Distributional proving problems

Edward A. Hirsch

(based on results of
E.A.H., D. Itsykson, I. Monakhov, V. Nikolaenko, A. Smal, D. Sokolov)

Steklov Institute of Mathematics at St.Petersburg, RAS
St.Petersburg Academic University, RAS

Motivation: Optimal algorithms

- Given a problem, even if there is no fast algorithm, what is the best one?
- Levin's optimal algorithm for **NP** search problems is known since 1973:
  - Run all algorithms $A_1, A_2, \ldots$ "in parallel".
  - Once A_i returns a satisfying assignment, verify it.

Motivation: Optimal algorithms

- Given a problem, even if there is no fast algorithm, what is the best one?
- Levin's optimal algorithm for **NP** search problems is known since 1973:
  - Run all algorithms $A_1, A_2, \ldots$ "in parallel".
  - Once A_i returns a satisfying assignment, verify it.
- The existence of optimal algorithms is not known for any decision problem in $\mathbf{NP} \setminus \mathbf{P}$.

  NB: search-to-decision reduction does not work: reduces to *different* instances

Motivation: Optimal algorithms

- Given a problem, even if there is no fast algorithm, what is the best one?
- Levin's optimal algorithm for **NP** search problems is known since 1973:
    - Run all algorithms $A_1, A_2, \ldots$ "in parallel".
    - Once A_i returns a satisfying assignment, verify it.
- The existence of optimal algorithms is not known for any decision problem in $\mathbf{NP} \setminus \mathbf{P}$.

  NB: search-to-decision reduction does not work: reduces to *different* instances
- ...and for any decision problem in $\mathbf{co\text{-}NP} \setminus \mathbf{P}$.

  Optimal algorithms for **TAUT** are tightly related to optimal proof systems

Motivation: Optimal algorithms

- Given a problem, even if there is no fast algorithm, what is the best one?
- Levin's optimal algorithm for **NP** search problems is known since 1973:
    - Run all algorithms $A_1, A_2, \ldots$ "in parallel".
    - Once A_i returns a satisfying assignment, verify it.
- The existence of optimal algorithms is not known for any decision problem in $\mathbf{NP} \setminus \mathbf{P}$.

  NB: search-to-decision reduction does not work: reduces to *different* instances
- ...and for any decision problem in $\mathbf{co}\text{-}\mathbf{NP} \setminus \mathbf{P}$.

  Optimal algorithms for **TAUT** are tightly related to optimal proof systems
- The best we can do is $\mathbf{E} \setminus \mathbf{P}$, for immune sets [Messner; Chen,Flum,Müller].

Distributional proving problems

Distributional proving problem (D, L) consists of

- a language L of "theorems",
- a polynomial-time samplable distribution $D = \{D_n\}_{n\in\mathbb{N}}$ on $\overline{L}$.

Motivation:

- a small (wrt D) amount of wrong theorems is acceptable;
- not interested in what happens on statements that are not claimed;
- polynomial-time samplable distributions are concentrated on **NP** languages, thus the definition is natural for $L \in \mathbf{co\text{-}NP}$.

Related concepts

Distributional problems

- A distribution on all inputs.
- Gives no information about the problem.

Distributional proving problems

- A distribution on negative instances.
- Allows to verify an algorithm on counterexamples.
- There are natural polynomial-time samplable distributions on all negative instances (e.g., planted **SAT**).

PAC learning

- A distribution providing correct answers.
- Allows to verify an algorithm on all samples.
- Polynomial-time samplable distributions on all inputs are unlikely to exist for **NP**-complete problems.

Heuristic acceptors

Definition

(Classical) acceptor A for L:

(completeness) A accepts every $x \in L$.

(correctness) A does not stop on any $x \notin L$.

Complexity parameter: running time <u>on L</u>.

Heuristic acceptors

Distributional proving problem (D, L) consists of a language L of "theorems" and a polynomial-time samplable distribution $D = \{D_n\}_{n\in\mathbb{N}}$ on $\overline{L}$.

Definition

Heuristic acceptor $A(x, d)$ for (D, L): (d is the desired "confidence")

(completeness) $A(x, d)$ accepts every $x \in L$:

(correctness) $A(r, d)$ makes few errors w.r.t. $r \leftarrow D_n$:

Heuristic acceptors

Distributional proving problem (D, L) consists of a language L of "theorems" and a polynomial-time samplable distribution $D = \{D_n\}_{n \in \mathbb{N}}$ on $\overline{L}$.

Definition

Heuristic acceptor $A(x, d)$ for (D, L): (d is the desired "confidence")

(completeness) $A(x, d)$ accepts every $x \in L$:

$$\forall x \in L \; \forall d \in \mathbb{N} \quad A(x, d) = 1.$$

(correctness) $A(r, d)$ makes few errors w.r.t. $r \leftarrow D_n$:

$$\Pr_{r \leftarrow D_n} \{A(r, d) = 1\} < \frac{1}{d} \text{ (deterministic acceptor)}$$

Heuristic acceptors

Distributional proving problem (D, L) consists of a language L of "theorems" and a polynomial-time samplable distribution $D = \{D_n\}_{n \in \mathbb{N}}$ on $\overline{L}$.

Definition

Heuristic acceptor $A(x, d)$ for (D, L): (d is the desired "confidence")

(completeness) $A(x, d)$ accepts every $x \in L$:

$$\forall x \in L \ \forall d \in \mathbb{N} \quad A(x, d) = 1.$$

(correctness) $A(r, d)$ makes few errors w.r.t. $r \leftarrow D_n$:

$$\Pr_{r \leftarrow D_n} \{A(r, d) = 1\} < \frac{1}{d} \text{ (deterministic acceptor)}$$

$$\Pr_{r \leftarrow D_n} \left\{\Pr_A\{A(r, d) = 1\} > \frac{1}{8}\right\} < \frac{1}{d} \text{ (randomized acceptor)}$$

Heuristic acceptors

Distributional proving problem (D, L) consists of a language L of "theorems" and a polynomial-time samplable distribution $D = \{D_n\}_{n \in \mathbb{N}}$ on $\overline{L}$.

Definition

Heuristic acceptor $A(x, d)$ for (D, L): (d is the desired "confidence")

(completeness) $A(x, d)$ accepts every $x \in L$:

$$\forall x \in L \ \forall d \in \mathbb{N} \quad A(x, d) = 1.$$

(correctness) $A(r, d)$ makes few errors w.r.t. $r \leftarrow D_n$:

$\Pr_{r \leftarrow D_n} \{A(r, d) = 1\} < \frac{1}{d}$ (deterministic acceptor)

$\Pr_{r \leftarrow D_n} \left\{\Pr_A\{A(r, d) = 1\} > \frac{1}{8}\right\} < \frac{1}{d}$ (randomized acceptor)

(correctness') $\Pr_{r \leftarrow D_n;\, A} \{A(r, d) = 1\} < \frac{1}{d}$.

Heuristic acceptors

Distributional proving problem (D, L) consists of a language L of "theorems" and a polynomial-time samplable distribution $D = \{D_n\}_{n\in\mathbb{N}}$ on $\overline{L}$.

Definition

Heuristic acceptor $A(x, d)$ for (D, L): (d is the desired "confidence")

(completeness) $A(x, d)$ accepts every $x \in L$:

$$\forall x \in L\ \forall d \in \mathbb{N} \quad A(x, d) = 1.$$

(correctness) $A(r, d)$ makes few errors w.r.t. $r \leftarrow D_n$:

$$\Pr_{r\leftarrow D_n}\{A(r, d) = 1\} < \tfrac{1}{d} \text{ (deterministic acceptor)}$$

$$\Pr_{r\leftarrow D_n}\left\{\Pr_A\{A(r, d) = 1\} > \tfrac{1}{8}\right\} < \tfrac{1}{d} \text{ (randomized acceptor)}$$

(correctness') $\Pr_{r\leftarrow D_n;\, A}\{A(r, d) = 1\} < \frac{1}{d}$.

- Time $\tau_A(x, d)$ is a random variable.
- $t_A(x, d)$ is the median (w.r.t. random bits) running time of $A(x, d)$.
- Polynomial time $\sim$ polynomial in $|x|$ and d.

Theorem

$\exists$ *polynomial-time samplable* D $\exists L \in \textbf{co-NP}$ $\nexists$ *polynomial-time heuristic acceptor for* $(D, L) \iff \exists$ *infinitely-often one-way function.*

Proof.

- i.o. o.w.f. $\Rightarrow$ i.o. PRG
  (similar to [Håstad, Impagliazzo, Levin, Luby])
- i.o. PRG $\Rightarrow$ hard problem for heuristic acceptors
  (hint: PRG is a distribution)
- hard problem for heuristic acceptors $\Rightarrow$ average-case o.w.f.
  (hint: the sampler is difficult to invert)
- average-case o.w.f. $\Rightarrow$ i.o. o.w.f.
  (padding)

Optimal heuristic acceptor

Definition

(Classical) acceptor S simulates W if it runs almost as fast for each x,
i.e., there is a polynomial p such that $\forall x \in L, \quad t_S(x) \leq p(t_W(x) \cdot |x|)$.

Optimal heuristic acceptor

Definition

Heuristic acceptor S simulates W if
it runs almost as fast for each x and ≈the same confidence,
i.e., there are polynomials p and q such that $\forall x \in L$, $\forall d \in \mathbb{N}$,

$$t_S(x, d) \leq \max_{d' \leq q(d \cdot |x|)} p(t_W(x, d') \cdot |x| \cdot d).$$

Idea: Certify A_i by testing it on samples $x \leftarrow D_n$.

Optimal heuristic acceptor

Definition

Heuristic acceptor S simulates W if
it runs almost as fast for each x and $\approx$the same confidence,
i.e., there are polynomials p and q such that $\forall x \in L$, $\forall d \in \mathbb{N}$,

$$t_S(x, d) \leq \max_{d' \leq q(d \cdot |x|)} p(t_W(x, d') \cdot |x| \cdot d).$$

Idea: Certify A_i by testing it on samples $x \leftarrow D_n$.

Optimal heuristic acceptor $U(x, d)$:

- For each $i \leq \log |x|$ in parallel:
    1. Execute $A_i(x, d')$.

Optimal heuristic acceptor

Definition

Heuristic acceptor S simulates W if
it runs almost as fast for each x and $\approx$the same confidence,
i.e., there are polynomials p and q such that $\forall x \in L$, $\forall d \in \mathbb{N}$,

$$t_S(x,d) \le \max_{d' \le q(d \cdot |x|)} p(t_W(x,d') \cdot |x| \cdot d).$$

Idea: Certify A_i by testing it on samples $x \leftarrow D_n$.

Optimal heuristic acceptor $U(x,d)$:

- For each $i \le \log|x|$ in parallel:
  1. Execute $A_i(x,d')$.
  2. If it accepts (<u>in T_i steps</u>), test its correctness:

     let $E_i = 0$ and execute k times:
     - $r \leftarrow D_{|x|}$,
     - if $A_i(r,d') = 1$ <u>in T_i steps</u>, then $E_i := E_i + 1$;

Optimal heuristic acceptor

Definition

Heuristic acceptor S simulates W if
it runs almost as fast for each x and $\approx$the same confidence,
i.e., there are polynomials p and q such that $\forall x \in L$, $\forall d \in \mathbb{N}$,

$$t_S(x, d) \leq \max_{d' \leq q(d \cdot |x|)} p(t_W(x, d') \cdot |x| \cdot d).$$

Idea: Certify A_i by testing it on samples $x \leftarrow D_n$.

Optimal heuristic acceptor $U(x, d)$:

- For each $i \leq \log |x|$ in parallel:
  1. Execute $A_i(x, d')$.
  2. If it accepts (in T_i steps), test its correctness:
     let $E_i = 0$ and execute k times:
     - $r \leftarrow D_{|x|}$,
     - if $A_i(r, d') = 1$ in T_i steps, then $E_i := E_i + 1$;
  3. If $E_i < \delta k$, output "1".

Optimal heuristic acceptor

Definition

Heuristic acceptor S simulates W if
it runs almost as fast for each x and $\approx$the same confidence,
i.e., there are polynomials p and q such that $\forall x \in L$, $\forall d \in \mathbb{N}$,

$$t_S(x,d) \le \max_{d' \le q(d\cdot|x|)} p(t_W(x,d') \cdot |x| \cdot d).$$

Let $d' = 4d|x|$, $k = 2d^3|x|^3$, $\delta = \frac{1}{2d|x|}$.

Optimal heuristic acceptor $U(x,d)$:

- For each $i \le \log|x|$ in parallel:
  1. Execute $A_i(x,d')$.
  2. If it accepts ($\underline{\text{in } T_i \text{ steps}}$), test its correctness:

     let $E_i = 0$ and execute k times:
     - $r \leftarrow D_{|x|}$,
     - if $A_i(r,d') = 1$ $\underline{\text{in } T_i \text{ steps}}$, then $E_i := E_i + 1$;
  3. If $E_i < \delta k$, output "1".

Heuristic proof systems

- Probabilistic proof verification (with bounded error).
- Small fraction $1/d$ of false theorems (unbounded error).

Heuristic proof systems

- Probabilistic proof verification (with bounded error).
- Small fraction $1/d$ of false theorems (unbounded error).

Definition

Heuristic proof system for (D, L) is a polynomial-time Π such that

(completeness) There is a proof accepted whp

(correctness) Most non-theorems don't have such proofs:

Heuristic proof systems

- Probabilistic proof verification (with bounded error).
- Small fraction $1/d$ of false theorems (unbounded error).

Definition

Heuristic proof system for (D, L) is a polynomial-time Π such that

(completeness) There is a proof accepted whp:

$$\forall x \in L\ \forall d \in \mathbb{N}\ \exists w \quad \Pr\{\Pi(x, w, d) = 1\} > \tfrac{1}{2}.$$

(Such w is a Π-proof with confidence d.)

(correctness) Most non-theorems don't have such proofs:

$$\Pr_{r \leftarrow D_n}\{\exists w\ \{\Pr\{\Pi(r, w, d) = 1\} > \tfrac{1}{8}\}\} < \tfrac{1}{d}.$$

Turning **AM** protocols into heuristic proof systems

- Assume $L \in \mathbf{AM}$.
  (E.g., $L = \mathbf{GNI}$, D samples random isomorphic graphs.)
- Consider a protocol (A, M) for L
  (w.l.o.g., with perfect completeness and exponentially small error):

$$\begin{aligned} x \in L &\quad\Longrightarrow\quad \forall r\ \exists w\ A(x, w, r) = 1, \\ x \notin L &\quad\Longrightarrow\quad \Pr_r\{\exists w\ A(x, w, r) = 1\} < 2^{-|x|}. \end{aligned}$$

Turning **AM** protocols into heuristic proof systems

- Assume $L \in \mathbf{AM}$.
  (E.g., $L = \mathbf{GNI}$, D samples random isomorphic graphs.)
- Consider a protocol (A, M) for L
  (w.l.o.g., with perfect completeness and exponentially small error):

$$\begin{aligned} x \in L &\implies \forall r\ \exists w\ A(x, w, r) = 1, \\ x \notin L &\implies \Pr_r\{\exists w\ A(x, w, r) = 1\} < 2^{-|x|}. \end{aligned}$$

- Consider $L' = \{(x, r) \mid x \in L\}$, where the length of r is enough to make the public random choices.
- Consider $D' = D \times U$, where D is any "original" distribution on $\overline{L}$ and U is the uniform distribution.

Turning **AM** protocols into heuristic proof systems

- Assume $L \in \mathbf{AM}$.
  (E.g., $L = \mathbf{GNI}$, D samples random isomorphic graphs.)
- Consider a protocol (A, M) for L
  (w.l.o.g., with perfect completeness and exponentially small error):

$$\begin{aligned} x \in L &\implies \forall r\ \exists w\ A(x, w, r) = 1, \\ x \notin L &\implies \Pr_r\{\exists w\ A(x, w, r) = 1\} < 2^{-|x|}. \end{aligned}$$

- Consider $L' = \{(x, r) \mid x \in L\}$, where the length of r is enough to make the public random choices.
- Consider $D' = D \times U$, where D is any "original" distribution on $\overline{L}$ and U is the uniform distribution.

Theorem (Itsykson, Sokolov)

(1) (L', D') has a polynomially bounded heuristic p.s.

. . .

- Proof: Simulate A (first round) using r.

Turning **AM** protocols into heuristic proof systems

Discussion

- Consider $L' = \{(x, r) \mid x \in L\}$, where the length of r is enough to make the public random choices.
- Consider $D' = D \times U$, where D is any "original" distribution on $\overline{L}$ and U is the uniform distribution.

Question: Is there a classical polynomially bounded proof system for L'?
Answer:

Theorem (Itsykson, Sokolov)

(1) (L', D') *has a polynomially bounded heuristic p.s.*

Turning **AM** protocols into heuristic proof systems

Discussion

- Consider $L' = \{(x, r) \mid x \in L\}$, where the length of r is enough to make the public random choices.
- Consider $D' = D \times U$, where D is any "original" distribution on $\overline{L}$ and U is the uniform distribution.

Question: Is there a classical polynomially bounded proof system for L'?
Answer: We don't know.

Theorem (Itsykson, Sokolov)

(1) (L', D') *has a polynomially bounded heuristic p.s.*

Turning **AM** protocols into heuristic proof systems

Discussion

- Consider $L' = \{(x, r) \mid x \in L\}$, where the length of r is enough to make the public random choices.
- Consider $D' = D \times U$, where D is any "original" distribution on $\overline{L}$ and U is the uniform distribution.

Question: Is there a classical polynomially bounded proof system for L'?
Answer: ~~We don't know.~~
Implies $L \in \mathbf{NP}$.

Theorem (Itsykson, Sokolov)

(1) (L', D') has a polynomially bounded heuristic p.s.
(2) if $L' \in \mathbf{NP}$, then $L \in \mathbf{NP}$.

Turning **AM** protocols into heuristic proof systems
Discussion

- Consider $L' = \{(x, r) \mid x \in L\}$, where the length of r is enough to make the public random choices.
- Consider $D' = D \times U$, where D is any "original" distribution on $\overline{L}$ and U is the uniform distribution.

Question: Is there a classical polynomially bounded proof system for L'?
Answer: ~~We don't know.~~
Implies $L \in \mathbf{NP}$.

Question: Is there a heuristic algorithm for (L', D')?
Answer:

Theorem (Itsykson, Sokolov)

(1) (L', D') *has a polynomially bounded heuristic p.s.*
(2) if $L' \in \mathbf{NP}$*, then* $L \in \mathbf{NP}$*.*

Turning **AM** protocols into heuristic proof systems

Discussion

- Consider $L' = \{(x, r) \mid x \in L\}$, where the length of r is enough to make the public random choices.
- Consider $D' = D \times U$, where D is any "original" distribution on $\overline{L}$ and U is the uniform distribution.

Question: Is there a classical polynomially bounded proof system for L'?
Answer: ~~We don't know.~~
Implies $L \in \mathbf{NP}$.

Question: Is there a heuristic algorithm for (L', D')?
Answer: We don't know.

Theorem (Itsykson, Sokolov)

(1) (L', D') has a polynomially bounded heuristic p.s.
(2) if $L' \in \mathbf{NP}$, then $L \in \mathbf{NP}$.

Turning **AM** protocols into heuristic proof systems

Discussion

- Consider $L' = \{(x, r) \mid x \in L\}$, where the length of r is enough to make the public random choices.
- Consider $D' = D \times U$, where D is any "original" distribution on $\overline{L}$ and U is the uniform distribution.

Question: Is there a classical polynomially bounded proof system for L'?
Answer: ~~We don't know.~~
Implies $L \in \mathbf{NP}$.

Question: Is there a heuristic algorithm for (L', D')?
Answer: ~~We don't know.~~
Implies randomized heuristic algorithm for (L, D).

Theorem (Itsykson, Sokolov)

(1) (L', D') has a polynomially bounded heuristic p.s.
(2) if $L' \in \mathbf{NP}$, then $L \in \mathbf{NP}$.
(3) if (L', D') has polynomial-time heuristic acceptor, then (L, D) does.

Turning **AM** protocols into heuristic proof systems

Discussion

- Consider $L' = \{(x, r) \mid x \in L\}$, where the length of r is enough to make the public random choices.
- Consider $D' = D \times U$, where D is any "original" distribution on $\overline{L}$ and U is the uniform distribution.

Question: Is there a classical polynomially bounded proof system for L'?
Answer: ~~We don't know.~~
Implies $L \in \mathbf{NP}$.

Question: Is there a heuristic algorithm for (L', D')?
Answer: ~~We don't know.~~
Implies randomized heuristic algorithm for (L, D).

Theorem (Itsykson, Sokolov)

(1) (L', D') *has a polynomially bounded heuristic p.s.*
(2) if $\forall L\ L' \in \mathbf{NP}$, *then* $\mathbf{NP} = \mathbf{co\text{-}NP}$.
(3) if $\forall L, D\ (L', D')$ *has poly-time h.acc., then* $(\mathbf{NP}, \mathrm{PSamp})$ *does.*

Optimal proof systems

Classical case

- A proof system Σ simulates a proof system Ω iff Σ-proofs are at most as long as Ω-proofs (up to a polynomial p):

$$\forall F \in L \quad |\text{shortest } \Sigma\text{-proof of } F| \leq p(|\text{shortest } \Omega\text{-proof of } F|, |F|).$$

- p-simulation is a constructive version: For any w-size Ω-proof, one can compute a $p(w)$-size Σ-proof in polynomial time.
- (p-)optimal proof system (p-)simulates any other proof system.
- **Does it exist?..**

Optimal proof systems

Classical case

- A proof system Σ simulates a proof system Ω iff Σ-proofs are at most as long as Ω-proofs (up to a polynomial p):

$$\forall F \in L \ \ |\text{shortest } \Sigma\text{-proof of } F| \leq p(|\text{shortest } \Omega\text{-proof of } F|, |F|).$$

- p-simulation is a constructive version: For any w-size Ω-proof, one can compute a $p(w)$-size Σ-proof in polynomial time.
- (p-)optimal proof system (p-)simulates any other proof system.
- **Does it exist?..**

Theorem

$\exists$ ***p-optimal proof system*** $\iff$ $\exists$ ***optimal acceptor.***

For **TAUT**: [Krajícek, Pudlák].

For paddable languages: [Messner].

For **co-NP**-complete languages: [Chen, Flüm, Müller].

From acceptors to proof systems

Definition

L is paddable if there is an injective non-length-decreasing polynomial-time padding function $\mathrm{pad}_L\colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ that is polynomial-time invertible on its image and such that $\forall x, w$ $(x \in L \iff \mathrm{pad}_L(x,w) \in L)$.

Optimal proof [Messner, 99]:

- A proof π of x in some system Π;
- padding.

Verification:

- run optimal acceptor on $\mathrm{pad}_L(x,\pi)$;
- for a correct proof π, it accepts in a polynomial time because for a correct system Π, the set $\{\mathrm{pad}_L(x,\pi) \mid x \in L,\ \Pi(x,\pi)=1\} \subseteq L$ can be accepted in a polynomial time.

From acceptors to proof systems

Optimal proof [Messner, 99]:

- A proof π of x in some system Π;
- padding.

Verification:

- run optimal acceptor on $\mathrm{pad}_L(x, \pi)$;
- for a correct proof π, it accepts in a polynomial time because for a correct system Π, the set $\{\mathrm{pad}_L(x, \pi) \mid x \in L,\ \Pi(x, \pi) = 1\} \subseteq L$ can be accepted in a polynomial time.

Applicability:

- Messner's proof goes for randomized algorithms.

From acceptors to proof systems

Optimal proof [Messner, 99]:

- A proof π of x in some system Π;
- padding.

Verification:

- run optimal acceptor on $\mathrm{pad}_L(x, \pi)$;
- for a correct proof π, it accepts in a polynomial time because for a correct system Π, the set $\{\mathrm{pad}_L(x, \pi) \mid x \in L,\ \Pi(x, \pi) = 1\} \subseteq L$ can be accepted in a polynomial time.

Applicability:

- Messner's proof goes for randomized algorithms.
- Does not go for heuristic, average-case algorithms.

Simulations

- pointwise simulation $\mathcal{A} \prec \mathcal{B}$:
  $\exists$ polynomial p $\forall x$
  $$t_{\mathcal{A}}(x) \leq p(t_{\mathcal{B}}(x) + |x|)$$

Simulations

- pointwise simulation $\mathcal{A} \prec \mathcal{B}$:
  $\exists$ polynomial p $\forall x$
  $$t_{\mathcal{A}}(x) \leq p(t_{\mathcal{B}}(x) + |x|)$$

- (yet weaker!) worst-case simulation $\mathcal{A} \prec_{wc} \mathcal{B}$:
  $\exists$ polynomials p, q $\forall x$
  $$t_{\mathcal{A}}(x) \leq p(\max_{\substack{|x'| \leq q(|x|) \\ x' \in L}} t_{\mathcal{B}}(x') + |x|)$$

Simulations

- pointwise simulation $\mathcal{A} \prec \mathcal{B}$:
  $\exists$ polynomial p $\forall x$
  $$t_{\mathcal{A}}(x) \leq p(t_{\mathcal{B}}(x) + |x|)$$
- (weaker) average-case simulation $\mathcal{A} \prec_D \mathcal{B}$ w.r.t. D:
  $\forall \epsilon > 0$ $\exists c > 0$
  $$\mathop{\mathbf{E}}_{x \leftarrow D_n}[{t_{\mathcal{A}}}^c(x)] = O(n \mathop{\mathbf{E}}_{y \leftarrow D_n}[{t_{\mathcal{B}}}^{\epsilon}(y)])$$
- (yet weaker!) worst-case simulation $\mathcal{A} \prec_{wc} \mathcal{B}$:
  $\exists$ polynomials p, q $\forall x$
  $$t_{\mathcal{A}}(x) \leq p(\max_{\substack{|x'| \leq q(|x|) \\ x' \in L}} t_{\mathcal{B}}(x') + |x|)$$

Simulations

- pointwise simulation $\mathcal{A} \prec \mathcal{B}$:
  $\exists$ polynomial p $\forall x$
  $$t_{\mathcal{A}}(x) \leq p(t_{\mathcal{B}}(x) + |x|)$$
- (weaker) average-case simulation $\mathcal{A} \prec_D \mathcal{B}$ w.r.t. D:
  $\forall \epsilon > 0$ $\exists c > 0$
  $$\mathop{\mathbf{E}}_{x \leftarrow D_n}[{t_{\mathcal{A}}}^c(x)] = O(n \mathop{\mathbf{E}}_{y \leftarrow D_n}[{t_{\mathcal{B}}}^\epsilon(y)])$$
- (weaker) simulation scheme:
  simulate everywhere except for the set of D-prob. $1/2d$.
- (yet weaker!) worst-case simulation $\mathcal{A} \prec_{wc} \mathcal{B}$:
  $\exists$ polynomials p, q $\forall x$
  $$t_{\mathcal{A}}(x) \leq p(\max_{\substack{|x'| \leq q(|x|) \\ x' \in L}} t_{\mathcal{B}}(x') + |x|)$$

Problems and complexities

- Decision problem L: is $x \in L$?
  Solved by decision algorithms: complexity measure = time.

Problems and complexities

- Decision problem L: is $x \in L$?
  Solved by decision algorithms: complexity measure = time.
- Same problem, solved by acceptors: complexity measure = time on L.

Problems and complexities

- Decision problem L: is $x \in L$?
  Solved by decision algorithms: complexity measure = time.
- Same problem, solved by acceptors: complexity measure = time on L.
  - worst-case optimal acceptor for **NP**-complete problems:

Problems and complexities

- Decision problem L: is $x \in L$?
  Solved by decision algorithms: complexity measure = time.
- Same problem, solved by acceptors: complexity measure = time on L.
  - worst-case optimal acceptor for **NP**-complete problems:
    Levin's universal search + self-to-decision reduction.
  - worst-case (and stronger) optimal randomized acceptor for GNI:
    verification by Goldwasser-Micali-Sipser protocol.

Problems and complexities

- Decision problem L: is $x \in L$?
  Solved by decision algorithms: complexity measure = time.
- Same problem, solved by acceptors: complexity measure = time on L.
  - worst-case optimal acceptor for **NP**-complete problems:
    Levin's universal search + self-to-decision reduction.
  - worst-case (and stronger) optimal randomized acceptor for GNI.
  - pointwise-optimal acceptor for Time(f)-immune sets [Messner],
    pointwise-optimal algorithm for bi-immune sets [Chen,Flum,Müller].

Problems and complexities

- Decision problem L: is $x \in L$?
  Solved by decision algorithms: complexity measure = time.
- Same problem, solved by acceptors: complexity measure = time on L.
  - worst-case optimal acceptor for **NP**-complete problems:
    Levin's universal search + self-to-decision reduction.
  - worst-case (and stronger) optimal randomized acceptor for GNI.
  - pointwise-optimal acceptor, algorithm for a set in $\mathbf{E} \setminus \mathbf{P}$.
- Distributional problem (D, L): is $x \in L$ with accuracy d?
  Complexity measure = $\text{time}(n, d)$.
  Errorless average-case complexity: estimate $\mathbf{E}$ or give up with D-prob. $1/d$.
  - average-case optimal randomized acceptor for GNI for some D.

Problems and complexities

- Decision problem L: is $x \in L$?
  Solved by decision algorithms: complexity measure = time.
- Same problem, solved by acceptors: complexity measure = time on L.
  - worst-case optimal acceptor for **NP**-complete problems:
    Levin's universal search + self-to-decision reduction.
  - worst-case (and stronger) optimal randomized acceptor for GNI.
  - pointwise-optimal acceptor, algorithm for a set in $\mathbf{E} \setminus \mathbf{P}$.
- Distributional problem (D, L): is $x \in L$ with accuracy d?
  Complexity measure = time(n, d).
  Errorless average-case complexity: estimate **E** or give up with D-prob. $1/d$.
  - average-case optimal randomized acceptor for GNI for some D.
- Same problem, solved by heuristic algorithms:
  allow false negatives and positives with D-prob. $1/d$.
  - pointwise optimal randomized *algorithm* for **Im** of an injective function,
  - "scheme-optimal" deterministic *algorithm* for —"—"—.

Problems and complexities

- Decision problem L: is $x \in L$?
  Solved by decision algorithms: complexity measure = time.
- Same problem, solved by acceptors: complexity measure = time on L.
  - worst-case optimal acceptor for **NP**-complete problems:
    Levin's universal search + self-to-decision reduction.
  - worst-case (and stronger) optimal randomized acceptor for GNI.
  - pointwise-optimal acceptor, algorithm for a set in $\mathbf{E} \setminus \mathbf{P}$.
- Distributional problem (D, L): is $x \in L$ with accuracy d?
  Complexity measure = $\text{time}(n, d)$.
  Errorless average-case complexity: estimate **E** or give up with D-prob. $1/d$.
  - average-case optimal randomized acceptor for GNI for some D.
- Same problem, solved by heuristic algorithms:
  allow false negatives and positives with D-prob. $1/d$.
  - pointwise optimal randomized *algorithm* for **Im** of an injective function,
  - "scheme-optimal" deterministic *algorithm* for —"—"—.
- Distributional proving problem (D, L): $\mathbf{supp}\, D \subseteq \overline{L}$.
  Solved by heuristic acceptors, may allow false positives only.
  - pointwise optimal randomized heuristic acceptor for p.-t.s. D, r.e. L.

Open questions

- $\exists$ optimal proof system $\iff$ $\exists$ optimal heuristic acceptor;
- $\exists$ optimal heuristic proof system $\overset{?}{\iff}$ $\exists$ optimal heuristic acceptor;
- $\exists$ optimal proof system with advice $\overset{?}{\iff}$ $\exists$ optimal acceptor with advice;
- $\exists$ average-case optimal acceptor?
- $\exists$ optimal acceptor for GNI or any other $\mathbf{co}$-$\mathbf{NP} \setminus \mathbf{P}$ problem?
- $\exists$ optimal proof system for any problem outside $\mathbf{P}$?
- $\exists (D, L) \in (\mathbf{co}$-$\mathbf{NP}, \mathrm{PSamplable})$ with no polynomially-bounded heuristic proof system $\iff$?
- **AM** protocols make deterministic (heuristic) proof systems with very small error; suggest another example: randomized and with larger error.