Lower bounds for deterministic communication complexity

Armando Matos

Departamento de Ciência de Computadores
Universidade de Porto

November 2006

Goals. . .

Goals. . .

- ⋆ Briefly review the concept of deterministic communication complexity (DCC)

Goals...

- ⋆ Briefly review the concept of deterministic communication complexity (DCC)
- ⋆ Explain two methods for obtaining DCC lower bounds: "fooling sets" and "log-rank"

Goals. . .

- ⋆ Briefly review the concept of deterministic communication complexity (DCC)
- ⋆ Explain two methods for obtaining DCC lower bounds: “fooling sets” and “log-rank”
- ⋆ Compare those lower bounds

Outline

Basics

Two lower bound techniques

How good is the log-rank lower bound? History. . .

How do “fooling sets” and “log-rank” compare?

Further study. . .

Deterministic communication complexity - protocol

Goal: $\mathcal{A}$ and $\mathcal{B}$ want to compute $f(x, y)$

Deterministic communication complexity - protocol

Goal: $\mathcal{A}$ and $\mathcal{B}$ want to compute $f(x, y)$

t: sequence of bits exchanged “so far”

Deterministic communication complexity - protocol

Goal: $\mathcal{A}$ and $\mathcal{B}$ want to compute $f(x, y)$

t: sequence of bits exchanged "so far"

- $\mathcal{A}$ knows x, $\mathcal{B}$ knows y, both know t.

Deterministic communication complexity - protocol

Goal: $\mathcal{A}$ and $\mathcal{B}$ want to compute $f(x, y)$

t: sequence of bits exchanged "so far"

- $\mathcal{A}$ knows x, $\mathcal{B}$ knows y, both know t.
  Protocol P:

Deterministic communication complexity - protocol

Goal: $\mathcal{A}$ and $\mathcal{B}$ want to compute $f(x, y)$

t: sequence of bits exchanged "so far"

- $\mathcal{A}$ knows x, $\mathcal{B}$ knows y, both know t.
  Protocol P:
  - $\mathcal{A}$ sends bit $g(x, t)$ to $\mathcal{B}$

Deterministic communication complexity - protocol

Goal: $\mathcal{A}$ and $\mathcal{B}$ want to compute $f(x, y)$

t: sequence of bits exchanged "so far"

- $\mathcal{A}$ knows x, $\mathcal{B}$ knows y, both know t.
  Protocol P:
  - $\mathcal{A}$ sends bit $g(x, t)$ to $\mathcal{B}$
  - $\mathcal{B}$ sends bit $h(y, t)$ to $\mathcal{A}$

Deterministic communication complexity - protocol

Goal: $\mathcal{A}$ and $\mathcal{B}$ want to compute $f(x, y)$

t: sequence of bits exchanged "so far"

- $\mathcal{A}$ knows x, $\mathcal{B}$ knows y, both know t.
  Protocol P:
  - $\mathcal{A}$ sends bit $g(x, t)$ to $\mathcal{B}$
  - $\mathcal{B}$ sends bit $h(y, t)$ to $\mathcal{A}$
  - ...

Deterministic communication complexity - protocol

Goal: $\mathcal{A}$ and $\mathcal{B}$ want to compute $f(x, y)$

t: sequence of bits exchanged "so far"

- $\mathcal{A}$ knows x, $\mathcal{B}$ knows y, both know t.
  Protocol P:
  - $\mathcal{A}$ sends bit $g(x, t)$ to $\mathcal{B}$
  - $\mathcal{B}$ sends bit $h(y, t)$ to $\mathcal{A}$
  - ...
  - ...

Deterministic communication complexity - protocol

Goal: $\mathcal{A}$ and $\mathcal{B}$ want to compute $f(x, y)$

t: sequence of bits exchanged "so far"

- $\mathcal{A}$ knows x, $\mathcal{B}$ knows y, both know t.
  Protocol P:
  - $\mathcal{A}$ sends bit $g(x, t)$ to $\mathcal{B}$
  - $\mathcal{B}$ sends bit $h(y, t)$ to $\mathcal{A}$
  - ...
  - ...
- ... until $\mathcal{A}$ (or $\mathcal{B}$) knows $f(x, y)$;
  she (he) sends that value to $\mathcal{B}$ ($\mathcal{A}$)

Some remarks. . .

Some remarks. . .

- $\mathcal{A}$ and $\mathcal{B}$ have unlimited computational power

Some remarks...

- $\mathcal{A}$ and $\mathcal{B}$ have unlimited computational power
- $x \in X$, $y \in Y$, $f : X \times Y \to Z$

Some remarks...

- $\mathcal{A}$ and $\mathcal{B}$ have unlimited computational power
- $x \in X$, $y \in Y$, $f : X \times Y \to Z$
  usually: $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$

Some remarks...

- $\mathcal{A}$ and $\mathcal{B}$ have unlimited computational power
- $x \in X$, $y \in Y$, $f : X \times Y \to Z$
  usually: $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$
- M_f is the matrix corresponding to f:
  M_f has $|X|$ lines, $|Y|$ columns and entries in Z.

Some remarks. . .

- $\mathcal{A}$ and $\mathcal{B}$ have unlimited computational power
- $x \in X$, $y \in Y$, $f : X \times Y \to Z$
  usually: $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$
- M_f is the matrix corresponding to f:
  M_f has $|X|$ lines, $|Y|$ columns and entries in Z.
  usually: $2^n \times 2^n$, each entry is 0 or 1

Some remarks...

- $\mathcal{A}$ and $\mathcal{B}$ have unlimited computational power
- $x \in X$, $y \in Y$, $f: X \times Y \to Z$
  usually: $f: \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$
- M_f is the matrix corresponding to f:
  M_f has $|X|$ lines, $|Y|$ columns and entries in Z.
  usually: $2^n \times 2^n$, each entry is 0 or 1
- Every (deterministic) protocol for f induces
  a partition of M_f in monochromatic rectangles

Some remarks...

- $\mathcal{A}$ and $\mathcal{B}$ have unlimited computational power
- $x \in X$, $y \in Y$, $f : X \times Y \to Z$
  usually: $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$
- M_f is the matrix corresponding to f:
  M_f has $|X|$ lines, $|Y|$ columns and entries in Z.
  usually: $2^n \times 2^n$, each entry is 0 or 1
- Every (deterministic) protocol for f induces
  a partition of M_f in monochromatic rectangles
- The converse is not true

$\mathrm{CC}_P(f)$, $\mathrm{CC}(f)$...

- $\mathrm{CC}_P(f)$ is the communication complexity of the protocol P:

$\mathrm{CC}_P(f)$, $\mathrm{CC}(f)$...

- $\mathrm{CC}_P(f)$ is the communication complexity of the protocol P:

$$\mathrm{CC}_P(f) = \text{maximum number of exchanged bits that guarantee that both } \mathcal{A} \text{ and } \mathcal{B} \text{ know } f(x, y)$$

$\mathrm{CC}_P(f)$, $\mathrm{CC}(f)$...

- $\mathrm{CC}_P(f)$ is the communication complexity of the protocol P:

$$\begin{aligned}\mathrm{CC}_P(f) &= \text{maximum number of exchanged bits that} \\ & \text{guarantee that both } \mathcal{A} \text{ and } \mathcal{B} \text{ know } f(x,y) \\ &= \text{height of the tree}\end{aligned}$$

$\mathrm{CC}_P(f)$, $\mathrm{CC}(f)$...

- $\mathrm{CC}_P(f)$ is the communication complexity of the protocol P:

$$\begin{aligned} \mathrm{CC}_P(f) &= \text{maximum number of exchanged bits that} \\ &\quad\ \text{guarantee that both } \mathcal{A} \text{ and } \mathcal{B} \text{ know } f(x,y) \\ &= \text{height of the tree} \end{aligned}$$

communication complexity of f

$$\mathrm{CC}(f) = \min_P \{\mathrm{CC}_P(f)\}$$

Outline

Basics

Two lower bound techniques

How good is the log-rank lower bound? History. . .

How do “fooling sets” and “log-rank” compare?

Further study. . .

Fooling set method

Fooling set method

- The height of a binary tree is $\geq \log N_l$ where N_l is the number of leaves.

Fooling set method

- The height of a binary tree is $\geq \log N_l$ where N_l is the number of leaves.
- The "leaves of a protocol" correspond to a partition of M_f into monochromatic rectangles.

Fooling set method

- The height of a binary tree is $\geq \log N_l$ where N_l is the number of leaves.
- The "leaves of a protocol" correspond to a partition of M_f into monochromatic rectangles.
- A monochromatic set $S \subset M_f$ is a fooling set if

$$\begin{array}{ccc} (x,y) \in S & \wedge & (x',y') \in S \\ & \Longrightarrow & \\ (S[x,y'] \neq m) & \vee & (S[x',y] \neq m) \end{array}$$

(thus two distinct elements of a fooling must disagree on lines <u>and</u> on columns)

Fooling set method (cont.)

The fooling set is blue, $\{(x, y), (x', y'), \ldots\}$

$$
\begin{array}{cc}
 & \begin{array}{ccccc} & y & & y' & \end{array} \\
\begin{array}{c} \\ x \\ \\ x' \\ \\ \end{array} &
\begin{bmatrix}
\ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & 1 & \ldots & 1 & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & 0 & \ldots & 1 & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots
\end{bmatrix}
\end{array}
$$

Fooling set method (cont.)

Fooling set method (cont.)

- No 2 elements of a fooling set S can belong to the same monochromatic rectangle.

Fooling set method (cont.)

- No 2 elements of a fooling set S can belong to the same monochromatic rectangle.
- ...every partition must contain at least $|S|$ rectangles...

Fooling set method (cont.)

- No 2 elements of a fooling set S can belong to the same monochromatic rectangle.
- ...every partition must contain at least $|S|$ rectangles...
- ...the tree of every protocol must have at least $|S|$ leaves...

Fooling set method (cont.)

- No 2 elements of a fooling set S can belong to the same monochromatic rectangle.
- ...every partition must contain at least $|S|$ rectangles...
- ...the tree of every protocol must have at least $|S|$ leaves...
- ...the tree of every protocol must have height $\geq \log |S|$ leaves...

Fooling set method (cont.)

- No 2 elements of a fooling set S can belong to the same monochromatic rectangle.
- ...every partition must contain at least $|S|$ rectangles...
- ...the tree of every protocol must have at least $|S|$ leaves...
- ...the tree of every protocol must have height $\geq \log |S|$ leaves...
- ...$\mathrm{CC}(f) \geq \log |S|$

Fooling set method (cont.)

Fooling set method (cont.)

A slightly better lower bound: for $Z = \{0, 1\}$, a partition must have at least one 0-rectangle; thus

Fooling set method (cont.)

A slightly better lower bound: for $Z = \{0, 1\}$, a partition must have at least one 0-rectangle; thus

Fooling set lowerbound

$$\mathrm{CC}(f) \geq \log(\lceil |S| + 1 \rceil)$$

The rank method

The rank method

Consider any monochromatic partition of M_f. Let the 1-rectangles be $R_1, \ldots, R_m$.

The rank method

Consider any monochromatic partition of M_f. Let the 1-rectangles be $R_1, \ldots, R_m$.

Let R_i' be the matrix with the dimensions of M_f and containing the rectangle R_i (1's):

$$M_i[k,l] = \begin{cases} 1 & \text{if } (k,l) \text{ is in } R_i \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The rank method (cont)

$+ \qquad\qquad +$

$$+ \qquad = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

The rank method (cont)

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \qquad +$$

$$+ \qquad = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

The rank method (cont)

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} +$$

$$+ \qquad = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

The rank method (cont)

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} +$$

$$+ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

The rank method (cont.)

The rank method (cont.)

$$M_f = \sum_{i=1}^{m} R_i'$$

The rank method (cont.)

$$
\begin{aligned}
M_f &= \textstyle\sum_{i=1}^{m} R'_i \\
\operatorname{rank}(M_f) &\leq \textstyle\sum_{i=1}^{m} \operatorname{rank}(R'_i) = m
\end{aligned}
$$

The rank method (cont.)

$$
\begin{aligned}
M_f &= \textstyle\sum_{i=1}^{m} R'_i \\
\operatorname{rank}(M_f) &\leq \textstyle\sum_{i=1}^{m} \operatorname{rank}(R'_i) = m \\
\log(\operatorname{rank}(M_f)) &\leq \log m \leq \mathrm{CC}(f)
\end{aligned}
$$

The rank method (cont.)

The rank method (cont.)

Counting also the 0-rectangles (using $\neg f$), we get a slightly better lower bound

The rank method (cont.)

Counting also the 0-rectangles (using $\neg f$), we get a slightly better lower bound

Log-rank lowerbound

$$\mathrm{CC}(f) \geq \lceil \log(2 \times \mathrm{rank}(M_f) - 1) \rceil$$

Notation. . .

Notation. . .

(simplifications. . .)

Notation. . .

(simplifications. . .)

- A function f denotes a family $f_{2n},\ n = 1, 2\ldots$ of functions with $2n$ variables.

Notation...

(simplifications...)

- A function f denotes a family $f_{2n},\ n = 1, 2 \ldots$ of functions with $2n$ variables.
- $\log \text{rank}(f)$ (instead of $\log \text{rank}(M_f)$)

Notation...

(simplifications...)

- A function f denotes a family f_{2n}, $n = 1, 2 \ldots$ of functions with $2n$ variables.
- $\log \operatorname{rank}(f)$ (instead of $\log \operatorname{rank}(M_f)$)
- $\operatorname{fs}(s) = \log |S|$, the fooling set lower bound

Notation...

(simplifications...)

- A function f denotes a family f_{2n}, $n = 1, 2 \ldots$ of functions with $2n$ variables.
- $\log \mathrm{rank}(f)$ (instead of $\log \mathrm{rank}(M_f)$)
- $\mathrm{fs}(s) = \log |S|$, the fooling set lower bound
- $\mathrm{r}(s) = \log \mathrm{rank}(f)$, the rank lower bound

Notation...

(simplifications...)

- A function f denotes a family f_{2n}, $n = 1, 2 \ldots$ of functions with $2n$ variables.
- $\log \text{rank}(f)$ (instead of $\log \text{rank}(M_f)$)
- $\text{fs}(s) = \log |S|$, the fooling set lower bound
- $\text{r}(s) = \log \text{rank}(f)$, the rank lower bound

Notes:

- S above is the largest fooling set

Notation. . .

(simplifications. . .)

- A function f denotes a family $f_{2n},\ n = 1, 2 \ldots$ of functions with $2n$ variables.
- $\log \text{rank}(f)$ (instead of $\log \text{rank}(M_f)$)
- $\text{fs}(s) = \log |S|$, the fooling set lower bound
- $\text{r}(s) = \log \text{rank}(f)$, the rank lower bound

Notes:

- S above is the largest fooling set
- To compute the rank, matrix M_f in considered over the field $\mathbb{Q}$ (or $\mathbb{R}$)

The log-rank conjecture

The log-rank conjecture

The log-rank conjecture says that the log-rank lower bound is not “very bad”,

The log-rank conjecture

The log-rank conjecture says that the log-rank lower bound is not "very bad",
CONJ1:

$$\forall f\ \exists k \in \mathbb{N}^+\text{: CC}(f) \text{ has order } O(\mathrm{r}(f))^k$$

(remember: $\mathrm{r}(f) = log(\mathrm{rank}(f))$)

The log-rank conjecture

The log-rank conjecture says that the log-rank lower bound is not “very bad”,
CONJ1:

$$\forall f \; \exists k \in \mathbb{N}^+ \text{: } \mathrm{CC}(f) \text{ has order } O(\mathrm{r}(f))^k$$

(remember: $\mathrm{r}(f) = \mathit{log}(\mathrm{rank}(f)))$
or: the gap between r and $\mathrm{CC}(f)$ can not be hyper-polinomial

The log-rank conjecture

The log-rank conjecture says that the log-rank lower bound is not "very bad",
CONJ1:

$$\forall f \; \exists k \in \mathbb{N}^+ \text{: CC}(f) \text{ has order } O(\text{r}(f))^k$$

(remember: $\text{r}(f) = log(\text{rank}(f)))$
or: the gap between r and CC(f) can not be hyper-polinomial

A stronger form (CONJ2) of the conjecture is: CC(f) has order $O(\text{r}(f))$

The log-rank conjecture

The log-rank conjecture says that the log-rank lower bound is not "very bad",
CONJ1:

$$\forall f \; \exists k \in \mathbb{N}^{+}\text{: } \mathrm{CC}(f) \text{ has order } O(\mathrm{r}(f))^{k}$$

(remember: $\mathrm{r}(f) = log(\mathrm{rank}(f)))$
or: the gap between r and $\mathrm{CC}(f)$ can not be hyper-polinomial

A stronger form (CONJ2) of the conjecture is: $\mathrm{CC}(f)$ has order $O(\mathrm{r}(f))$ ... this form is false – no more a conjecture.

Notes:

Notes:

- Why is the log-rank lower bound important?

Notes:

- Why is the log-rank lower bound important?
  Because linear algebra is a very well developed branch of mathematics

Notes:

- Why is the log-rank lower bound important?
  Because linear algebra is a very well developed branch of mathematics
- The log-rank conjecture is related to another conjecture: relation between the chromatic number of a graph and the rank of its adjacency matrix (Lovász, Saks).

Outline

Basics

Two lower bound techniques

How good is the log-rank lower bound? History…

How do "fooling sets" and "log-rank" compare?

Further study…

log-rank lower bound (1989)

log-rank lower bound (1989)

1989, N. Alon and P. D. Seymour
A counter-example to the rank-coloring conjecture
J. Graph Theory 13

log-rank lower bound (1989)

1989, N. Alon and P. D. Seymour
A counter-example to the rank-coloring conjecture
J. Graph Theory 13

The conjecture (as we stated it) may still be true!

log-rank lower bound (1992)

log-rank lower bound (1992)

1992, A. A. Razborov
The gap between the chromatic number of a graph and the rank of its adjacency matrix is super linear
Discrete Mathematics Vol 108.

log-rank lower bound (1992)

1992, A. A. Razborov
The gap between the chromatic number of a graph and the rank of its adjacency matrix is super linear
Discrete Mathematics Vol 108.

Presents a function f such that:

- $\mathrm{CC}(f) \leq k\mathrm{r}(f)$ is false for every $k \in \mathbb{R}^+$.

log-rank lower bound (1992)

1992, A. A. Razborov
The gap between the chromatic number of a graph and the rank of its adjacency matrix is super linear
Discrete Mathematics Vol 108.

Presents a function f such that:

- $\mathrm{CC}(f) \leq k\mathrm{r}(f)$ is false for every $k \in \mathbb{R}^+$.

The gap is not constant.

log-rank lower bound (1992)

1992, A. A. Razborov
The gap between the chromatic number of a graph and the rank of its adjacency matrix is super linear
Discrete Mathematics Vol 108.

Presents a function f such that:

- $\mathrm{CC}(f) \leq k\mathrm{r}(f)$ is false for every $k \in \mathbb{R}^+$.

The gap is not constant.
The conjecture (CONJ1) may still be true! (but CONJ2 is false)

log-rank lower bound (1995)

log-rank lower bound (1995)

1995, R. Raz and B.Spieker
(with the help of John H. Conway and Laci Lovász!)
On the log-rank conjecture in communication complexity
Combinatorica 15(4).

log-rank lower bound (1995)

1995, R. Raz and B.Spieker
(with the help of John H. Conway and Laci Lovász!)
On the log-rank conjecture in communication complexity
Combinatorica 15(4).

Presents a f such that

- NCC is $\Omega(n \log \log n)$ (non-deterministic communication complexity); the same is of course true for deterministic communication complexity!

log-rank lower bound (1995)

1995, R. Raz and B.Spieker
(with the help of John H. Conway and Laci Lovász!)
On the log-rank conjecture in communication complexity
Combinatorica 15(4).

Presents a f such that

- NCC is $\Omega(n \log\log n)$ (non-deterministic communication complexity); the same is of course true for deterministic communication complexity!
- $\mathrm{r}(f)$ is $O(n)$

log-rank lower bound (1995)

1995, R. Raz and B.Spieker
(with the help of John H. Conway and Laci Lovász!)
On the log-rank conjecture in communication complexity
Combinatorica 15(4).

Presents a f such that

- NCC is $\Omega(n \log \log n)$ (non-deterministic communication complexity); the same is of course true for deterministic communication complexity!
- r(f) is $O(n)$

$$\text{gap: } \boxed{\leq kn} \longrightarrow \boxed{\geq k' n \log \log n}$$

log-rank lower bound (1995)

1995, R. Raz and B.Spieker
(with the help of John H. Conway and Laci Lovász!)
On the log-rank conjecture in communication complexity
Combinatorica 15(4).

Presents a f such that

- NCC is $\Omega(n \log \log n)$ (non-deterministic communication complexity); the same is of course true for deterministic communication complexity!
- r(f) is $O(n)$

$$\text{gap: } \boxed{\leq kn} \longrightarrow \boxed{\geq k' n \log \log n}$$

The conjecture may still be true!

log-rank lower bound (1994)

log-rank lower bound (1994)

1994, Noam Nisan, Avi Widgerson
On rank vs. communication complexity

log-rank lower bound (1994)

1994, Noam Nisan, Avi Widgerson
On rank vs. communication complexity

Presents an f such that

log-rank lower bound (1994)

1994, Noam Nisan, Avi Widgerson
On rank vs. communication complexity

Presents an f such that

- $\mathrm{CC}(f)$ is $\Omega(n)$

log-rank lower bound (1994)

1994, Noam Nisan, Avi Widgerson
On rank vs. communication complexity

Presents an f such that

- CC(f) is $\Omega(n)$
- r(f) is $O(n^{0.631})$

log-rank lower bound (1994)

1994, Noam Nisan, Avi Widgerson
On rank vs. communication complexity

Presents an f such that

- CC(f) is $\Omega(n)$
- r(f) is $O(n^{0.631})$

The conjecture may still be true

Outline

Basics

Two lower bound techniques

How good is the log-rank lower bound? History...

How do “fooling sets” and “log-rank” compare?

Further study...

Comparing “fooling set” with “log-rank”. . .

Comparing “fooling set” with “log-rank”. . .

A summary of the results in

1996, M. Dietzfelbinger, J. Hromkovilmage, G, Schnitger
A comparison of two lower-bound methods for communication complexity, TCS 168 (1)

Comparing “fooling set” with “log-rank”. . .

A summary of the results in

1996, M. Dietzfelbinger, J. Hromkovilmage, G, Schnitger
A comparison of two lower-bound methods for communication complexity, TCS 168 (1)

FSM - “fooling set” method
LRM - “log-rank” method

Comparisons (cont.)

Comparisons (cont.)

- For some problems FSM is very weak:
  There is a function f such that for n large enough we have $\mathrm{CC}(f) = n$ and $\mathrm{fs}(f) = O(\log n)$.

Comparisons (cont.)

- For some problems FSM is very weak:
  There is a function f such that for n large enough we have $\mathrm{CC}(f) = n$ and $\mathrm{fs}(f) = O(\log n)$.
- LRM is almost always much better than FSM:
  for almost all functions f: $\mathrm{r}(f) \approx n$ and $\mathrm{fs}(f) \leq \log n + \log 10$

Comparisons (cont.)

- For some problems FSM is very weak:
  There is a function f such that for n large enough we have $\mathrm{CC}(f) = n$ and $\mathrm{fs}(f) = O(\log n)$.
- LRM is almost always much better than FSM:
  for almost all functions f: $\mathrm{r}(f) \approx n$ and $\mathrm{fs}(f) \leq \log n + \log 10$
- LRM cannot be much worse than the FSM:
  $\forall f: \ \mathrm{fs}(f) \leq 2\mathrm{r}(f)$

Comparisons (cont.)

- For some problems FSM is very weak:
  There is a function f such that for n large enough we have $\mathrm{CC}(f) = n$ and $\mathrm{fs}(f) = O(\log n)$.
- LRM is almost always much better than FSM:
  for almost all functions f: $\mathrm{r}(f) \approx n$ and $\mathrm{fs}(f) \leq \log n + \log 10$
- LRM cannot be much worse than the FSM:
  $\forall f: \ \mathrm{fs}(f) \leq 2\mathrm{r}(f)$
- LRM can be (slightly) worse than FSM:
  There is a function f such that $\mathrm{fs}(f) = n$ and $\mathrm{r}(f) = (\log 3)/2n \approx 0.79n$

Comparisons (cont.)

Comparisons (cont.)

About these results see also

1992, M. Karchmer, E. Kushilevitz, N. Nisan
Fractional covers and communication complexity
Structure in Complexity Theory Conference

Comparisons (cont.)

About these results see also

1992, M. Karchmer, E. Kushilevitz, N. Nisan
Fractional covers and communication complexity
Structure in Complexity Theory Conference

also in
SIAM J. Discrete Mathematics, 1995

Comparisons (cont.)

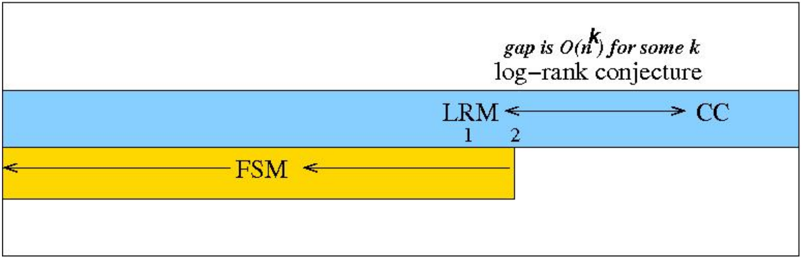
Comparisons (cont.)

- For some problems FSM is very weak
- LRM is almost always much better than FSM
- LRM cannot be much worse than the FSM
- LRM can be (slightly) worse than FSM
- LRM can be (slightly) worse than FSM

Comparisons (cont.)

- For some problems FSM is very weak
- LRM is almost always much better than FSM
- LRM cannot be much worse than the FSM
- LRM can be (slightly) worse than FSM
- LRM can be (slightly) worse than FSM
- LRM can be relatively week
  $\exists\, f$ with $\mathrm{CC}(f) \in \Omega(n)$ and $\mathrm{r}(f) \in O(n^{0.631})$

Summary. . .



Outline

Basics

Two lower bound techniques

How good is the log-rank lower bound? History...

How do “fooling sets” and “log-rank” compare?

Further study...

Comments and further study…

Comments and further study. . .

- Is there any recent (after 1996) important work?

Comments and further study...

- Is there any recent (after 1996) important work?
- Study lower bound information obtained by using Kolmogorov complexity methods (see "Individual communication complexity" by Buhrman, Klauck, Vereshchagin and Vitanyi, 2004).

Comments and further study...

- Is there any recent (after 1996) important work?
- Study lower bound information obtained by using Kolmogorov complexity methods (see "Individual communication complexity" by Buhrman, Klauck, Vereshchagin and Vitanyi, 2004).
- Statements like "almost all functions have property P" are not relevant for "real world" functions like gt($\overline{x}, \overline{y}$) or median($\overline{x}, \overline{y}$) because "real world" functions are not random, $K(f|n) = c \ll 2^{2n}$.

Comments and further study…

- Is there any recent (after 1996) important work?
- Study lower bound information obtained by using Kolmogorov complexity methods (see "Individual communication complexity" by Buhrman, Klauck, Vereshchagin and Vitanyi, 2004).
- Statements like "almost all functions have property P" are not relevant for "real world" functions like gt($\overline{x}, \overline{y}$) or median($\overline{x}, \overline{y}$) because "real world" functions are not random, $K(f|n) = c \ll 2^{2n}$.
- Study lower bounds for uniform communication complexity.

the end