Average-case Computational Complexity

Algorithms and Complexity

Andreas Mantis

MPLA

July 10, 2014

Motivation and Structure of the talk

- Conventional complexity theory deals with worst-case.

Motivation and Structure of the talk

- Conventional complexity theory deals with worst-case.
- However, we are only interested in instances that arise in practice.

Motivation and Structure of the talk

- Conventional complexity theory deals with worst-case.
- However, we are only interested in instances that arise in practice.
- First, we define what are distributional problems and the class of "easy" problems(distP).

Motivation and Structure of the talk

- Conventional complexity theory deals with worst-case.
- However, we are only interested in instances that arise in practice.
- First, we define what are distributional problems and the class of "easy" problems(distP).
- What are some realistic distributions?

Motivation and Structure of the talk

- Conventional complexity theory deals with worst-case.
- However, we are only interested in instances that arise in practice.
- First, we define what are distributional problems and the class of "easy" problems(distP).
- What are some realistic distributions?
- How can we define reductions on average case?

Motivation and Structure of the talk

- Conventional complexity theory deals with worst-case.
- However, we are only interested in instances that arise in practice.
- First, we define what are distributional problems and the class of "easy" problems(distP).
- What are some realistic distributions?
- How can we define reductions on average case?
- A class of "hard" problems (distNP) and a complete problem for that class.

Distributional Problems and distP

Distributional Problem

A distributional problem is a pair $< L, D >$ where $L \subseteq \{0,1\}^*$ is a language, and $D = \{D_n\}$ is a sequence of distributions, with D_n being a distribution over $\{0,1\}^n$

Distributional Problems and distP

Distributional Problem

A distributional problem is a pair $< L, D >$ *where* $L \subseteq \{0,1\}^*$ *is a language, and* $D = \{D_n\}$ *is a sequence of distributions, with* D_n *being a distribution over* $\{0,1\}^n$

For example: SAT only on inputs with more than 10n clauses u.a.r

distP definition

- Defining a class of easy problems on average is tricky.

distP definition

- Defining a class of easy problems on average is tricky.

distP first attempt

$< L, D >$ is poly-time solvable on average if there is an algorithm A such that $A(x) = L(x)$ for every x and a polynomial p such that for every n,

$$\mathbb{E}_{x \sim D_n}[time_A(x)] \leq p(n).$$

distP definition

- The definition is not robust.

distP definition

- The definition is not robust.
- If we change the model of computation to one with quadratic slow-down, we lose exponentially on time.

distP definition

- The definition is not robust.
- If we change the model of computation to one with quadratic slow-down, we lose exponentially on time.
- Suppose that an algorithm A halts in n steps on every input except for the all-zeros input, on which it runs for 2^n steps.

distP definition

- The definition is not robust.
- If we change the model of computation to one with quadratic slow-down, we lose exponentially on time.
- Suppose that an algorithm A halts in n steps on every input except for the all-zeros input, on which it runs for 2^n steps.
- $\mathbb{E}_{x\sim\{0,1\}^n}[time_A(x)] = (1-2^{-n})n + 2^{-n}2^n \leq n+1.$

distP definition

- The definition is not robust.
- If we change the model of computation to one with quadratic slow-down, we lose exponentially on time.
- Suppose that an algorithm A halts in n steps on every input except for the all-zeros input, on which it runs for 2^n steps.
- $\mathbb{E}_{x\sim\{0,1\}^n}[time_A(x)] = (1-2^{-n})n + 2^{-n}2^n \leq n+1.$
- $\mathbb{E}_{x\sim\{0,1\}^n}[time_A^2(x)] = (1-2^{-n})n^2 + 2^{-n}2^{2n} \geq 2^n.$

distP definition

distP

A distributional problem $< L, D >$ *is in distP if there is an algorithm* A *for* L *and constants* C *and* $\epsilon > 0$ *such that for every* n*,*

$$\mathbb{E}_{x \sim D_n}\left[\frac{time_A(x)^{\epsilon}}{n}\right] \leq C.$$

distP definition observations

- $P \subseteq distP$. Indeed, if L is decided by A in $O(|x|^c)$, then $time_A(x)^{1/c} = O(|x|)$ and the expectation would be bounded by a constant.

distP definition observations

- $P \subseteq distP$. Indeed, if L is decided by A in $O(|x|^c)$, then $time_A(x)^{1/c} = O(|x|)$ and the expectation would be bounded by a constant.
- High probability to run in polytime. Indeed, by Markov's inequality, for every $K > 1$, $Prob[\frac{time_A(x)^{\epsilon}}{n} \geq KC] = Prob[time_A(x) \geq (KCn)^{1/\epsilon}]$ is at most $1/K$.

distP definition observations

- $P \subseteq distP$. Indeed, if L is decided by A in $O(|x|^c)$, then $time_A(x)^{1/c} = O(|x|)$ and the expectation would be bounded by a constant.
- High probability to run in polytime. Indeed, by Markov's inequality, for every $K > 1$, $Prob[\frac{time_A(x)^\epsilon}{n} \geq KC] = Prob[time_A(x) \geq (KCn)^{1/\epsilon}]$ is at most $1/K$.
- Robust in minor changes. Indeed, for every $d > 0$ our definition is equivalent with: there exist ϵ, C, such that

$$\mathbb{E}_{x \sim D_n}[\frac{time_A(x)^\epsilon}{n^d}] \leq C.$$

Realistic distributions

Realistic distributions

- The world is indifferent to our algorithm.

Realistic distributions

- The world is indifferent to our algorithm.

P-computable distributions

The P-computable distributions have an associated deterministic polynomial time machine that, given input $x \in \{0,1\}^n$, can compute the cumulative probability $\mu_{D_n}(x)$, where

$$\mu_{D_n}(x) = \sum_{y \in \{0,1\}^n : y \leq x} Pr[y].$$

Realistic distributions

P-samplable distributions

The P-samplable distributions have an associated probabilistic polynomial time machine that can produce samples from the distribution. Specifically, we say that $D = \{D_n\}$ is P-samplable if there is a polynomial p and a probabilistic $p(n)$-time algorithm A such that for every n, the random variables $A(1^n)$ and D_n are identically distributed.

Realistic distributions

Theorem 2

Every P-computable distribution is also P-samplable

Realistic distributions

Theorem 2

Every P-computable distribution is also P-samplable

Proof.

- Generate a truncated $\rho \in [0, 1]$.

Realistic distributions

Theorem 2

Every P-computable distribution is also P-samplable

Proof.

- Generate a truncated $\rho \in [0, 1]$.
- Look via binary search for the unique x, such that $\mu_{D_n}(x-1) < \rho \leq \mu_{D_n}(x)$.

Realistic distributions

Theorem 2

Every P-computable distribution is also P-samplable

Proof.

- Generate a truncated $\rho \in [0, 1]$.
- Look via binary search for the unique x, such that $\mu_{D_n}(x-1) < \rho \leq \mu_{D_n}(x)$.
- Output x.

□

Realistic distributions

Theorem 2

Every P-computable distribution is also P-samplable

Proof.

- Generate a truncated $\rho \in [0,1]$.
- Look via binary search for the unique x, such that $\mu_{D_n}(x-1) < \rho \leq \mu_{D_n}(x)$.
- Output x.

$\square$

NOTE: The converse is not true unless $P = P^{\sharp P}$.

distNP and its complete problems

distNP

A distributional problem $< L, D >$ is in distNP if $L \in NP$ and D is P-computable.

distNP and its complete problems

distNP

A distributional problem $< L, D >$ is in distNP if $L \in NP$ and D is P-computable.

Average-case reduction

We say that a distributional problem $< L, D >$ average-case reduces to a distributional problem $< L', D' >$, if there is a polynomial-time computable f and polynomials p,q: $\mathbb{N} \to \mathbb{N}$ satisfying:

1. (Correctness) For every $x \in \{0,1\}^*$, $x \in L \Leftrightarrow f(x) \in L'$.

2. (Length Regularity) For every $x \in \{0,1\}^*$, $|f(x)| = p(|x|)$.

3. (Domination) For every $n \in \mathbb{N}$ and $y \in \{0,1\}^{p(n)}$,

$$Pr[y = f(D_n)] \leq q(n) Pr[y = D'_{p(n)}].$$

distNP and its complete problems

Notes on the definition:

- (Correctness) For every $x \in \{0,1\}^*$, $x \in L \Leftrightarrow f(x) \in L'$.

Notes on the definition:

- (Correctness) For every $x \in \{0,1\}^*$, $x \in L \Leftrightarrow f(x) \in L'$.
  Like any other reduction!

distNP and its complete problems

Notes on the definition:

- (Correctness) For every $x \in \{0,1\}^*$, $x \in L \Leftrightarrow f(x) \in L'$.
  Like any other reduction!
- (Length Regularity) For every $x \in \{0,1\}^*$, $|f(x)| = p(|x|)$.

distNP and its complete problems

Notes on the definition:

- (Correctness) For every $x \in \{0,1\}^*$, $x \in L \Leftrightarrow f(x) \in L'$.
  Like any other reduction!
- (Length Regularity) For every $x \in \{0,1\}^*$, $|f(x)| = p(|x|)$.
  Technical. Useful for proving transitivity.

Notes on the definition:

- (Correctness) For every $x \in \{0,1\}^*$, $x \in L \Leftrightarrow f(x) \in L'$. Like any other reduction!
- (Length Regularity) For every $x \in \{0,1\}^*$, $|f(x)| = p(|x|)$. Technical. Useful for proving transitivity.
- (Domination) For every $n \in \mathbb{N}$ and $y \in \{0,1\}^{p(n)}$,

$$Pr[y = f(D_n)] \leq q(n)Pr[y = D'_{p(n)}].$$

distNP and its complete problems

Notes on the definition:

- (Correctness) For every $x \in \{0,1\}^*$, $x \in L \Leftrightarrow f(x) \in L'$.
  Like any other reduction!
- (Length Regularity) For every $x \in \{0,1\}^*$, $|f(x)| = p(|x|)$.
  Technical. Useful for proving transitivity.
- (Domination) For every $n \in \mathbb{N}$ and $y \in \{0,1\}^{p(n)}$,

$$Pr[y = f(D_n)] \leq q(n) Pr[y = D'_{p(n)}].$$

If A' is for $< L', D' >$, we want the obvious algorithm A for $< L, D >$ to work: on $x \sim D$, we compute $f(x) = y$ and run algorithm A' on y.

Theorem 1

distP is closed under the average-case reduction.

distNP and its complete problems

Theorem 1

distP is closed under the average-case reduction.

Proof:

- A' solves polynomially $< L', D' >$. Therefore, there are constants C, $\epsilon > 0$, such that for every m,

$$\mathbb{E}_{x \sim D'_m}[\frac{time_{A'}(x)^{\epsilon}}{m}] \leq C.$$

distNP and its complete problems

Theorem 1

distP is closed under the average-case reduction.

Proof:

- A' solves polynomially $< L', D' >$. Therefore, there are constants C, $\epsilon > 0$, such that for every m,

$$\mathbb{E}_{x \sim D'_m}[\frac{time_{A'}(x)^{\epsilon}}{m}] \leq C.$$

- We need to prove that $< L, D >$ is in distP.

distNP and its complete problems

- Assume that for every x, $|f(x)| = |x|^d$.

distNP and its complete problems

- Assume that for every x, $|f(x)| = |x|^d$.
- Computing f on input length n is faster than A' on input length n^d. Therefore $time_A(x) \leq 2time_{A'}(f(x))$.

distNP and its complete problems

- Assume that for every x, $|f(x)| = |x|^d$.
- Computing f on input length n is faster than A' on input length n^d. Therefore $time_A(x) \leq 2time_{A'}(f(x))$.
- It suffices to show that

$$\mathbb{E}_{x \sim D_n}[\frac{(\frac{1}{2}time_A(x))^{\epsilon}}{q(n)n^d}] \leq C.$$

where $q(n)$ is the polynomial in the dominating condition.

distNP and its complete problems

$$\mathbb{E}_{x\sim D_n}\left[\frac{\left(\frac{1}{2}time_A(x)\right)^{\epsilon}}{q(n)n^d}\right] \leq \sum_{y\in\{0,1\}^{n^d}} Pr[y = f(D_n)]\frac{time_{A'}(y)^{\epsilon}}{q(n)n^d}$$

distNP and its complete problems

$$\mathbb{E}_{x\sim D_n}\left[\frac{\left(\frac{1}{2}time_A(x)\right)^{\epsilon}}{q(n)n^d}\right] \leq \sum_{y\in\{0,1\}^{n^d}} Pr[y = f(D_n)]\frac{time_{A'}(y)^{\epsilon}}{q(n)n^d}$$

$$\leq \sum_{y\in\{0,1\}^{n^d}} Pr[y = D'_{n^d}]\frac{time_{A'}(y)^{\epsilon}}{n^d}$$

distNP and its complete problems

$$\mathbb{E}_{x\sim D_n}[\frac{(\frac{1}{2}time_A(x))^{\epsilon}}{q(n)n^d}] \leq \sum_{y\in\{0,1\}^{n^d}} Pr[y = f(D_n)]\frac{time_{A'}(y)^{\epsilon}}{q(n)n^d}$$

$$\leq \sum_{y\in\{0,1\}^{n^d}} Pr[y = D'_{n^d}]\frac{time_{A'}(y)^{\epsilon}}{n^d}$$

$$= \mathbb{E}[\frac{time_{A'}(D'_{n^d})^{\epsilon}}{n^d}] \leq C.$$

distNP and its complete problems

And now a complete problem... albeit artificial

distNP and its complete problems

And now a complete problem... albeit artificial

Theorem 2

(Existence of a distNP-complete problem) Let U contain all tuples $< M, x, 1^t >$ where there exists a string $y \in \{0,1\}^l$ such that a nondeterministic TM M outputs 1 on input x on t steps.
For every n, we let $\mathcal{U}_n$ be the following distribution on length n tuples $< M, x, 1^t >$: the string representing M is chosen at random from all strings of length at most $\log n$, t is chosen at random in the set $\{0, ..., n - |M|\}$ and x is chosen at random from $\{0,1\}^{n-t-|M|}$.
This distribution is polynomial-time computable.
Then $< U, \mathcal{U} >$ is distNP-complete.

distNP and its complete problems

- L decidable by a $p(n)$-time NDTM M.

distNP and its complete problems

- L decidable by a $p(n)$-time NDTM M.
- f: $x \mapsto < M, x, 1^{p(n)} >$.

distNP and its complete problems

- L decidable by a $p(n)$-time NDTM M.
- f: $x \mapsto < M, x, 1^{p(n)} >$.
- Correctness?

distNP and its complete problems

- L decidable by a $p(n)$-time NDTM M.
- f: $x \mapsto < M, x, 1^{p(n)} >$.
- Correctness? **YES**. The problem is NP-complete!

distNP and its complete problems

- L decidable by a $p(n)$-time NDTM M.
- f: $x \mapsto < M, x, 1^{p(n)} >$.
- Correctness? **YES**. The problem is NP-complete!
- Length regularity?

distNP and its complete problems

- L decidable by a $p(n)$-time NDTM M.
- f: $x \mapsto < M, x, 1^{p(n)} >$.
- Correctness? **YES**. The problem is NP-complete!
- Length regularity? **YES**

distNP and its complete problems

- L decidable by a $p(n)$-time NDTM M.
- f: $x \mapsto < M, x, 1^{p(n)} >$.
- Correctness? **YES**. The problem is NP-complete!
- Length regularity? **YES**
- Domination?

distNP and its complete problems

- L decidable by a $p(n)$-time NDTM M.
- f: $x \mapsto < M, x, 1^{p(n)} >$.
- Correctness? **YES**. The problem is NP-complete!
- Length regularity? **YES**
- Domination? Not necessarily. We have a problem with peaks. What if D has an input of probability much higher than 2^{-n}, whereas $\mathcal{U}$'s probability is at most 2^{-n}.

distNP and its complete problems

- L decidable by a $p(n)$-time NDTM M.
- f: $x \mapsto < M, x, 1^{p(n)} >$.
- Correctness? **YES**. The problem is NP-complete!
- Length regularity? **YES**
- Domination? Not necessarily. We have a problem with peaks. What if D has an input of probability much higher than 2^{-n}, whereas $\mathcal{U}$'s probability is at most 2^{-n}.
- We deal with that with our next lemma

distNP and its complete problems

Lemma (Peak elimination)

Let $D = \{D_n\}$ be a P-computable distribution. Then there is a polynomial-time computable function $g : \{0,1\}^* \to \{0,1\}^*$ such that:
1. g is one-to-one: $g(x) = g(z)$ iff $x = z$.
2. For every $x \in \{0,1\}^*$, $|g(x)| \le |x| + 1$.
3. For every string $y \in \{0,1\}^m$, $Pr[y = g(D_m)] \le 2^{-m+1}$.

Proof:

- For any x of length n, $h(x)$ is the largest common prefix of the binary numbers $\mu_{D_n}(x)$ and $\mu_{D_n}(x-1)$.

Proof:

- For any x of length n, $h(x)$ is the largest common prefix of the binary numbers $\mu_{D_n}(x)$ and $\mu_{D_n}(x-1)$.
- We know that $Pr_{D_n}[x] = \mu_{D_n}(x) - \mu_{D_n}(x-1)$ and let's assume that $Pr_{D_n}[x] \geq 2^{-k}$. Then the difference must be in the first k digits, namely $|h(x)| \leq k$.

Proof:

- For any x of length n, $h(x)$ is the largest common prefix of the binary numbers $\mu_{D_n}(x)$ and $\mu_{D_n}(x-1)$.
- We know that $Pr_{D_n}[x] = \mu_{D_n}(x) - \mu_{D_n}(x-1)$ and let's assume that $Pr_{D_n}[x] \geq 2^{-k}$. Then the difference must be in the first k digits, namely $|h(x)| \leq k$.
- D is P-computable, thus h is computable in polynomial time.

Proof:

- For any x of length n, $h(x)$ is the largest common prefix of the binary numbers $\mu_{D_n}(x)$ and $\mu_{D_n}(x-1)$.
- We know that $Pr_{D_n}[x] = \mu_{D_n}(x) - \mu_{D_n}(x-1)$ and let's assume that $Pr_{D_n}[x] \geq 2^{-k}$. Then the difference must be in the first k digits, namely $|h(x)| \leq k$.
- D is P-computable, thus h is computable in polynomial time.
- h is one-to-one because only two strings can have a specific longest prefix.

distNP and its complete problems

$$g(x) = \begin{cases} 0x & , Prob_{D_n}[x] \leq 2^{-n} \\ 1h(x) & , otherwise \end{cases}$$

Example:

$$Prob_{D_n}[1010] = 0.10000$$

$$Prob_{D_n}[1011] = 0.10101111$$

$$\Rightarrow g(1011) = 1h(1011) = 110.$$

distNP and its complete problems

$$g(x) = \begin{cases} 0x & , Prob_{D_n}[x] \leq 2^{-n} \\ 1h(x) & , otherwise \end{cases}$$

- g in one-to-one.

distNP and its complete problems

$$g(x) = \begin{cases} 0x & , Prob_{D_n}[x] \leq 2^{-n} \\ 1h(x) & , otherwise \end{cases}$$

- g in one-to-one.
- $|g(x)| \leq |x| + 1$.

distNP and its complete problems

$$g(x) = \begin{cases} 0x & , Prob_{D_n}[x] \leq 2^{-n} \\ 1h(x) & , otherwise \end{cases}$$

- g in one-to-one.
- $|g(x)| \leq |x| + 1$.
- If y is not g(x), then $Pr[y = g(D_m)] = 0 \leq 2^{-m+1}$.

distNP and its complete problems

$$g(x) = \begin{cases} 0x & , Prob_{D_n}[x] \leq 2^{-n} \\ 1h(x) & , otherwise \end{cases}$$

- g in one-to-one.
- $|g(x)| \leq |x| + 1$.
- If y is not g(x), then $Pr[y = g(D_m)] = 0 \leq 2^{-m+1}$.
- If $y = 0x$ when $Pr_{D_n}[x] \leq 2^{-|x|}$, then $Pr[y = g(D_m)] \leq 2^{-|y|+1}$.

distNP and its complete problems

$$g(x) = \begin{cases} 0x & , Prob_{D_n}[x] \leq 2^{-n} \\ 1h(x) & , otherwise \end{cases}$$

- g in one-to-one.
- $|g(x)| \leq |x| + 1$.
- If y is not g(x), then $Pr[y = g(D_m)] = 0 \leq 2^{-m+1}$.
- If $y = 0x$ when $Pr_{D_n}[x] \leq 2^{-|x|}$, then $Pr[y = g(D_m)] \leq 2^{-|y|+1}$.
- Let $y = 1h(x)$. $Pr_{D_n}][x] > 2^{-|x|}$ and therefore $|h(x)| \leq |x|$. So $Pr[y = g(D_m)] \leq 2^{-|y|+1}$.

And now we can prove the following theorem

distNP and its complete problems

And now we can prove the following theorem

Theorem 3

(Existence of a distNP-complete problem) Let U contain all tuples $< M, x, 1^t >$ *where there exists a string* $y \in \{0,1\}^l$ *such that a nondeterministic TM M outputs 1 on input x on t steps.*
For every n, we let $\mathcal{U}_n$ *be the following distribution on length n tuples* $< M, x, 1^t >$*: the string representing M is chosen at random from all strings of length at most* $\log n$*, t is chosen at random in the set* $\{0, ..., n - |M|\}$ *and x is chosen at random from* $\{0,1\}^{n-t-|M|}$.
This distribution is polynomial-time computable.
Then $< U, \mathcal{U} >$ *is distNP-complete.*

Theorem Proof:

- Let $< L, D >$ be in distNP and let M be a nondeterministic TM M accepting L.

distNP and its complete problems

Theorem Proof:

- Let $< L, D >$ be in distNP and let M be a nondeterministic TM M accepting L.
- NDTM M': On input y, guess x such that $g(x) = y$ and execute $M(x)$. Let p the polynomial running time of M'.

Theorem Proof:

- Let $< L, D >$ be in distNP and let M be a nondeterministic TM M accepting L.
- NDTM M': On input y, guess x such that $g(x) = y$ and execute $M(x)$. Let p the polynomial running time of M'.
- Reduction: $x \mapsto < M', g(x), 1^k >$, where $k = p(n) + \log n + n - |M'| - |g(x)|$.

distNP and its complete problems

Theorem Proof:

- Let $< L, D >$ be in distNP and let M be a nondeterministic TM M accepting L.
- NDTM M': On input y, guess x such that $g(x) = y$ and execute $M(x)$. Let p the polynomial running time of M'.
- Reduction: $x \mapsto < M', g(x), 1^k >$, where $k = p(n) + \log n + n - |M'| - |g(x)|$.
- Correctness?

distNP and its complete problems

Theorem Proof:

- Let $< L, D >$ be in distNP and let M be a nondeterministic TM M accepting L.
- NDTM M': On input y, guess x such that $g(x) = y$ and execute $M(x)$. Let p the polynomial running time of M'.
- Reduction: $x \mapsto < M', g(x), 1^k >$, where $k = p(n) + \log n + n - |M'| - |g(x)|$.
- Correctness? Yes! g is one-to-one.

Theorem Proof:

- Let $< L, D >$ be in distNP and let M be a nondeterministic TM M accepting L.
- NDTM M': On input y, guess x such that $g(x) = y$ and execute $M(x)$. Let p the polynomial running time of M'.
- Reduction: $x \mapsto < M', g(x), 1^k >$, where $k = p(n) + \log n + n - |M'| - |g(x)|$.
- Correctness? Yes! g is one-to-one.
- Length regularity?

distNP and its complete problems

Theorem Proof:

- Let $< L, D >$ be in distNP and let M be a nondeterministic TM M accepting L.
- NDTM M': On input y, guess x such that $g(x) = y$ and execute $M(x)$. Let p the polynomial running time of M'.
- Reduction: $x \mapsto < M', g(x), 1^k >$, where $k = p(n) + \log n + n - |M'| - |g(x)|$.
- Correctness? Yes! g is one-to-one.
- Length regularity? Yes!

distNP and its complete problems

Theorem Proof:

- Let $< L, D >$ be in distNP and let M be a nondeterministic TM M accepting L.
- NDTM M': On input y, guess x such that $g(x) = y$ and execute $M(x)$. Let p the polynomial running time of M'.
- Reduction: $x \mapsto < M', g(x), 1^k >$, where $k = p(n) + \log n + n - |M'| - |g(x)|$.
- Correctness? Yes! g is one-to-one.
- Length regularity? Yes!
- Domination?

Theorem Proof:

- Let $< L, D >$ be in distNP and let M be a nondeterministic TM M accepting L.
- NDTM M': On input y, guess x such that $g(x) = y$ and execute $M(x)$. Let p the polynomial running time of M'.
- Reduction: $x \mapsto < M', g(x), 1^k >$, where $k = p(n) + \log n + n - |M'| - |g(x)|$.
- Correctness? Yes! g is one-to-one.
- Length regularity? Yes!
- Domination? Yes! A legnth m tuple $< M', y, 1^t >$ is obtained by the reduction with probability at most $2^{-|y|+1}$. This tuple is however obtained by U_m with probability at least $2^{-\log m} 2^{-|y|} \frac{1}{m}$.

Question:

Question: Why not use the polynomially samplable distribution since it is stronger?

sampNP and its complete problems

Question: Why not use the polynomially samplable distribution since it is stronger?

sampNP

A distributional problem $< L, D >$ is in sampNP if $L \in NP$ and D is P-samplable.

sampNP and its complete problems

Question: Why not use the polynomially samplable distribution since it is stronger?

sampNP

A distributional problem $< L, D >$ is in sampNP if $L \in NP$ and D is P-samplable.

Theorem 4

if $< L, D >$ is distNP-complete, then it is also sampNP-complete.

sampNP and its complete problems

Question: Why not use the polynomially samplable distribution since it is stronger?

sampNP

A distributional problem $< L, D >$ is in sampNP if $L \in NP$ and D is P-samplable.

Theorem 4

if $< L, D >$ is distNP-complete, then it is also sampNP-complete.

The choice of P-computable distributions was suitable for average-case completeness results.

Philosophical and Practical Implications

Impagliazzo has considered the following possible scenarios for the world of Complexity:

Philosophical and Practical Implications

Impagliazzo has considered the following possible scenarios for the world of Complexity:

Algorithmica

Philosophical and Practical Implications

Impagliazzo has considered the following possible scenarios for the world of Complexity:

Algorithmica

- P = NP or something equivalent like $NP \subseteq BPP$

Philosophical and Practical Implications

Impagliazzo has considered the following possible scenarios for the world of Complexity:

Algorithmica

- P = NP or something equivalent like $NP \subseteq BPP$
- Computational utopia.

Philosophical and Practical Implications

Impagliazzo has considered the following possible scenarios for the world of Complexity:

Algorithmica

- P = NP or something equivalent like $NP \subseteq BPP$
- Computational utopia.
- Quite an array of tasks could be automated.

Philosophical and Practical Implications

Impagliazzo has considered the following possible scenarios for the world of Complexity:

Algorithmica

- P = NP or something equivalent like $NP \subseteq BPP$
- Computational utopia.
- Quite an array of tasks could be automated.
- All the current cryptographic applications will break down.

Philosophical and Practical Implications

Heuristica

Philosophical and Practical Implications

Heuristica

- We have $P \neq NP$ and yet $distNP \subseteq distP$.

Philosophical and Practical Implications

Heuristica

- We have $P \neq NP$ and yet $distNP \subseteq distP$.
- We have efficient algorithms that "almost" solves every NP problem.

Philosophical and Practical Implications

Heuristica

- We have $P \neq NP$ and yet $distNP \subseteq distP$.
- We have efficient algorithms that "almost" solves every NP problem.
- The kinds of inputs that fail are very rare to find in practice.

Philosophical and Practical Implications

Heuristica

- We have $P \neq NP$ and yet $distNP \subseteq distP$.
- We have efficient algorithms that "almost" solves every NP problem.
- The kinds of inputs that fail are very rare to find in practice.
- Very similar to Algorithmica. Many NP optimization problems solved in practice.

Philosophical and Practical Implications

Heuristica

- We have $P \neq NP$ and yet $distNP \subseteq distP$.
- We have efficient algorithms that "almost" solves every NP problem.
- The kinds of inputs that fail are very rare to find in practice.
- Very similar to Algorithmica. Many NP optimization problems solved in practice.
- Many cryptographic applications break down.

Pessiland

Philosophical and Practical Implications

Pessiland

- distNP is not in distP and there do not exist any one-way functions.

Pessiland

- distNP is not in distP and there do not exist any one-way functions.
- Worst possible world!

Philosophical and Practical Implications

Pessiland

- distNP is not in distP and there do not exist any one-way functions.
- Worst possible world!
- No wonderous results and no cryptography.

Philosophical and Practical Implications

Pessiland

- distNP is not in distP and there do not exist any one-way functions.
- Worst possible world!
- No wonderous results and no cryptography.

Minicrypt

Philosophical and Practical Implications

Pessiland

- distNP is not in distP and there do not exist any one-way functions.
- Worst possible world!
- No wonderous results and no cryptography.

Minicrypt

- One-way functions exist (and therefore distNP not in distP) but all the highly structured problems in NP such as integer factorization are solvable in polynomial time.

Philosophical and Practical Implications

Pessiland

- distNP is not in distP and there do not exist any one-way functions.
- Worst possible world!
- No wonderous results and no cryptography.

Minicrypt

- One-way functions exist (and therefore distNP not in distP) but all the highly structured problems in NP such as integer factorization are solvable in polynomial time.
- Although one-way functions exist, there are no public key encryption schemes or key exchange protocols.

Philosophical and Practical Implications

Pessiland

- distNP is not in distP and there do not exist any one-way functions.
- Worst possible world!
- No wonderous results and no cryptography.

Minicrypt

- One-way functions exist (and therefore distNP not in distP) but all the highly structured problems in NP such as integer factorization are solvable in polynomial time.
- Although one-way functions exist, there are no public key encryption schemes or key exchange protocols.
- Those cryptographic applications achievalbe only by one-way functions such as private key encryption and pseudorandom generators still work.

Philosophical and Practical Implications

Cryptomania

Philosophical and Practical Implications

Cryptomania

- The problem on factoring large integers is exponentially hard on average.

Philosophical and Practical Implications

Cryptomania

- The problem on factoring large integers is exponentially hard on average.
- Most believe that this is the world we live in!

Philosophical and Practical Implications

Cryptomania

- The problem on factoring large integers is exponentially hard on average.
- Most believe that this is the world we live in!
- We don't have general-purpose algorithms and we have to resort to heuristics, approximation, creativity and hard work.

Philosophical and Practical Implications

Cryptomania

- The problem on factoring large integers is exponentially hard on average.
- Most believe that this is the world we live in!
- We don't have general-purpose algorithms and we have to resort to heuristics, approximation, creativity and hard work.
- We have a host of exciting cryptographic applications.