

Sublinear Protocols for Multi-Party Pointer Jumping

Joshua Brody and Amit Chakrabarti

DARTMOUTH COLLEGE

HANOVER, NH, USA

25th STACS, 2008, Bordeaux

Multi-Player Communication Protocols



Multiparty Communication Games

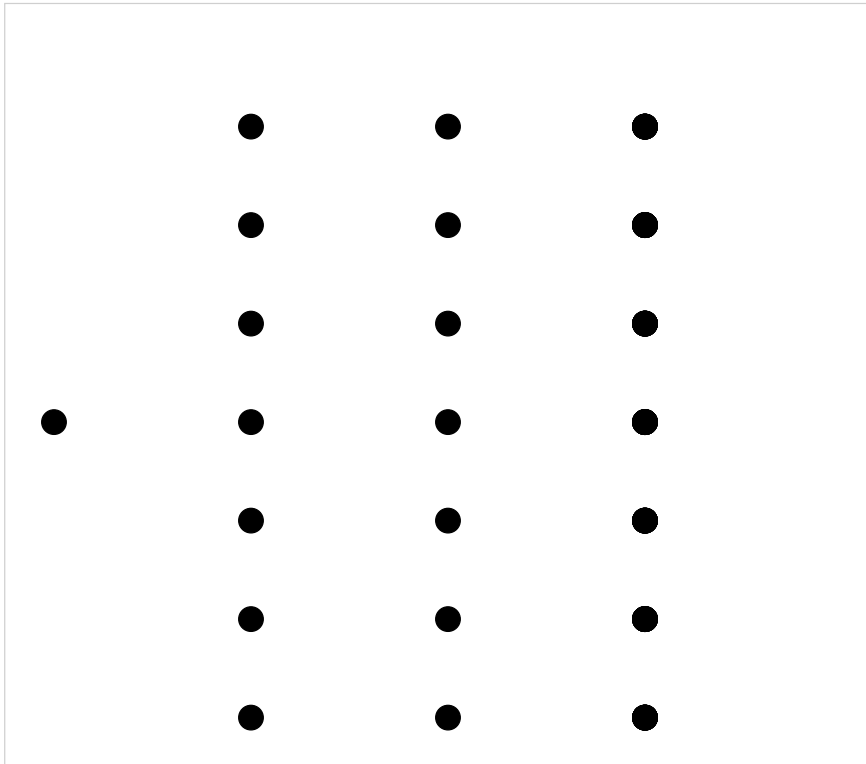
Input $x = (x_1, \dots, x_k)$ is split between k players.

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

Our model of communication:

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

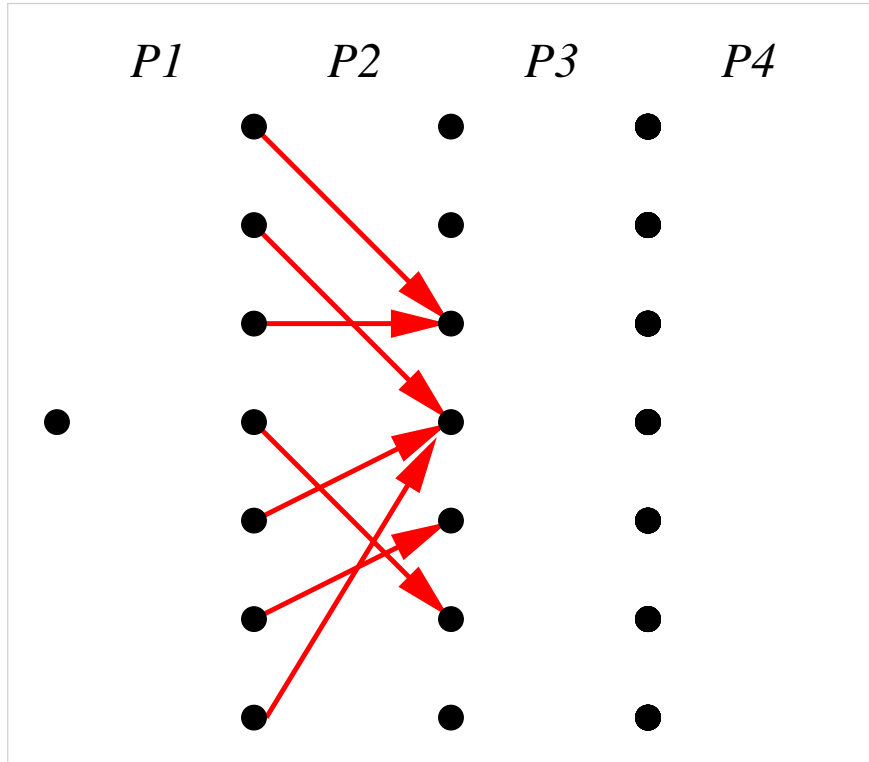
Pointer Jumping



Vertices: $k - 1$ layers, plus start vertex

- layers have n vertices

Pointer Jumping



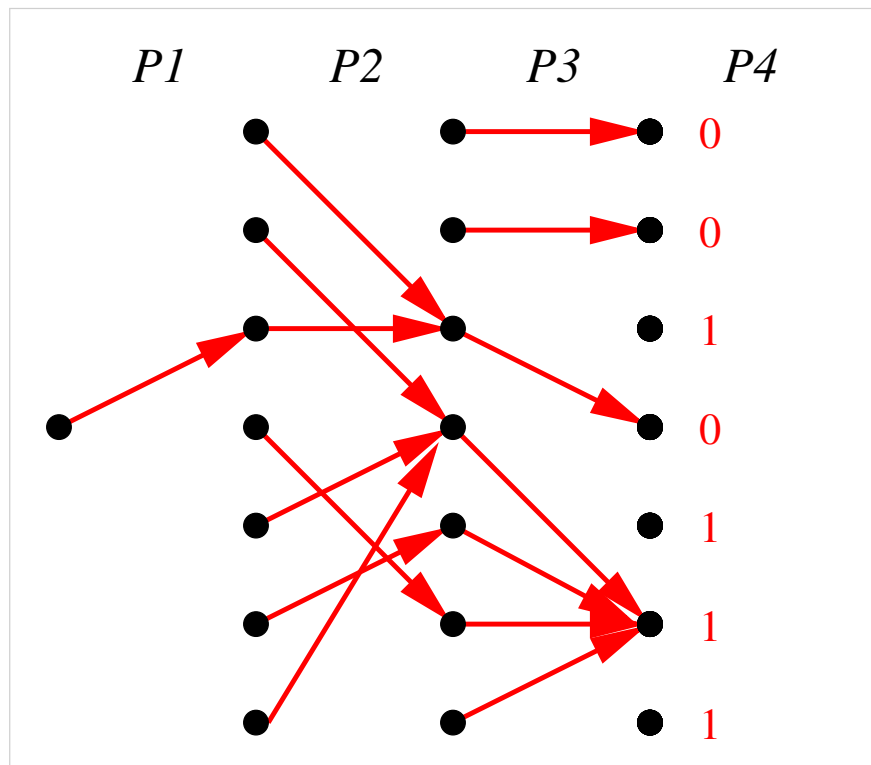
Vertices: $k - 1$ layers, plus start vertex

- layers have n vertices

Input:

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

Pointer Jumping



Vertices: $k - 1$ layers, plus start vertex

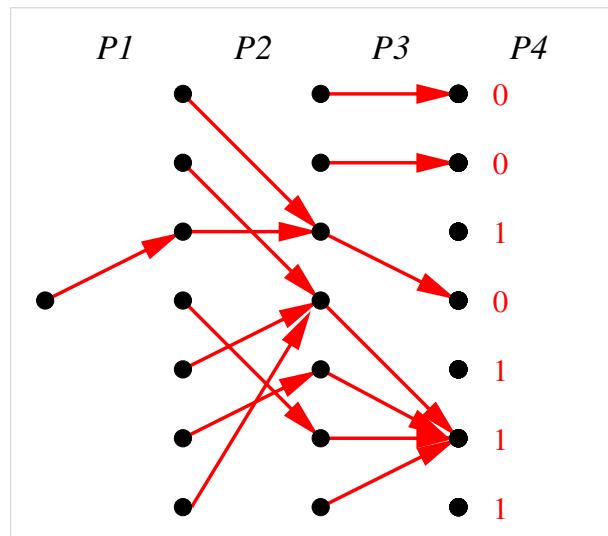
- layers have n vertices

Input:

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

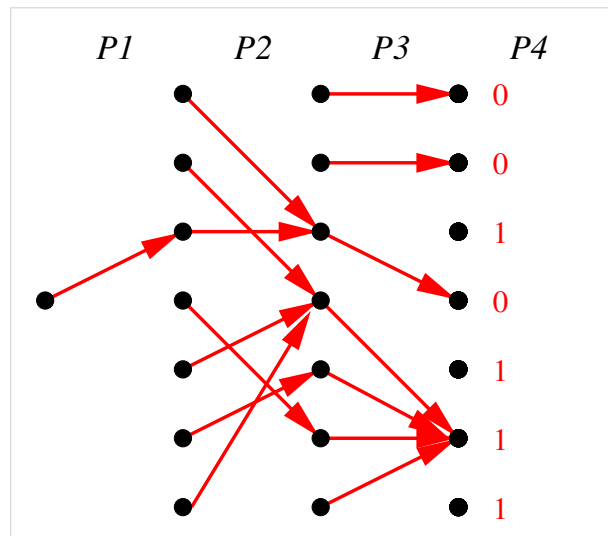
Compute $\text{MPJ}_k = \text{bit reached by following pointers from start vertex.}$

Pointer Jumping: Trivial Bounds



- One-way: any order except $P1, P2, \dots, Pk$: $O(\log n)$
- One way: in the order $P1, P2, \dots, Pk$: $O(n)$

Pointer Jumping: Trivial Bounds



- One-way: any order except $P1, P2, \dots, Pk$: $O(\log n)$
- One way: in the order $P1, P2, \dots, Pk$: $O(n)$
 - Problem seems hard: maybe $\Omega(n)$ lower bound?

Motivation

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

- No function $f \notin ACC^0$ is known.
- If $f : \{0, 1\}^n \rightarrow \{0, 1\}$ and $f \in ACC^0$, then f has 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]
- median finding in randomly-ordered streams [Chakrabarti-Cormode-McGregor'08]

Previous Results

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

- Simultaneous: $\Omega(n^{1/(k-1)})$ for MPJ_k [Babai-Gál-Kimmel-Lokam'95]
- One-way: $O(n \frac{\log \log n}{\log n})$ for $\text{MPJ}_3^{\text{perm}}$ [Pudlák-Rödl-Sgall'97]
 - middle layer must be permutation.
- One-way: $\Omega(n^{1/2})$ for MPJ_3 [Wigderson'97]
- One-way: $\Omega(n^{1/(k-1)}/k^k)$ for MPJ_k [Viola-Wigderson'07]

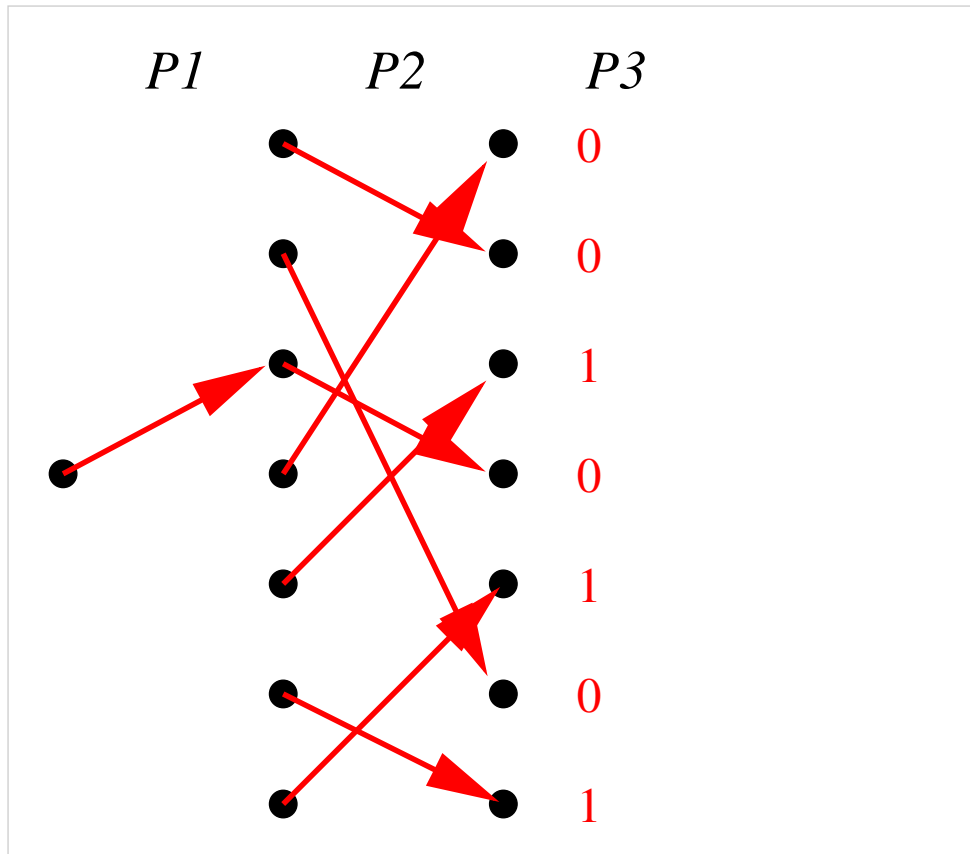
Our Results

- First sublinear protocol for MPJ_k .
- Defined a restricted form of information sharing: **Collapsing Protocols**.
- Derived $n - O(\log n)$ lower bound for **collapsing protocols**.

Our Results

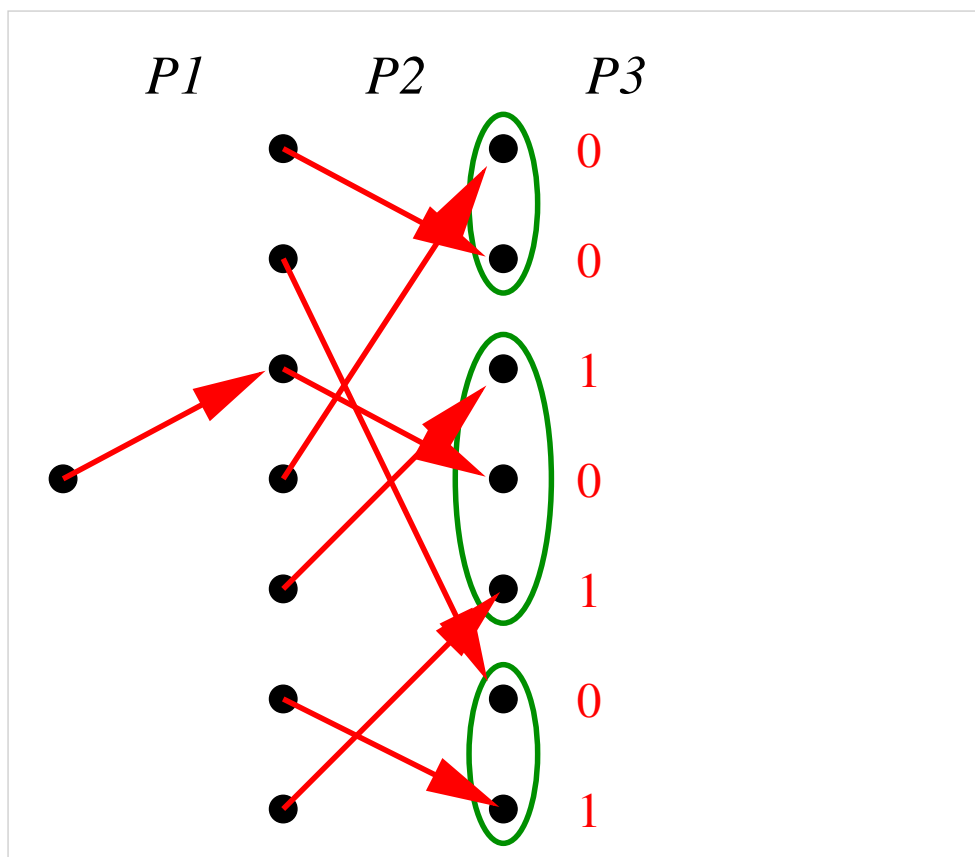
- First sublinear protocol for MPJ_k .
- Defined a restricted form of information sharing: **Collapsing Protocols**.
- Derived $n - O(\log n)$ lower bound for **collapsing protocols**.

The Pudlák-Rödl-Sgall Protocol



- Works for MPJ_3 when middle layer is permutation
- Uses Probabilistic Method and random graphs

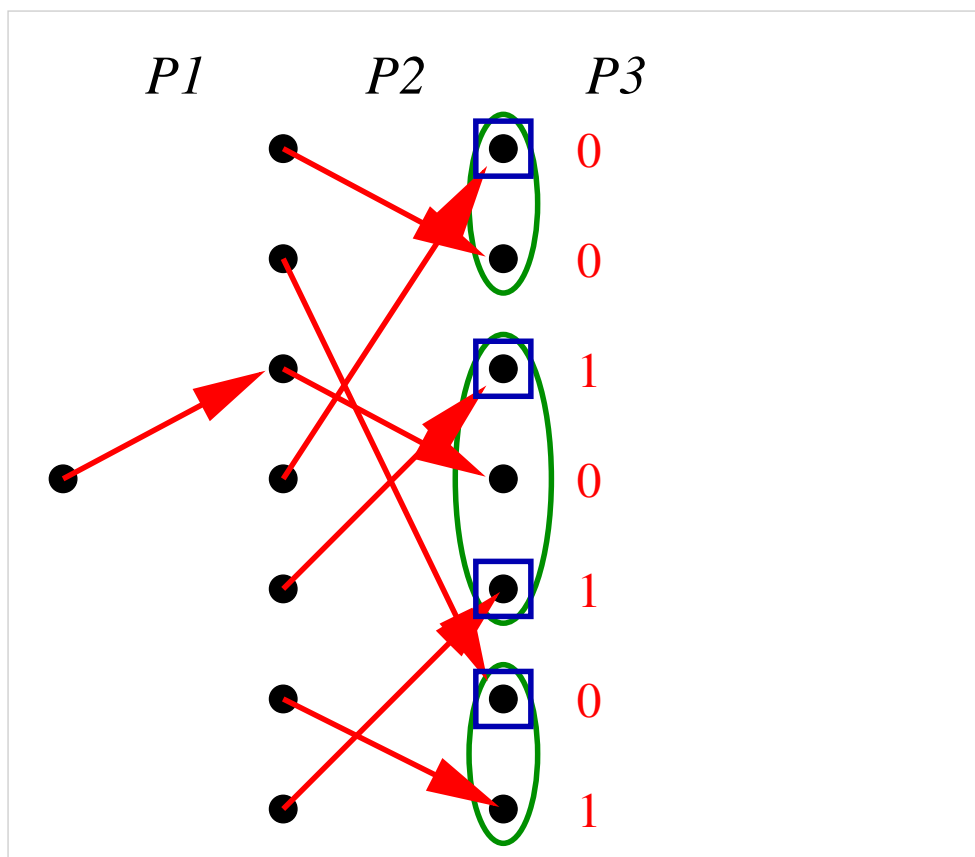
The Pudlák-Rödl-Sgall Protocol



- Works for MPJ_3 when middle layer is permutation
- Uses Probabilistic Method and random graphs

$P1$ sends parity of $O\left(n \frac{\log \log n}{\log n}\right)$ components of nodes in layer 2.

The Pudlák-Rödl-Sgall Protocol

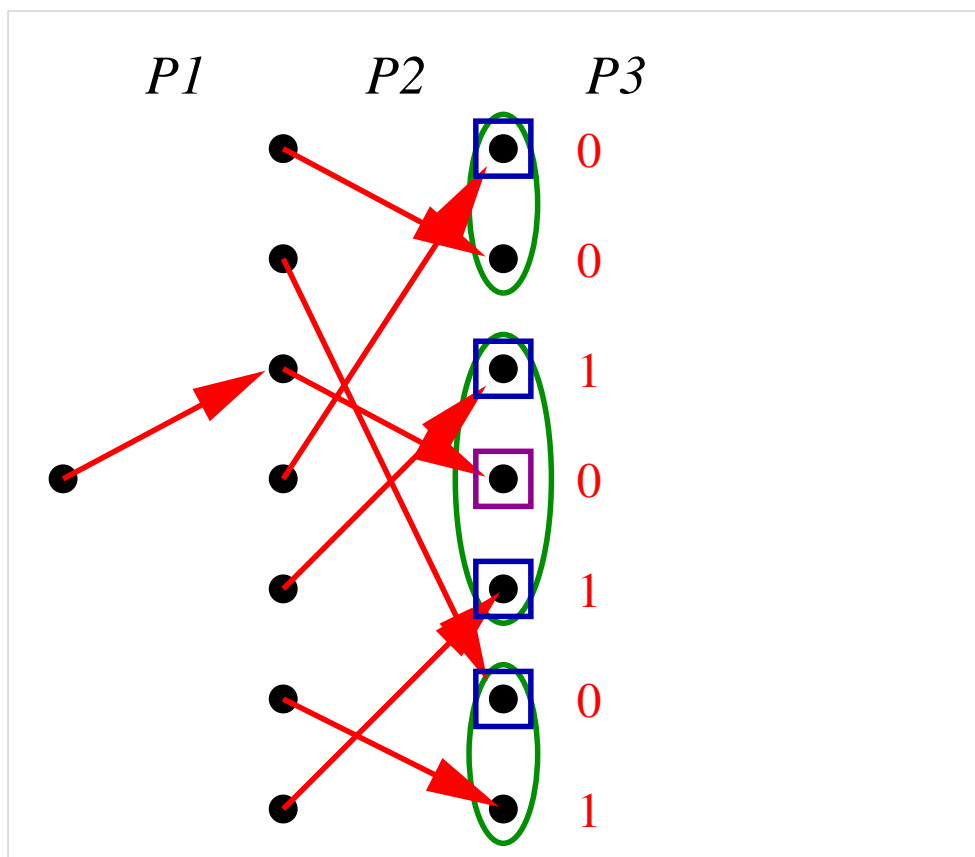


- Works for MPJ_3 when middle layer is permutation
- Uses Probabilistic Method and random graphs

$P1$ sends parity of $O\left(n \frac{\log \log n}{\log n}\right)$ components of nodes in layer 2.

$P2$ sends $O\left(n \frac{\log \log n}{\log n}\right)$ bits.

The Pudlák-Rödl-Sgall Protocol



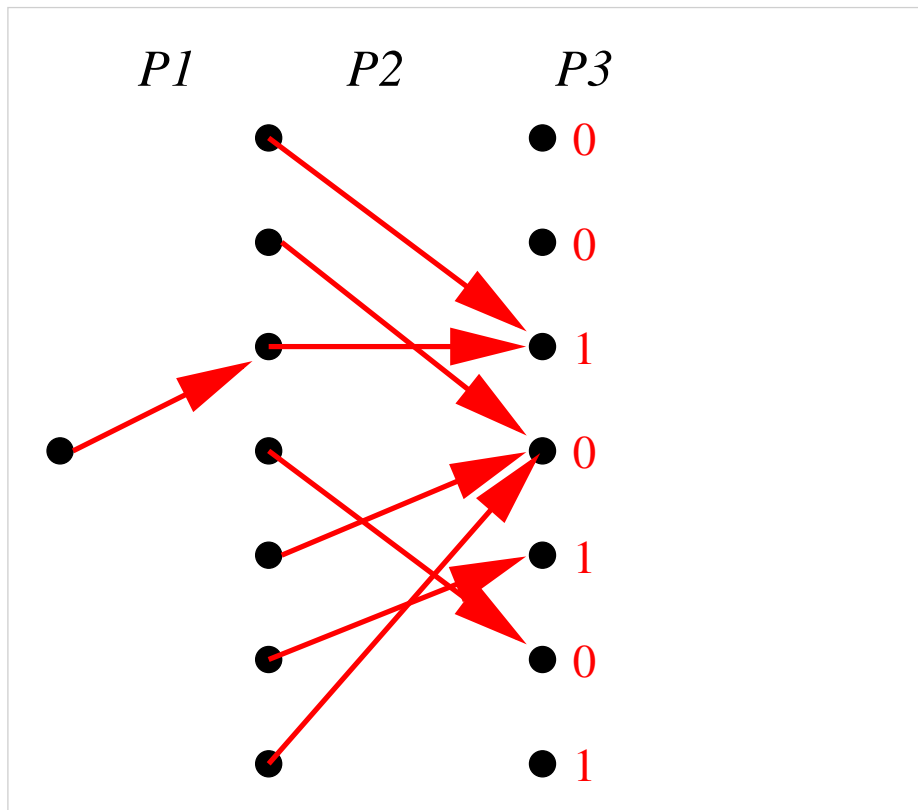
- Works for MPJ_3 when middle layer is permutation
- Uses Probabilistic Method and random graphs

$P1$ sends parity of $O\left(n \frac{\log \log n}{\log n}\right)$ components of nodes in layer 2.

$P2$ sends $O\left(n \frac{\log \log n}{\log n}\right)$ bits.

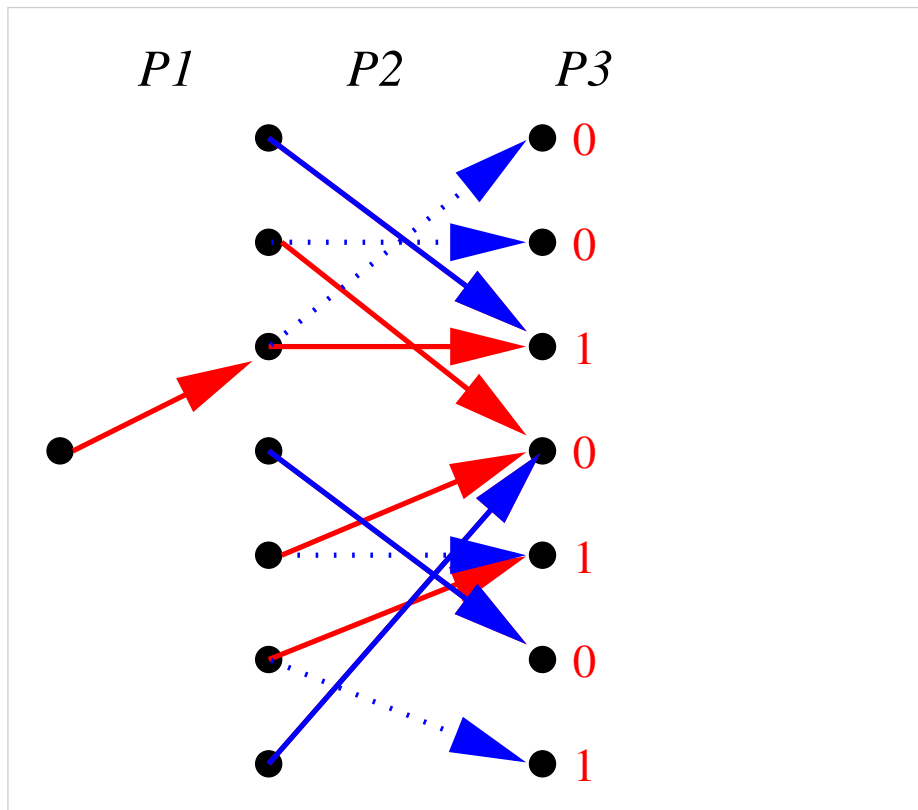
$P2$ sends every bit in the component except for the answer!

A General Protocol: 3 players



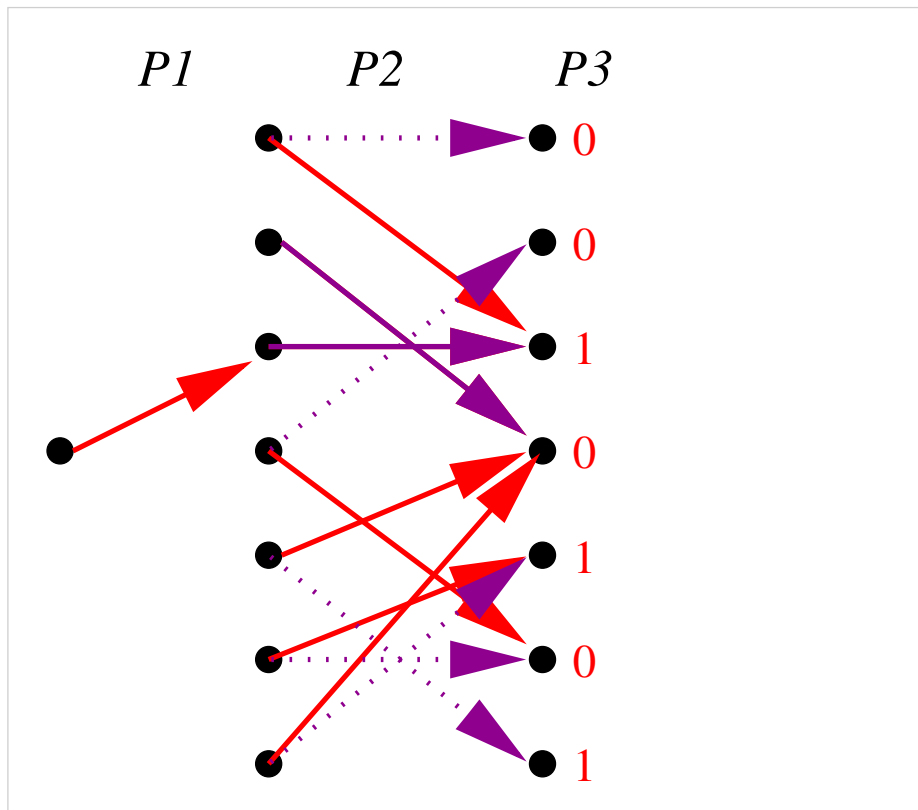
Idea: - Run PRS several times in parallel.

A General Protocol: 3 players



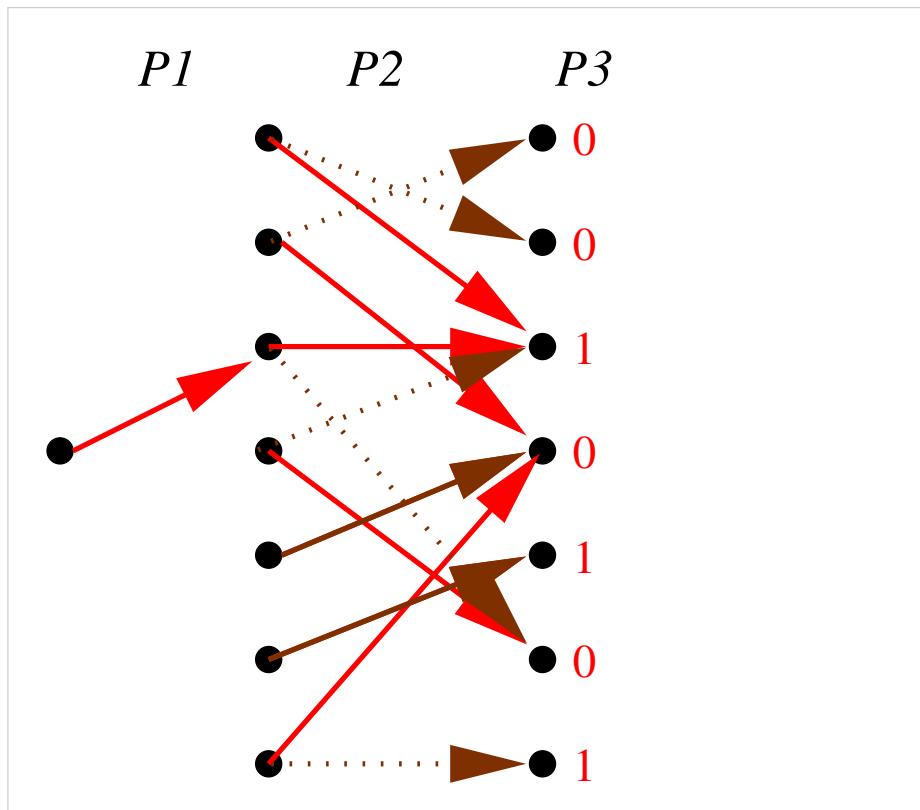
- Idea: - Run PRS several times in parallel.
- Pick permutations $\pi_1, \pi_2, \dots, \pi_d$ such that $f(i) = \pi_j(i)$ for some permutation.

A General Protocol: 3 players



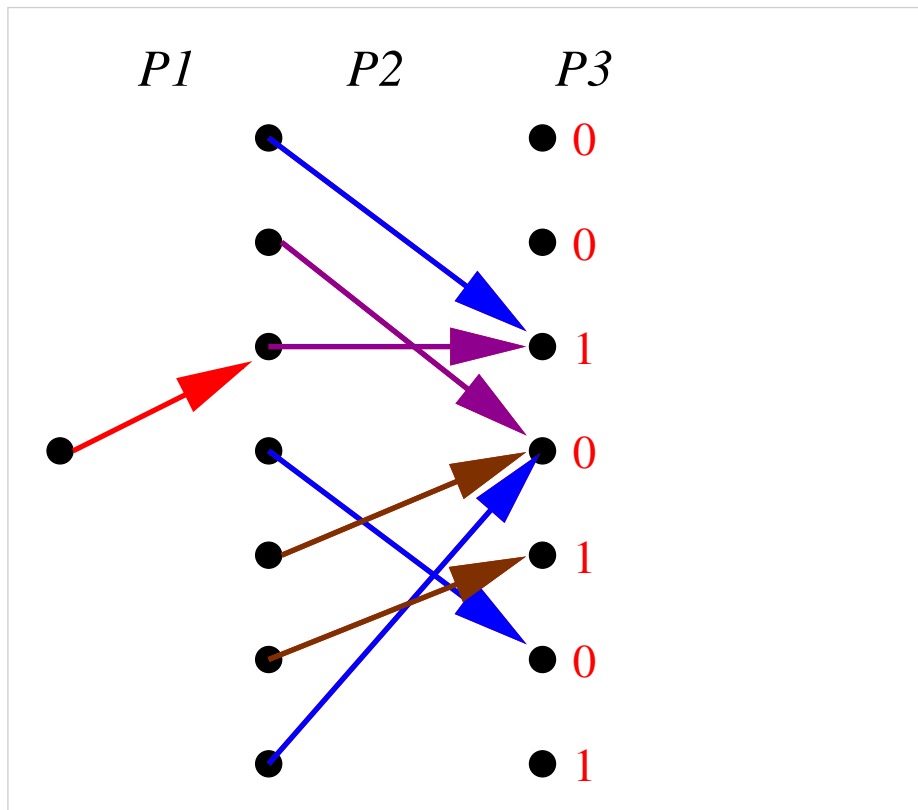
- Idea: - Run PRS several times in parallel.
- Pick permutations $\pi_1, \pi_2, \dots, \pi_d$ such that $f(i) = \pi_j(i)$ for some permutation.

A General Protocol: 3 players



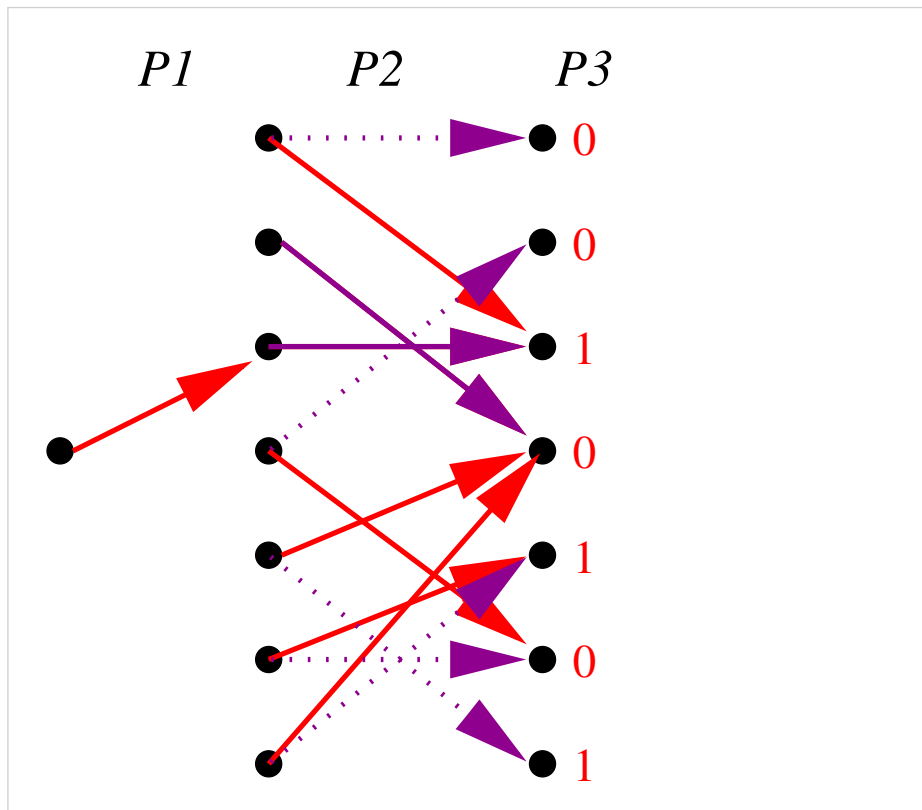
- Idea: - Run PRS several times in parallel.
- Pick permutations $\pi_1, \pi_2, \dots, \pi_d$ such that $f(i) = \pi_j(i)$ for some permutation.

A General Protocol: 3 players



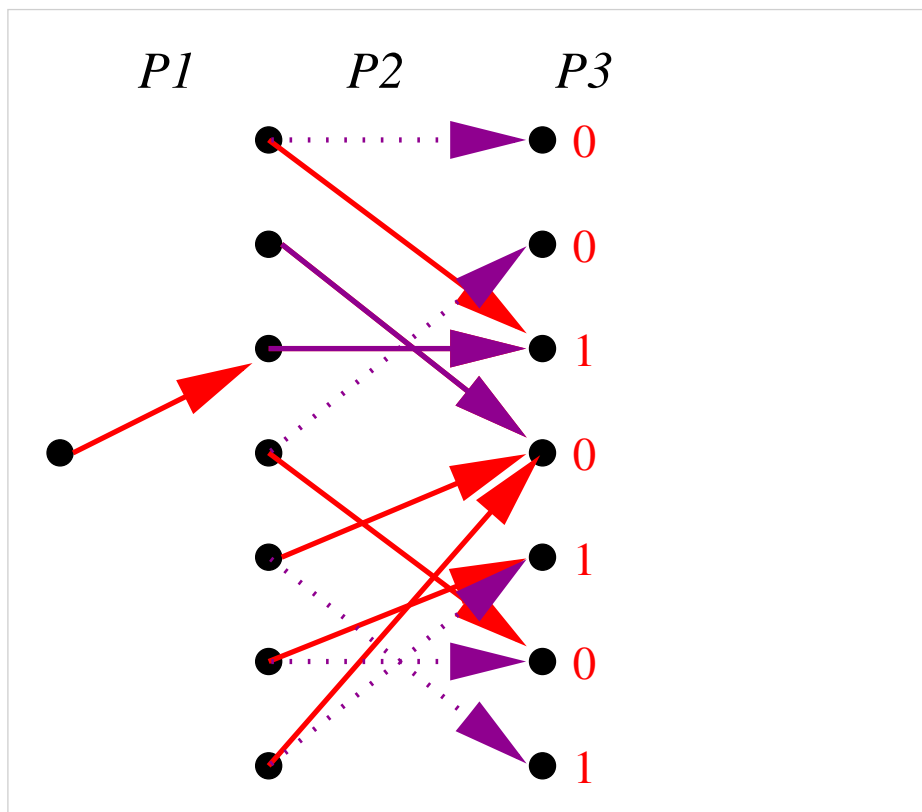
- Idea: - Run PRS several times in parallel.
- Pick permutations $\pi_1, \pi_2, \dots, \pi_d$ such that $f(i) = \pi_j(i)$ for some permutation.

A General Protocol: 3 players



- Idea:
- Run PRS several times in parallel.
 - Pick permutations $\pi_1, \pi_2, \dots, \pi_d$ 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 $\pi_1, \pi_2, \dots, \pi_d$ 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

Definition 1 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

Definition 1 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 2 We can always find a set of d permutations that d -covers f .

Technical Details

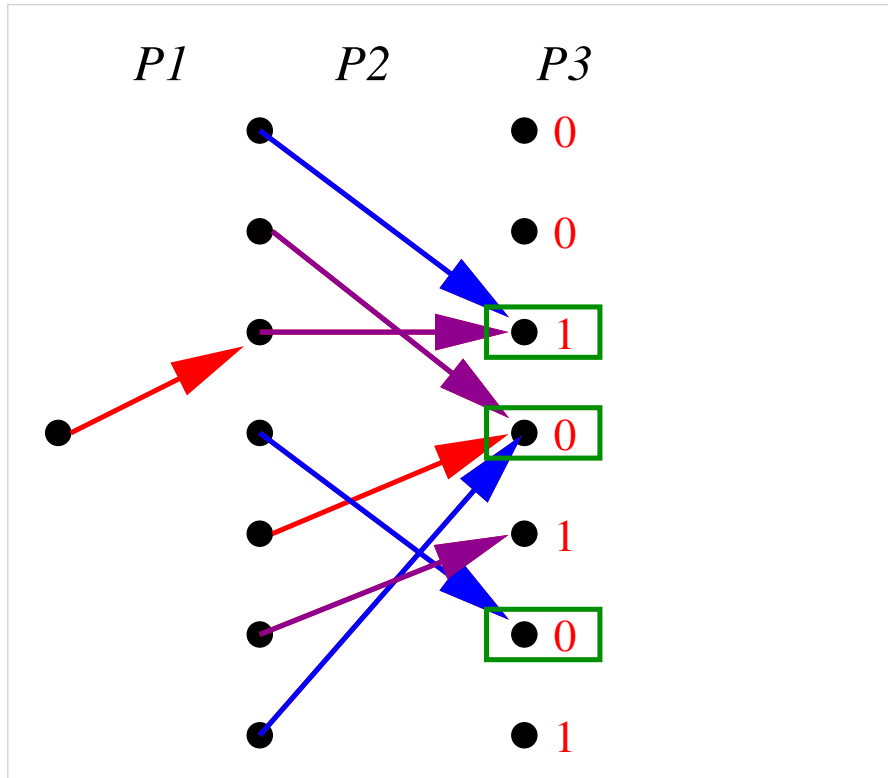
Definition 1 A set of permutations $A \subseteq S_n$ d -covers f if for all $i \in [n]$, one of the following conditions holds:

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

Lemma 2 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.

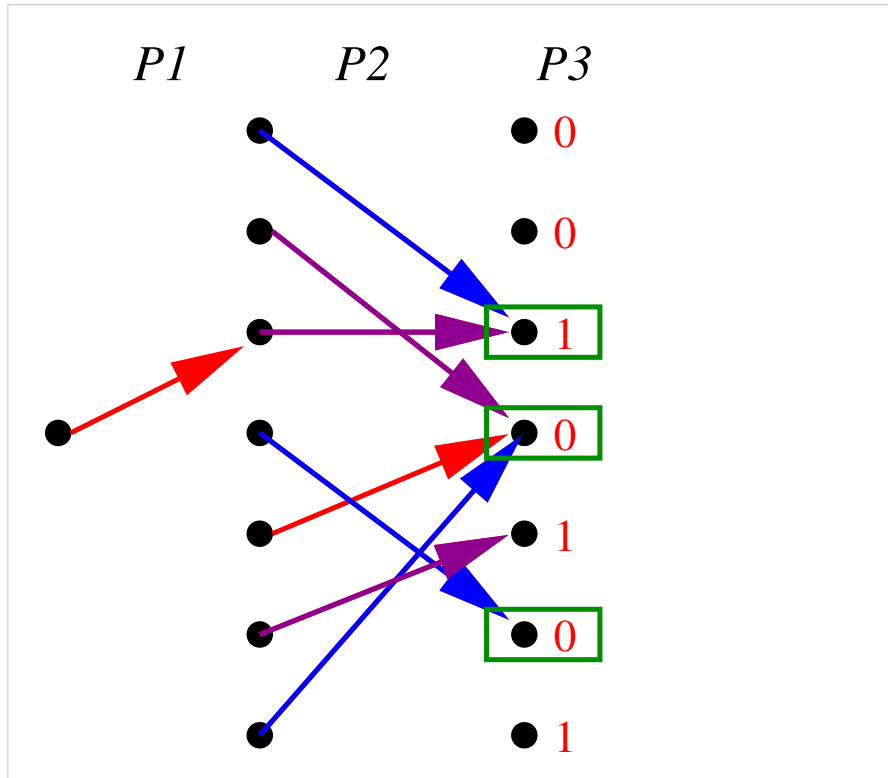
A General Protocol: 3 players



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

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

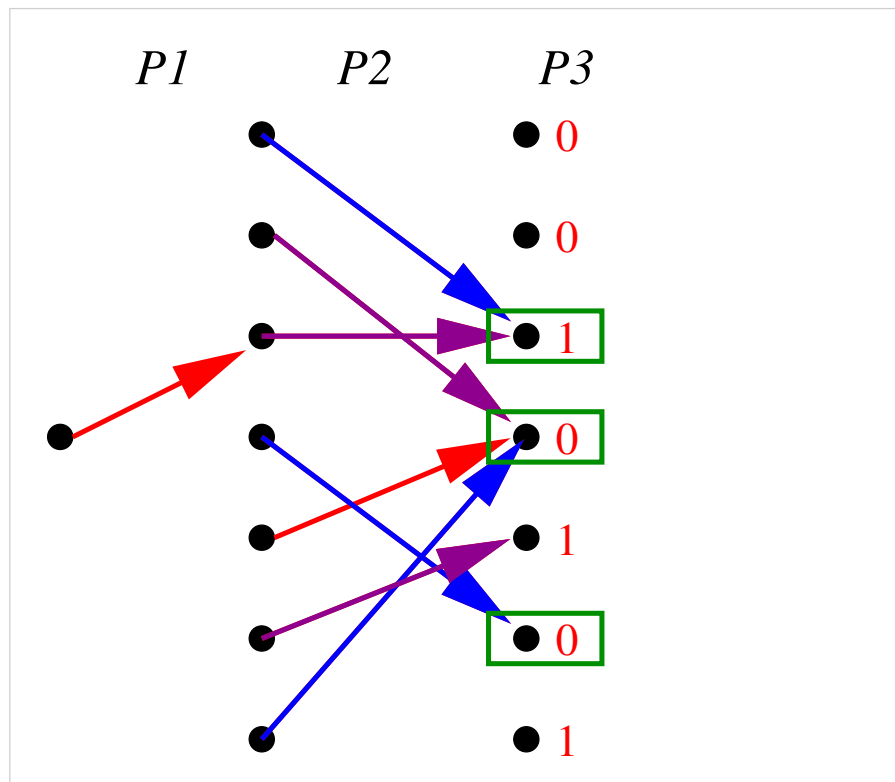
A General Protocol: 3 players



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 $\{\beta(i, x, \alpha)\}_{\alpha}$.

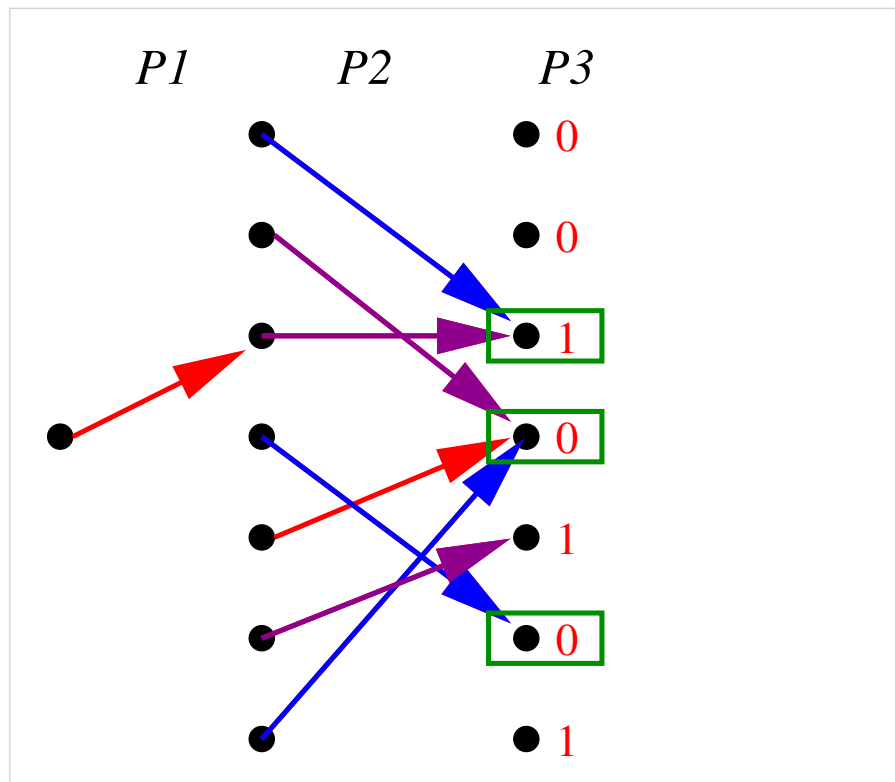
A General Protocol: 3 players



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 $O\left(n \sqrt{\frac{\log \log n}{\log n}}\right)$.

Collapsing Protocols

- **Collapsing Protocols:** P_j sees layers $(1, \dots, j - 1)$ of graph, plus composition of layers $(j + 1, \dots, k)$. Doesn't see individual layers $(j + 1, \dots, k)$.
- Note: In protocol for MPJ_k , every player but P_1 is collapsing.
- A sketch of the lower bound:
 - Fix a collapsing protocol \mathcal{P} .
 - Come up with inputs $(i, f_2, \dots, f_{k-1}, x, x')$ such that the same messages are sent on $(i, f_2, \dots, f_{k-1}, x)$ and $(i, f_2, \dots, f_{k-1}, x')$, but $\text{MPJ}_k(i, f_2, \dots, f_{k-1}, x) \neq \text{MPJ}_k(i, f_2, \dots, f_{k-1}, x')$.

Collapsing Protocols

- **Collapsing Protocols:** P_j sees layers $(1, \dots, j - 1)$ of graph, plus composition of layers $(j + 1, \dots, k)$. Doesn't see individual layers $(j + 1, \dots, k)$.
- Note: In protocol for MPJ_k , every player but P_1 is collapsing.
- A sketch of the lower bound:
 - Fix a collapsing protocol \mathcal{P} .
 - Come up with inputs $(i, f_2, \dots, f_{k-1}, x, x')$ such that the same messages are sent on $(i, f_2, \dots, f_{k-1}, x)$ and $(i, f_2, \dots, f_{k-1}, x')$, but $\text{MPJ}_k(i, f_2, \dots, f_{k-1}, x) \neq \text{MPJ}_k(i, f_2, \dots, f_{k-1}, x')$.
- Crossing Pair:

x1:	0	1	0	0	1	1	0	1
x2:	0	0	1	0	1	0	1	1

A lower bound for collapsing protocols

Definition 3 $y \in \{0, 1\}^n$ is consistent with $(i, f_2, \dots, f_j, \alpha_1, \dots, \alpha_j)$ if for all $h \leq j$, P_h sends α_h on seeing input $(i, f_2, \dots, f_{h-1}, y \circ f_j \circ \dots \circ f_{h+1})$ and messages $\alpha_1, \dots, \alpha_{h-1}$.

A lower bound for collapsing protocols

Definition 3 $y \in \{0, 1\}^n$ is consistent with $(i, f_2, \dots, f_j, \alpha_1, \dots, \alpha_j)$ if for all $h \leq j$, P_h sends α_h on seeing input $(i, f_2, \dots, f_{h-1}, y \circ f_j \circ \dots \circ f_{h+1})$ and messages $\alpha_1, \dots, \alpha_{h-1}$.

Lemma 4 If P_j sends less than $n - O(\log n)$ bits, he sends the same message α_j for a crossing pair (y_j, y'_j) .

A lower bound for collapsing protocols

Definition 3 $y \in \{0, 1\}^n$ is consistent with $(i, f_2, \dots, f_j, \alpha_1, \dots, \alpha_j)$ if for all $h \leq j$, P_h sends α_h on seeing input $(i, f_2, \dots, f_{h-1}, y \circ f_j \circ \dots \circ f_{h+1})$ and messages $\alpha_1, \dots, \alpha_{h-1}$.

Lemma 4 If P_j sends less than $n - O(\log n)$ bits, he sends the same message α_j for a crossing pair (y_j, y'_j) .

Lemma 5 If (y_j, y'_j) is a crossing pair consistent with $(i, f_2, \dots, f_j, \alpha_1, \dots, \alpha_j)$ and P_{j+1} sends α_{j+1} on a crossing pair (y_{j+1}, y'_{j+1}) , then we can fix f_{j+1} such that (y_{j+1}, y'_{j+1}) is consistent with $(i, f_2, \dots, f_{j+1}, \alpha_1, \dots, \alpha_{j+1})$.

Conclusions

Contributions:

- Created the first $o(n)$ one-way protocol for pointer jumping.
- Derived a strong $n - O(\log n)$ lower bound for one-way collapsing protocols.

Open problems:

- Improve the $\Omega(\sqrt{n})$ lower bound for MPJ_3 .
- Improve the $O\left(n\sqrt{\frac{\log \log n}{\log n}}\right)$ upper bound for MPJ_3 .