Electronic Colloquium on Computational Complexity, Report No. 88 (2018)

A story of AM and UNIQUE-SAT

Ilya Volkovich *

Abstract

In the seminal work of [Bab85], Babai have introduced *Arthur-Merlin Protocols* and in particular the complexity classes MA and AM as randomized extensions of the class NP. While it is easy to see that NP $\subseteq$ MA $\subseteq$ AM, it has been a long standing open question whether these classes are actually different. In [BGM06], Böhler et al. introduced the probabilistic class of SBP and showed that MA $\subseteq$ SBP $\subseteq$ AM. Indeed, this is the only known natural complexity class that lies between MA and AM. In this work we show that the class AM collapses to SBP if UNIQUE-SAT is NP-hard, where UNIQUE-SAT stands for the problem of deciding whether a Boolean formula has a unique satisfying assignment or none.

1 Introduction

For more than three decades, the question of whether the classes MA, NP and AM are different remains open. While AM and MA are conjectured to be equal and, moreover, both are conjectured to collapse to NP, there has been only a mild advancement on this front. In particular, Arvind et al. [AKSS95] showed that AM = MA if NP $\subseteq$ P/poly. Yet, the same premises imply a collapse of the Polynomial Hierarchy (see e.g. [KL80]) and hence are not *believed* to be true.

A different line of work has been involved with the study of the computational power of *unambiguous* polynomial-time machines, captured by the complexity class UP. This class was introduced by Valiant [Val76] and it consists of NP problems with unique solutions. A typical example of such a problem is the problem of UNIQUE-SAT. As in the standard SAT problem, we are given a Boolean formula φ as input, and we need to determine if φ is satisfiable. Yet, in UNIQUE-SAT we are given an addition promise that φ is either unsatisfiable or contains **exactly** one satisfying assignment. Indeed, UP should be thought of a class of "promise problems" (see Definition 2.1 for more details).

It is easy to see that UNIQUE-SAT $\in$ NP. Conversely, Valiant & Vazirani [VV86] have shown that UNIQUE-SAT is NP-hard under randomized reductions. That is, NP $\subseteq$ $\mathsf{RP}^{\text{UNIQUE-SAT}}$. Nonetheless, whether UNIQUE-SAT is NP-hard under (the regular notion of) many-to-one reduction, remains an open question. Blass & Gurevich [BG82] have shown an oracle relative to which UNIQUE-SAT is not NP-hard under many-to-one reductions, implying that this kind of hardness should be obtained by a non-relativizing techniques.

In [BGM06], Böhler et al. introduced the class of SBP, which stands for *small bounded-error probability*. They also showed that SBP lies between MA and AM. To the best of our knowledge, SBP is the only natural class with this property. Moreover, the only known conditional collapse results of either AM to SBP or SBP to MA are actually those that collapse AM all the way to MA (for example, see above). Our main result links the aforementioned lines of work:

*Department of EECS, CSE Division, University of Michigan, Ann Arbor, MI. Email: ilyavol@umich.edu.

ISSN 1433-8092

Theorem 1. *Suppose* UNIQUE-SAT *is* NP*-hard under many-to-one reductions. Then* AM = SBP.

We also show some technical extension of the result. For more details see Section 3.1. We would now like to elaborate on the context of the result.

In [Ver92], Vereshchagin have shown that while MA ⊆ PP for every oracle, there exists an oracle that separates AM from PP. In [BGM06], Böhler et al. have extended this inclusion to MA ⊆ SBP ⊆ PP ∩ AM. As a corollary, they have concluded that the same oracle separates SBP from AM and PP. Furthermore, it is an easy exercise to see that AM is closed under union and intersection. Yet, whether SBP is closed under intersection remains an open question. In [GLM+16], Göös et al. have shown an oracle relative to which SBP is not closed under intersection. In conclusion, the collapse of AM to SBP should evade numerous relativization barriers. As was mentioned earlier, Blass & Gurevich [BG82] have shown an oracle relative to which UNIQUE-SAT is not NP-hard under many-to-one reductions, indicating that our assumption, indeed, evades these barriers.

1.1 Ideas and Techniques

We show the collapse by identifying complete sets of AM and SBP[1]. In [GS86], Goldwasser & Sipser considered the problem of determining whether a set S is of size at least m or at most $m/2$, where membership of x in S can be efficiently determined given a witness w. They show an AM protocol for the problem.

Our first observation is that this problem is, in fact, hard for the class AM. In conclusion, we obtain a natural AM-complete problem - WSSE (see Definition 2.4 for more details). Next, we observe that the class SBP corresponds to a simpler version of the problem - SSE. As before, we would like to determine whether a set S is of size at least m or at most $m/2$. Yet, in this version of the problem, the membership of x can be efficiently determined given (just) x (see Definition 2.6 for more details).

In what follows, we show a polynomial-time reduction from WSSE to SSE, under the assumption that UNIQUE-SAT is NP-hard. The natural approach would be to regard the set S as a set of tuples (x, w) such that w is a witness for membership of x in S. By definition, each $x \in S$ has at least one witness w associated with it. Yet, the actual number of such witnesses could be arbitrary. To illustrate this, consider the following two sets: S_1 contains only one element x_1 with $K \gg 2$ witnesses of membership; S_2 contains two elements element e_1, e_2 with 1 witnesses of membership each. Suppose $m = 2$. Viewing S_1 and S_2 as above, we obtain sets with K and 2 elements, respectively. One approach to over to over this issue would be to count the actually number of witness. Yet this turns out to be a #P-hard problem. Instead, we will use utilize the NP-hardness of UNIQUE-SAT to prevent the inversion.

We will operate by "flipping the chair". Fix x and consider the set W_x of witnesses associated with x. By definition, if $x \notin S$ then $W_x = \emptyset$; otherwise, if $x \in S$ then $|W_x| \geq 1$. Moreover, observe that the membership of w in W_x can be efficiently determined given (just) w. We will apply the mapping f, from the NP-hardness reduction of UNIQUE-SAT, on W_x. Observe that in the former case, $f(W_x) = \emptyset$ and in the latter case $|f(W_x)| = 1$. In other words, every $x \in S$ will have a unique witness w of its membership in S.

[1] Technically, we are looking at the promise versions of AM and SBP.

2 Preliminaries

For a unary relation $R(x)$, we define $\#_x R \stackrel{\Delta}{=} |\{x \mid x \in R\}|$. For a binary relation $R(x,w)$, we define $\exists_w \#_x R \stackrel{\Delta}{=} |\{x \mid \exists w \text{ s.t. } (x,w) \in R\}|$. For technical reasons will we need to consider promise problems. A *promise problem* is a relaxation of a language. Formally:

Definition 2.1 (Promise Problems)**.** $\Pi = (\Pi_{YES}, \Pi_{NO})$ *is a* promise problem *if* $\Pi_{YES} \cap \Pi_{NO} = \emptyset$.

We will be mostly concerned with the two following complexity classes. We refer the reader to [AB09] for the definitions of other standard complexity classes.

Definition 2.2 ([Bab85])**.** *A language L is in* AM *if there exists a polynomial-time computable predicate $A(x,r,w)$ such that:*

$$\begin{aligned} x \in L &\implies \Pr_r[\exists w : A(x,r,w) = 1] \geq 2/3 \\ x \notin L &\implies \Pr_r[\exists w : A(x,r,w) = 1] \leq 1/3. \end{aligned}$$

Definition 2.3 ([BGM06])**.** *A language L is in* SBP *if there exists $\varepsilon > 0, k \in \mathbb{N}$ and a polynomial-time computable predicate $B(x,r)$ such that:*

$$\begin{aligned} x \in L &\implies \Pr_r[B(x,r)) = 1] \geq (1+\varepsilon) \cdot \tfrac{1}{2^{n^k}} \\ x \notin L &\implies \Pr_r[B(x,r)) = 1] \leq (1-\varepsilon) \cdot \tfrac{1}{2^{n^k}}. \end{aligned}$$

where $n = |x|$.

In [GS86], Goldwasser & Sipser consider the problem of determining whether a set S is of size at least m or at most $m/2$, where membership of x can be efficiently determined given x and witness w. Formally, we define the following promise problem.

Definition 2.4. *Witnessed Set-Size Estimation.* $\text{WSSE} = (\text{WSSE}_{YES}, \text{WSSE}_{NO})$ *where,*
$\text{WSSE}_{YES} = \{(C,m) \mid \exists_w \#_x C \geq m\}, \text{WSSE}_{NO} = \{(C,m) \mid \exists_w \#_x C \leq m/2\}$.
Here $C(x,w)$ is a Boolean circuit and m is an integer given in binary representation.

In the same paper, an AM protocol for the problem was given. In other words, it was shown that WSSE $\in$ PromiseAM. We begin by observing that WSSE is also hard for the class AM. Recall Definition 2.2. Let $L \in$ AM and suppose $r \in \{0,1\}^\ell$. Furthermore, set $A_x(r,w) \stackrel{\Delta}{=} A(x,r,w)$ and $m = 2^{\ell+1}/3$. We observe that:

$$\begin{aligned} x \in L &\implies \exists_w \#_r A_x \geq m \\ x \notin L &\implies \exists_w \#_r A_x \leq m/2. \end{aligned}$$

Corollary 2.5. WSSE *is* PromiseAM*-complete.*

In this paper, we also study a simpler version of the problem. As before, we would like to determine whether a set S is of size at least m or at most $m/2$. Yet, in this version of the problem, the membership of x can be efficiently determined given (just) x. Formally, we define the following promise problem.

Definition 2.6. *Set-Size Estimation.* $\text{SSE} = (\text{SSE}_{YES}, \text{SSE}_{NO})$ *where,*
$\text{SSE}_{YES} = \{(C,m) \mid \#_x C \geq m\}, \text{SSE}_{NO} = \{(C,m) \mid \#_x C \leq m/2\}$.
Here $C(x)$ is a Boolean circuit and m is an integer given in binary representation.

Lemma 2.7 (Implicit in [BGM06])**.** SSE *is* PromiseSBP*-complete.*

Fact 2.8. *There exists a polynomial-time algorithm that given a program P, that runs in time t, converts it into a circuit of size* $\text{poly}(t)$ *with the same functionality.*

3 Relation to UNIQUE-SAT

In this section we prove our main result and discuss some extension. We begin by formally introducing the problem of UNIQUE-SAT.

Definition 3.1. UNIQUE-SAT $= (\text{UNIQUE-SAT}_{YES}, \text{UNIQUE-SAT}_{NO})$ *where,*
$\text{UNIQUE-SAT}_{YES} = \{\varphi \mid \#_x\varphi = 0\}, \text{UNIQUE-SAT}_{YES} = \{\varphi \mid \#_x\varphi = 1\}$.
Here φ is Boolean formula.

The next lemma is our main technical contribution. The proof implements the idea outlined in Section 1.1. In particular, if UNIQUE-SAT is NP-hard, it can be reduced from Circuit-SAT. That is, the problem of decided whether a given Boolean circuit C is satisfiable. Let f be the corresponding mapping reduction and suppose f can be computed in time $|C|^k$.

Before moving to the proof, we would like to address a technical issue. Given two circuits C_1, C_2 of the same size s, their corresponding images $\hat{\varphi}_1 = f(C_1)$ and $\hat{\varphi}_2 = f(C_2)$ could, potentially, depend on a different number of variables $t_1 \neq t_2$, respectively. Clearly, $t_1, t_2 \leq s^k$ and hence we can regard both $\hat{\varphi}_1$ and $\hat{\varphi}_2$ as circuits defined on s^k variables. While this would not affect the satisfiability of $\hat{\varphi}_1$ and $\hat{\varphi}_2$, it might affect the uniqueness property. For example, while the formula $\hat{\varphi}_1(x) = \neg x$ has a unique satisfying assignment, regarding $\hat{\varphi}_1$ as $\hat{\varphi}_1(x, y, z) = \neg x$ introduces additional satisfying assignments. We resolve this issue by connecting the remaining variables to the circuit via the $\wedge$ connector. Going back to our example, observe that $\hat{\varphi}_1(x) \wedge y \wedge z = \neg x \wedge y \wedge z$ has (again) a unique satisfying assignment.

Lemma 3.2. *Suppose* UNIQUE-SAT *is* NP*-hard. Then there exist an algorithm that given a Boolean circuit $C(x, w)$ outputs (another) Boolean circuit $\hat{C}(x, z)$ such that $\#_{(x,z)}\hat{C} = \exists_w \#_x C$, in time* $\text{poly}(|C|)$.

Proof. By the assumption, there exists $k \in \mathbb{N}$ and a mapping $\hat{\varphi}(u) = f(C)$, computable in time $|C|^k$ satisfying:

$$\begin{aligned} \#_x C > 1 &\implies \#_u\hat{\varphi} = 1 \\ \#_x C = 0 &\implies \#_u\hat{\varphi} = 0. \end{aligned}$$

Consider the following program P.

Input: Description of a Boolean circuit $\langle C(x, w)\rangle$
Boolean vector x
Boolean variables $z_1, \ldots, z_{|C|^k}$

1 $C'(w) \leftarrow C(x, w)$; `/* plugging in the value of x */`
2 $\hat{\varphi}(u_1, \ldots, u_t) \leftarrow f(C')$;
3 **return** $\hat{\varphi}(z_1, \ldots, z_t) \wedge z_{t+1} \wedge \ldots \wedge z_{|C|^k}$ `/* By definition,` $t \leq |C|^k$ `*/`

Let $\hat{C}(x, z)$ denote the circuit that results from converting the program P into a Boolean circuit (applying Fact 2.8) having $\langle C(x, w)\rangle$ hard-coded. Suppose x is such that there exists a witness w for which $C(x, w) = 1$. We show that for each such, and only such, x there exists a unique z satisfying $C(x, z) = 1$.
By definition, $\#_w C' > 1$. Therefore, by the properties of f, there exists a unique assignment $u = (u_1, \ldots, u_t)$ such that $\hat{\varphi}(u) = 1$. Consequently, $C(x, z) = 1$ iff $z = (u_1, \ldots, u_t, 1, \ldots, 1)$.
Conversely, suppose x is such that $\forall w : C(x, w) = 0$. By definition, $\#_w C' = 0$. Again, by the properties of f, $\#_u\hat{\varphi} = 0$ and consequently, $\forall z : C(x, z) = 0$. □

We now prove our main theorem. We restate it for completeness.

Theorem 3.3 (Theorem 1 restated)**.** *Suppose* UNIQUE-SAT *is* NP*-hard. Then* AM = SBP*.*

Proof. We show that AM $\subseteq$ SBP by showing that WSSE $\leq_p$ SSE. We map an instance $(C(x,w),m)$ of WSSE to $\left(\hat{C}(x,z),m\right)$ via the procedure described in Lemma 3.2. The claim regarding the runtime and the correctness follow from Lemma 3.2. □

3.1 Extensions

In this section we discuss some extension to the main result. First, we observe that φ is not required to be a formula. We can extend the result further to the case when there exists a mapping $\hat{\varphi}(u) = f(C)$ computable polynomial-time satisfying:

$$\begin{aligned} \#_x C > 1 \quad &\implies \#_u \hat{\varphi} = \alpha \\ \#_x C = 0 \quad &\implies \#_u \hat{\varphi} = 0, \end{aligned}$$

for some universal constant $\alpha \in \mathbb{N}^+$. By repeating the previous arguments, we can show that in this case $\#_{(x,z)}\hat{C} = \exists_w \#_x C \cdot v$. Hence, in this case, map an instance $(C(x,w),m)$ to $\left(\hat{C}(x,z), m \cdot \alpha\right)$. We show that we can handle a certain kind of randomized algorithms, where we have a bound in expectation.

Lemma 3.4. *Suppose there exists a universal constant $\alpha \in \mathbb{R}^+$ and a mapping $\hat{\varphi}(u) = f(C)$ computable in randomized polynomial time satisfying:*

$$\begin{aligned} \#_x C > 1 \quad &\implies \mathrm{EX}[\#_u \hat{\varphi}] = \alpha \\ \#_x C = 0 \quad &\implies \mathrm{EX}[\#_u \hat{\varphi}] = 0. \end{aligned}$$

Then there exists a mapping $\hat{C} = \hat{f}(C)$ computable in deterministic polynomial time such that $\#_{(x,z)}\hat{C} = \exists_w \#_x C \cdot \alpha \cdot 2^\ell$, when $\ell = \mathrm{poly}(|C|)$.

Proof. We regard the mapping f as $f(C;\tau)$, when $\tau \in \{0,1\}^\ell$ is treated as a random string. Let $C(x)$ be a Boolean circuit. Consider the following program P.

Input: Description of a Boolean circuit $\langle C(x) \rangle$
Boolean variables $z_1, \ldots, z_{|C|^k}$ and $\tau_1, \ldots, \tau_\ell$

1 $\hat{\varphi}(u_1, \ldots, u_t) \leftarrow f(C;\tau)$;
2 **return** $\hat{\varphi}(u) \wedge z_{t+1} \wedge \ldots \wedge z_{|C|^k}$;

Let $\hat{C}'(u,\tau)$ denote the circuit that results by converting the program P into a Boolean circuit having $\langle C(x) \rangle$ hard-coded. Observe that

$$\begin{aligned} \#_x C > 1 \quad &\implies \#_{(z,\tau)}\hat{C}' = \alpha \cdot 2^\ell \\ \#_x C = 0 \quad &\implies \#_{(z,\tau)}\hat{C}' = 0. \end{aligned}$$

The remainder of the claim follows by repeating the previous arguments. □

4 Discussion & Open Question

There is another interesting link between the complexity class AM and the UNIQUE-SAT problem: both the AM protocol for WSSE of [GS86] and the proof that $\mathsf{NP} \subseteq \mathsf{RP}^{\text{UNIQUE-SAT}}$ of [VV86] utilize pairwise independent hash functions. A natural question to ask is whether pairwise independent hash functions could be used to obtain the premises of Lemma 3.4.

One could extend these ideas to construct a mapping $\hat{\varphi}(u) = f(C)$, computable in randomized polynomial time, satisfying $\mathrm{EX}[\#_u\hat{\varphi}] = \#_x C/\alpha$ for some constant $\alpha \in \mathbb{R}^+$. It not clear, thought, how this construction could be used for the proof of our main result. Moreover, it is to be noted that both [GS86] and [VV86] utilize pairwise independent hash functions in a relativizable fashion, which precludes direct application of these ideas to our proof, due to oracle separations (see discussion after Theorem 1 for more details).

Another natural question is to identify a corresponding MA-complete problem in the flavor of WSSE for AM and SSE for SBP. Could the presented collapse, then, be extended to MA? Conversely, could one show that any collapse either AM, SBP or MA to a subclass implies that UNIQUE-SAT is NP-hard? Perhaps under an even strong assumption that $\mathsf{NP} \subseteq \mathsf{P/poly}$?

References

[AB09] S. Arora and B. Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[AKSS95] V. Arvind, J. Köbler, U. Schöning, and R. Schuler. If NP has polynomial-size circuits, then ma=am. *Theor. Comput. Sci.*, 137(2):279–282, 1995.

[Bab85] L. Babai. Trading group theory for randomness. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing (STOC)*, pages 421–429, 1985.

[BG82] A. Blass and Y. Gurevich. On the unique satisfiability problem. *Information and Control*, 55(1-3):80–88, 1982.

[BGM06] E. Böhler, C. Glaßer, and D. Meister. Error-bounded probabilistic computations between MA and AM. *J. Comput. Syst. Sci.*, 72(6):1043–1076, 2006.

[GLM+16] M. Göös, Sh. Lovett, R. Meka, T. Watson, and D. Zuckerman. Rectangles are nonnegative juntas. *SIAM J. Comput.*, 45(5):1835–1869, 2016.

[GS86] S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing (STOC)*, pages 59–68, 1986.

[KL80] R. M. Karp and R. J. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proceedings of the 12th Annual ACM Symposium on Theory of Computing, April 28-30, 1980, Los Angeles, California, USA*, pages 302–309, 1980.

[Val76] L. G. Valiant. Relative complexity of checking and evaluating. *Inf. Process. Lett.*, 5(1):20–23, 1976.

[Ver92] N. K. Vereshchagin. On the power of PP. In *Proceedings of the Seventh Annual Structure in Complexity Theory Conference*, pages 138–143, 1992.

[VV86] L. G. Valiant and V. V. Vazirani. NP is as easy as detecting unique solutions. *Theor. Comput. Sci.*, 47(3):85–93, 1986.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il