Communication vs. Computation

Prahladh Harsha
MIT

Yuval Ishai
Technion

Joe Kilian
NEC

Kobbi Nissim
Microsoft SVC

S Venkatesh
Univ. Victoria

Presentation by Piotr Indyk (MIT)

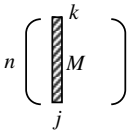
Main Question

- Two important resources (in distributed computing)
  - Amount of communication between processors
  - Time spent in local computation by each processor
- Question: Is there a computational task that shows a strong tradeoff behaviour between these two resources (communication and computation)?

Main Question

- Question: Is there a computational task that shows a strong tradeoff behaviour between these two resources (communication and computation)?
- Main Result: Yes, under certain standard complexity assumptions in the following models
  - 2-party randomized communication complexity model
  - Query complexity model
  - Property Testing model

A Motivating Riddle [BGKL '03]



- M – $n \times k$ matrix over field $\boldsymbol{F}$ ($k > n$)
- k players, one referee
- Player j knows all columns of M except j^{th}
  aka: Input on the forehead model [CFL '83]
- Goal: compute product of the n row sums:

$$PS(M) = \prod_{i=1}^{n} \sum_{j=1}^{k} M_{ij}$$

Computing *PS(M)*

$$PS(M) = \prod_{i=1}^{n} \sum_{j=1}^{k} M_{ij}$$

- Expansion of product *PS(M)* contains k^n terms
  - Since $k > n$, each term can be computed by some player [Recall: Player j has all columns except j^{th}]
- Protocol [BGKL '03]:
  - Assign each term to first player that can compute it.
  - Each player computes the sum of all terms assigned to him and sends sum to referee.
  - Referee publishes the sum of all the messages he receives.

Properties of Protocol

$$PS(M) = \prod_{i=1}^{n} \sum_{j=1}^{k} M_{ij}$$

- Communication: very efficient
  - Each player sends a single element of the field $\boldsymbol{F}$ as a message.
- Computation: inefficient
  - Player (n +1) computes the permanent of the $n \times n$ sub-matrix of M (*#P* computation).

The Riddle

$PS(M) = \prod_{i=1}^{n} \sum_{j=1}^{k} M_{ij}$

- Question: Does there exist a protocol for this problem
  - Each player sends a single element of ***F***
  - Local computation for each player is polynomial in *n, k* ?
- Answer: YES !!
  - Solution: later….

7

Two party Communication Model [Yao '79]

$f : X \times Y \rightarrow Z$

•Alice gets $x \in X$ and Bob gets $y \in Y$
• They compute $z = f(x,y)$ using a protocol and with some local (possibly randomized) computation
• Complexity Measures
  • Communication Complexity: Number of bits communicated by Alice and Bob
  • Round Complexity: Number of rounds of communication
  • Time Complexity

8

Tradeoff Results in Communication Model

- Round Complexity vs. Communication [PS '84, DGS '87, NW '93]

  Pointer chasing problem: k-rounds with O(log n) communication, k -1 rounds with $\Omega(n)$ communication
- Space vs Communication [BTY '94]
- Randomness vs. Communication [CG '93]
- Computation vs. Communication [this paper]

9

Communication vs. Computation

Is there a function $f : X \times Y \rightarrow Z$ such that

- f can be computed efficiently given both its inputs, with no restriction on communication
- f has a protocol with low communication complexity given no restriction on computation
- There is no protocol for f which simultaneously has low communication and efficient computation

- [This paper] YES!, if one-way permutations exist

10

One-way Permutations

A family of permutations

$$\{p_n\}; \quad p_n : \{0,1\}^n \rightarrow \{0,1\}^n$$

is said to be one-way if

- They are easy to compute – there is a deterministic polynomial time algorithm, that given x, can compute $p_n(x)$
- They are hard to invert – any probabilistic algorithm that, given $p_n(x)$, can compute x with probability at least ¾ requires at least $2^{\Omega(n)}$ time on inputs of length n

11

Main Theorem

Assuming one-way permutations exist, there is a boolean function $f : X \times Y \rightarrow \{0,1\}$ such that

– f is computable in polynomial time

– There exists a randomized protocol that computes f with just O(log n) bits of communication

– If Alice and Bob are computationally bounded (i.e., prob. poly-time machines), then any randomized protocol for f (even with multiple rounds) requires $\Omega(n)$ bits of communication

12

The function

Suppose $p : \{0,1\}^n \to \{0,1\}^n$ is a one-way permutation, then define

• Alice's input : $(y, z) \in \{0,1\}^n \times \{0,1\}^n$

•Bob's input : $x \in \{0,1\}^n$

$$f((y,z),x) = \begin{cases} \langle x,z \rangle & \text{if } y = p(x) \\ 0 & \text{otherwise} \end{cases}$$

where $\langle x, z \rangle = \sum x_i \cdot z_i$

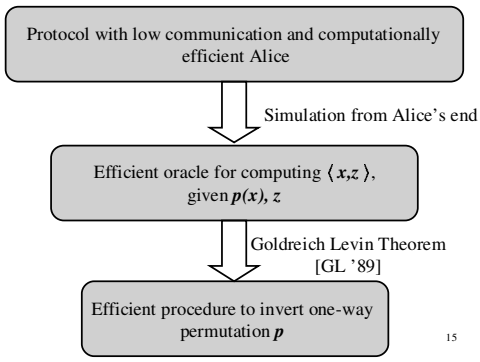
13

Proof of Main Theorem: Upper Bounds

- $f((y,z),x)$ is computable in polynomial time with O(n) of communication
  - Bob sends x to Alice. Alice checks if $p(x)=y$ and if so outputs $\langle x,z \rangle$ else outputs 0.
- One-round randomized protocol computing $f((y,z),x)$ with O(log n) communication with unbounded Alice:
  - (unbounded) Alice computes $w = p^{-1}(x)$ and sends $b = \langle w,z \rangle$ to Bob
  - Alice and Bob engage in equality test protocol comparing w and x
    - One round protocol -- O(log n) communication
  - If comparison succeeds Bob outputs b, otherwise outputs 0

14

Lower Bound Sketch



15

Goldreich-Levin Theorem [GL '89]

- Let h: $\{0,1\}^n \to \{0,1\}$ be a randomized algorithm such that

  $$\Pr[\ h(z)=\langle x,z \rangle] \geq 0.5+\varepsilon$$

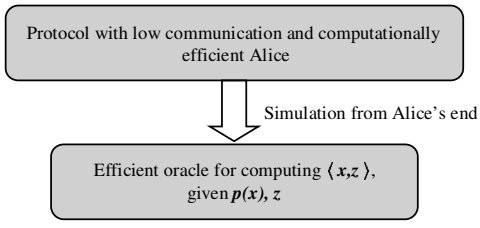
  where the probability is taken over choice of z and the coin tosses of h.
- Then there exists a randomized algorithm GL that outputs a list of elements with oracle access to h such that

  $$\Pr[\ GL^h(n,\varepsilon) \text{ contains } x\ ] \geq 3/4$$

  GL also runs in polynomial in n and $1/\varepsilon$.

16

Converting protocols into oracles



Need to construct efficient oracle such that Given $y = p(x)$ and z, computes $\langle x, z \rangle$

17

Converting transcripts into oracles

Fix a transcript τ of the protocol. Then Oracle h_τ is as follows:

- Simulate the protocol from Alice's end with inputs $y=p(x)$ and z.
- Whenever, a message from Bob is required, use the transcript τ to obtain the corresponding message.
- If at any point, the message generated by Alice deviates the transcript, output a random bit as an answer. Otherwise, output the answer of the protocol.

18

A Simple Claim

- For any y, there exists a transcript τ^* such that

  $\Pr [h_{\tau^*}(z) = \langle x,z \rangle] \geq 0.5 + 1/2^{(b+1)}$

  where the probability is taken over choice of z and the coin tosses of h_{τ^*} and b is the size of the transcript τ^*.
- Hence, given τ^* we can compute $\langle x, z \rangle$ efficiently

  But we do not know τ^* !!

19

Trying every transcript

- If we start with a communication protocol with $b(n)$ bits of communication, we have a set of only $2^{b(n)}$ possible oracles. Try all of them !
  - We can verify which is the right one by checking $y = p(x)$
- Using the Goldreich-Levin Theorem, p can be inverted by a probabilisitic algorithm running in time poly$(n,2^b)$.
- Since p requires $2^{\Omega(n)}$ time to invert, $b(n) \geq \Omega(n)$.

QED

20

Related Models

- Query complexity model and the property testing model
- Information is stored in the form of a table and the queries are answered by probes to the table.
- We view the probes as communication between the storage and query scheme and the computation of the query scheme as local computation.

21

Query complexity

Under a cryptographic assumption, there exists a language L, such that on inputs of size n,

- A query scheme with unlimited computation makes only O(log n) queries.
- However, any query scheme with efficient local computation requires $\Omega(n^{\varepsilon})$ queries for some fixed $\varepsilon < 1$.

22

Property testing

Assuming NP is not contained in BPP, given any $\varepsilon > 0$, there exists a property P such that on inputs of size n,

- A tester with unlimited computation makes only O(n^{ε}) queries.
- However, a tester with efficient local computation requires $\Omega(n^{1-\varepsilon})$ queries.

23

Recall Our Riddle

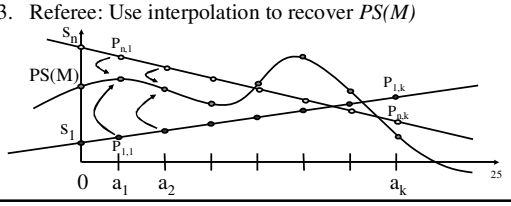
$PS(M) = \prod_{i=1}^{n} \sum_{j=1}^{k} M_{ij}$

- $k > n$
- Player j holds all M but the j^{th} column
- Theorem:
  - The function *PS(M)* admits a protocol where each player runs in polynomial time and sends a single field element to the referee
- Preliminaries:
  - wlog $|F| \geq k+1$ (otherwise, work in extension field)
    - Let $a_1,\ldots,a_k$ be k distinct non-zero elements of F
  - Define row sums $s_i = \Sigma_j M_{i,j}$; Hence $PS(M) = \Pi_i s_i$

24

The Protocol

1. Players compute for each row i=1,…,n elements $P_{i,j}$ s.t. $(a_j, P_{i,j})_{j = 1,\ldots,k}$ lie on a line with free coefficient s_i
2. Player j: Send $q_j = \Pi_i P_{i,j}$ to referee
   - The points $(a_j, P_{i,j})_{j = 1,\ldots,k}$ lie on a degree n polynomial whose free coefficient is $PS(M) = \Pi_i s_i$
3. Referee: Use interpolation to recover $PS(M)$



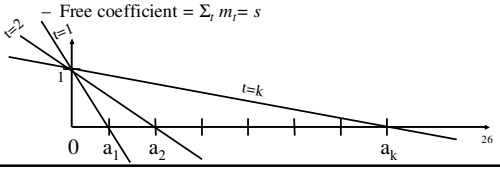
25

Computing the Values $P_{i,j}$

Input: $m_1,\ldots,m_k$ where m_j hidden from j^{th} player

Goal: (a_j, P_j) lie on a line whose free coefficient is $s = \Sigma m_j$

- Let $L_{r,t} = 1 - a_r a_t^{-1}$ for $r,t = 1,\ldots,k$
- $(a_1,L_{1,t}),\ldots,(a_k,L_{k,t})$ lie on a line with Free coefficient = 1
- Player j computes $P_j = \Sigma_t m_t L_{j,t}$
  - Can be computed locally as $L_{j,j}=0$
- By linearity, the points $(a_1,P_1),\ldots, (a_k,P_k)$ lie on a line
  - Free coefficient = $\Sigma_t m_t = s$



26

Summarizing….

- Communication vs. Computation tradeoffs in several communication models
- Open Questions:
  - Can we prove a strong tradeoff result in the two-party communication model under a weaker complexity assumption?
  - Can we show that unconditional results are not possible?
  - Can we prove unconditional results for restricted models of communication and computation?

27

The End

28