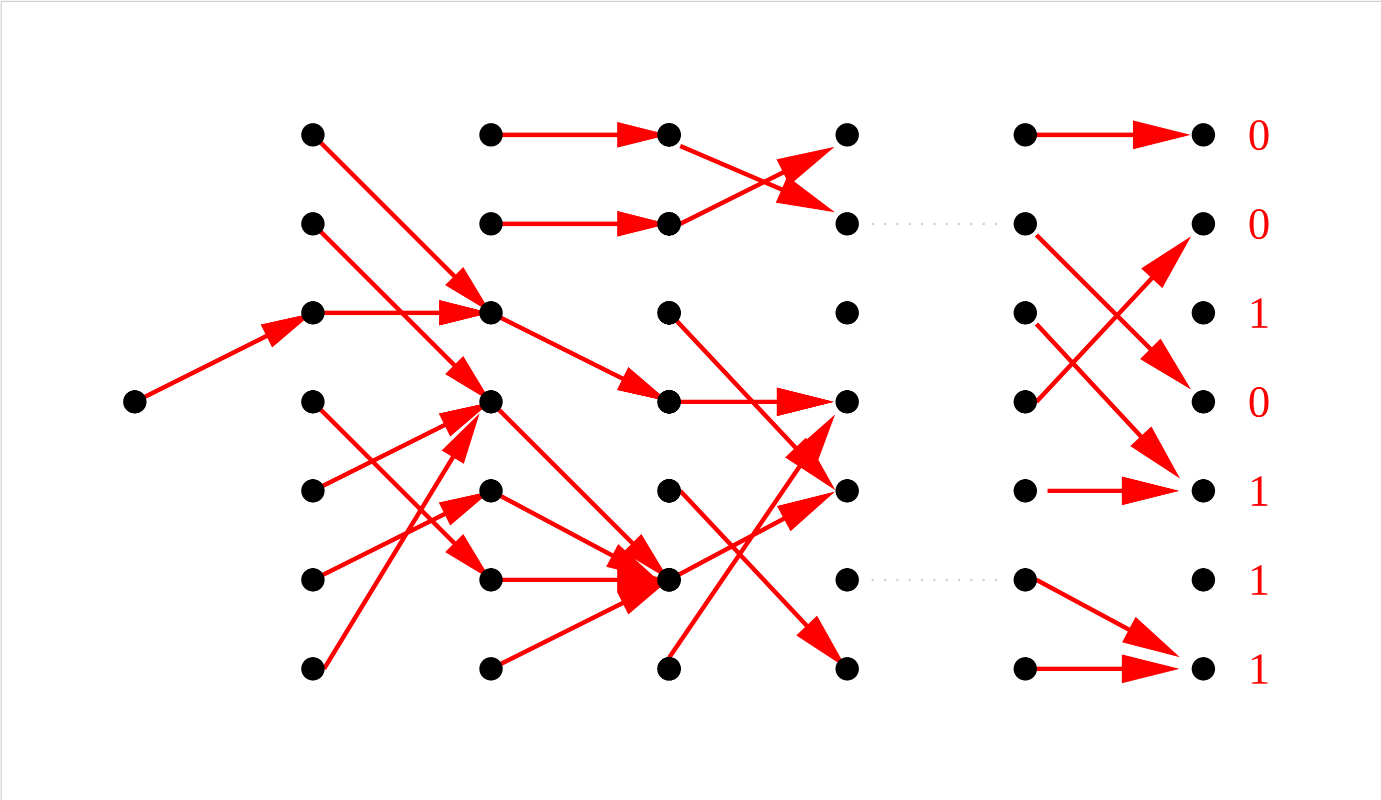
$$\text{cost}(\mathcal{P}) \geq n \left(\log^{(k-1)} n\right) (1 - o(1)).$$

Theorem: Any randomized protocol for MPJ_k has

$$\text{cost}(\mathcal{P}) = \Omega\left(\frac{n}{k \log n}\right).$$

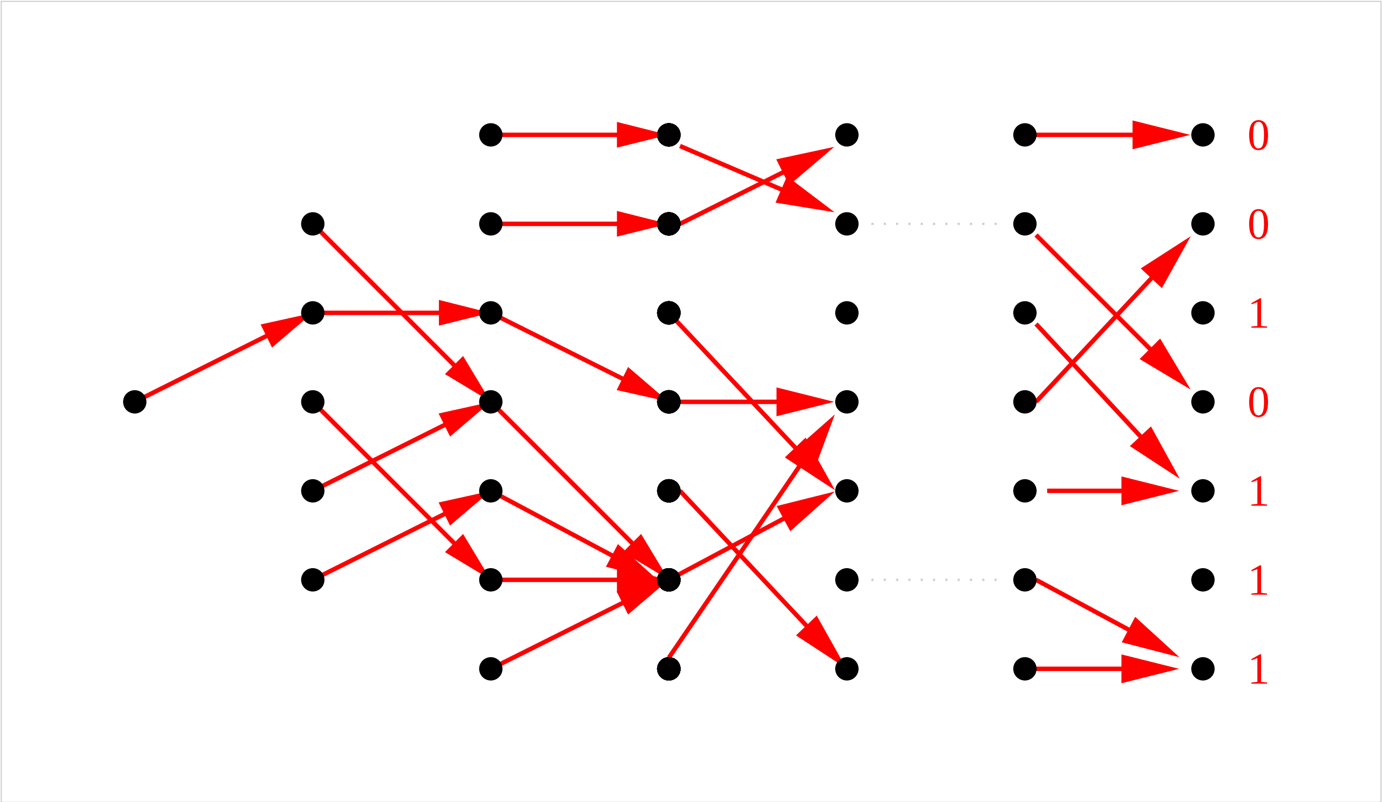
Generalized Pointer Jumping

$\mathrm{MPJ}_{m,k}$: just like MPJ_k, except $m \leq n$ vertices in first layer.



Generalized Pointer Jumping

$\mathrm{MPJ}_{m,k}$: just like MPJ_k, except $m \leq n$ vertices in first layer.



Round Elimination Lemma

Base Case Lemma: Any protocol $\mathcal{P}$ for $\textsc{mpj}_{m,2}$ has $\mathrm{cost}(\mathcal{P}) \geq m$ (INDEX)

Round Elimination Lemma: Let $k \geq 3$. If there is a δn-bit protocol $\mathcal{P}$ for $\textsc{mpj}_{m,k}$, then there is a δn-bit protocol $\mathcal{Q}$ for $\textsc{mpj}_{m',k-1}$ with $m' = n \cdot 2^{-\delta n/m}$.

Round Elimination Lemma

Base Case Lemma: Any protocol $\mathcal{P}$ for $\text{MPJ}_{m,2}$ has $\text{cost}(\mathcal{P}) \geq m$ (INDEX)

Round Elimination Lemma: Let $k \geq 3$. If there is a δn-bit protocol $\mathcal{P}$ for $\text{MPJ}_{m,k}$, then there is a δn-bit protocol $\mathcal{Q}$ for $\text{MPJ}_{m',k-1}$ with $m' = n \cdot 2^{-\delta n/m}$.

Message Sets:

- P1's input: $f_2 \in [n]^{[m]}$
- $M := M_{\mathsf{m}} = \{f_2 :$ P1 sends m on input $f_2\}$.
- Fix m to maximize $|M|$; then $|M| \geq \frac{n^m}{2^{\delta n}}$.

Round Elimination Lemma

Base Case Lemma: Any protocol $\mathcal{P}$ for $\mathrm{MPJ}_{m,2}$ has $\mathrm{cost}(\mathcal{P}) \geq m$ (INDEX)

Round Elimination Lemma: Let $k \geq 3$. If there is a δn-bit protocol $\mathcal{P}$ for $\mathrm{MPJ}_{m,k}$, then there is a δn-bit protocol $\mathcal{Q}$ for $\mathrm{MPJ}_{m',k-1}$ with $m' = n \cdot 2^{-\delta n/m}$.

Message Sets:

- P1's input: $f_2 \in [n]^{[m]}$
- $M := M_{\mathsf{m}} = \{f_2 :$ P1 sends m on input $f_2\}$.
- Fix m to maximize $|M|$; then $|M| \geq \frac{n^m}{2^{\delta n}}$.

Definition: For $\mathcal{F} \subseteq [n]^{[m]}$, $\mathrm{Range}(i, \mathcal{F}) := \{f_2(i) : f_2 \in \mathcal{F}\}$

Round Elimination Lemma

Base Case Lemma: Any protocol $\mathcal{P}$ for $\text{MPJ}_{m,2}$ has $\text{cost}(\mathcal{P}) \geq m$ (INDEX)

Round Elimination Lemma: Let $k \geq 3$. If there is a δn-bit protocol $\mathcal{P}$ for $\text{MPJ}_{m,k}$, then there is a δn-bit protocol $\mathcal{Q}$ for $\text{MPJ}_{m',k-1}$ with $m' = n \cdot 2^{-\delta n/m}$.

Message Sets:

- P1's input: $f_2 \in [n]^{[m]}$
- $M := M_{\mathsf{m}} = \{f_2 : \text{P1 sends } \mathsf{m} \text{ on input } f_2\}$.
- Fix m to maximize $|M|$; then $|M| \geq \frac{n^m}{2^{\delta n}}$.

Definition: For $\mathcal{F} \subseteq [n]^{[m]}$, $\text{Range}(i, \mathcal{F}) := \{f_2(i) : f_2 \in \mathcal{F}\}$

Range Lemma: If $|\mathcal{F}| \geq (m')^m$, then $\exists$ i with $|\text{Range}(i, \mathcal{F})| \geq m'$

Proof of Round Elimination Lemma

Base Case Lemma: Any protocol $\mathcal{P}$ for $\textsc{mpj}_{m,2}$ has $\mathrm{cost}(\mathcal{P}) \geq m$ (INDEX)

Round Elimination Lemma: Let $k \geq 3$. If there is a δn-bit protocol $\mathcal{P}$ for $\textsc{mpj}_{m,k}$, then there is a δn-bit protocol $\mathcal{Q}$ for $\textsc{mpj}_{m',k-1}$ with $m' = n \cdot 2^{-\delta n/m}$.

Proof:

Proof of Round Elimination Lemma

Base Case Lemma: Any protocol $\mathcal{P}$ for $\textsc{mpj}_{m,2}$ has $\mathrm{cost}(\mathcal{P}) \geq m$ ($\textsc{index}$)

Round Elimination Lemma: Let $k \geq 3$. If there is a δn-bit protocol $\mathcal{P}$ for $\textsc{mpj}_{m,k}$, then there is a δn-bit protocol $\mathcal{Q}$ for $\textsc{mpj}_{m',k-1}$ with $m' = n \cdot 2^{-\delta n/m}$.

Proof:

- Fix M. Note: $|M| \geq \frac{n^m}{2^{\delta n}} = 2^{m \log n - \delta n} = (m')^m$.
- By Range Lemma, $\exists\ i \in [m]$ s.t. $|\mathrm{Range}(i, M)| \geq m'$. Fix i.
- For each $j \in [m']$, fix $g_j \in M$ s.t. $g_j(i) = j$.
- Protocol $\mathcal{Q}$: on input $(j, f_3, \ldots, f_{k-1}, x)$, players simulate $\mathcal{P}$ on input $(i, g_j, f_3, \ldots, f_{k-1}, x)$.

Analysis

Define

- $a_0 := 0, a_\ell := \delta 2^{a_{\ell-1}}$
- $m_\ell := n2^{-a_\ell}$

$a_0 = 0$

Definition: Let $\phi(k) :=$ least δ such that $a_{k-1} \geq 1$

Analysis

Define

- $a_0 := 0, a_\ell := \delta 2^{a_{\ell-1}}$
- $m_\ell := n 2^{-a_\ell}$

$a_1 = \delta$

Definition: Let $\phi(k) :=$ least δ such that $a_{k-1} \geq 1$

Analysis

Define

- $a_0 := 0, a_\ell := \delta 2^{a_{\ell-1}}$
- $m_\ell := n 2^{-a_\ell}$

$$a_2 = \delta 2^\delta$$

Definition: Let $\phi(k) :=$ least δ such that $a_{k-1} \geq 1$

Analysis

Define

- $a_0 := 0, a_\ell := \delta 2^{a_{\ell-1}}$
- $m_\ell := n2^{-a_\ell}$

$$a_3 = \delta 2^{\delta 2^{\delta}}$$

Definition: Let $\phi(k) :=$ least δ such that $a_{k-1} \geq 1$

Analysis

Define

- $a_0 := 0, a_\ell := \delta 2^{a_{\ell-1}}$
- $m_\ell := n2^{-a_\ell}$

$$a_4 = \delta 2^{\delta 2^{\delta 2^{\delta}}}$$

Definition: Let $\phi(k) :=$ least δ such that $a_{k-1} \geq 1$

Analysis

Define

- $a_0 := 0, a_\ell := \delta 2^{a_{\ell-1}}$
- $m_\ell := n2^{-a_\ell}$

$$a_\ell = \delta 2^{\delta 2^{\delta 2^{\delta 2^{\cdot^{\cdot^{\cdot}}}}}}$$

Definition: Let $\phi(k) :=$ least δ such that $a_{k-1} \geq 1$

Analysis

Define

- $a_0 := 0, a_\ell := \delta 2^{a_{\ell-1}}$
- $m_\ell := n2^{-a_\ell}$

$$a_\ell = \delta 2^{\delta 2^{\delta 2^{\delta 2^{\cdot^{\cdot^{\cdot}}}}}}$$

Definition: Let $\phi(k) :=$ least δ such that $a_{k-1} \geq 1$

Claim: $\lim_{k\to\infty} \phi(k) = 1/2$ (Induction)

Round elimination ($m = m_\ell$):

$$m' = n2^{-\frac{\delta n}{m_\ell}} = n2^{-\delta n / n2^{-a_\ell}} = n2^{-\delta 2^{a_\ell}} = n2^{-a_{\ell+1}} = m_{\ell+1}$$

Proof of Main Theorem

Theorem: Any myopic protocol $\mathcal{P}$ for $\text{MPJ}_k = \text{MPJ}_{n,k}$ has

$$\text{cost}(\mathcal{P}) \geq n\phi(k).$$

Proof:

Proof of Main Theorem

Theorem: Any myopic protocol $\mathcal{P}$ for $\mathrm{MPJ}_k = \mathrm{MPJ}_{n,k}$ has

$$\mathrm{cost}(\mathcal{P}) \geq n\phi(k).$$

Proof:

δn-bit protocol for $\mathrm{MPJ}_{m_0,k} \Rightarrow$

$\ldots k-2$ round eliminations $\ldots \Rightarrow$

δn-bit protocol for $\mathrm{MPJ}_{m_{k-2},2} \Rightarrow$

$\delta n \geq n2^{-a_{k-2}} = m_{k-2}$ (Base Case Lemma) $\Rightarrow$

$a_{k-1} = \delta 2^{a_{k-2}} \geq 1 \Rightarrow$

$\delta \geq \phi(k)$ (by def. of $\phi(k)$)

A Sketch of Matching Upper Bound

Idea: Cover $[n]^{[m]}$ with sets $S_1, \ldots, S_t \subseteq [n]^{[m]}$ s.t.

$$|\text{Range}(i, S)| = m' \text{ for all } i, S.$$

Packing lower bound: $t \geq 2^{\delta n}$.

Claim: $t \leq 2^{\delta n + o(n)}$. (Prob. Method)

Protocol:

- P1 sends $S \ni f_2$. (cost $= \delta n + o(n)$)
- Players 2,. . . ,k see i, set $[m'] := Range(i, S)$.
- Players 2,. . . ,k run $\text{MPJ}_{m',k-1}$ protocol on $(f_2(i), f_3, \ldots, x)$.

Randomizing the Lower Bound

Round Elimination Lemma: Let $k \geq 3$. If there is a δn-bit, ε-error distributional protocol $\mathcal{P}$ for $\text{MPJ}_{m,k}$, then there is a δn-bit, ε'-error protocol $\mathcal{Q}$ for $\text{MPJ}_{m',k-1}$ with $m' = n \cdot 2^{-2\delta n/m}$ and $\varepsilon' = 2n\varepsilon$.

Proof:

Randomizing the Lower Bound

Round Elimination Lemma: Let $k \geq 3$. If there is a δn-bit, ε-error distributional protocol $\mathcal{P}$ for $\mathrm{MPJ}_{m,k}$, then there is a δn-bit, ε'-error protocol $\mathcal{Q}$ for $\mathrm{MPJ}_{m',k-1}$ with $m' = n \cdot 2^{-2\delta n/m}$ and $\varepsilon' = 2n\varepsilon$.

Proof:

- $z := (f_3, \ldots, f_{k-1}, x)$
- Call (i, f_2) bad if $\Pr_z[\text{error } |(i, f_2)] > 2n\varepsilon$
- Call f_2 bad if $\Pr_i[(i, f_2) \text{ bad } | f_2] \geq 1/n$

Randomizing the Lower Bound

Round Elimination Lemma: Let $k \geq 3$. If there is a δn-bit, ε-error distributional protocol $\mathcal{P}$ for $\text{MPJ}_{m,k}$, then there is a δn-bit, ε'-error protocol $\mathcal{Q}$ for $\text{MPJ}_{m',k-1}$ with $m' = n \cdot 2^{-2\delta n/m}$ and $\varepsilon' = 2n\varepsilon$.

Proof:

- $z := (f_3, \ldots, f_{k-1}, x)$
- Call (i, f_2) bad if $\Pr_z[\text{error } |(i, f_2)] > 2n\varepsilon$

  $\Rightarrow \Pr[(i, f_2) \text{ bad}] < 1/2n$ (Markov)
- Call f_2 bad if $\Pr_i[(i, f_2) \text{ bad } |f_2] \geq 1/n$

  $\Rightarrow \Pr[f_2 \text{ bad}] < 1/2$ (Markov)

Randomizing the Lower Bound

Round Elimination Lemma: Let $k \geq 3$. If there is a δn-bit, ε-error distributional protocol $\mathcal{P}$ for $\text{MPJ}_{m,k}$, then there is a δn-bit, ε'-error protocol $\mathcal{Q}$ for $\text{MPJ}_{m',k-1}$ with $m' = n \cdot 2^{-2\delta n/m}$ and $\varepsilon' = 2n\varepsilon$.

Proof:

- $z := (f_3, \ldots, f_{k-1}, x)$
- Call (i, f_2) bad if $\Pr_z[\text{error } |(i, f_2)] > 2n\varepsilon$

  $\Rightarrow \Pr[(i, f_2) \text{ bad}] < 1/2n$ (Markov)
- Call f_2 bad if $\Pr_i[(i, f_2) \text{ bad } |f_2] \geq 1/n$

  $\Rightarrow \Pr[f_2 \text{ bad}] < 1/2$ (Markov)

  Note: f_2 good $\Rightarrow (i, f_2)$ good for all i.

Randomizing the Lower Bound

Round Elimination Lemma: Let $k \geq 3$. If there is a δn-bit, ε-error distributional protocol $\mathcal{P}$ for $\text{MPJ}_{m,k}$, then there is a δn-bit, ε'-error protocol $\mathcal{Q}$ for $\text{MPJ}_{m',k-1}$ with $m' = n \cdot 2^{-2\delta n/m}$ and $\varepsilon' = 2n\varepsilon$.

Proof:

- $z := (f_3, \ldots, f_{k-1}, x)$
- Call (i, f_2) bad if $\Pr_z[\text{error } |(i, f_2)] > 2n\varepsilon$

  $\Rightarrow \Pr[(i, f_2) \text{ bad}] < 1/2n$ (Markov)
- Call f_2 bad if $\Pr_i[(i, f_2) \text{ bad } | f_2] \geq 1/n$

  $\Rightarrow \Pr[f_2 \text{ bad}] < 1/2$ (Markov)

  Note: f_2 good $\Rightarrow (i, f_2)$ good for all i.
- Follow deterministic proof

  $M := M_{\mathsf{m}} = \{\text{good } f_2 : \text{P1 sends } \mathsf{m} \text{ on input } f_2\} \ \ldots$

Conclusions/Open Problems

Conclusions

- Still far from proving $\text{MPJ}_k \notin \mathsf{ACC}^0$
- Provided the first $o(n)$ protocol for MPJ_k
- Characterized maximum communication complexity of myopic protocols up to $1 + o(1)$ factors.
- Lower bound technique applies to MPJ_k and $\widehat{\text{MPJ}}_k$ and does randomize; seems promising for other problems.

Open Problems

1. Settle $D(\text{MPJ}_k)$
2. Possible first step: improve bound on MPJ_3
3. Relax protocol restrictions: 2-myopic, ...

Thank you!

Questions?

Contact jbrody@cs.dartmouth.edu