| | |
|---|---|
| **Average Case Complexity** | March 16, 2011 |

Impagliazzo's Five Worlds

| | |
|---|---|
| Professor: Valentine Kabanets | Scribe: Shawn Andrews |

# The Five Worlds of Impagliazzo

(based on "A Personal View of Average Case Complexity" by R. Impagliazzo, 1995)

## 1 Algorithmica:

$$P = NP$$
$$(\text{or } NP \subseteq BPP)$$

- "Paradise"
- AI rules
- No secrets
- Cryptography is dead

$P = NP \Rightarrow$ we can efficiently invert any poly-time function $f : \{0,1\}^{\ell(n)} \rightarrow \{0,1\}^n$. Given $y \in \{0,1\}^n$, non-deterministically guess an $x \in \{0,1\}^{\ell(n)}$ and check if $f(x) = y$, if so, output $x$. Since $P = NP$, this search can be done in polytime.

It is impossible to have cryptography without a one-way function (OWF).

**Oracle:** $\exists$ an oracle $O$ s.t. $NP^O = P^O$ [Baker, Bill, Solovay, '75].

## 2 Heuristica:

$$NP \neq P \text{ but } (NP, U) \subseteq \text{AvgP}$$
$$NP \nsubseteq BPP \text{ but } (NP, U) \subseteq \text{HeurBPP}$$

"Paradox": $\exists$ hard instances of NP-Hard problems, but they are hard to find.

**VLSI minimization:** Given a circuit specification $C(x_1, \ldots, x_n)$ to compute some $f : \{0,1\}^n \rightarrow \{0,1\}$, we want the minimum sized circuit $C_{min}$ s.t.

$$\forall x \in \{0,1\}^n,\ C_{min}(x) = C(x)\ .$$

In Algorithmica, we can define the predicate:

$$Check(\tilde{C}) = \begin{cases} T \text{ if } \underbrace{\forall x\ \tilde{C}(x) = C(x)}_{\in coNP = P} \\ \\ F \text{ otherwise} \end{cases}$$

Then we find the smallest $\tilde{C}$ s.t. $Check(\tilde{C})$, which is an NPSearch problem, and thus can be done in polynomial time. This approach doesn't work in Heuristica.

**Open Question:** Suppose $(NP, U) \subseteq \text{AvgP}$. Is then $(PH, U) \subseteq \text{AvgP}$? (cf. $P = NP \Rightarrow P = PH$).

**Claim 1.** $(NP, U) \in$ HEURBPP $\Rightarrow \nexists$ OWF

Given a candidate polytime function $f : \{0,1\}^{\ell(n)} \to \{0,1\}^n$, define

$$L = \{y | \exists x \; f(x) = y\}$$

and define $D = \{D_n\} \in$ PSAMP, $D_n$ is sampled: "Given a random $x$, output $f(x)$".

By [Impagliazzo, Levin], $(NP, U) \in$ HEURBPP $\Rightarrow$ Can solve any $(NP, \text{PSAMP})$ search problem including $(L, D)$

## 3 Pessiland:

$$(NP, \text{PSAMP}) \not\subseteq \text{HEURBPP}\text{ , but } \nexists \text{OWF}$$

Is it true that $(NP, \text{PSAMP}) \not\subseteq$ HEURBPP $\Rightarrow \exists$ OWF? We know:

$$(NP, \text{PSAMP}) \not\subseteq \text{HEURBPP} \Rightarrow \exists (BH, U) \text{ s.t. } \forall\, BPP \text{ type algorithm } A,$$
$$\exists\, \delta \text{ s.t. } A \text{ fails on } (BH, U) \text{ on } \geq \delta \text{ fraction of inputs.}$$

Thus we can easily generate/sample hard instances for any $A$.

The existence of a OWF is equivalent to being able to generate hard "solved instances": suppose $f : \{0,1\}^{\ell(n)} \to \{0,1\}^n$ is a OWF, then generating algorithm:

---

1: Pick a random $x \in \{0,1\}^{\ell(n)}$
2: Compute $b = f(x) \in \{0,1\}^n$
3: Create a SAT formula $\psi(z_1, \ldots, z_{\ell(n)}) \equiv \left[f(z_1, \ldots, z_{\ell(n)}) = b\right]$ (can be done by Cook-Levin)
4: Output $(x, \psi)$

---

Note $\psi(x) = T$. For a random $x$, $\psi$ must be hard to find witnesses for, otherwise we get a pre-image of $f(x)$.

Suppose we have a generating algorithm $S : \{0,1\}^{r(n)} \to \{(x, \psi) \mid \psi(x) = T\}$ and it's hard to find a witness for $\psi$ for random input to $S$. Define $f : \{0,1\}^{r(n)} \to \{0,1\}^{poly(n)}$, $f(z) = S(z)|_2$, the formula part of $S$'s output. $f$ is a OWF. If $S(z)|_2 = S(z')|_2$, $S(z')|_1$ is a satisfying assignment for $f(z)$.

If $\nexists$ OWF ($\exists$ a generic inverter for any polytime computable function)

- We can learn efficiently the behaviour of an unknown algorithm by observing its input/output behaviour on some sample distribution.
- We can use randomized compression to compress samplable distributions.
- More?

How to know we are in Pessiland: $(NP, U) \not\subseteq$ HEURBPP, but we can invert any polytime function, $f : \{0,1\}^* \to \{0,1\}^*$.

Levin: there is a complete OWF $f_{Levin}$ s.t.

$$\exists \text{ OWF} \Leftrightarrow f_{Levin} \text{ is a OWF .}$$

## 4 Minicrypt

$\exists$ OWF, but no public key cryptography (cannot agree on a secret with a stranger using only public channels) $\Rightarrow$ private-key cryptography (PRG, digital signatures, zero-knowledge, etc.)

Oracle: [Impagliazzo, Rudich]

## 5 Cryptomania:

Public-key crypto is possible. All sorts of privacy is possible (via math), but maybe not by the government.

- Secure e-voting
- Joint computation of secret inputs without revealing the secrets
- ...

Oracle: [Brassard]

## Levin's Universal Search Algorithm

**Theorem 1.** *Suppose $\exists$ $t(n)$-time algorithm $A$ for SAT. Then the following algorithm will solve SAT in time $O(t(n))$: More precisely, time $2^{|A|+1} \cdot t(n)$, exponential in the size of $A$.*

**Algorithm 1** Levin-Search

On input $\psi(x_1, \ldots, x_n)$
**for** $i = 1$ to $\infty$ **do**
    Run each TM $M$ of size $|M| \leq i$ for $\leq 2^{i-|M|}$ steps
    **if** any $M$ halted with a satisfying assignment $\alpha$ **then**
        **return** $\alpha$
    **end if**
**end for**

*Proof.* Consider prefix free encodings of TM's. Calculate the time up to and including stage $i$: A TM $M$ of size $|M|$ is run for:

$$\sum_{j=1}^{i} 2^{j-|M|} = 2^{i+1-|M|} \text{ steps}$$

Over all TM's, $|M| \leq i$, the time taken is

$$\begin{aligned} \sum_{M:|M|\leq i} 2^{i+1-|M|} &= 2^{i+1} \sum_{M:|M|\leq i} 2^{-|M|} \\ &\leq 2^{i+1} \quad \text{by Kraft} \end{aligned}$$

At stage $i$, $M$ is run for $2^{i-|M|}$ steps, so the algorithm terminates at stage $i_0$ s.t.

$$2^{i_0-|A|} = t \Rightarrow i_0 = \log(t) + |A|$$

Thus the overall time for stage $i_0$ is $\leq 2^{i_0+1} = t \cdot 2^{|A|+1}$.

□

## Levin's OWF-Complete Function

Recall that a polytime function $f : \{0,1\}^* \to \{0,1\}^*$ is called *weakly one-way function (OWF)* if there is some constant $c > 0$ such that for every polytime algorithm $A$, $\mathbb{P}_x[A(f(x)) \notin f^{-1}(x)] > |x|^{-c}$, i.e., every efficient inverter fails to invert $f(x)$ on at least $n^{-c}$ fraction of uniformly random inputs $x$ of length $n$. (A strongly OWF is a polytime function $f$ such that, for every $c > 0$, every polytime inverter fails at inverting $f(x)$ on at least $1 - n^{-c}$ fraction of inputs $x$ of length $n$. In the later lecture, we show that weakly OWFs can be used to construct strongly OWFs.)

Here we show that there is a particular polytime function such that it is a weakly OWF if any weakly OWF exists. So in a sense, this is a function "complete for one-wayness". This function was defined by Levin as follows.

$f_{Levin}(y)$: "Interpret $y = \langle M, x\rangle$ where $|M| \leq \log|x|$. Run $M$ on $x$ for $\leq |y|^3$ steps. If $M$ terminates, output $\langle M, M(x)\rangle$. Otherwise, output $\perp$."

**Claim 2.** *If there exists a OWF, then $f_{Levin}$ is a weakly* OWF*.*

*Proof.* Suppose $g$ is a (weakly) OWF. Without loss of generality, $g$ is computable in time $O(n^2)$ on inputs of size $n$. (If $g$ takes time $n^c$, define a new function $g'(a,x) = (a, g(x))$, where $|a| = |x|^c$. Then $g'$ is computable in linear time. Also, $g'$ is OWF if $g$ is. Suppose not. Let $I$ be an inverter for $g'$ that succeeds on at least $p$ fraction of random inputs $a, x$ to $g'$. We'll use $I$ to invert $g$: "Given $y = g(x)$ (for a random $x$), pick random $a$ (of appropriate length), and run $I(a, y)$." Then this algorithm inverts $g(x)$ with probability at least $p$, where the probability is over uniform input $x$, as well as the internal randomness of the algorithm.)

Let $y = \langle M_g, x\rangle$ for random $x$, then $f_{Levin}(y) = \langle M_g, g(x)\rangle$. We know any polytime algorithm inverting $g$ fails with probability $> \frac{1}{|x|^c}$ for some $c$, given $|x|$ is large enough. Thus any inverter for $f_{Levin}$ must fail on $y$ with probability $> \frac{1}{|x|^c}$. But inputs to $f_{Levin}$ of the form $\langle M_g, x\rangle$ have density $\geq 2^{-|M_g|} \sim \frac{1}{|x|}$, so any inverter for $f_{Levin}$ must fail on a random input with probability $> \frac{1}{|x|^{c+1}}$. $\square$