Complexity lower bounds for algebraic computation trees

Dima Grigoriev (Lille)

CNRS

14/06/2011, Saint-Petersbourg

Several non-linear lower bounds are known in algebraic complexity unlike its boolean counterpart

Computational models

Let F be a ground field.

Algebraic d-decision tree

- Input $x = (x_1, \ldots, x_n) \in F^n$ is attributed to the root of tree T.
- To each node v of T (except leaves) a (testing) polynomial $g_v \in F[X_1, \ldots, X_n]$ with $\deg(g_v) \leq d$ is assigned. The algebraic decision tree branches at node v according to whether $g_v(x) = 0$. In case $F = \mathbb{R}$ one uses alternatively $g_v(x) > 0$ as a condition of branching.
- To each leaf L an output "*accept*" or "*reject*" is assigned.
- Denote by $S_L \subset F^n$ the set of inputs which arrive at L. They are pairwise disjoint and the set $S \subset F^n$ **accepted by the algebraic decision tree** is the union of S_L for all leaves L assigned with "*accept*".
- The **complexity** C_d is the depth of T.

For $d = 1$ they are called linear decision trees.

Computational models

Let F be a ground field.

Algebraic d-decision tree

- Input $x = (x_1, \ldots, x_n) \in F^n$ is attributed to the root of tree T.
- To each node v of T (except leaves) a (testing) polynomial $g_v \in F[X_1, \ldots, X_n]$ with $\deg(g_v) \le d$ is assigned. The algebraic decision tree branches at node v according to whether $g_v(x) = 0$. In case $F = \mathbb{R}$ one uses alternatively $g_v(x) > 0$ as a condition of branching.
- To each leaf L an output "*accept*" or "*reject*" is assigned.
- Denote by $S_L \subset F^n$ the set of inputs which arrive at L. They are pairwise disjoint and the set $S \subset F^n$ **accepted by the algebraic decision tree** is the union of S_L for all leaves L assigned with "*accept*".
- The **complexity** C_d is the depth of T.

For $d = 1$ they are called linear decision trees.

Computational models

Let F be a ground field.

Algebraic d-decision tree

- Input $x = (x_1, \ldots, x_n) \in F^n$ is attributed to the root of tree T.
- To each node v of T (except leaves) a (testing) polynomial $g_v \in F[X_1, \ldots, X_n]$ with $\deg(g_v) \leq d$ is assigned. The algebraic decision tree branches at node v according to whether $g_v(x) = 0$. In case $F = \mathbb{R}$ one uses alternatively $g_v(x) > 0$ as a condition of branching.
- To each leaf L an output "*accept*" or "*reject*" is assigned.
- Denote by $S_L \subset F^n$ the set of inputs which arrive at L. They are pairwise disjoint and the set $S \subset F^n$ **accepted by the algebraic decision tree** is the union of S_L for all leaves L assigned with "*accept*".
- The **complexity** C_d is the depth of T.

For $d = 1$ they are called linear decision trees.

Computational models

Let F be a ground field.

Algebraic d-decision tree

- Input $x = (x_1, \ldots, x_n) \in F^n$ is attributed to the root of tree T.
- To each node v of T (except leaves) a (testing) polynomial $g_v \in F[X_1, \ldots, X_n]$ with $\deg(g_v) \leq d$ is assigned. The algebraic decision tree branches at node v according to whether $g_v(x) = 0$. In case $F = \mathbb{R}$ one uses alternatively $g_v(x) > 0$ as a condition of branching.
- To each leaf L an output "*accept*" or "*reject*" is assigned.
- Denote by $S_L \subset F^n$ the set of inputs which arrive at L. They are pairwise disjoint and the set $S \subset F^n$ **accepted by the algebraic decision tree** is the union of S_L for all leaves L assigned with "*accept*".
- The **complexity** C_d is the depth of T.

For $d = 1$ they are called linear decision trees.

Computational models

Let F be a ground field.

Algebraic d-decision tree

- Input $x = (x_1, \dots, x_n) \in F^n$ is attributed to the root of tree T.
- To each node v of T (except leaves) a (testing) polynomial $g_v \in F[X_1, \dots, X_n]$ with $\deg(g_v) \le d$ is assigned. The algebraic decision tree branches at node v according to whether $g_v(x) = 0$. In case $F = \mathbb{R}$ one uses alternatively $g_v(x) > 0$ as a condition of branching.
- To each leaf L an output "*accept*" or "*reject*" is assigned.
- Denote by $S_L \subset F^n$ the set of inputs which arrive at L. They are pairwise disjoint and the set $S \subset F^n$ **accepted by the algebraic decision tree** is the union of S_L for all leaves L assigned with "*accept*".
- The **complexity** C_d is the depth of T.

For $d = 1$ they are called linear decision trees.

Computational models

Let F be a ground field.

Algebraic d-decision tree

- Input $x = (x_1, \dots, x_n) \in F^n$ is attributed to the root of tree T.
- To each node v of T (except leaves) a (testing) polynomial $g_v \in F[X_1, \dots, X_n]$ with $\deg(g_v) \leq d$ is assigned. The algebraic decision tree branches at node v according to whether $g_v(x) = 0$. In case $F = \mathbb{R}$ one uses alternatively $g_v(x) > 0$ as a condition of branching.
- To each leaf L an output "*accept*" or "*reject*" is assigned.
- Denote by $S_L \subset F^n$ the set of inputs which arrive at L. They are pairwise disjoint and the set $S \subset F^n$ **accepted by the algebraic decision tree** is the union of S_L for all leaves L assigned with "*accept*".
- The **complexity** C_d is the depth of T.

For $d = 1$ they are called linear decision trees.

Computational models

Let F be a ground field.

Algebraic d-decision tree

- Input $x = (x_1, \ldots, x_n) \in F^n$ is attributed to the root of tree T.
- To each node v of T (except leaves) a (testing) polynomial $g_v \in F[X_1, \ldots, X_n]$ with $\deg(g_v) \leq d$ is assigned. The algebraic decision tree branches at node v according to whether $g_v(x) = 0$. In case $F = \mathbb{R}$ one uses alternatively $g_v(x) > 0$ as a condition of branching.
- To each leaf L an output "*accept*" or "*reject*" is assigned.
- Denote by $S_L \subset F^n$ the set of inputs which arrive at L. They are pairwise disjoint and the set $S \subset F^n$ **accepted by the algebraic decision tree** is the union of S_L for all leaves L assigned with "*accept*".
- The **complexity** C_d is the depth of T.

For $d = 1$ they are called linear decision trees.

Computational models

Let F be a ground field.

Algebraic d-decision tree

- Input $x = (x_1, \ldots, x_n) \in F^n$ is attributed to the root of tree T.
- To each node v of T (except leaves) a (testing) polynomial $g_v \in F[X_1, \ldots, X_n]$ with $\deg(g_v) \leq d$ is assigned. The algebraic decision tree branches at node v according to whether $g_v(x) = 0$. In case $F = \mathbb{R}$ one uses alternatively $g_v(x) > 0$ as a condition of branching.
- To each leaf L an output "*accept*" or "*reject*" is assigned.
- Denote by $S_L \subset F^n$ the set of inputs which arrive at L. They are pairwise disjoint and the set $S \subset F^n$ **accepted by the algebraic decision tree** is the union of S_L for all leaves L assigned with "*accept*".
- The **complexity** C_d is the depth of T.

For $d = 1$ they are called linear decision trees.

Algebraic computation tree

The difference with algebraic decision trees is that testing polynomial g_v is calculated as $g_v = a \circ b$ where

- operation $\circ \in \{+, \times\}$
- $a, b \in F \cup \{X_1, \dots, X_n\} \cup \{g_u\}$ where u runs the nodes on the path from the root to v.

In particular, $\deg(g_v) \leq 2^{k_v}$ where k_v is the depth of v.
Denote by C the complexity of algebraic computation trees. For any constant d the model of computation trees is stronger than one of d-decision trees.
We study the complexities $C_d(S)$, $C(S)$ of the membership problem to S.

Algebraic computation tree

The difference with algebraic decision trees is that testing polynomial g_v is calculated as $g_v = a \circ b$ where

- operation $\circ \in \{+, \times\}$
- $a, b \in F \cup \{X_1, \ldots, X_n\} \cup \{g_u\}$ where u runs the nodes on the path from the root to v.

In particular, $\deg(g_v) \leq 2^{k_v}$ where k_v is the depth of v.
Denote by C the complexity of algebraic computation trees. For any constant d the model of computation trees is stronger than one of d-decision trees.
We study the complexities $C_d(S)$, $C(S)$ of the membership problem to S.

Algebraic computation tree

The difference with algebraic decision trees is that testing polynomial g_v is calculated as $g_v = a \circ b$ where

- operation $\circ \in \{+, \times\}$
- $a, b \in F \cup \{X_1, \dots, X_n\} \cup \{g_u\}$ where u runs the nodes on the path from the root to v.

In particular, $\deg(g_v) \leq 2^{k_v}$ where k_v is the depth of v.
Denote by C the complexity of algebraic computation trees. For any constant d the model of computation trees is stronger than one of d-decision trees.
We study the complexities $C_d(S)$, $C(S)$ of the membership problem to S.

Algebraic computation tree

The difference with algebraic decision trees is that testing polynomial g_v is calculated as $g_v = a \circ b$ where

- operation $\circ \in \{+, \times\}$
- $a, b \in F \cup \{X_1, \ldots, X_n\} \cup \{g_u\}$ where u runs the nodes on the path from the root to v.

In particular, $\deg(g_v) \leq 2^{k_v}$ where k_v is the depth of v.

Denote by C the complexity of algebraic computation trees. For any constant d the model of computation trees is stronger than one of d-decision trees.

We study the complexities $C_d(S)$, $C(S)$ of the membership problem to S.

Algebraic computation tree

The difference with algebraic decision trees is that testing polynomial g_v is calculated as $g_v = a \circ b$ where

- operation $\circ \in \{+, \times\}$
- $a, b \in F \cup \{X_1, \dots, X_n\} \cup \{g_u\}$ where u runs the nodes on the path from the root to v.

In particular, $\deg(g_v) \leq 2^{k_v}$ where k_v is the depth of v.
Denote by C the complexity of algebraic computation trees. For any constant d the model of computation trees is stronger than one of d-decision trees.
We study the complexities $C_d(S)$, $C(S)$ of the membership problem to S.

Algebraic computation tree

The difference with algebraic decision trees is that testing polynomial g_v is calculated as $g_v = a \circ b$ where

- operation $\circ \in \{+, \times\}$
- $a, b \in F \cup \{X_1, \ldots, X_n\} \cup \{g_u\}$ where u runs the nodes on the path from the root to v.

In particular, $\deg(g_v) \leq 2^{k_v}$ where k_v is the depth of v.
Denote by C the complexity of algebraic computation trees. For any constant d the model of computation trees is stronger than one of d-decision trees.
We study the complexities $C_d(S), C(S)$ of the membership problem to S.

Complexity lower bounds via topological invariants

There are several lower bounds for algebraic computation trees over the field either $F = \mathbb{C}$ or $F = \mathbb{R}$ of the form
$C(S) \geq \log_2(c_i) - n,\ i = 1, 2, 3$ where

- c_1 is the maximum of the number of connected components in S and in $F^n \setminus S$ (Ben-Or [1983]);
- c_2 is the Euler characteristic of S (Björner-Lovasz-Yao [1992]);
- c_3 is the sum of Betti numbers (ranks of homological groups) of S (Yao [1994], Montaña-Morais-Pardo [1996]).

$c_1, c_2 \leq c_3$

Complexity lower bounds via topological invariants

There are several lower bounds for algebraic computation trees over the field either $F = \mathbb{C}$ or $F = \mathbb{R}$ of the form $C(S) \geq \log_2(c_i) - n$, $i = 1, 2, 3$ where

- c_1 is the maximum of the number of connected components in S and in $F^n \setminus S$ (Ben-Or [1983]);
- c_2 is the Euler characteristic of S (Björner-Lovasz-Yao [1992]);
- c_3 is the sum of Betti numbers (ranks of homological groups) of S (Yao [1994], Montaña-Morais-Pardo [1996]).

$c_1, c_2 \leq c_3$

Complexity lower bounds via topological invariants

There are several lower bounds for algebraic computation trees over the field either $F = \mathbb{C}$ or $F = \mathbb{R}$ of the form
$C(S) \geq \log_2(c_i) - n,\ i = 1, 2, 3$ where

- c_1 is the maximum of the number of connected components in S and in $F^n \setminus S$ (Ben-Or [1983]);
- c_2 is the Euler characteristic of S (Björner-Lovasz-Yao [1992]);
- c_3 is the sum of Betti numbers (ranks of homological groups) of S (Yao [1994], Montaña-Morais-Pardo [1996]).

$c_1, c_2 \leq c_3$

Complexity lower bounds via topological invariants

There are several lower bounds for algebraic computation trees over the field either $F = \mathbb{C}$ or $F = \mathbb{R}$ of the form $C(S) \geq \log_2(c_i) - n, \ i = 1, 2, 3$ where

- c_1 is the maximum of the number of connected components in S and in $F^n \setminus S$ (Ben-Or [1983]);
- c_2 is the Euler characteristic of S (Björner-Lovasz-Yao [1992]);
- c_3 is the sum of Betti numbers (ranks of homological groups) of S (Yao [1994], Montaña-Morais-Pardo [1996]).

$c_1, c_2 \leq c_3$

Complexity lower bounds via topological invariants

There are several lower bounds for algebraic computation trees over the field either $F = \mathbb{C}$ or $F = \mathbb{R}$ of the form
$C(S) \geq \log_2(c_i) - n, \ i = 1, 2, 3$ where

- c_1 is the maximum of the number of connected components in S and in $F^n \setminus S$ (Ben-Or [1983]);
- c_2 is the Euler characteristic of S (Björner-Lovasz-Yao [1992]);
- c_3 is the sum of Betti numbers (ranks of homological groups) of S (Yao [1994], Montaña-Morais-Pardo [1996]).

$c_1, c_2 \leq c_3$

Combinatorial problems

- EQUALITY is the set $\{(x_1, \ldots, x_n, y_1, \ldots, y_n) \subset F^{2n} \,:\, \{x_1, \ldots, x_n\} = \{y_1, \ldots, y_n\}\}$;
- DISTINCTNESS is the set $\{(x_1, \ldots, x_n) \,:\, x_i \neq x_j, i \neq j\}$;
- KNAPSACK is the set $\bigcup_{I \subset \{1,\ldots,n\}} \{(x_1, \ldots, x_n) \,:\, \sum_{i \in I} x_i = 1\}$.

Corollary

- *$C(EQUALITY) \asymp n \cdot \log n$ (QuickSort ($F = \mathbb{R}$), elementary symmetric functions ($F = \mathbb{C}$) Strassen [1973], Ben-Or [1983]);*
- *$C(DISTINCTNESS) \asymp n \cdot \log n$ (QuickSort ($F = \mathbb{R}$), discriminant ($F = \mathbb{C}$) Strassen [1973], Ben-Or [1983]);*
- *$C(KNAPSACK) \geq \Omega(n^2)$ (Ben-Or [1983]).*

Combinatorial problems

- EQUALITY is the set $\{(x_1, \ldots, x_n, y_1, \ldots, y_n) \subset F^{2n} : \{x_1, \ldots, x_n\} = \{y_1, \ldots, y_n\}\}$;
- DISTINCTNESS is the set $\{(x_1, \ldots, x_n) : x_i \neq x_j, i \neq j\}$;
- KNAPSACK is the set $\bigcup_{I \subset \{1,\ldots,n\}} \{(x_1, \ldots, x_n) : \sum_{i \in I} x_i = 1\}$.

Corollary

- $C(EQUALITY) \asymp n \cdot \log n$ (QuickSort ($F = \mathbb{R}$), elementary symmetric functions ($F = \mathbb{C}$) Strassen [1973], Ben-Or [1983]);
- $C(DISTINCTNESS) \asymp n \cdot \log n$ (QuickSort ($F = \mathbb{R}$), discriminant ($F = \mathbb{C}$) Strassen [1973], Ben-Or [1983]);
- $C(KNAPSACK) \geq \Omega(n^2)$ (Ben-Or [1983]).

Combinatorial problems

- EQUALITY is the set $\{(x_1, \ldots, x_n, y_1, \ldots, y_n) \subset F^{2n} : \{x_1, \ldots, x_n\} = \{y_1, \ldots, y_n\}\}$;
- DISTINCTNESS is the set $\{(x_1, \ldots, x_n) : x_i \neq x_j, i \neq j\}$;
- KNAPSACK is the set $\bigcup_{I \subset \{1,\ldots,n\}} \{(x_1, \ldots, x_n) : \sum_{i \in I} x_i = 1\}$.

Corollary

- *$C(EQUALITY) \asymp n \cdot \log n$ (QuickSort ($F = \mathbb{R}$), elementary symmetric functions ($F = \mathbb{C}$) Strassen [1973], Ben-Or [1983]);*
- *$C(DISTINCTNESS) \asymp n \cdot \log n$ (QuickSort ($F = \mathbb{R}$), discriminant ($F = \mathbb{C}$) Strassen [1973], Ben-Or [1983]);*
- *$C(KNAPSACK) \geq \Omega(n^2)$ (Ben-Or [1983]).*

Combinatorial problems

- EQUALITY is the set $\{(x_1, \ldots, x_n, y_1, \ldots, y_n) \subset F^{2n} : \{x_1, \ldots, x_n\} = \{y_1, \ldots, y_n\}\}$;
- DISTINCTNESS is the set $\{(x_1, \ldots, x_n) : x_i \neq x_j, i \neq j\}$;
- KNAPSACK is the set $\bigcup_{I \subset \{1,\ldots,n\}} \{(x_1, \ldots, x_n) : \sum_{i \in I} x_i = 1\}$.

Corollary

- *$C(EQUALITY) \asymp n \cdot \log n$ (QuickSort ($F = \mathbb{R}$), elementary symmetric functions ($F = \mathbb{C}$) Strassen [1973], Ben-Or [1983]);*
- *$C(DISTINCTNESS) \asymp n \cdot \log n$ (QuickSort ($F = \mathbb{R}$), discriminant ($F = \mathbb{C}$) Strassen [1973], Ben-Or [1983]);*
- *$C(KNAPSACK) \geq \Omega(n^2)$ (Ben-Or [1983]).*

Combinatorial problems

- EQUALITY is the set $\{(x_1, \ldots, x_n, y_1, \ldots, y_n) \subset F^{2n} : \{x_1, \ldots, x_n\} = \{y_1, \ldots, y_n\}\}$;
- DISTINCTNESS is the set $\{(x_1, \ldots, x_n) : x_i \neq x_j, i \neq j\}$;
- KNAPSACK is the set $\bigcup_{I \subset \{1,\ldots,n\}} \{(x_1, \ldots, x_n) : \sum_{i \in I} x_i = 1\}$.

Corollary

- *$C(EQUALITY) \asymp n \cdot \log n$ (QuickSort ($F = \mathbb{R}$), elementary symmetric functions ($F = \mathbb{C}$) Strassen [1973], Ben-Or [1983]);*
- *$C(DISTINCTNESS) \asymp n \cdot \log n$ (QuickSort ($F = \mathbb{R}$), discriminant ($F = \mathbb{C}$) Strassen [1973], Ben-Or [1983]);*
- *$C(KNAPSACK) \geq \Omega(n^2)$ (Ben-Or [1983]).*

Combinatorial problems

- EQUALITY is the set $\{(x_1, \dots, x_n, y_1, \dots, y_n) \subset F^{2n} : \{x_1, \dots, x_n\} = \{y_1, \dots, y_n\}\}$;
- DISTINCTNESS is the set $\{(x_1, \dots, x_n) : x_i \neq x_j, i \neq j\}$;
- KNAPSACK is the set $\bigcup_{I \subset \{1,\dots,n\}} \{(x_1, \dots, x_n) : \sum_{i \in I} x_i = 1\}$.

Corollary

- *$C(EQUALITY) \asymp n \cdot \log n$ (QuickSort ($F = \mathbb{R}$), elementary symmetric functions ($F = \mathbb{C}$) Strassen [1973], Ben-Or [1983]);*
- *$C(DISTINCTNESS) \asymp n \cdot \log n$ (QuickSort ($F = \mathbb{R}$), discriminant ($F = \mathbb{C}$) Strassen [1973], Ben-Or [1983]);*
- *$C(KNAPSACK) \geq \Omega(n^2)$ (Ben-Or [1983]).*

Algebraic computation trees over positive characteristic $p > 0$

Let F be an algebraically closed field of characteristic $p > 0$. For a variety $S \subset F^n$ introduce $B(S)$ which replaces the sum of Betti numbers (over $\mathbb{C}$ or $\mathbb{R}$).

Zeta function

- Denote $N_k = N_k(S) = \sharp(S \cap \mathbb{F}^n_{p^k})$.
- **Zeta function** $Z(S,t) = \exp(\sum_{1 \le k < \infty} N_k t^k / k) = P(t)/Q(t)$ is a rational function (due to Dwork, Deligne).
- Define $B(S) = \deg(P) + \deg(Q)$.

Theorem

$C(S) \ge a \cdot \log(B(S)) - b \cdot n$ for some constants a, b (Ben-Or [1994])

Corollary

$C(EQUALITY), C(DISTINCTNESS) \asymp n \cdot \log n$ (Strassen [1973], Ben-Or [1994])

Algebraic computation trees over positive characteristic $p > 0$

Let F be an algebraically closed field of characteristic $p > 0$. For a variety $S \subset F^n$ introduce $B(S)$ which replaces the sum of Betti numbers (over $\mathbb{C}$ or $\mathbb{R}$).

Zeta function

- Denote $N_k = N_k(S) = \sharp(S \cap \mathbb{F}^n_{p^k})$.
- **Zeta function** $Z(S,t) = \exp(\sum_{1 \leq k < \infty} N_k t^k / k) = P(t)/Q(t)$ is a rational function (due to Dwork, Deligne).
- Define $B(S) = \deg(P) + \deg(Q)$.

Theorem

$C(S) \geq a \cdot \log(B(S)) - b \cdot n$ for some constants a, b (Ben-Or [1994])

Corollary

$C(EQUALITY), C(DISTINCTNESS) \asymp n \cdot \log n$ (Strassen [1973], Ben-Or [1994])

Algebraic computation trees over positive characteristic $p > 0$

Let F be an algebraically closed field of characteristic $p > 0$. For a variety $S \subset F^n$ introduce $B(S)$ which replaces the sum of Betti numbers (over $\mathbb{C}$ or $\mathbb{R}$).

Zeta function

- Denote $N_k = N_k(S) = \sharp(S \cap \mathbb{F}^n_{p^k})$.
- **Zeta function** $Z(S,t) = \exp(\sum_{1 \le k < \infty} N_k t^k / k) = P(t)/Q(t)$ is a rational function (due to Dwork, Deligne).
- Define $B(S) = \deg(P) + \deg(Q)$.

Theorem

$C(S) \ge a \cdot \log(B(S)) - b \cdot n$ for some constants a, b (Ben-Or [1994])

Corollary

$C(EQUALITY), C(DISTINCTNESS) \asymp n \cdot \log n$ (Strassen [1973], Ben-Or [1994])

Algebraic computation trees over positive characteristic $p > 0$

Let F be an algebraically closed field of characteristic $p > 0$. For a variety $S \subset F^n$ introduce $B(S)$ which replaces the sum of Betti numbers (over $\mathbb{C}$ or $\mathbb{R}$).

Zeta function

- Denote $N_k = N_k(S) = \sharp(S \cap \mathbb{F}^n_{p^k})$.
- **Zeta function** $Z(S,t) = \exp(\sum_{1 \le k < \infty} N_k t^k / k) = P(t)/Q(t)$ is a rational function (due to Dwork, Deligne).
- Define $B(S) = \deg(P) + \deg(Q)$.

Theorem

$C(S) \geq a \cdot \log(B(S)) - b \cdot n$ for some constants a, b (Ben-Or [1994])

Corollary

$C(EQUALITY), C(DISTINCTNESS) \asymp n \cdot \log n$ (Strassen [1973], Ben-Or [1994])

Algebraic computation trees over positive characteristic $p > 0$

Let F be an algebraically closed field of characteristic $p > 0$. For a variety $S \subset F^n$ introduce $B(S)$ which replaces the sum of Betti numbers (over $\mathbb{C}$ or $\mathbb{R}$).

Zeta function

- Denote $N_k = N_k(S) = \sharp(S \cap \mathbb{F}^n_{p^k})$.
- **Zeta function** $Z(S, t) = \exp(\sum_{1 \le k < \infty} N_k t^k / k) = P(t)/Q(t)$ is a rational function (due to Dwork, Deligne).
- Define $B(S) = \deg(P) + \deg(Q)$.

Theorem

$C(S) \geq a \cdot \log(B(S)) - b \cdot n$ *for some constants* a, b *(Ben-Or [1994])*

Corollary

$C(EQUALITY), C(DISTINCTNESS) \asymp n \cdot \log n$ *(Strassen [1973], Ben-Or [1994])*

Algebraic computation trees over positive characteristic $p > 0$

Let F be an algebraically closed field of characteristic $p > 0$. For a variety $S \subset F^n$ introduce $B(S)$ which replaces the sum of Betti numbers (over $\mathbb{C}$ or $\mathbb{R}$).

Zeta function

- Denote $N_k = N_k(S) = \sharp(S \cap \mathbb{F}^n_{p^k})$.
- **Zeta function** $Z(S,t) = \exp(\sum_{1 \le k < \infty} N_k t^k / k) = P(t)/Q(t)$ is a rational function (due to Dwork, Deligne).
- Define $B(S) = \deg(P) + \deg(Q)$.

Theorem

$C(S) \geq a \cdot \log(B(S)) - b \cdot n$ *for some constants* a, b *(Ben-Or [1994])*

Corollary

$C(EQUALITY), C(DISTINCTNESS) \asymp n \cdot \log n$ *(Strassen [1973], Ben-Or [1994])*

Complexity upper bound for membership to a linear complex

An **arrangement** $S \subset F^n$ is a union of hyperplanes $\cup_i H_i$. A face of S is an intersection of some of the hyperplanes. DISTINCTNESS and KNAPSACK are arrangements.

Theorem

Let S be either an arrangement or a (possibly unbounded) polyhedron (when $F = \mathbb{R}$). Denote by N the number of faces (of all the dimensions) of S. Then $C(S) \leq O(n^3 \cdot \log N)$ (Meyer auf der Heide [1985]).

One can generalize the theorem to arbitrary **linear complexes**, i. e. the unions of polyhedra.

Corollary

$C(KNAPSACK) \leq O(n^5)$ (Meyer auf der Heide [1985]).

The corollary does not directly imply P=NP since the construction of an algebraic computation tree is non-uniform (depends on n).

Complexity upper bound for membership to a linear complex

An **arrangement** $S \subset F^n$ is a union of hyperplanes $\cup_i H_i$. A face of S is an intersection of some of the hyperplanes. DISTINCTNESS and KNAPSACK are arrangements.

Theorem

Let S be either an arrangement or a (possibly unbounded) polyhedron (when $F = \mathbb{R}$). Denote by N the number of faces (of all the dimensions) of S. Then $C(S) \leq O(n^3 \cdot \log N)$ (Meyer auf der Heide [1985]).

One can generalize the theorem to arbitrary **linear complexes**, i. e. the unions of polyhedra.

Corollary

$C(KNAPSACK) \leq O(n^5)$ (Meyer auf der Heide [1985]).

The corollary does not directly imply P=NP since the construction of an algebraic computation tree is non-uniform (depends on n).

Complexity upper bound for membership to a linear complex

An **arrangement** $S \subset F^n$ is a union of hyperplanes $\cup_i H_i$. A face of S is an intersection of some of the hyperplanes. DISTINCTNESS and KNAPSACK are arrangements.

Theorem

Let S be either an arrangement or a (possibly unbounded) polyhedron (when $F = \mathbb{R}$). Denote by N the number of faces (of all the dimensions) of S. Then $C(S) \leq O(n^3 \cdot \log N)$ (Meyer auf der Heide [1985]).

One can generalize the theorem to arbitrary **linear complexes**, i. e. the unions of polyhedra.

Corollary

$C(KNAPSACK) \leq O(n^5)$ (Meyer auf der Heide [1985]).

The corollary does not directly imply P=NP since the construction of an algebraic computation tree is non-uniform (depends on n).

Complexity upper bound for membership to a linear complex

An **arrangement** $S \subset F^n$ is a union of hyperplanes $\cup_i H_i$. A face of S is an intersection of some of the hyperplanes. DISTINCTNESS and KNAPSACK are arrangements.

Theorem

Let S be either an arrangement or a (possibly unbounded) polyhedron (when $F = \mathbb{R}$). Denote by N the number of faces (of all the dimensions) of S. Then $C(S) \leq O(n^3 \cdot \log N)$ (Meyer auf der Heide [1985]).

One can generalize the theorem to arbitrary **linear complexes**, i. e. the unions of polyhedra.

Corollary

$C(KNAPSACK) \leq O(n^5)$ (Meyer auf der Heide [1985]).

The corollary does not directly imply P=NP since the construction of an algebraic computation tree is non-uniform (depends on n).

Complexity upper bound for membership to a linear complex

An **arrangement** $S \subset F^n$ is a union of hyperplanes $\cup_i H_i$. A face of S is an intersection of some of the hyperplanes. DISTINCTNESS and KNAPSACK are arrangements.

Theorem

Let S be either an arrangement or a (possibly unbounded) polyhedron (when $F = \mathbb{R}$). Denote by N the number of faces (of all the dimensions) of S. Then $C(S) \leq O(n^3 \cdot \log N)$ (Meyer auf der Heide [1985]).

One can generalize the theorem to arbitrary **linear complexes**, i. e. the unions of polyhedra.

Corollary

$C(KNAPSACK) \leq O(n^5)$ *(Meyer auf der Heide [1985]).*

The corollary does not directly imply P=NP since the construction of an algebraic computation tree is non-uniform (depends on n).

Complexity upper bound for membership to a linear complex

An **arrangement** $S \subset F^n$ is a union of hyperplanes $\cup_i H_i$. A face of S is an intersection of some of the hyperplanes. DISTINCTNESS and KNAPSACK are arrangements.

Theorem

Let S be either an arrangement or a (possibly unbounded) polyhedron (when $F = \mathbb{R}$). Denote by N the number of faces (of all the dimensions) of S. Then $C(S) \leq O(n^3 \cdot \log N)$ (Meyer auf der Heide [1985]).

One can generalize the theorem to arbitrary **linear complexes**, i. e. the unions of polyhedra.

Corollary

$C(KNAPSACK) \leq O(n^5)$ (Meyer auf der Heide [1985]).

The corollary does not directly imply P=NP since the construction of an algebraic computation tree is non-uniform (depends on n).

Membership to a polyhedron

Let $P \subset \mathbb{R}^n$ be a convex polyhedron (with N faces of all the dimensions). Then the sum of its Betti numbers equals 1.

Theorem

Complexity of linear decision trees $C_1(P) \geq \log_2 N$ (Rivest-Yao [1980])

Theorem

$C_d(P) \geq \Omega(\log N)$, provided that $N \geq (dn)^{\Omega(n)}$ (G.-Karpinski-Vorobjov [1994]).

Membership to a polyhedron

Let $P \subset \mathbb{R}^n$ be a convex polyhedron (with N faces of all the dimensions). Then the sum of its Betti numbers equals 1.

Theorem

Complexity of linear decision trees $C_1(P) \geq \log_2 N$ (Rivest-Yao [1980])

Theorem

$C_d(P) \geq \Omega(\log N)$, provided that $N \geq (dn)^{\Omega(n)}$ (G.-Karpinski-Vorobjov [1994]).

Membership to a polyhedron

Let $P \subset \mathbb{R}^n$ be a convex polyhedron (with N faces of all the dimensions). Then the sum of its Betti numbers equals 1.

Theorem

Complexity of linear decision trees $C_1(P) \geq \log_2 N$ (Rivest-Yao [1980])

Theorem

$C_d(P) \geq \Omega(\log N)$, provided that $N \geq (dn)^{\Omega(n)}$ (G.-Karpinski-Vorobjov [1994]).

Pfaffian decision trees

As a testing function g_v at a node v of a Pfaffian d-decision tree appears a Pfaffian function.

Let $v_0, \dots, v_k$ be the nodes on the path from the root v_0 to v_k.
$dg_{v_k} = \sum_{1 \le j \le n} h_{kj}(g_{v_0}, \dots, g_{v_k}, X_1, \dots, X_n) \cdot dX_j$
for some polynomials $h_{kj} \in \mathbb{R}[Z_0, \dots, Z_k, X_1, \dots, X_n]$, $\deg(h_{kj}) \le d$.

The complexity denote by C_d^{pfaff}.

Examples of Pfaffian functions: polynomials, $\exp$, $\log$ (on the positive half-line), $\sin$ (on the interval $(-\pi, \pi)$), $\sqrt{X}$ (on the positive half-line), X^{-1} on $\mathbb{R} \setminus \{0\}$.

Theorem

- *Let $S \subset \mathbb{R}^n$ be semi-pfaffian (defined by inequalities of the form $g \ge 0$ for Pfaffian functions g. Denote by c_3 the sum of its Betti numbers. Then $C_d^{pfaff} \ge \Omega(\sqrt{\log c_3})$ (G.-Vorobjov [1994]);*
- *For a polyhedron P with N faces of all the dimensions $C_d^{pfaff}(P) \ge \Omega(\sqrt{\log N})$, provided that $N \ge (dn)^{\Omega(n^4 \cdot \log d)}$ (G.-Vorobjov [1994]).*

Pfaffian decision trees

As a testing function g_v at a node v of a Pfaffian d-decision tree appears a Pfaffian function.

Let $v_0, \dots, v_k$ be the nodes on the path from the root v_0 to v_k.

$dg_{v_k} = \sum_{1 \le j \le n} h_{kj}(g_{v_0}, \dots, g_{v_k}, X_1, \dots, X_n) \cdot dX_j$

for some polynomials $h_{kj} \in \mathbb{R}[Z_0, \dots, Z_k, X_1, \dots, X_n]$, $\deg(h_{kj}) \le d$.

The complexity denote by C_d^{pfaff}.

Examples of Pfaffian functions: polynomials, $\exp$, $\log$ (on the positive half-line), $\sin$ (on the interval $(-\pi, \pi)$), $\sqrt{X}$ (on the positive half-line), X^{-1} on $\mathbb{R} \setminus \{0\}$.

Theorem

- *Let $S \subset \mathbb{R}^n$ be semi-pfaffian (defined by inequalities of the form $g \ge 0$ for Pfaffian functions g. Denote by c_3 the sum of its Betti numbers. Then $C_d^{pfaff} \ge \Omega(\sqrt{\log c_3})$ (G.-Vorobjov [1994]);*
- *For a polyhedron P with N faces of all the dimensions $C_d^{pfaff}(P) \ge \Omega(\sqrt{\log N})$, provided that $N \ge (dn)^{\Omega(n^4 \cdot \log d)}$ (G.-Vorobjov [1994]).*

Pfaffian decision trees

As a testing function g_v at a node v of a Pfaffian d-decision tree appears a Pfaffian function.

Let $v_0, \dots, v_k$ be the nodes on the path from the root v_0 to v_k.

$dg_{v_k} = \sum_{1 \le j \le n} h_{kj}(g_{v_0}, \dots, g_{v_k}, X_1, \dots, X_n) \cdot dX_j$

for some polynomials $h_{kj} \in \mathbb{R}[Z_0, \dots, Z_k, X_1, \dots, X_n]$, $\deg(h_{kj}) \le d$.

The complexity denote by C_d^{pfaff}.

Examples of Pfaffian functions: polynomials, $\exp$, $\log$ (on the positive half-line), $\sin$ (on the interval $(-\pi, \pi)$), $\sqrt{X}$ (on the positive half-line), X^{-1} on $\mathbb{R} \setminus \{0\}$.

Theorem

- *Let $S \subset \mathbb{R}^n$ be semi-pfaffian (defined by inequalities of the form $g \ge 0$ for Pfaffian functions g. Denote by c_3 the sum of its Betti numbers. Then $C_d^{pfaff} \ge \Omega(\sqrt{\log c_3})$ (G.-Vorobjov [1994]);*
- *For a polyhedron P with N faces of all the dimensions $C_d^{pfaff}(P) \ge \Omega(\sqrt{\log N})$, provided that $N \ge (dn)^{\Omega(n^4 \cdot \log d)}$ (G.-Vorobjov [1994]).*

Pfaffian decision trees

As a testing function g_v at a node v of a Pfaffian d-decision tree appears a Pfaffian function.

Let $v_0, \dots, v_k$ be the nodes on the path from the root v_0 to v_k.

$dg_{v_k} = \sum_{1 \le j \le n} h_{kj}(g_{v_0}, \dots, g_{v_k}, X_1, \dots, X_n) \cdot dX_j$

for some polynomials $h_{kj} \in \mathbb{R}[Z_0, \dots, Z_k, X_1, \dots, X_n]$, $\deg(h_{kj}) \le d$.

The complexity denote by C_d^{pfaff}.

Examples of Pfaffian functions: polynomials, $\exp$, $\log$ (on the positive half-line), $\sin$ (on the interval $(-\pi, \pi)$), $\sqrt{X}$ (on the positive half-line), X^{-1} on $\mathbb{R} \setminus \{0\}$.

Theorem

- *Let $S \subset \mathbb{R}^n$ be semi-pfaffian (defined by inequalities of the form $g \ge 0$ for Pfaffian functions g. Denote by c_3 the sum of its Betti numbers. Then $C_d^{pfaff} \ge \Omega(\sqrt{\log c_3})$ (G.-Vorobjov [1994]);*
- *For a polyhedron P with N faces of all the dimensions $C_d^{pfaff}(P) \ge \Omega(\sqrt{\log N})$, provided that $N \ge (dn)^{\Omega(n^4 \cdot \log d)}$ (G.-Vorobjov [1994]).*

Pfaffian decision trees

As a testing function g_v at a node v of a Pfaffian d-decision tree appears a Pfaffian function.

Let $v_0, \dots, v_k$ be the nodes on the path from the root v_0 to v_k.

$dg_{v_k} = \sum_{1 \le j \le n} h_{kj}(g_{v_0}, \dots, g_{v_k}, X_1, \dots, X_n) \cdot dX_j$

for some polynomials $h_{kj} \in \mathbb{R}[Z_0, \dots, Z_k, X_1, \dots, X_n]$, $\deg(h_{kj}) \le d$.

The complexity denote by C_d^{pfaff}.

Examples of Pfaffian functions: polynomials, $\exp$, $\log$ (on the positive half-line), $\sin$ (on the interval $(-\pi, \pi)$), $\sqrt{X}$ (on the positive half-line), X^{-1} on $\mathbb{R} \setminus \{0\}$.

Theorem

- *Let $S \subset \mathbb{R}^n$ be semi-pfaffian (defined by inequalities of the form $g \ge 0$ for Pfaffian functions g. Denote by c_3 the sum of its Betti numbers. Then $C_d^{pfaff} \ge \Omega(\sqrt{\log c_3})$ (G.-Vorobjov [1994]);*
- *For a polyhedron P with N faces of all the dimensions $C_d^{pfaff}(P) \ge \Omega(\sqrt{\log N})$, provided that $N \ge (dn)^{\Omega(n^4 \cdot \log d)}$ (G.-Vorobjov [1994]).*

Pfaffian decision trees

As a testing function g_v at a node v of a Pfaffian d-decision tree appears a Pfaffian function.

Let $v_0, \dots, v_k$ be the nodes on the path from the root v_0 to v_k.

$dg_{v_k} = \sum_{1 \leq j \leq n} h_{kj}(g_{v_0}, \dots, g_{v_k}, X_1, \dots, X_n) \cdot dX_j$

for some polynomials $h_{kj} \in \mathbb{R}[Z_0, \dots, Z_k, X_1, \dots, X_n]$, $\deg(h_{kj}) \leq d$.

The complexity denote by C_d^{pfaff}.

Examples of Pfaffian functions: polynomials, $\exp$, $\log$ (on the positive half-line), $\sin$ (on the interval $(-\pi, \pi)$), $\sqrt{X}$ (on the positive half-line), X^{-1} on $\mathbb{R} \setminus \{0\}$.

Theorem

- *Let $S \subset \mathbb{R}^n$ be semi-pfaffian (defined by inequalities of the form $g \geq 0$ for Pfaffian functions g. Denote by c_3 the sum of its Betti numbers. Then $C_d^{pfaff} \geq \Omega(\sqrt{\log c_3})$ (G.-Vorobjov [1994]);*
- *For a polyhedron P with N faces of all the dimensions $C_d^{pfaff}(P) \geq \Omega(\sqrt{\log N})$, provided that $N \geq (dn)^{\Omega(n^4 \cdot \log d)}$ (G.-Vorobjov [1994]).*

Probabilistic decision and computation trees

Let $\{T_i\}_{1\leq i\leq s}$ be a collection of either algebraic d-decision or, respectively computation trees with attributed probabilities $p_1, \dots, p_s$, $\sum_{1\leq i\leq s} p_i = 1$. It is called **probabilistic algebraic d-decision** or, respectively **probabilistic computation tree.**

A probabilistic tree accepts set $S \subset F^n$ if for any input $x \in F^n$ the output is correct with probability $> 2/3$.

The complexity C_d^{prob} is defined as expectation $\sum_{1\leq i\leq s} p_i \cdot C_d(T_i)$. Similarly, one defines C^{prob}.

One can also consider a continuous distribution of trees.

Speed-up by probabilistic computation trees

- $C^{prob}(EQUALITY) \asymp n$ (Bürgisser-Karpinski-Lickteig [1992]);
- Consider polynomials $f(Z) = \prod_{1\leq i\leq n}(Z - x_i)$, $h(Z) = \prod_{1\leq i\leq n}(Z - y_i)$. Sets $\{x_1, \dots, x_n\} = \{y_1, \dots, y_n\} \Leftrightarrow f \equiv h$;
- Choose randomly $z_0 \in F$ and test whether $f(z_0) = h(z_0)$. If yes, return "*accept*", else "*reject*".

Probabilistic decision and computation trees

Let $\{T_i\}_{1\le i\le s}$ be a collection of either algebraic d-decision or, respectively computation trees with attributed probabilities $p_1,\dots,p_s$, $\sum_{1\le i\le s} p_i = 1$. It is called **probabilistic algebraic d-decision** or, respectively **probabilistic computation tree.**

A probabilistic tree accepts set $S\subset F^n$ if for any input $x\in F^n$ the output is correct with probability $>2/3$.

The complexity C_d^{prob} is defined as expectation $\sum_{1\le i\le s} p_i\cdot C_d(T_i)$.

Similarly, one defines C^{prob}.

One can also consider a continuous distribution of trees.

Speed-up by probabilistic computation trees

- $C^{prob}(EQUALITY)\asymp n$ (Bürgisser-Karpinski-Lickteig [1992]);
- Consider polynomials $f(Z)=\prod_{1\le i\le n}(Z-x_i)$, $h(Z)=\prod_{1\le i\le n}(Z-y_i)$. Sets $\{x_1,\dots,x_n\}=\{y_1,\dots,y_n\}\Leftrightarrow f\equiv h$;
- Choose randomly $z_0\in F$ and test whether $f(z_0)=h(z_0)$. If yes, return "*accept*", else "*reject*".

Probabilistic decision and computation trees

Let $\{T_i\}_{1 \leq i \leq s}$ be a collection of either algebraic d-decision or, respectively computation trees with attributed probabilities $p_1, \ldots, p_s$, $\sum_{1 \leq i \leq s} p_i = 1$. It is called **probabilistic algebraic** ***d*-decision** or, respectively **probabilistic computation tree**.

A probabilistic tree accepts set $S \subset F^n$ if for any input $x \in F^n$ the output is correct with probability $> 2/3$.

The complexity C_d^{prob} is defined as expectation $\sum_{1 \leq i \leq s} p_i \cdot C_d(T_i)$. Similarly, one defines C^{prob}.

One can also consider a continuous distribution of trees.

Speed-up by probabilistic computation trees

- $C^{prob}(EQUALITY) \asymp n$ (Bürgisser-Karpinski-Lickteig [1992]);
- Consider polynomials $f(Z) = \prod_{1 \leq i \leq n}(Z - x_i)$, $h(Z) = \prod_{1 \leq i \leq n}(Z - y_i)$. Sets $\{x_1, \ldots, x_n\} = \{y_1, \ldots, y_n\} \Leftrightarrow f \equiv h$;
- Choose randomly $z_0 \in F$ and test whether $f(z_0) = h(z_0)$. If yes, return "*accept*", else "*reject*".

Probabilistic decision and computation trees

Let $\{T_i\}_{1\le i\le s}$ be a collection of either algebraic d-decision or, respectively computation trees with attributed probabilities $p_1,\dots,p_s$, $\sum_{1\le i\le s} p_i = 1$. It is called **probabilistic algebraic d-decision** or, respectively **probabilistic computation tree**.

A probabilistic tree accepts set $S \subset F^n$ if for any input $x \in F^n$ the output is correct with probability $> 2/3$.

The complexity C_d^{prob} is defined as expectation $\sum_{1\le i\le s} p_i \cdot C_d(T_i)$. Similarly, one defines C^{prob}.

One can also consider a continuous distribution of trees.

Speed-up by probabilistic computation trees

- $C^{prob}(EQUALITY) \asymp n$ (Bürgisser-Karpinski-Lickteig [1992]);
- Consider polynomials $f(Z) = \prod_{1\le i\le n}(Z - x_i)$, $h(Z) = \prod_{1\le i\le n}(Z - y_i)$. Sets $\{x_1,\dots,x_n\} = \{y_1,\dots,y_n\} \Leftrightarrow f \equiv h$;
- Choose randomly $z_0 \in F$ and test whether $f(z_0) = h(z_0)$. If yes, return "*accept*", else "*reject*".

Probabilistic decision and computation trees

Let $\{T_i\}_{1 \le i \le s}$ be a collection of either algebraic d-decision or, respectively computation trees with attributed probabilities $p_1, \dots, p_s$, $\sum_{1 \le i \le s} p_i = 1$. It is called **probabilistic algebraic d-decision** or, respectively **probabilistic computation tree**.

A probabilistic tree accepts set $S \subset F^n$ if for any input $x \in F^n$ the output is correct with probability $> 2/3$.

The complexity C_d^{prob} is defined as expectation $\sum_{1 \le i \le s} p_i \cdot C_d(T_i)$. Similarly, one defines C^{prob}.

One can also consider a continuous distribution of trees.

Speed-up by probabilistic computation trees

- $C^{prob}(EQUALITY) \asymp n$ (Bürgisser-Karpinski-Lickteig [1992]);
- Consider polynomials $f(Z) = \prod_{1 \le i \le n}(Z - x_i)$, $h(Z) = \prod_{1 \le i \le n}(Z - y_i)$. Sets $\{x_1, \dots, x_n\} = \{y_1, \dots, y_n\} \Leftrightarrow f \equiv h$;
- Choose randomly $z_0 \in F$ and test whether $f(z_0) = h(z_0)$. If yes, return "*accept*", else "*reject*".

Probabilistic decision and computation trees

Let $\{T_i\}_{1\leq i\leq s}$ be a collection of either algebraic d-decision or, respectively computation trees with attributed probabilities $p_1,\dots,p_s$, $\sum_{1\leq i\leq s} p_i = 1$. It is called **probabilistic algebraic d-decision** or, respectively **probabilistic computation tree.**

A probabilistic tree accepts set $S \subset F^n$ if for any input $x \in F^n$ the output is correct with probability $> 2/3$.

The complexity C_d^{prob} is defined as expectation $\sum_{1\leq i\leq s} p_i \cdot C_d(T_i)$.

Similarly, one defines C^{prob}.

One can also consider a continuous distribution of trees.

Speed-up by probabilistic computation trees

- $C^{prob}(EQUALITY) \asymp n$ (Bürgisser-Karpinski-Lickteig [1992]);
- Consider polynomials $f(Z) = \prod_{1\leq i\leq n}(Z - x_i)$, $h(Z) = \prod_{1\leq i\leq n}(Z - y_i)$. Sets $\{x_1,\dots,x_n\} = \{y_1,\dots,y_n\} \Leftrightarrow f \equiv h$;
- Choose randomly $z_0 \in F$ and test whether $f(z_0) = h(z_0)$. If yes, return "*accept*", else "*reject*".

Probabilistic decision and computation trees

Let $\{T_i\}_{1\leq i\leq s}$ be a collection of either algebraic d-decision or, respectively computation trees with attributed probabilities $p_1,\dots,p_s$, $\sum_{1\leq i\leq s} p_i = 1$. It is called **probabilistic algebraic d-decision** or, respectively **probabilistic computation tree**.

A probabilistic tree accepts set $S \subset F^n$ if for any input $x \in F^n$ the output is correct with probability $> 2/3$.

The complexity C_d^{prob} is defined as expectation $\sum_{1\leq i\leq s} p_i \cdot C_d(T_i)$.

Similarly, one defines C^{prob}.

One can also consider a continuous distribution of trees.

Speed-up by probabilistic computation trees

- $C^{prob}(EQUALITY) \asymp n$ (Bürgisser-Karpinski-Lickteig [1992]);
- Consider polynomials $f(Z) = \prod_{1\leq i\leq n}(Z - x_i)$, $h(Z) = \prod_{1\leq i\leq n}(Z - y_i)$. Sets $\{x_1,\dots,x_n\} = \{y_1,\dots,y_n\} \Leftrightarrow f \equiv h$;
- Choose randomly $z_0 \in F$ and test whether $f(z_0) = h(z_0)$. If yes, return "*accept*", else "*reject*".

Lower bounds for probabilistic algebraic trees

The topological methods and differential-geometric methods fail for probabilistic trees (another evidence is the latter example).

Let $S \subset F^n$ be a polyhedron (when $F = \mathbb{R}$) or an arrangement determined by m hyperplanes having $N \geq m^{\Omega(n)}$ faces.

Theorem

1) $C_d^{prob}(S) \geq \Omega(\log N)$ *for constant d (G.-Karpinski-Meyer auf der Heide-Smolensky [1995]);*
2) $C^{prob}(S) \geq \Omega(\log N)$ *(G. [1997]).*

Corollary

1) $C_d^{prob}(DISTINCTNESS) \asymp n \cdot \log n$, $C_d^{prob}(KNAPSACK) \geq \Omega(n^2)$ *(G.-Karpinski-Meyer auf der Heide-Smolensky [1995]);*
2) $C^{prob}(DISTINCTNESS) \asymp n \cdot \log n$, $C^{prob}(KNAPSACK) \geq \Omega(n^2)$ *(G.).*

EQUALITY is a union of n-dimensional planes in F^{2n} (not an arrangement), thus the Theorem is not applicable.

Lower bounds for probabilistic algebraic trees

The topological methods and differential-geometric methods fail for probabilistic trees (another evidence is the latter example).

Let $S \subset F^n$ be a polyhedron (when $F = \mathbb{R}$) or an arrangement determined by m hyperplanes having $N \geq m^{\Omega(n)}$ faces.

Theorem

1) $C_d^{prob}(S) \geq \Omega(\log N)$ for constant d (G.-Karpinski-Meyer auf der Heide-Smolensky [1995]);
2) $C^{prob}(S) \geq \Omega(\log N)$ (G. [1997]).

Corollary

1) $C_d^{prob}(DISTINCTNESS) \asymp n \cdot \log n$, $C_d^{prob}(KNAPSACK) \geq \Omega(n^2)$ (G.-Karpinski-Meyer auf der Heide-Smolensky [1995]);
2) $C^{prob}(DISTINCTNESS) \asymp n \cdot \log n$, $C^{prob}(KNAPSACK) \geq \Omega(n^2)$ (G.).

EQUALITY is a union of n-dimensional planes in F^{2n} (not an arrangement), thus the Theorem is not applicable.

Lower bounds for probabilistic algebraic trees

The topological methods and differential-geometric methods fail for probabilistic trees (another evidence is the latter example).
Let $S \subset F^n$ be a polyhedron (when $F = \mathbb{R}$) or an arrangement determined by m hyperplanes having $N \geq m^{\Omega(n)}$ faces.

Theorem

1) $C_d^{prob}(S) \geq \Omega(\log N)$ for constant d (G.-Karpinski-Meyer auf der Heide-Smolensky [1995]);
2) $C^{prob}(S) \geq \Omega(\log N)$ (G. [1997]).

Corollary

1) $C_d^{prob}(DISTINCTNESS) \asymp n \cdot \log n$, $C_d^{prob}(KNAPSACK) \geq \Omega(n^2)$ (G.-Karpinski-Meyer auf der Heide-Smolensky [1995]);
2) $C^{prob}(DISTINCTNESS) \asymp n \cdot \log n$, $C^{prob}(KNAPSACK) \geq \Omega(n^2)$ (G.).

EQUALITY is a union of n-dimensional planes in F^{2n} (not an arrangement), thus the Theorem is not applicable.

Lower bounds for probabilistic algebraic trees

The topological methods and differential-geometric methods fail for probabilistic trees (another evidence is the latter example).
Let $S \subset F^n$ be a polyhedron (when $F = \mathbb{R}$) or an arrangement determined by m hyperplanes having $N \geq m^{\Omega(n)}$ faces.

Theorem

1) $C_d^{prob}(S) \geq \Omega(\log N)$ for constant d (G.-Karpinski-Meyer auf der Heide-Smolensky [1995]);
2) $C^{prob}(S) \geq \Omega(\log N)$ (G. [1997]).

Corollary

1) $C_d^{prob}(DISTINCTNESS) \asymp n \cdot \log n$, $C_d^{prob}(KNAPSACK) \geq \Omega(n^2)$ (G.-Karpinski-Meyer auf der Heide-Smolensky [1995]);
2) $C^{prob}(DISTINCTNESS) \asymp n \cdot \log n$, $C^{prob}(KNAPSACK) \geq \Omega(n^2)$ (G.).

EQUALITY is a union of n-dimensional planes in F^{2n} (not an arrangement), thus the Theorem is not applicable.

Lower bounds for probabilistic algebraic trees

The topological methods and differential-geometric methods fail for probabilistic trees (another evidence is the latter example).
Let $S \subset F^n$ be a polyhedron (when $F = \mathbb{R}$) or an arrangement determined by m hyperplanes having $N \geq m^{\Omega(n)}$ faces.

Theorem

1) $C_d^{prob}(S) \geq \Omega(\log N)$ for constant d (G.-Karpinski-Meyer auf der Heide-Smolensky [1995]);
2) $C^{prob}(S) \geq \Omega(\log N)$ (G. [1997]).

Corollary

1) $C_d^{prob}(DISTINCTNESS) \asymp n \cdot \log n$, $C_d^{prob}(KNAPSACK) \geq \Omega(n^2)$ (G.-Karpinski-Meyer auf der Heide-Smolensky [1995]);
2) $C^{prob}(DISTINCTNESS) \asymp n \cdot \log n$, $C^{prob}(KNAPSACK) \geq \Omega(n^2)$ (G.).

EQUALITY is a union of n-dimensional planes in F^{2n} (not an arrangement), thus the Theorem is not applicable.

Lower bounds for probabilistic algebraic trees

The topological methods and differential-geometric methods fail for probabilistic trees (another evidence is the latter example).
Let $S \subset F^n$ be a polyhedron (when $F = \mathbb{R}$) or an arrangement determined by m hyperplanes having $N \geq m^{\Omega(n)}$ faces.

Theorem

1) $C_d^{prob}(S) \geq \Omega(\log N)$ for constant d (G.-Karpinski-Meyer auf der Heide-Smolensky [1995]);
2) $C^{prob}(S) \geq \Omega(\log N)$ (G. [1997]).

Corollary

1) $C_d^{prob}(DISTINCTNESS) \asymp n \cdot \log n$, $C_d^{prob}(KNAPSACK) \geq \Omega(n^2)$ (G.-Karpinski-Meyer auf der Heide-Smolensky [1995]);
2) $C^{prob}(DISTINCTNESS) \asymp n \cdot \log n$, $C^{prob}(KNAPSACK) \geq \Omega(n^2)$ (G.).

EQUALITY is a union of n-dimensional planes in F^{2n} (not an arrangement), thus the Theorem is not applicable.

Lower bounds for probabilistic algebraic trees

The topological methods and differential-geometric methods fail for probabilistic trees (another evidence is the latter example).
Let $S \subset F^n$ be a polyhedron (when $F = \mathbb{R}$) or an arrangement determined by m hyperplanes having $N \geq m^{\Omega(n)}$ faces.

Theorem

1) $C_d^{prob}(S) \geq \Omega(\log N)$ *for constant d (G.-Karpinski-Meyer auf der Heide-Smolensky [1995]);*
2) $C^{prob}(S) \geq \Omega(\log N)$ *(G. [1997]).*

Corollary

1) $C_d^{prob}(DISTINCTNESS) \asymp n \cdot \log n$, $C_d^{prob}(KNAPSACK) \geq \Omega(n^2)$ *(G.-Karpinski-Meyer auf der Heide-Smolensky [1995]);*
2) $C^{prob}(DISTINCTNESS) \asymp n \cdot \log n$, $C^{prob}(KNAPSACK) \geq \Omega(n^2)$ *(G.).*

EQUALITY is a union of n-dimensional planes in F^{2n} (not an arrangement), thus the Theorem is not applicable.

Probabilistic computation trees over algebraically closed fields of zero characteristic

Let $S = H_1 \cup \cdots \cup H_m \subset F^n$ be an arrangement.

Theorem

Assume that for a certain $c > 0$ any subarrangement $H_{i_1} \cup \cdots \cup H_{i_{\lfloor cm \rfloor}}$ containing $\lfloor cm \rfloor$ hyperplanes, has $\geq N$ faces (of all the dimensions). Then $C^{prob}(S) \geq \Omega(\log N - 2 \cdot n)$ (G. [1997]).

Corollary

- $C^{prob}(DISTINCTNESS) \asymp n \cdot \log n$;
- $C^{prob}(KNAPSACK) \geq \Omega(n^2)$.

Probabilistic computation trees over algebraically closed fields of zero characteristic

Let $S = H_1 \cup \dots \cup H_m \subset F^n$ be an arrangement.

Theorem

Assume that for a certain $c > 0$ any subarrangement $H_{i_1} \cup \dots \cup H_{i_{\lfloor cm \rfloor}}$ containing $\lfloor cm \rfloor$ hyperplanes, has $\geq N$ faces (of all the dimensions). Then $C^{prob}(S) \geq \Omega(\log N - 2 \cdot n)$ (G. [1997]).

Corollary

- $C^{prob}(DISTINCTNESS) \asymp n \cdot \log n$;
- $C^{prob}(KNAPSACK) \geq \Omega(n^2)$.

Probabilistic computation trees over algebraically closed fields of zero characteristic

Let $S = H_1 \cup \cdots \cup H_m \subset F^n$ be an arrangement.

Theorem

Assume that for a certain $c > 0$ any subarrangement $H_{i_1} \cup \cdots \cup H_{i_{\lfloor cm \rfloor}}$ containing $\lfloor cm \rfloor$ hyperplanes, has $\geq N$ faces (of all the dimensions). Then $C^{prob}(S) \geq \Omega(\log N - 2 \cdot n)$ (G. [1997]).

Corollary

- $C^{prob}(DISTINCTNESS) \asymp n \cdot \log n$;
- $C^{prob}(KNAPSACK) \geq \Omega(n^2)$.

Analytic and topological decision trees

If a decision tree admits arbitrary polynomials as testing functions we call it **topological tree** and its complexity denote by $C_{top} \leq C, C_d$ (Shub-Smale).

I. e. computations are gratis, only branchings are counted.

If a decision tree admits arbitrary analytic functions as testing functions we call it **analytic tree** and its complexity denote by $C_{an} \leq C_{top}$.

MAX

- ORTANT $= \mathbb{R}^n_+ = \{(x_1, \dots, x_n) : x_1 \geq 0, \dots, x_n \geq 0\} \Leftrightarrow$
- MAX= $= \{(x_1, \dots, x_n) : x_1 = \max\{x_1, \dots, x_n\}\}$ is a weaker problem than
- MAX: to compute $\max\{x_1, \dots, x_n\}$ by means of a modification of a decision (or computation) tree in which the outputs of leaves are functions (polynomials) rather than labels "*accept*" or "*reject*".

Analytic and topological decision trees

If a decision tree admits arbitrary polynomials as testing functions we call it **topological tree** and its complexity denote by $C_{top} \leq C, C_d$ (Shub-Smale).
I. e. computations are gratis, only branchings are counted.

If a decision tree admits arbitrary analytic functions as testing functions we call it **analytic tree** and its complexity denote by $C_{an} \leq C_{top}$.

MAX

- ORTANT $= \mathbb{R}^n_+ = \{(x_1, \ldots, x_n) : x_1 \geq 0, \ldots, x_n \geq 0\} \Leftrightarrow$
- MAX= $= \{(x_1, \ldots, x_n) : x_1 = \max\{x_1, \ldots, x_n\}\}$
  is a weaker problem than
- MAX: to compute $\max\{x_1, \ldots, x_n\}$ by means of a modification of a decision (or computation) tree in which the outputs of leaves are functions (polynomials) rather than labels "*accept*" or "*reject*".

Analytic and topological decision trees

If a decision tree admits arbitrary polynomials as testing functions we call it **topological tree** and its complexity denote by $C_{top} \leq C, C_d$ (Shub-Smale).
I. e. computations are gratis, only branchings are counted.

If a decision tree admits arbitrary analytic functions as testing functions we call it **analytic tree** and its complexity denote by $C_{an} \leq C_{top}$.

MAX

- ORTANT $= \mathbb{R}^n_+ = \{(x_1, \dots, x_n) : x_1 \geq 0, \dots, x_n \geq 0\} \Leftrightarrow$
- MAX= $= \{(x_1, \dots, x_n) : x_1 = \max\{x_1, \dots, x_n\}\}$ is a weaker problem than
- MAX: to compute $\max\{x_1, \dots, x_n\}$ by means of a modification of a decision (or computation) tree in which the outputs of leaves are functions (polynomials) rather than labels "*accept*" or "*reject*".

Analytic and topological decision trees

If a decision tree admits arbitrary polynomials as testing functions we call it **topological tree** and its complexity denote by $C_{top} \leq C, C_d$ (Shub-Smale).
I. e. computations are gratis, only branchings are counted.

If a decision tree admits arbitrary analytic functions as testing functions we call it **analytic tree** and its complexity denote by $C_{an} \leq C_{top}$.

MAX

- ORTANT $= \mathbb{R}^n_+ = \{(x_1, \ldots, x_n) \, : \, x_1 \geq 0, \ldots, x_n \geq 0\} \Leftrightarrow$
- MAX= $= \{(x_1, \ldots, x_n) \, : \, x_1 = \max\{x_1, \ldots, x_n\}\}$
  is a weaker problem than
- MAX: to compute $\max\{x_1, \ldots, x_n\}$ by means of a modification of a decision (or computation) tree in which the outputs of leaves are functions (polynomials) rather than labels "*accept*" or "*reject*".

$C_1(MAX) \leq n - 1$

Theorem

- $C_{an}(MAX =) = n - 1$ *(Rabin [1972]; corrected proof Montaña-Pardo-Recio [1994]; short proof G.-Karpinski-Smolensky [1995]);*
- $C_d^{prob}(MAX =) \geq n/(2 \cdot d)$, $C^{prob}(MAX =) \geq n/4$ *(G.-K.-S. [1995]);*
- $C_{top}^{prob}(MAX =) \leq C_n^{prob}(MAX =) \leq O(\log^2 n)$ *(G.-K.-S. [1995]);*
- $C_{top}^{prob}(MAX) \leq C_n^{prob}(MAX) \leq O(\log^2 n)$ *(Ben-Or [1996];* $\leq O(\log^5 n)$ *G.-K.-S. [1995]).*

$C_1(MAX) \leq n - 1$

Theorem

- $C_{an}(MAX =) = n - 1$ *(Rabin [1972]; corrected proof Montaña-Pardo-Recio [1994]; short proof G.-Karpinski-Smolensky [1995]);*
- $C_d^{prob}(MAX =) \geq n/(2 \cdot d)$, $C^{prob}(MAX =) \geq n/4$ *(G.-K.-S. [1995]);*
- $C_{top}^{prob}(MAX =) \leq C_n^{prob}(MAX =) \leq O(\log^2 n)$ *(G.-K.-S. [1995]);*
- $C_{top}^{prob}(MAX) \leq C_n^{prob}(MAX) \leq O(\log^2 n)$ *(Ben-Or [1996];* $\leq O(\log^5 n)$ *G.-K.-S. [1995]).*

$C_1(MAX) \leq n-1$

Theorem

- $C_{an}(MAX =) = n - 1$ *(Rabin [1972]; corrected proof Montaña-Pardo-Recio [1994]; short proof G.-Karpinski-Smolensky [1995]);*
- $C_d^{prob}(MAX =) \geq n/(2 \cdot d)$, $\quad C^{prob}(MAX =) \geq n/4$ *(G.-K.-S. [1995]);*
- $C_{top}^{prob}(MAX =) \leq C_n^{prob}(MAX =) \leq O(\log^2 n)$ *(G.-K.-S. [1995]);*
- $C_{top}^{prob}(MAX) \leq C_n^{prob}(MAX) \leq O(\log^2 n)$ *(Ben-Or [1996];* $\leq O(\log^5 n)$ *G.-K.-S. [1995]).*

$C_1(MAX) \leq n-1$

Theorem

- $C_{an}(MAX =) = n-1$ *(Rabin [1972]; corrected proof Montaña-Pardo-Recio [1994]; short proof G.-Karpinski-Smolensky [1995]);*
- $C_d^{prob}(MAX =) \geq n/(2 \cdot d)$, $C^{prob}(MAX =) \geq n/4$ *(G.-K.-S. [1995]);*
- $C_{top}^{prob}(MAX =) \leq C_n^{prob}(MAX =) \leq O(\log^2 n)$ *(G.-K.-S. [1995]);*
- $C_{top}^{prob}(MAX) \leq C_n^{prob}(MAX) \leq O(\log^2 n)$ *(Ben-Or [1996];* $\leq O(\log^5 n)$ *G.-K.-S. [1995]).*

$C_1(MAX) \leq n-1$

Theorem

- $C_{an}(MAX =) = n-1$ *(Rabin [1972]; corrected proof Montaña-Pardo-Recio [1994]; short proof G.-Karpinski-Smolensky [1995]);*
- $C_d^{prob}(MAX =) \geq n/(2 \cdot d)$, $\quad C^{prob}(MAX =) \geq n/4$ *(G.-K.-S. [1995]);*
- $C_{top}^{prob}(MAX =) \leq C_n^{prob}(MAX =) \leq O(\log^2 n)$ *(G.-K.-S. [1995]);*
- $C_{top}^{prob}(MAX) \leq C_n^{prob}(MAX) \leq O(\log^2 n)$ *(Ben-Or [1996];* $\leq O(\log^5 n)$ *G.-K.-S. [1995]).*

Topological complexity of the range searching

For $f_1, \ldots, f_m \in \mathbb{R}[X_1, \ldots, X_n]$ we say that a decision/computation tree T solves the RANGE SEARCHING problem if any two inputs $x, y \in \mathbb{R}^n$ with different sign vectors
$(\mathrm{sgn}(f_1), \ldots, \mathrm{sgn}(f_n))(x) \neq (\mathrm{sgn}(f_1), \ldots, \mathrm{sgn}(f_n))(y)$
arrive to different leaves of T. Denote by N the number of sign vectors.

Theorem

C_{top} *(RANGE SEARCHING)* $\asymp \log N$ *(G. [1998]).*

Topological complexity of the range searching

For $f_1, \ldots, f_m \in \mathbb{R}[X_1, \ldots, X_n]$ we say that a decision/computation tree T solves the RANGE SEARCHING problem if any two inputs $x, y \in \mathbb{R}^n$ with different sign vectors
$(\mathrm{sgn}(f_1), \ldots, \mathrm{sgn}(f_n))(x) \neq (\mathrm{sgn}(f_1), \ldots, \mathrm{sgn}(f_n))(y)$
arrive to different leaves of T. Denote by N the number of sign vectors.

Theorem

C_{top} (RANGE SEARCHING) $\asymp \log N$ (G. [1998]).

Lower bound for probabilistic analytic trees

We have seen that for MAX tossing a coin can (exponentially) speed-up computation. Here is an example of a set for which it is not the case.

Let an integer $q \neq 2^m$. Consider set $MOD_q = \{(x_1, \ldots, x_n) \in \mathbb{R}^n : \prod_{1 \le i \le n} x_i \neq 0,\ q \mid \sharp\{i : x_i < 0\}\}$.

Theorem

$C_{an}^{prob}(MOD_q) \geq \Omega(\sqrt{n})$ *(G.-Karpinski-Smolensly [1995]).*

Lower bound for probabilistic analytic trees

We have seen that for MAX tossing a coin can (exponentially) speed-up computation. Here is an example of a set for which it is not the case.

Let an integer $q \neq 2^m$. Consider set $\mathrm{MOD}_q = \{(x_1, \ldots, x_n) \in \mathbb{R}^n : \prod_{1 \leq i \leq n} x_i \neq 0,\ q \mid \sharp\{i : x_i < 0\}\}$.

Theorem

$C_{an}^{prob}(MOD_q) \geq \Omega(\sqrt{n})$ *(G.-Karpinski-Smolensly [1995]).*

Lower bound for probabilistic analytic trees

We have seen that for MAX tossing a coin can (exponentially) speed-up computation. Here is an example of a set for which it is not the case.

Let an integer $q \neq 2^m$. Consider set
$MOD_q = \{(x_1, \ldots, x_n) \in \mathbb{R}^n : \prod_{1 \leq i \leq n} x_i \neq 0,\ q \mid \sharp\{i : x_i < 0\}\}$.

Theorem

$C_{an}^{prob}(MOD_q) \geq \Omega(\sqrt{n})$ (G.-Karpinski-Smolensly [1995]).

Size of decision trees

So far, we studied the depth of decision/computation trees. Since *depth* $\geq \log($*size*$)$, the bounds on the size are stronger.

Consider set
$EXACT_n = \{(x_1, \dots, x_{2 \cdot n}) \in \mathbb{R}^{2 \cdot n} \, : \, \prod_{1 \leq i \leq 2 \cdot n} x_i \neq 0, \sharp\{i \, : \, x_i < 0\} = n\}$.

Theorem

For analytic decision trees $size_{an}(EXACT_n) \geq 2^n / n$ *(G.-K.-S. [1995]).*

For linear decision trees $size_1(MAX =) \leq 2 \cdot n$;

Alternatively to usual (binary) trees one can consider **ternary** decision trees in which a node v with a testing function g_v branches into 3 nodes according to whether $g_v < 0$, or $g_v = 0$, or $g_v > 0$.

Theorem

Ternary $size_d(MAX =) \geq 2^{c_d \cdot n}$ *for d-decision trees,* $c_d > 0$ *(G.-Karpinski-Yao [1994])*

Open question: size (MAX) for binary decision/computation trees?

Size of decision trees

So far, we studied the depth of decision/computation trees. Since *depth* $\geq \log(size)$, the bounds on the size are stronger.

Consider set

$\text{EXACT}_n = \{(x_1, \ldots, x_{2\cdot n}) \in \mathbb{R}^{2\cdot n} : \prod_{1 \leq i \leq 2\cdot n} x_i \neq 0, \sharp\{i : x_i < 0\} = n\}$.

Theorem

For analytic decision trees $size_{an}(EXACT_n) \geq 2^n/n$ *(G.-K.-S. [1995]).*

For linear decision trees $size_1(MAX =) \leq 2 \cdot n$;

Alternatively to usual (binary) trees one can consider **ternary** decision trees in which a node v with a testing function g_v branches into 3 nodes according to whether $g_v < 0$, or $g_v = 0$, or $g_v > 0$.

Theorem

Ternary $size_d(MAX =) \geq 2^{c_d \cdot n}$ *for d-decision trees,* $c_d > 0$ *(G.-Karpinski-Yao [1994])*

Open question: size (MAX) for binary decision/computation trees?

Size of decision trees

So far, we studied the depth of decision/computation trees. Since *depth* $\geq \log(size)$, the bounds on the size are stronger.

Consider set

$\text{EXACT}_n = \{(x_1, \ldots, x_{2\cdot n}) \in \mathbb{R}^{2\cdot n} : \prod_{1\leq i\leq 2\cdot n} x_i \neq 0, \sharp\{i : x_i < 0\} = n\}$.

Theorem

For analytic decision trees $size_{an}(EXACT_n) \geq 2^n/n$ *(G.-K.-S. [1995]).*

For linear decision trees $size_1(MAX =) \leq 2 \cdot n$;

Alternatively to usual (binary) trees one can consider **ternary** decision trees in which a node v with a testing function g_v branches into 3 nodes according to whether $g_v < 0$, or $g_v = 0$, or $g_v > 0$.

Theorem

Ternary $size_d(MAX =) \geq 2^{c_d \cdot n}$ *for d-decision trees,* $c_d > 0$ *(G.-Karpinski-Yao [1994])*

Open question: size (MAX) for binary decision/computation trees?

Size of decision trees

So far, we studied the depth of decision/computation trees. Since *depth* $\geq \log(size)$, the bounds on the size are stronger.

Consider set

$\mathrm{EXACT}_n = \{(x_1, \ldots, x_{2\cdot n}) \in \mathbb{R}^{2\cdot n} : \prod_{1\leq i\leq 2\cdot n} x_i \neq 0, \sharp\{i : x_i < 0\} = n\}$.

Theorem

For analytic decision trees $size_{an}(EXACT_n) \geq 2^n/n$ *(G.-K.-S. [1995]).*

For linear decision trees $size_1(MAX =) \leq 2 \cdot n$;

Alternatively to usual (binary) trees one can consider **ternary** decision trees in which a node v with a testing function g_v branches into 3 nodes according to whether $g_v < 0$, or $g_v = 0$, or $g_v > 0$.

Theorem

Ternary $size_d(MAX =) \geq 2^{c_d \cdot n}$ *for d-decision trees,* $c_d > 0$ *(G.-Karpinski-Yao [1994])*

Open question: size (MAX) for binary decision/computation trees?

Size of decision trees

So far, we studied the depth of decision/computation trees. Since *depth* $\geq \log(size)$, the bounds on the size are stronger.
Consider set
$\text{EXACT}_n = \{(x_1, \ldots, x_{2\cdot n}) \in \mathbb{R}^{2\cdot n} : \prod_{1\leq i\leq 2\cdot n} x_i \neq 0, \sharp\{i : x_i < 0\} = n\}$.

Theorem

For analytic decision trees $size_{an}(EXACT_n) \geq 2^n/n$ *(G.-K.-S. [1995]).*

For linear decision trees $size_1(MAX =) \leq 2 \cdot n$;

Alternatively to usual (binary) trees one can consider **ternary** decision trees in which a node v with a testing function g_v branches into 3 nodes according to whether $g_v < 0$, or $g_v = 0$, or $g_v > 0$.

Theorem

Ternary $size_d(MAX =) \geq 2^{c_d \cdot n}$ *for d-decision trees,* $c_d > 0$ *(G.-Karpinski-Yao [1994])*

Open question: size (MAX) for binary decision/computation trees?

Size of decision trees

So far, we studied the depth of decision/computation trees. Since *depth* $\geq \log(size)$, the bounds on the size are stronger.

Consider set

$\mathrm{EXACT}_n = \{(x_1, \ldots, x_{2\cdot n}) \in \mathbb{R}^{2\cdot n} : \prod_{1\leq i\leq 2\cdot n} x_i \neq 0, \sharp\{i : x_i < 0\} = n\}$.

Theorem

For analytic decision trees $size_{an}(EXACT_n) \geq 2^n/n$ *(G.-K.-S. [1995]).*

For linear decision trees $size_1(MAX =) \leq 2 \cdot n$;

Alternatively to usual (binary) trees one can consider **ternary** decision trees in which a node v with a testing function g_v branches into 3 nodes according to whether $g_v < 0$, or $g_v = 0$, or $g_v > 0$.

Theorem

Ternary $size_d(MAX =) \geq 2^{c_d \cdot n}$ *for* d*-decision trees,* $c_d > 0$ *(G.-Karpinski-Yao [1994])*

Open question: size (MAX) for binary decision/computation trees?

Size of decision trees

So far, we studied the depth of decision/computation trees. Since *depth* $\geq \log(size)$, the bounds on the size are stronger.

Consider set

$\text{EXACT}_n = \{(x_1, \ldots, x_{2\cdot n}) \in \mathbb{R}^{2\cdot n} : \prod_{1\leq i\leq 2\cdot n} x_i \neq 0, \sharp\{i : x_i < 0\} = n\}$.

Theorem

For analytic decision trees $size_{an}(EXACT_n) \geq 2^n/n$ *(G.-K.-S. [1995]).*

For linear decision trees $size_1(MAX =) \leq 2 \cdot n$;

Alternatively to usual (binary) trees one can consider **ternary** decision trees in which a node v with a testing function g_v branches into 3 nodes according to whether $g_v < 0$, or $g_v = 0$, or $g_v > 0$.

Theorem

Ternary $size_d(MAX =) \geq 2^{c_d \cdot n}$ *for* d*-decision trees,* $c_d > 0$ *(G.-Karpinski-Yao [1994])*

Open question: size (MAX) for binary decision/computation trees?

Communication complexity of decision trees

Communication tree

Let $F = \mathbb{R}$ (also $F = \mathbb{C}$ can be studied). Let input variables be divided into 2 groups: $X_1, \dots, X_{n_1}$ and $Y_1, \dots, Y_{n_2}$. An input $(x, y) \in \mathbb{R}^{n_1+n_2}$.

To each node v with an even (respectively, odd) depth a **calculating polynomial** $a_v \in \mathbb{R}[X_1, \dots, X_{n_1}]$ (respectively, $b_v \in \mathbb{R}[Y_1, \dots, Y_{n_2}]$) is attached.

Also a family of **testing polynomials** $\{g_{vi}\}_{1 \le i \le N_v}$ is assigned.

There are 2 players; the first one has access to x and the second to y.

Let $v_0, \dots, v_k = v$ be the nodes on the path from the root v_0 to $v_k = v$ and k be odd (for definiteness).

Then at v the second player computes $b_v(y)$, transmits it to the first player and branches according to the vector $\{\mathrm{sgn}(g_{vi}(a_{v_0}(x), b_{v_1}(y), a_{v_2}(x), \dots, a_{v_{k-1}}(x), b_{v_k}(y)\}_{1 \le i \le N_v}$

Communication complexity CC is the depth of the communication tree.

One defines also **probabilistic communication complexity** CC^{prob}.

Communication complexity of decision trees

Communication tree

Let $F = \mathbb{R}$ (also $F = \mathbb{C}$ can be studied). Let input variables be divided into 2 groups: $X_1, \dots, X_{n_1}$ and $Y_1, \dots, Y_{n_2}$. An input $(x, y) \in \mathbb{R}^{n_1+n_2}$.

To each node v with an even (respectively, odd) depth a **calculating polynomial** $a_v \in \mathbb{R}[X_1, \dots, X_{n_1}]$ (respectively, $b_v \in \mathbb{R}[Y_1, \dots, Y_{n_2}]$) is attached.

Also a family of **testing polynomials** $\{g_{vi}\}_{1 \le i \le N_v}$ is assigned.

There are 2 players; the first one has access to x and the second to y.

Let $v_0, \dots, v_k = v$ be the nodes on the path from the root v_0 to $v_k = v$ and k be odd (for definiteness).

Then at v the second player computes $b_v(y)$, transmits it to the first player and branches according to the vector $\{\mathrm{sgn}(g_{vi}(a_{v_0}(x), b_{v_1}(y), a_{v_2}(x), \dots, a_{v_{k-1}}(x), b_{v_k}(y)\}_{1 \le i \le N_v}$

Communication complexity CC is the depth of the communication tree.

One defines also **probabilistic communication complexity** CC^{prob}.

Communication complexity of decision trees

Communication tree

Let $F = \mathbb{R}$ (also $F = \mathbb{C}$ can be studied). Let input variables be divided into 2 groups: $X_1, \ldots, X_{n_1}$ and $Y_1, \ldots, Y_{n_2}$. An input $(x, y) \in \mathbb{R}^{n_1+n_2}$.
To each node v with an even (respectively, odd) depth a **calculating polynomial** $a_v \in \mathbb{R}[X_1, \ldots, X_{n_1}]$ (respectively, $b_v \in \mathbb{R}[Y_1, \ldots, Y_{n_2}]$) is attached.
Also a family of **testing polynomials** $\{g_{vi}\}_{1 \le i \le N_v}$ is assigned.
There are 2 players; the first one has access to x and the second to y.
Let $v_0, \ldots, v_k = v$ be the nodes on the path from the root v_0 to $v_k = v$ and k be odd (for definiteness).
Then at v the second player computes $b_v(y)$, transmits it to the first player and branches according to the vector
$\{\mathrm{sgn}(g_{vi}(a_{v_0}(x), b_{v_1}(y), a_{v_2}(x), \ldots, a_{v_{k-1}}(x), b_{v_k}(y)\}_{1 \le i \le N_v}$
Communication complexity CC is the depth of the communication tree.

One defines also **probabilistic communication complexity** CC^{prob}.

Communication complexity of decision trees

Communication tree

Let $F = \mathbb{R}$ (also $F = \mathbb{C}$ can be studied). Let input variables be divided into 2 groups: $X_1, \ldots, X_{n_1}$ and $Y_1, \ldots, Y_{n_2}$. An input $(x, y) \in \mathbb{R}^{n_1+n_2}$.
To each node v with an even (respectively, odd) depth a **calculating polynomial** $a_v \in \mathbb{R}[X_1, \ldots, X_{n_1}]$ (respectively, $b_v \in \mathbb{R}[Y_1, \ldots, Y_{n_2}]$) is attached.
Also a family of **testing polynomials** $\{g_{vi}\}_{1 \leq i \leq N_v}$ is assigned.
There are 2 players; the first one has access to x and the second to y.
Let $v_0, \ldots, v_k = v$ be the nodes on the path from the root v_0 to $v_k = v$ and k be odd (for definiteness).
Then at v the second player computes $b_v(y)$, transmits it to the first player and branches according to the vector
$\{\mathrm{sgn}(g_{vi}(a_{v_0}(x), b_{v_1}(y), a_{v_2}(x), \ldots, a_{v_{k-1}}(x), b_{v_k}(y)\}_{1 \leq i \leq N_v}$
Communication complexity CC is the depth of the communication tree.

One defines also **probabilistic communication complexity** CC^{prob}.

Communication complexity of decision trees

Communication tree

Let $F = \mathbb{R}$ (also $F = \mathbb{C}$ can be studied). Let input variables be divided into 2 groups: $X_1, \dots, X_{n_1}$ and $Y_1, \dots, Y_{n_2}$. An input $(x, y) \in \mathbb{R}^{n_1+n_2}$.
To each node v with an even (respectively, odd) depth a **calculating polynomial** $a_v \in \mathbb{R}[X_1, \dots, X_{n_1}]$ (respectively, $b_v \in \mathbb{R}[Y_1, \dots, Y_{n_2}]$) is attached.
Also a family of **testing polynomials** $\{g_{vi}\}_{1 \le i \le N_v}$ is assigned.
There are 2 players; the first one has access to x and the second to y.
Let $v_0, \dots, v_k = v$ be the nodes on the path from the root v_0 to $v_k = v$ and k be odd (for definiteness).
Then at v the second player computes $b_v(y)$, transmits it to the first player and branches according to the vector
$\{\mathrm{sgn}(g_{vi}(a_{v_0}(x), b_{v_1}(y), a_{v_2}(x), \dots, a_{v_{k-1}}(x), b_{v_k}(y)\}_{1 \le i \le N_v}$
Communication complexity CC is the depth of the communication tree.

One defines also **probabilistic communication complexity** CC^{prob}.

Communication complexity of decision trees

Communication tree

Let $F = \mathbb{R}$ (also $F = \mathbb{C}$ can be studied). Let input variables be divided into 2 groups: $X_1, \dots, X_{n_1}$ and $Y_1, \dots, Y_{n_2}$. An input $(x, y) \in \mathbb{R}^{n_1+n_2}$.
To each node v with an even (respectively, odd) depth a **calculating polynomial** $a_v \in \mathbb{R}[X_1, \dots, X_{n_1}]$ (respectively, $b_v \in \mathbb{R}[Y_1, \dots, Y_{n_2}]$) is attached.
Also a family of **testing polynomials** $\{g_{vi}\}_{1 \le i \le N_v}$ is assigned.
There are 2 players; the first one has access to x and the second to y.
Let $v_0, \dots, v_k = v$ be the nodes on the path from the root v_0 to $v_k = v$ and k be odd (for definiteness).
Then at v the second player computes $b_v(y)$, transmits it to the first player and branches according to the vector
$\{\mathrm{sgn}(g_{vi}(a_{v_0}(x), b_{v_1}(y), a_{v_2}(x), \dots, a_{v_{k-1}}(x), b_{v_k}(y)\}_{1 \le i \le N_v}$
Communication complexity CC is the depth of the communication tree.

One defines also **probabilistic communication complexity** CC^{prob}.

Communication complexity of decision trees

Communication tree

Let $F = \mathbb{R}$ (also $F = \mathbb{C}$ can be studied). Let input variables be divided into 2 groups: $X_1, \dots, X_{n_1}$ and $Y_1, \dots, Y_{n_2}$. An input $(x, y) \in \mathbb{R}^{n_1+n_2}$.
To each node v with an even (respectively, odd) depth a **calculating polynomial** $a_v \in \mathbb{R}[X_1, \dots, X_{n_1}]$ (respectively, $b_v \in \mathbb{R}[Y_1, \dots, Y_{n_2}]$) is attached.
Also a family of **testing polynomials** $\{g_{vi}\}_{1 \le i \le N_v}$ is assigned.
There are 2 players; the first one has access to x and the second to y.
Let $v_0, \dots, v_k = v$ be the nodes on the path from the root v_0 to $v_k = v$ and k be odd (for definiteness).
Then at v the second player computes $b_v(y)$, transmits it to the first player and branches according to the vector
$\{\mathrm{sgn}(g_{vi}(a_{v_0}(x), b_{v_1}(y), a_{v_2}(x), \dots, a_{v_{k-1}}(x), b_{v_k}(y)\}_{1 \le i \le N_v}$
Communication complexity CC is the depth of the communication tree.

One defines also **probabilistic communication complexity** CC^{prob}.

Lower bound on communication complexity

$\mathbb{R}_{>}^{n_1+n_2} = \{(x,y) \in \mathbb{R}^{n_1+n_2} : \forall i,j\, x_i > 0,\ y_j > 0\} \subset \mathbb{R}_{+}^{n_1+n_2} = \overline{\mathbb{R}_{>}^{n_1+n_2}}$.

Proposition

- $CC(\mathbb{R}_{>}^{n_1+n_2}),\ CC(\mathbb{R}_{+}^{n_1+n_2}) = n_1 + n_2$;
- $CC^{prob}(\mathbb{R}_{>}^{n_1+n_2}) \le 4, \quad CC^{prob}(\mathbb{R}_{+}^{n_1+n_2}) \le \log^{O(1)}(n_1+n_2)$.

Lower bound on probabilistic communication complexity

Polyhedron $S = \{(x,y) \in \mathbb{R}^{2\cdot n} : \forall i\, x_i + y_i > 0\}$.
Arrangement $Q = \bigcup_{1 \le i \le n} \{X_i + Y_i = 0\} \subset \mathbb{R}^{2\cdot n}$.

Theorem

$CC^{prob}(S)$, $CC^{prob}(Q)$, $CC^{prob}(EQUALITY)$, $CC^{prob}(KNAPSACK) \ge n$ (G. [2006]).

Problem; to obtain a lower bound for communication trees with branching condition $\{\mathrm{sgn}(g_{vi}(a_{v_0}(x), a_{v_2}(x), \ldots, a_{v_{k-1}}(x), y))\}_{1 \le i \le N}$

Lower bound on communication complexity

$\mathbb{R}_>^{n_1+n_2} = \{(x,y) \in \mathbb{R}^{n_1+n_2} : \forall i,j\, x_i > 0,\ y_j > 0\} \subset \mathbb{R}_+^{n_1+n_2} = \overline{\mathbb{R}_>^{n_1+n_2}}$.

Proposition

- $CC(\mathbb{R}_>^{n_1+n_2}),\ CC(\mathbb{R}_+^{n_1+n_2}) = n_1 + n_2;$
- $CC^{prob}(\mathbb{R}_>^{n_1+n_2}) \le 4, \quad CC^{prob}(\mathbb{R}_+^{n_1+n_2}) \le \log^{O(1)}(n_1+n_2).$

Lower bound on probabilistic communication complexity

Polyhedron $S = \{(x,y) \in \mathbb{R}^{2\cdot n} : \forall i\, x_i + y_i > 0\}$.
Arrangement $Q = \bigcup_{1\le i\le n}\{X_i + Y_i = 0\} \subset \mathbb{R}^{2\cdot n}$.

Theorem

$CC^{prob}(S),\ CC^{prob}(Q),\ CC^{prob}(EQUALITY),\ CC^{prob}(KNAPSACK) \ge n$ (G. [2006]).

Problem; to obtain a lower bound for communication trees with branching condition $\{\mathrm{sgn}(g_{vi}(a_{v_0}(x), a_{v_2}(x), \ldots, a_{v_{k-1}}(x), y))\}_{1\le i\le N}$

Lower bound on communication complexity

$\mathbb{R}_{>}^{n_1+n_2} = \{(x, y) \in \mathbb{R}^{n_1+n_2} : \forall i, j\, x_i > 0,\ y_j > 0\} \subset \mathbb{R}_{+}^{n_1+n_2} = \overline{\mathbb{R}_{>}^{n_1+n_2}}$.

Proposition

- $CC(\mathbb{R}_{>}^{n_1+n_2}),\ CC(\mathbb{R}_{+}^{n_1+n_2}) = n_1 + n_2;$
- $CC^{prob}(\mathbb{R}_{>}^{n_1+n_2}) \leq 4, \quad CC^{prob}(\mathbb{R}_{+}^{n_1+n_2}) \leq \log^{O(1)}(n_1 + n_2).$

Lower bound on probabilistic communication complexity

Polyhedron $S = \{(x, y) \in \mathbb{R}^{2 \cdot n} : \forall i\, x_i + y_i > 0\}$.
Arrangement $Q = \bigcup_{1 \leq i \leq n} \{X_i + Y_i = 0\} \subset \mathbb{R}^{2 \cdot n}$.

Theorem

$CC^{prob}(S)$, $CC^{prob}(Q)$, $CC^{prob}(EQUALITY)$, $CC^{prob}(KNAPSACK) \geq n$ (G. [2006]).

Problem; to obtain a lower bound for communication trees with branching condition $\{\mathrm{sgn}(g_{vi}(a_{v_0}(x), a_{v_2}(x), \ldots, a_{v_{k-1}}(x), y))\}_{1 \leq i \leq N}$

Lower bound on communication complexity

$\mathbb{R}_>^{n_1+n_2} = \{(x,y) \in \mathbb{R}^{n_1+n_2} : \forall i,j\, x_i > 0,\ y_j > 0\} \subset \mathbb{R}_+^{n_1+n_2} = \overline{\mathbb{R}_>^{n_1+n_2}}$.

Proposition

- $CC(\mathbb{R}_>^{n_1+n_2}),\ CC(\mathbb{R}_+^{n_1+n_2}) = n_1 + n_2;$
- $CC^{prob}(\mathbb{R}_>^{n_1+n_2}) \le 4, \quad CC^{prob}(\mathbb{R}_+^{n_1+n_2}) \le \log^{O(1)}(n_1+n_2).$

Lower bound on probabilistic communication complexity

Polyhedron $S = \{(x,y) \in \mathbb{R}^{2\cdot n} : \forall i\, x_i + y_i > 0\}$.

Arrangement $Q = \bigcup_{1\le i\le n}\{X_i + Y_i = 0\} \subset \mathbb{R}^{2\cdot n}$.

Theorem

$CC^{prob}(S)$, $CC^{prob}(Q)$, $CC^{prob}(EQUALITY)$, $CC^{prob}(KNAPSACK) \ge n$ (G. [2006]).

Problem; to obtain a lower bound for communication trees with branching condition $\{\mathrm{sgn}(g_{vi}(a_{v_0}(x), a_{v_2}(x), \dots, a_{v_k}(x), y))\}_{1\le i\le N}$

Lower bound on communication complexity

$\mathbb{R}_{>}^{n_1+n_2} = \{(x,y) \in \mathbb{R}^{n_1+n_2} : \forall i,j\, x_i > 0,\ y_j > 0\} \subset \mathbb{R}_{+}^{n_1+n_2} = \overline{\mathbb{R}_{>}^{n_1+n_2}}$.

Proposition

- $CC(\mathbb{R}_{>}^{n_1+n_2}),\ CC(\mathbb{R}_{+}^{n_1+n_2}) = n_1 + n_2$;
- $CC^{prob}(\mathbb{R}_{>}^{n_1+n_2}) \leq 4, \quad CC^{prob}(\mathbb{R}_{+}^{n_1+n_2}) \leq \log^{O(1)}(n_1+n_2)$.

Lower bound on probabilistic communication complexity

Polyhedron $S = \{(x,y) \in \mathbb{R}^{2\cdot n} : \forall i\, x_i + y_i > 0\}$.
Arrangement $Q = \bigcup_{1\leq i\leq n}\{X_i + Y_i = 0\} \subset \mathbb{R}^{2\cdot n}$.

Theorem

$CC^{prob}(S), CC^{prob}(Q), CC^{prob}(EQUALITY), CC^{prob}(KNAPSACK) \geq n$ (G. [2006]).

Problem; to obtain a lower bound for communication trees with branching condition $\{\mathrm{sgn}(g_{vi}(a_{v_0}(x), a_{v_2}(x), \ldots, a_{v_{k-1}}(x), y))\}_{1\leq i\leq N}$

Lower bound on communication complexity

$\mathbb{R}_{>}^{n_1+n_2} = \{(x,y) \in \mathbb{R}^{n_1+n_2} : \forall i,j\, x_i > 0,\ y_j > 0\} \subset \mathbb{R}_{+}^{n_1+n_2} = \overline{\mathbb{R}_{>}^{n_1+n_2}}$.

Proposition

- $CC(\mathbb{R}_{>}^{n_1+n_2}),\ CC(\mathbb{R}_{+}^{n_1+n_2}) = n_1 + n_2;$
- $CC^{prob}(\mathbb{R}_{>}^{n_1+n_2}) \leq 4, \quad CC^{prob}(\mathbb{R}_{+}^{n_1+n_2}) \leq \log^{O(1)}(n_1+n_2).$

Lower bound on probabilistic communication complexity

Polyhedron $S = \{(x,y) \in \mathbb{R}^{2 \cdot n} : \forall i\, x_i + y_i > 0\}$.
Arrangement $Q = \bigcup_{1 \leq i \leq n} \{X_i + Y_i = 0\} \subset \mathbb{R}^{2 \cdot n}$.

Theorem

$CC^{prob}(S),\ CC^{prob}(Q),\ CC^{prob}(EQUALITY),\ CC^{prob}(KNAPSACK) \geq n$ *(G. [2006]).*

Problem; to obtain a lower bound for communication trees with branching condition $\{\mathrm{sgn}(g_{vi}(a_{v_0}(x), a_{v_2}(x), \ldots, a_{v_{k-1}}(x), y))\}_{1 \leq i \leq N}$

Lower bound on communication complexity

$\mathbb{R}_{>}^{n_1+n_2} = \{(x,y) \in \mathbb{R}^{n_1+n_2} : \forall i,j\, x_i > 0,\ y_j > 0\} \subset \mathbb{R}_{+}^{n_1+n_2} = \overline{\mathbb{R}_{>}^{n_1+n_2}}$.

Proposition

- $CC(\mathbb{R}_{>}^{n_1+n_2}),\ CC(\mathbb{R}_{+}^{n_1+n_2}) = n_1 + n_2$;
- $CC^{prob}(\mathbb{R}_{>}^{n_1+n_2}) \leq 4, \quad CC^{prob}(\mathbb{R}_{+}^{n_1+n_2}) \leq \log^{O(1)}(n_1+n_2)$.

Lower bound on probabilistic communication complexity

Polyhedron $S = \{(x,y) \in \mathbb{R}^{2\cdot n} : \forall i\, x_i + y_i > 0\}$.
Arrangement $Q = \bigcup_{1 \leq i \leq n} \{X_i + Y_i = 0\} \subset \mathbb{R}^{2\cdot n}$.

Theorem

$CC^{prob}(S),\ CC^{prob}(Q),\ CC^{prob}(EQUALITY),\ CC^{prob}(KNAPSACK) \geq n$ (G. [2006]).

Problem; to obtain a lower bound for communication trees with branching condition $\{\mathrm{sgn}(g_{vi}(a_{v_0}(x), a_{v_2}(x), \dots, a_{v_{k-1}}(x), y))\}_{1 \leq i \leq N_v}$

Parallel networks

To the root of a **network** input $x \in \mathbb{R}^n$ is attributed.

To each node v of depth k lead edges from (at most) two nodes v_1, v_2 of depth $k-1$.

To each node v a **computing polynomial** $f_v \in \mathbb{R}[X_1, \dots, x_n]$ and a **boolean indicator** b_v are attached.

Informally, $b_v = 1$ ("*active*") means that one of the processors is located in v. We impose the condition that the number of "*active*" v of depth k is bounded by $p \le 2^k$.

$f_v = A \circ B$ where $\circ \in \{+, \times\}$ and $A, B \in F \cup \{X_1, \dots, X_n\} \cup \{f_{v_1}, f_{v_2}\}$

To v two boolean functions B_{v1}, B_{v2} are also attached.

- if both v_1, v_2 are "*passive*" then v is "*passive*" as well;
- if v_1, v_2 are "*active*" then $b_v = B_{v2}(\mathrm{sgn}(f_{v_1}(x)), \mathrm{sgn}(f_{v_2}(x)))$;
- if only v_i, $i = 1, 2$ is "*active*" then $b_v = B_{v1}(\mathrm{sgn}(f_{v_i}(x)))$.

Exactly one node w with the largest depth is "*active*", $\mathrm{sgn}(f_w(x))$ is treated as an output of the parallel network.

The **parallel complexity** PC is the depth of the network.

Parallel networks

To the root of a **network** input $x \in \mathbb{R}^n$ is attributed.

To each node v of depth k lead edges from (at most) two nodes v_1, v_2 of depth $k-1$.

To each node v a **computing polynomial** $f_v \in \mathbb{R}[X_1, \dots, x_n]$ and a **boolean indicator** b_v are attached.

Informally, $b_v = 1$ ("*active*") means that one of the processors is located in v. We impose the condition that the number of "*active*" v of depth k is bounded by $p \leq 2^k$.

$f_v = A \circ B$ where $\circ \in \{+, \times\}$ and $A, B \in F \cup \{X_1, \dots, X_n\} \cup \{f_{v_1}, f_{v_2}\}$

To v two boolean functions B_{v1}, B_{v2} are also attached.

- if both v_1, v_2 are "*passive*" then v is "*passive*" as well;
- if v_1, v_2 are "*active*" then $b_v = B_{v2}(\mathrm{sgn}(f_{v_1}(x)), \mathrm{sgn}(f_{v_2}(x)))$;
- if only v_i, $i = 1, 2$ is "*active*" then $b_v = B_{v1}(\mathrm{sgn}(f_{v_i}(x)))$.

Exactly one node w with the largest depth is "*active*", $\mathrm{sgn}(f_w(x))$ is treated as an output of the parallel network.

The **parallel complexity** *PC* is the depth of the network.

Parallel networks

To the root of a **network** input $x \in \mathbb{R}^n$ is attributed.
To each node v of depth k lead edges from (at most) two nodes v_1, v_2 of depth $k-1$.
To each node v a **computing polynomial** $f_v \in \mathbb{R}[X_1, \ldots, x_n]$ and a **boolean indicator** b_v are attached.

Informally, $b_v = 1$ ("*active*") means that one of the processors is located in v. We impose the condition that the number of "*active*" v of depth k is bounded by $p \leq 2^k$.

$f_v = A \circ B$ where $\circ \in \{+, \times\}$ and $A, B \in F \cup \{X_1, \ldots, X_n\} \cup \{f_{v_1}, f_{v_2}\}$

To v two boolean functions B_{v1}, B_{v2} are also attached.

- if both v_1, v_2 are "*passive*" then v is "*passive*" as well;
- if v_1, v_2 are "*active*" then $b_v = B_{v2}(\mathrm{sgn}(f_{v_1}(x)), \mathrm{sgn}(f_{v_2}(x)))$;
- if only v_i, $i = 1, 2$ is "*active*" then $b_v = B_{v1}(\mathrm{sgn}(f_{v_i}(x)))$.

Exactly one node w with the largest depth is "*active*", $\mathrm{sgn}(f_w(x))$ is treated as an output of the parallel network.

The **parallel complexity** PC is the depth of the network.

Parallel networks

To the root of a **network** input $x \in \mathbb{R}^n$ is attributed.
To each node v of depth k lead edges from (at most) two nodes v_1, v_2 of depth $k-1$.
To each node v a **computing polynomial** $f_v \in \mathbb{R}[X_1, \dots, x_n]$ and a **boolean indicator** b_v are attached.

Informally, $b_v = 1$ ("*active*") means that one of the processors is located in v. We impose the condition that the number of "*active*" v of depth k is bounded by $p \leq 2^k$.

$f_v = A \circ B$ where $\circ \in \{+, \times\}$ and $A, B \in F \cup \{X_1, \dots, X_n\} \cup \{f_{v_1}, f_{v_2}\}$

To v two boolean functions B_{v1}, B_{v2} are also attached.

- if both v_1, v_2 are "*passive*" then v is "*passive*" as well;
- if v_1, v_2 are "*active*" then $b_v = B_{v2}(\mathrm{sgn}(f_{v_1}(x)), \mathrm{sgn}(f_{v_2}(x)))$;
- if only v_i, $i = 1, 2$ is "*active*" then $b_v = B_{v1}(\mathrm{sgn}(f_{v_i}(x)))$.

Exactly one node w with the largest depth is "*active*", $\mathrm{sgn}(f_w(x))$ is treated as an output of the parallel network.

The **parallel complexity** *PC* is the depth of the network.

Parallel networks

To the root of a **network** input $x \in \mathbb{R}^n$ is attributed.
To each node v of depth k lead edges from (at most) two nodes v_1, v_2 of depth $k-1$.
To each node v a **computing polynomial** $f_v \in \mathbb{R}[X_1, \dots, x_n]$ and a **boolean indicator** b_v are attached.

Informally, $b_v = 1$ (”*active*”) means that one of the processors is located in v. We impose the condition that the number of ”*active*” v of depth k is bounded by $p \le 2^k$.

$f_v = A \circ B$ where $\circ \in \{+, \times\}$ and $A, B \in F \cup \{X_1, \dots, X_n\} \cup \{f_{v_1}, f_{v_2}\}$

To v two boolean functions B_{v1}, B_{v2} are also attached.

- if both v_1, v_2 are ”*passive*” then v is ”*passive*” as well;
- if v_1, v_2 are ”*active*” then $b_v = B_{v2}(\mathrm{sgn}(f_{v_1}(x)), \mathrm{sgn}(f_{v_2}(x)))$;
- if only v_i, $i = 1, 2$ is ”*active*” then $b_v = B_{v1}(\mathrm{sgn}(f_{v_i}(x)))$.

Exactly one node w with the largest depth is ”*active*”, $\mathrm{sgn}(f_w(x))$ is treated as an output of the parallel network.

The **parallel complexity** *PC* is the depth of the network.

Parallel networks

To the root of a **network** input $x \in \mathbb{R}^n$ is attributed.
To each node v of depth k lead edges from (at most) two nodes v_1, v_2 of depth $k-1$.
To each node v a **computing polynomial** $f_v \in \mathbb{R}[X_1, \dots, x_n]$ and a **boolean indicator** b_v are attached.

Informally, $b_v = 1$ ("*active*") means that one of the processors is located in v. We impose the condition that the number of "*active*" v of depth k is bounded by $p \leq 2^k$.

$f_v = A \circ B$ where $\circ \in \{+, \times\}$ and $A, B \in F \cup \{X_1, \dots, X_n\} \cup \{f_{v_1}, f_{v_2}\}$

To v two boolean functions B_{v1}, B_{v2} are also attached.

- if both v_1, v_2 are "*passive*" then v is "*passive*" as well;
- if v_1, v_2 are "*active*" then $b_v = B_{v2}(\mathrm{sgn}(f_{v_1}(x)), \mathrm{sgn}(f_{v_2}(x)))$;
- if only v_i, $i = 1, 2$ is "*active*" then $b_v = B_{v1}(\mathrm{sgn}(f_{v_i}(x)))$.

Exactly one node w with the largest depth is "*active*", $\mathrm{sgn}(f_w(x))$ is treated as an output of the parallel network.

The **parallel complexity** *PC* is the depth of the network.

Parallel networks

To the root of a **network** input $x \in \mathbb{R}^n$ is attributed.
To each node v of depth k lead edges from (at most) two nodes v_1, v_2 of depth $k-1$.
To each node v a **computing polynomial** $f_v \in \mathbb{R}[X_1, \dots, x_n]$ and a **boolean indicator** b_v are attached.

Informally, $b_v = 1$ ("*active*") means that one of the processors is located in v. We impose the condition that the number of "*active*" v of depth k is bounded by $p \leq 2^k$.

$f_v = A \circ B$ where $\circ \in \{+, \times\}$ and $A, B \in F \cup \{X_1, \dots, X_n\} \cup \{f_{v_1}, f_{v_2}\}$

To v two boolean functions B_{v1}, B_{v2} are also attached.

- if both v_1, v_2 are "*passive*" then v is "*passive*" as well;
- if v_1, v_2 are "*active*" then $b_v = B_{v2}(\mathrm{sgn}(f_{v_1}(x)), \mathrm{sgn}(f_{v_2}(x)))$;
- if only $v_i,\ i = 1, 2$ is "*active*" then $b_v = B_{v1}(\mathrm{sgn}(f_{v_i}(x)))$.

Exactly one node w with the largest depth is "*active*", $\mathrm{sgn}(f_w(x))$ is treated as an output of the parallel network.

The **parallel complexity** *PC* is the depth of the network.

Parallel networks

To the root of a **network** input $x \in \mathbb{R}^n$ is attributed.
To each node v of depth k lead edges from (at most) two nodes v_1, v_2 of depth $k - 1$.
To each node v a **computing polynomial** $f_v \in \mathbb{R}[X_1, \dots, x_n]$ and a **boolean indicator** b_v are attached.

Informally, $b_v = 1$ ("*active*") means that one of the processors is located in v. We impose the condition that the number of "*active*" v of depth k is bounded by $p \leq 2^k$.

$f_v = A \circ B$ where $\circ \in \{+, \times\}$ and $A, B \in F \cup \{X_1, \dots, X_n\} \cup \{f_{v_1}, f_{v_2}\}$

To v two boolean functions B_{v1}, B_{v2} are also attached.

- if both v_1, v_2 are "*passive*" then v is "*passive*" as well;
- if v_1, v_2 are "*active*" then $b_v = B_{v2}(\mathrm{sgn}(f_{v_1}(x)), \mathrm{sgn}(f_{v_2}(x)))$;
- if only v_i, $i = 1, 2$ is "*active*" then $b_v = B_{v1}(\mathrm{sgn}(f_{v_i}(x)))$.

Exactly one node w with the largest depth is "*active*", $\mathrm{sgn}(f_w(x))$ is treated as an output of the parallel network.

The **parallel complexity** *PC* is the depth of the network.

Informally: at depth k the network can contain $\leq 2^{2^k}$ nodes with potential possibility of their use in computation, while for every particular input x just $\leq 2^k$ are involved (the processors are located in them).

Which nodes to use in a computation for x is determined by the boolean indicators.

Lower bound on the parallel complexity

For set $S \subset \mathbb{R}^n$ parallel complexity $PC(S) \geq \Omega(\sqrt{(\log N)/n})$ where N is

- either the sum of Betti numbers of S (Mulmuley [1994], Montaña-Morais-Pardo [1996].)
- or the number of faces of S of all the dimensions when S is a polyhedron, provided that $N \geq n^{\Omega(n)}$ (G. [1996])

Informally: at depth k the network can contain $\leq 2^{2^k}$ nodes with potential possibility of their use in computation, while for every particular input x just $\leq 2^k$ are involved (the processors are located in them).
Which nodes to use in a computation for x is determined by the boolean indicators.

Lower bound on the parallel complexity

For set $S \subset \mathbb{R}^n$ parallel complexity $PC(S) \geq \Omega(\sqrt{(\log N)/n})$ where N is

- either the sum of Betti numbers of S (Mulmuley [1994], Montaña-Morais-Pardo [1996].)
- or the number of faces of S of all the dimensions when S is a polyhedron, provided that $N \geq n^{\Omega(n)}$ (G. [1996])

Informally: at depth k the network can contain $\leq 2^{2^k}$ nodes with potential possibility of their use in computation, while for every particular input x just $\leq 2^k$ are involved (the processors are located in them).
Which nodes to use in a computation for x is determined by the boolean indicators.

Lower bound on the parallel complexity

For set $S \subset \mathbb{R}^n$ parallel complexity $PC(S) \geq \Omega(\sqrt{(\log N)/n})$ where N is

- either the sum of Betti numbers of S (Mulmuley [1994], Montaña-Morais-Pardo [1996].)
- or the number of faces of S of all the dimensions when S is a polyhedron, provided that $N \geq n^{\Omega(n)}$ (G. [1996])

Informally: at depth k the network can contain $\leq 2^{2^k}$ nodes with potential possibility of their use in computation, while for every particular input x just $\leq 2^k$ are involved (the processors are located in them).
Which nodes to use in a computation for x is determined by the boolean indicators.

Lower bound on the parallel complexity

For set $S \subset \mathbb{R}^n$ parallel complexity $PC(S) \geq \Omega(\sqrt{(\log N)/n})$ where N is

- either the sum of Betti numbers of S (Mulmuley [1994], Montaña-Morais-Pardo [1996].)
- or the number of faces of S of all the dimensions when S is a polyhedron, provided that $N \geq n^{\Omega(n)}$ (G. [1996])

Informally: at depth k the network can contain $\leq 2^{2^k}$ nodes with potential possibility of their use in computation, while for every particular input x just $\leq 2^k$ are involved (the processors are located in them).
Which nodes to use in a computation for x is determined by the boolean indicators.

Lower bound on the parallel complexity

For set $S \subset \mathbb{R}^n$ parallel complexity $PC(S) \geq \Omega(\sqrt{(\log N)/n})$ where N is

- either the sum of Betti numbers of S (Mulmuley [1994], Montaña-Morais-Pardo [1996].)
- or the number of faces of S of all the dimensions when S is a polyhedron, provided that $N \geq n^{\Omega(n)}$ (G. [1996])

Let a **linear complex** $S \subset \mathbb{R}^n$ be given as a boolean combination of linear inequalities $L_j \geq 0$, $1 \leq j \leq m$.

Upper bound on the parallel complexity

$p, CP(S) \leq O(\sqrt{\log m \cdot \log\log m} \cdot 2^n)$ (G. [1996]).

Thus, the bound is interesting for small n. The following corollary provides a nearly (sharp) quadratic gap between the parallel complexity and the usual (sequential) one.

Corollary

Let $S \subset \mathbb{R}^2$ be an m-gon. Then

- $C(S) \asymp \log m$ *(Steele-Yao [1982], Meyer auf der Heide [1985]);*
- $\Omega(\sqrt{\log m}) \leq PC(S) \leq O(\sqrt{\log m \cdot \log\log m})$ *(Mulmuley [1994], G. [1996]).*

Open problem: obtain an upper bound on the parallel complexity for more general semi-algebraic sets (rather than for linear complexes).

Let a **linear complex** $S \subset \mathbb{R}^n$ be given as a boolean combination of linear inequalities $L_j \geq 0$, $1 \leq j \leq m$.

Upper bound on the parallel complexity

$p, CP(S) \leq O(\sqrt{\log m \cdot \log\log m} \cdot 2^n)$ (G. [1996]).

Thus, the bound is interesting for small n. The following corollary provides a nearly (sharp) quadratic gap between the parallel complexity and the usual (sequential) one.

Corollary

Let $S \subset \mathbb{R}^2$ be an m-gon. Then

- $C(S) \asymp \log m$ *(Steele-Yao [1982], Meyer auf der Heide [1985]);*
- $\Omega(\sqrt{\log m}) \leq PC(S) \leq O(\sqrt{\log m \cdot \log\log m})$ *(Mulmuley [1994], G. [1996]).*

Open problem: obtain an upper bound on the parallel complexity for more general semi-algebraic sets (rather than for linear complexes).

Let a **linear complex** $S \subset \mathbb{R}^n$ be given as a boolean combination of linear inequalities $L_j \geq 0$, $1 \leq j \leq m$.

Upper bound on the parallel complexity

$p, CP(S) \leq O(\sqrt{\log m \cdot \log\log m} \cdot 2^n)$ (G. [1996]).

Thus, the bound is interesting for small n. The following corollary provides a nearly (sharp) quadratic gap between the parallel complexity and the usual (sequential) one.

Corollary

Let $S \subset \mathbb{R}^2$ be an m-gon. Then

- $C(S) \asymp \log m$ *(Steele-Yao [1982], Meyer auf der Heide [1985]);*
- $\Omega(\sqrt{\log m}) \leq PC(S) \leq O(\sqrt{\log m \cdot \log\log m})$ *(Mulmuley [1994], G. [1996]).*

Open problem: obtain an upper bound on the parallel complexity for more general semi-algebraic sets (rather than for linear complexes).

Let a **linear complex** $S \subset \mathbb{R}^n$ be given as a boolean combination of linear inequalities $L_j \geq 0$, $1 \leq j \leq m$.

Upper bound on the parallel complexity

$p, CP(S) \leq O(\sqrt{\log m \cdot \log\log m} \cdot 2^n)$ (G. [1996]).

Thus, the bound is interesting for small n. The following corollary provides a nearly (sharp) quadratic gap between the parallel complexity and the usual (sequential) one.

Corollary

Let $S \subset \mathbb{R}^2$ be an m-gon. Then

- $C(S) \asymp \log m$ *(Steele-Yao [1982], Meyer auf der Heide [1985]);*
- $\Omega(\sqrt{\log m}) \leq PC(S) \leq O(\sqrt{\log m \cdot \log\log m})$ *(Mulmuley [1994], G. [1996]).*

Open problem: obtain an upper bound on the parallel complexity for more general semi-algebraic sets (rather than for linear complexes).

Let a **linear complex** $S \subset \mathbb{R}^n$ be given as a boolean combination of linear inequalities $L_j \geq 0$, $1 \leq j \leq m$.

Upper bound on the parallel complexity

$p, CP(S) \leq O(\sqrt{\log m \cdot \log\log m} \cdot 2^n)$ (G. [1996]).

Thus, the bound is interesting for small n. The following corollary provides a nearly (sharp) quadratic gap between the parallel complexity and the usual (sequential) one.

Corollary

Let $S \subset \mathbb{R}^2$ be an m-gon. Then

- $C(S) \asymp \log m$ *(Steele-Yao [1982], Meyer auf der Heide [1985]);*
- $\Omega(\sqrt{\log m}) \leq PC(S) \leq O(\sqrt{\log m \cdot \log\log m})$ *(Mulmuley [1994], G. [1996]).*

Open problem: obtain an upper bound on the parallel complexity for more general semi-algebraic sets (rather than for linear complexes).

Let a **linear complex** $S \subset \mathbb{R}^n$ be given as a boolean combination of linear inequalities $L_j \geq 0$, $1 \leq j \leq m$.

Upper bound on the parallel complexity

$p, CP(S) \leq O(\sqrt{\log m \cdot \log\log m} \cdot 2^n)$ (G. [1996]).

Thus, the bound is interesting for small n. The following corollary provides a nearly (sharp) quadratic gap between the parallel complexity and the usual (sequential) one.

Corollary

Let $S \subset \mathbb{R}^2$ be an m-gon. Then

- $C(S) \asymp \log m$ *(Steele-Yao [1982], Meyer auf der Heide [1985]);*
- $\Omega(\sqrt{\log m}) \leq PC(S) \leq O(\sqrt{\log m \cdot \log\log m})$ *(Mulmuley [1994], G. [1996]).*

Open problem: obtain an upper bound on the parallel complexity for more general semi-algebraic sets (rather than for linear complexes).