On Worst-Case to Average-Case Reductions for NP Problems

Sai Kishan Pampana, Sarthak Garg, Drishti Wali

Indian Institute of Technology Kanpur

April 13, 2016

Motivation

- The author tries to find whether the existence of problems in NP which have no polytime heuristic algorithm can be related to the NP $\subseteq$ BPP question.
- Whether $\exists$ a reduction R that converts a heuristic polytime algorithm for an NP-Complete problem (or inverting a one-way function) into a BPP algorithm for NP

Acknowledgement

This is a presentation that is made from the content in the paper[On Worst-Case to Average-Case Reductions for NP Problems]

Concepts and terms used in Paper

- **Distributional NP** A problem in this class consists on the ordered pair (L, D), where L is a NP problem and D is a sample distribution.
- **Intractable Problems** A problem (L ,D) in Distribution NP is intractable if on every Poly Time algo, we fail with a probability of at least $\frac{1}{p(n)}$ when input is of length n.
  Note - Saying that a problem is intractable is equivalent to saying that we have no poly time heuristic algorithm

Preliminaries

- **Locally Random Reductions** An LRR from L to (L', D) is a PPTM R such that $R^{L'}$ solves L and each oracle query of $R^{L'}$ is distributed according to D
- **Smooth Reductions** An SMR from L to (L', D) is same as an LRR except that the oracle queries can be distributed according to any smooth distribution ie Pr[x $\in \{0,1\}^{n'}$ is queried] $\leq \frac{c}{2^{n'}}$
- **Worst Case to Average Case Reductions** A WCAC(δ) reduction is similiar to a LRR except two differences. First, instead to having an oracle to L', we have an any oracle A that agrees with L' on atleast δ fraction of the inputs. Secondly there is no restriction on how the queries should be distributed.

  In this paper we consider all the reductions to be non-adaptive in nature.

Preliminaries

- **Non Uniform AM protocol** AM^{poly} is a class of languages for which there exists an AM protocol which is non uniform and access to an oracle that gives it an polynomial length "advice".
- If coNP $\subseteq AM^{poly}$ then $\Sigma^3 = \Pi^3$
- **FF Protocol** Fortnow and Feigenbaum showed that if $\exists$ an LRR from an 3SAT to a problem (L, D) $\in$ Dist-NP then the polynomial Heirarchy collapes. They do this by showing a non uniform coAM protocol for L
- **Paper's result** The paper shows how to generalize the above result for WCAC reductions. It also gives a non-uniform coAM protocol for L

FF protocol

There is an LRR $R^{L'}$ from L $\in$ NP to (L', D) *in* Dist-NP. The following is an AM^{poly} protocol for $L^{\complement}$

- **Verifier** Generates k independant computations of $R^{L'}$ each making q queries ($\in \{0,1\}^{n'}$)(without loss of generality). It sends to prover all the kq queries and asks for certificates of the YES instances.
- **Prover** Provides answers for all the queries and certificates for all the YES answers
- **Verifier** Has non uniformly access to fraction p of YES instances in $\{0,1\}^{n'}$. If the prover provides less than kqp - $0(q\sqrt[2]{k})$ then REJECT. If any of the certificates are wrong then REJECT. If any computation of $R^{L'}$ ACCEPTS, then REJECT. Else ACCEPT

Upper Bound and Lower bound protocols

We will use these 2 protocols in the final protocol.

- **Set Lower Bound Protocol** Given an NP set S $\subseteq \{0,1\}^n$ and a bound s. Then by the Goldwasser Sipser protocol, $\exists$ an AM protocol st
- If $|S| \geq s$, there exists a prover that makes the verfier accept with probability $1 - \frac{9}{\epsilon^2 k}$. If $|S| \leq (1-\epsilon)s$, no prover makes the verifier accept with probability more than $\frac{9}{\epsilon^2 k}$

Upper Bound and Lower bound protocols

- **Set Upper Bound Protocol** Given an NP set S $\subseteq \{0,1\}^n$ and a bound s, if the verfier has access to a secret "r", chosen uniformly at random from the set S, then due to Aiello and Hastad, $\exists$ an AM protocol st
- If $|S| \leq s$, there exists a prover that makes the verifier accept with probability $1 - \frac{9}{\epsilon^2 k}$. If $|S| \geq (1+\epsilon)s$, no prover makes the verifier accept with probability more than $1 + \frac{9}{\epsilon^2 k} - \frac{\epsilon}{6}$

Handling Smooth Reductions

The FF protocol can be used, if given x, we can get a good estimate of the average number of oracle queries of $R^{L'}(x)$ that are answered YES. We will describe a 3 phase protocol for the main theorem. The second phase deals with the task of estimating the above fraction.(q^*)

Handling general reductions

Let R^A be a WCAC(δ) from L to (L', D'). We construct a smooth reduction roughly according to the following idea.

- Fix a threshhold $t = \frac{q}{\delta}$ and then for every query i of length m made by R(x), the verifier asks the prover for the probability that the query can be generated by R(x)(say p_i).
- Partition the queries into 2 parts, Heavy ($p_i \geq \frac{t}{2^m}$)(Verified using the set lower bound protocol) and light ($p_i \leq \frac{t}{2^m}$) (Verified by set upper bound protocol). Ask light queries to the oracle, but proceed as if the heavy ones had been answered NO. Let this modified procedure be R'. Then $R'^{L'}(x)$ behaves as $R^A(x)$. R' is smooth by construction. Let the fraction of queries that are answered incorrectly be p'. Then $p'.2^{m'}.\frac{t}{2^m} \leq 1 \Rightarrow p' \leq \frac{\delta}{q} \leq \delta$

Handling general reductions

The idea in the previous slide is implemented in 2 phases, the first and the third phase.

- **First phase** The set upper bounds and lower bounds are able to estimate the probability within some gap of error. If there are a large number of queries present in this gap then the reduction from WCAC to SMR would fail. To avoid this the verifier chooses the threshhold randomly. Then the verfier starts a protocol that finds the fraction of light queries (p^*).
- **Third Phase** The third phases uses p^* to estimate whether a given query is heavy or light. It obtains the fraction q^* from the second phase. It then runs a modified FF protocol

References

Andrej Bogdanov, Luca Trevisan

The End