Heuristic time hierarchies via hierarchies for sampling distributions

Authors:
Dmitry Itsykson, Alexander Knop, Dmitry Sokolov

Institute:
St. Petersburg Department of V.A. Steklov Institute of Mathematics of the Russian Academy of Sciences
First steps

HARTMANIS AND STEARNS, 1965

For any $k > 0$ we have that

\[
P \not\subset \text{DTime}(n^k).
\]

COOK, 1973; ZAK, 1983

For any $k > 0$ holds

\[
\text{NP} \not\subset \text{NTime}(n^k).
\]
First steps

HARTMANIS AND STEARNS, 1965

For any $k > 0$ we have that

$$P \not\subseteq \text{DTime}(n^k).$$

COOK, 1973; ZAK, 1983

For any $k > 0$ holds

$$NP \not\subseteq \text{NTime}(n^k).$$
Probabilistic algorithms

BOUNDDED PROBABILISTIC ALGORITHMS

Language \( L \in \text{BPTime}(n^k) \) iff there is randomized \( O(n^k) \)-time algorithm \( A \) such that
\[
\forall x \in \{0,1\}^* \Pr[A(x) = L(x)] > \frac{3}{4}.
\]

We also denote \( \text{BPP} = \bigcup_k \text{BPTime}(n^k) \).

OPEN QUESTION

Is it true that for any \( k > 0 \) holds that
\[
\text{BPP} \nsubseteq \text{BPTime}(n^k).
\]
Probabilistic algorithms

BOUNDED PROBABILISTIC ALGORITHMS

Language $L \in \text{BPTime}(n^k)$ iff there is randomized $O(n^k)$-time algorithm $A$ such that

$$\forall x \in \{0, 1\}^* \Pr[A(x) = L(x)] > \frac{3}{4}.$$ 

We also denote $\text{BPP} = \bigcup_k \text{BPTime}(n^k)$.

OPEN QUESTION

Is it true that for any $k > 0$ holds that

$$\text{BPP} \nsubseteq \text{BPTime}(n^k).$$
Derandomization

FOLKLORE

If there is pseudorandom generator that maps $\log(n)$ bits to $\text{poly}(n)$ then $\text{BPP} \not\subseteq \text{BPTime}(n^k)$.

ITSYKSON, KNOP, SOKOLOV, 2015

If there is pseudorandom generator that maps $n$ bits to $\text{poly}(n)$ then $\text{BPP} \not\subseteq \text{BPTime}(n^k)$. 

Derandomization

FOLKLORE

If there is pseudorandom generator that maps $\log(n)$ bits to $\text{poly}(n)$ then $\text{BPP} \not\subseteq \text{BPTime}(n^k)$.

ITSYKSON, KNOP, SOKOLOV, 2015

If there is pseudorandom generator that maps $n$ bits to $\text{poly}(n)$ then $\text{BPP} \not\subseteq \text{BPTime}(n^k)$.
Deterministic algorithms

**HEURISTIC DETERMINISTIC ALGORITHM**

Language $L \in \text{Heur}_\delta \text{DTIME}(n^k)$ iff there is $O(n^k)$-time algorithm $A$ such that

$$\forall n \in \mathbb{N} \quad \text{Pr}_{x \in \{0,1\}^n} [A(x) = L(x)] > 1 - \delta.$$

**FOLKLORE**

For every $k > 0$ and $\epsilon > 0$ holds that

$$\mathsf{P} \not\subseteq \text{Heur}_{1-\epsilon} \text{DTIME}(n^k).$$
Deterministic algorithms

**HEURISTIC DETERMINISTIC ALGORITHM**

Language $L \in \text{Heur}_\delta \text{DTime}(n^k)$ iff there is $O(n^k)$-time algorithm $A$ such that

$$\forall n \in \mathbb{N} \quad \Pr_{x \in \{0,1\}^n} [A(x) = L(x)] > 1 - \delta.$$ 

**FOLKLORE**

For every $k > 0$ and $\epsilon > 0$ holds that

$$\mathbf{P} \nsubseteq \text{Heur}_{1-\epsilon} \text{DTime}(n^k).$$
Bounded probabilistic algorithms

HEURISTIC BOUNDED PROBABILISTIC ALGORITHMS

Language $L \in \text{Heur}_\delta \text{BPTime}(n^k)$ iff there is randomized $O(n^k)$-time algorithm $A$ such that

$$\forall n \in \mathbb{N} \quad \Pr_{x \in \{0,1\}^n} \left[ \Pr[A(x) = L(x)] > \frac{3}{4} \right] > 1 - \delta.$$ 

We also denote $\text{Heur}_\delta \text{BPP} = \bigcup_k \text{Heur}_\delta \text{BPTime}(n^k)$

FORTNOW AND SANTHANAM, 2004

For each $k > 0$ there is $\epsilon > 0$ such that

$$\text{Heur}_\epsilon \text{BPP} \not\subseteq \text{Heur}_\epsilon \text{BPTime}(n^k).$$
**Bounded probabilistic algorithms**

**HEURISTIC BOUNDED PROBABILISTIC ALGORITHMS**

Language $L \in \text{Heur}_\delta \text{BPTime}(n^k)$ iff there is randomized $O(n^k)$-time algorithm $A$ such that

$$\forall n \in \mathbb{N} \quad \Pr_{x \in \{0,1\}^n} [\Pr[A(x) = L(x)] > \frac{3}{4}] > 1 - \delta.$$ 

We also denote $\text{Heur}_\delta \text{BPP} = \bigcup_k \text{Heur}_\delta \text{BPTime}(n^k)$

**FORTNOW AND SANTHANAM, 2004**

For each $k > 0$ there is $\epsilon > 0$ such that

$$\text{Heur}_\epsilon \text{BPP} \not\subseteq \text{Heur}_\epsilon \text{BPTime}(n^k).$$
Bounded probabilistic algorithms

HEURISTIC BOUNDED PROBABILISTIC ALGORITHMS

Language \( L \in \text{Heur}_\delta \text{BPTime}(n^k) \) iff there is randomized \( O(n^k) \)-time algorithm \( A \) such that

\[
\forall n \in \mathbb{N} \quad \Pr_{x \in \{0,1\}^n} \left[ \Pr[A(x) = L(x)] > \frac{3}{4} \right] > 1 - \delta.
\]

We also denote \( \text{Heur}_\delta \text{BPP} = \bigcup_k \text{Heur}_\delta \text{BPTime}(n^k) \)

PERVYSHEV, 2006

For each \( k > 0 \) and \( \epsilon > 0 \) holds that

\[
\text{Heur}_\epsilon \text{BPP} \nsubseteq \text{Heur}_{\frac{1}{2} - \epsilon} \text{BPTime}(n^k).
\]
Nondeterministic algorithms

**HEURISTIC NONDETERMINISTIC ALGORITHMS**

Language $L \in \text{Heur}_\delta \text{NTime}(n^k)$ iff there is nondeterministic $O(n^k)$-time algorithm $A$ such that

$$\forall n \in \mathbb{N} \quad \Pr_{x \in \{0,1\}^n} [A(x) = L(x)] > 1 - \delta.$$ 

**PERVYSHEV, 2006**

For each $k > 0$ and $\epsilon > 0$ holds that

$$\text{NP} \not\subseteq \text{Heur}_{1 - \epsilon} \text{NTime}(n^k).$$
Non-deterministic algorithms

HEURISTIC NONDETERMINISTIC ALGORITHMS

Language \( L \in \text{Heur}_\delta \text{NTime}(n^k) \) iff there is non-deterministic \( O(n^k) \)-time algorithm \( A \) such that

\[
\forall n \in \mathbb{N} \quad \Pr_{x \in \{0,1\}^n}[A(x) = L(x)] > 1 - \delta.
\]

PERVYSHEV, 2006

For each \( k > 0 \) and \( \epsilon > 0 \) holds that

\[
\text{NP} \not\subseteq \text{Heur}_{\frac{1}{2} - \epsilon} \text{NTime}(n^k).
\]
State of art for heuristic hierarchies

**FOLKLORE**

For any $k > 0$ and $\epsilon > 0$ holds

$$P \nsubseteq \text{Heur}_{1-\epsilon} \text{DTime}(n^k).$$

**PERVYSHEV, 2006**

For any $k > 0$ and $\epsilon > 0$ holds

$$\text{Heur}_{\epsilon} \text{BPP} \nsubseteq \text{Heur}_{\frac{1}{2}-\epsilon} \text{BPTime}(n^k) \text{ and } \text{NP} \nsubseteq \text{Heur}_{\frac{1}{2}-\epsilon} \text{NTime}(n^k).$$
Generalized hierarchy

For any $k > 0$, $\epsilon > 0$ and $a > 1$ holds that

$$\text{Heur}_\epsilon \text{FBPP} \not\subseteq \text{Heur}_{1 - \frac{1}{a}} \epsilon \text{FBPTime}(n^k).$$

Moreover there is $F : \{0, 1\}^n \rightarrow \{0, \ldots, b - 1\}$ such that $F \in \text{Heur}_\epsilon \text{FBPP}$ and $F \not\in \text{Heur}_{1 - \frac{1}{a}} \epsilon \text{FBPTime}(n^k)$. 
Generalized hierarchy

ITSYKSON, KNOP, SOKOLOV, 2015

For any $k > 0$, $\epsilon > 0$ and $a > 1$ holds that

$$\text{Heur}_\epsilon \text{FBPP} \not\subseteq \text{Heur}_{1 - \frac{1}{a} - \epsilon} \text{FBPTime}(n^k).$$

Moreover there is $F : \{0, 1\}^n \rightarrow \{0, \ldots, b - 1\}$ such that $F \in \text{Heur}_\epsilon \text{FBPP}$ and $F \not\in \text{Heur}_{1 - \frac{1}{a} - \epsilon} \text{FBPTime}(n^k)$. 
SAMPLABLE RANDOM VARIABLES

Ensemble of random variables \( \gamma \in \text{DSamp}(n^k) \) iff there is a randomized \( O(n^k) \)-time algorithm \( A \) such that \( \gamma_n \) and \( A(1^n) \) are equally distributed. We also denote \( \text{PSamp} = \bigcup_k \text{DSamp}(n^k) \).

WATSON, 2014

For any \( k > 0, \epsilon > 0 \) and \( a > 1 \) there is an ensemble of random variables \( \gamma \in \text{PSamp} \) such that for every \( \beta \in \text{DSamp}(n^k) \) holds \( \Delta(\gamma, \delta) > 1 - \frac{1}{a} - \epsilon \) and \( \text{supp}(\gamma) = \{0, \ldots, a - 1\} \).
Samplable random variables

**SAMPLABLE RANDOM VARIABLES**

Ensemble of random variables $\gamma \in \text{DSamp}(n^k)$ iff there is a randomized $O(n^k)$-time algorithm $A$ such that $\gamma_n$ and $A(1^n)$ are equally distributed.

We also denote $\text{PSamp} = \bigcup_k \text{DSamp}(n^k)$.

**WATSON, 2014**

For any $k > 0$, $\epsilon > 0$ and $a > 1$ there is an ensemble of random variables $\gamma \in \text{PSamp}$ such that for every $\beta \in \text{DSamp}(n^k)$ holds $\Delta(\gamma, \delta) > 1 - \frac{1}{a} - \epsilon$ and $\text{supp}(\gamma) = \{0, \ldots, a - 1\}$. 
Proof for $a = 2$

1. Consider the language $L = \{r \mid 0.r > \Pr[\gamma_n = 1]\}$.

2. Note that $L \in \text{Heur}_B \text{BPP}$. Consider the following algorithm:
   - Sample $r_1, \ldots, r_m$ from $\gamma_n$;
   - Return 1 if $0.r \geq \frac{1}{m} \sum_{i=0}^{m} r_i$;
   - Return 0 in other case.

3. Note that $L \notin \text{Heur}_{1-\frac{1}{a}} \text{BPTime}(n^k)$. Let us assume the opposite that $L$ are decidable by algorithm $D$ and consider the following algorithm:
   - sample random $r \in \{0, 1\}^n$;
   - return 1 if $D(r) = 1$ else return 0.
Proof for \( a = 2 \)

1. Consider the language \( L = \{ r \mid 0.r > \Pr[\gamma_n = 1] \} \).

2. Note that \( L \in \text{Heur}_\epsilon \text{BPP} \).
   Consider the following algorithm:
   - Sample \( r_1, \ldots, r_m \) from \( \gamma_n \);
   - Return 1 if \( 0.r \geq \frac{1}{m} \sum_{i=0}^{m} r_i \);
   - Return 0 in other case.

3. Note that \( L \notin \text{Heur}_{1-\frac{1}{a}-\epsilon} \text{BPTime}(n^k) \).
   Let us assume the opposite that \( L \) are decidable by algorithm \( D \) and consider the following algorithm:
   - sample random \( r \in \{0,1\}^n \);
   - return 1 if \( D(r) = 1 \) else return 0.
Proof for $a = 2$

1. Consider the language $L = \{r \mid 0.r > \Pr[\gamma_n = 1]\}$.

2. Note that $L \in \text{Heur}_\epsilon \text{BPP}$.
   Consider the following algorithm:
   - Sample $r_1, \ldots, r_m$ from $\gamma_n$;
   - Return 1 if $0.r \geq \frac{1}{m} \sum_{i=0}^{m} r_i$;
   - Return 0 in other case.

3. Note that $L \notin \text{Heur}_{1-\frac{1}{a} - \epsilon} \text{BPTime}(n^k)$.
   Let us assume the opposite that $L$ are decidable by algorithm $D$ and consider the following algorithm:
   - sample random $r \in \{0, 1\}^n$;
   - return 1 if $D(r) = 1$ else return 0.
Proof for $a = 2$

1. Consider the language $L = \{ r \mid 0.r > \Pr[\gamma_n = 1] \}$.

2. Note that $L \in \text{Heur}_\epsilon \text{BPP}$. Consider the following algorithm:
   - Sample $r_1, \ldots, r_m$ from $\gamma_n$;
   - Return 1 if $0.r \geq \frac{1}{m} \sum_{i=0}^{m} r_i$;
   - Return 0 in other case.

3. Note that $L \not\in \text{Heur}_{1-\frac{1}{a}} \text{BPTime}(n^k)$.
   Let us assume the opposite that $L$ are decidable by algorithm $D$ and consider the following algorithm:
   - sample random $r \in \{0, 1\}^n$;
   - return 1 if $D(r) = 1$ else return 0.
Proof for $a = 2$

1. Consider the language $L = \{ r \mid 0.r > \Pr[\gamma_n = 1] \}$.

2. Note that $L \in \text{Heur}_\epsilon \text{BPP}$.
   Consider the following algorithm:
   - Sample $r_1, \ldots, r_m$ from $\gamma_n$;
   - Return 1 if $0.r \geq \frac{1}{m} \sum_{i=0}^{m} r_i$;
   - Return 0 in other case.

3. Note that $L \not\in \text{Heur}_{1-\frac{1}{a} - \epsilon} \text{BPTime}(n^k)$.
   Let us assume the opposite that $L$ are decidable by algorithm $D$ and consider the following algorithm:
   - sample random $r \in \{0, 1\}^n$;
   - return 1 if $D(r) = 1$ else return 0.
Proof for $a = 2$

1. Consider the language $L = \{ r \mid 0.r > \text{Pr}[\gamma_n = 1] \}$.

2. Note that $L \in \text{Heur}_\epsilon \text{BPP}$.
   Consider the following algorithm:
   - Sample $r_1, \ldots, r_m$ from $\gamma_n$;
   - Return 1 if $0.r \geq \frac{1}{m} \sum_{i=0}^{m} r_i$;
   - Return 0 in other case.

3. Note that $L \not\in \text{Heur}_{1-\frac{1}{a}} \text{BPTIME}(n^k)$.
   Let us assume the opposite that $L$ are decidable by algorithm $D$ and consider the following algorithm:
   - sample random $r \in \{0, 1\}^n$;
   - return 1 if $D(r) = 1$ else return 0.
Proof for $a = 2$

1. Consider the language $L = \{ r \mid 0.r > \Pr[\gamma_n = 1] \}$.

2. Note that $L \in \text{Heur}_{\epsilon, \text{BPP}}$. Consider the following algorithm:
   - Sample $r_1, \ldots, r_m$ from $\gamma_n$;
   - Return 1 if $0.r \geq \frac{1}{m} \sum_{i=0}^{m} r_i$;
   - Return 0 in other case.

3. Note that $L \notin \text{Heur}_{1 - \frac{1}{a} - \epsilon, \text{BPTIME}}(n^k)$.

   Let us assume the opposite that $L$ are decidable by algorithm $D$ and consider the following algorithm:
   - sample random $r \in \{0, 1\}^n$;
   - return 1 if $D(r) = 1$ else return 0.
Proof for \( a = 2 \)

1. Consider the language \( L = \{ r \mid 0.r > \Pr[\gamma_n = 1]\} \).

2. Note that \( L \in \text{Heur}_\epsilon \text{BPP} \).
   Consider the following algorithm:
   - Sample \( r_1, \ldots, r_m \) from \( \gamma_n \);
   - Return 1 if \( 0.r \geq \frac{1}{m} \sum_{i=0}^{m} r_i \);
   - Return 0 in other case.

3. Note that \( L \not\in \text{Heur}_{1 - \frac{1}{a - \epsilon}} \text{BPTime}(n^k) \).
   Let us assume the opposite that \( L \) are decidable by algorithm \( D \) and consider the following algorithm:
   - sample random \( r \in \{0, 1\}^n \);
   - return 1 if \( D(r) = 1 \) else return 0.
Proof for $a = 2$

1. Consider the language $L = \{ r \mid 0.r > \Pr[\gamma_n = 1] \}$.

2. Note that $L \in \text{Heur}_\epsilon \text{BPP}$.
   Consider the following algorithm:
   - Sample $r_1, \ldots, r_m$ from $\gamma_n$;
   - Return 1 if $0.r \geq \frac{1}{m} \sum_{i=0}^{m} r_i$;
   - Return 0 in other case.

3. Note that $L \notin \text{Heur}_{1 - \frac{1}{a} - \epsilon} \text{BPTIME}(n^k)$.
   Let us assume the opposite that $L$ are decidable by algorithm $D$ and consider the following algorithm:
   - sample random $r \in \{0, 1\}^n$;
   - return 1 if $D(r) = 1$ else return 0.
Open questions

- Hierarchy theorem for $\text{BPTime}(n^k)$ or $\text{RTime}(n^k)$;
- Hierarchy theorem for heuristic version of $\text{RTime}(n^k)$;
- Prove hierarchy for heuristic version of $\text{BPTime}(n^k)$ for bigger confidence parameter.
Open questions

- Hierarchy theorem for $\text{BPTime}(n^k)$ or $\text{RTime}(n^k)$;
- Hierarchy theorem for heuristic version of $\text{RTime}(n^k)$;
- Prove hierarchy for heuristic version of $\text{BPTime}(n^k)$ for bigger confidence parameter.
Open questions

- Hierarchy theorem for $\text{BPTime}(n^k)$ or $\text{RTime}(n^k)$;
- Hierarchy theorem for heuristic version of $\text{RTime}(n^k)$;
- Prove hierarchy for heuristic version of $\text{BPTime}(n^k)$ for bigger confidence parameter.