Review on Circuit Minimization Problem by Kabanets-Cai

Date: November 16, 2020

(First Draft)

1 Introduction

In the Minimum Circuit Size Problem (MCSP), we are given a truth table of some Boolean function together with a positive integer s_n as input, and our task is to answer the question whether there exists a circuit of size at most s_n that solves the function represented by the given truth table. Formally speaking, given a Boolean function $f_n : \{0,1\}^n \to \{0,1\}$, an instance of MCSP is a tuple $\langle T_n, s_n \rangle$ where T_n is the string of length 2^n representing the truth table of f_n and f_n is computable by a circuit of size at most s_n. It is easy to see that MCSP is in **NP**. Namely, we can define a certificate as some proposed circuit C of size at most s_n, and verify whether C computes each entry of the truth table correctly in polynomial time. With that being said, a natural question arise: Is MCSP **NP**-complete?

In the paper "Circuit Minimization Problem," Valentine Kabanets and Jin-Yi Cai addressed the difficulty of showing MCSP to be **NP**-hard. Namely, they showed some consequences that are unlikely to happen if there exists a polynomial-time reduction function R from SAT to MCSP that is "natural," in the sense that the size of the output depends on the size of the inputs only, and these sizes are polynomial related. Furthermore, the authors pointed out why it is challenging to prove $\mathsf{MCSP} \notin \mathbf{P}$ (again, by providing some consequences whose likelihood are questionable). Clearly, such proof would imply $\mathbf{P} \neq \mathbf{NP}$ which goes beyond the currently known techniques.

The rest of the review: In Section 2, we will provide some definitions required to understand the main results and theorems of the paper. Section 3 introduces some main consequences when $\mathsf{MCSP} \notin \mathbf{P}$ and MCSP is **NP**-hard under the "natural" reduction. Finally, we conclude the review by providing some insightful remarks and directions for further research on this topic in Section 4.

2 Some definitions:

1. The class Sub-Exponential: $\mathbf{SUBEXP} = \bigcap_{\epsilon>0} \mathbf{DTIME}(2^{n^\epsilon})$
2. The class **QP** of languages decided by a TM in *quasi-polynomial* time defined by

$$\mathbf{QP} \coloneqq \mathbf{DTIME}(n^{\mathrm{polylog}(n)}) = \bigcup_{c>1} \mathbf{DTIME}(2^{\log^c n}) = \bigcup_{c>1} \mathbf{DTIME}(n^{\log^c n})$$

where we obtain the last equality since

$$2^{(\log n)^{c+1}} = \exp(\log^{c+1} n) = \exp(\log n \log^c n) = n^{\log^c n}.$$

Note that **QP** contains **P** since $\text{polylog}(n) \in \Omega(1)$.

3. The class exponential time with linear exponential: $\mathbf{E} = \mathbf{DTIME}(2^{O(n)})$

4. **Natural Reduction**: For two problems A and B and a Karp reduction from A to B, we say the reduction R is natural if, for any instance I of A, the length of the output $|R(I)|$, as well as all possible output parameters s_n, depend only on the input length $|I|$. Furthermore, $|I|$ and $|R(I)|$ are polynomial related.

   Example: $\mathsf{SAT} \leq_p \mathsf{3SAT}$ is natural.

   Question(s): What is output parameter s_n? Example?

Lemma 1. *If* $\mathbf{NP} \subseteq \mathbf{QP}$ *then* $\mathbf{PH} \subseteq \mathbf{QP}$.

Proof. We first show that $\mathbf{NP} = \exists\mathbf{P} \subseteq \mathbf{QP}$ also implies that $\exists\mathbf{QP} \subseteq \mathbf{QP}$ by a padding argument.[1] To this end, assume that $L \in \exists\mathbf{QP}$ such that it is decided by a $\exp(\log^c n)$-time verifier M. We define our padded language as $L_{\text{pad}} \coloneqq \{\, x1^{\exp(\log^c n)} \mid x \in L \,\}$. We obtain that $L_{\text{pad}} \in \exists\mathbf{P}$ since applying M to a padded element takes takes only linear time. It follows by our assumption that $L_{\text{pad}} \in \mathbf{QP}$; say L_{pad} is decided by an $\exp(\log^{c'} n)$-time TM M'. Now, the key idea lies in showing that this already implies $L \in \mathbf{QP}$: Let $x \in \{0,1\}^*$ be of length $n \coloneqq |x|$ and its padded version $y \coloneqq x1^{\exp(\log^c n)}$ of length $m = \exp(\log^c n)$. Then, the time needed by M' to decide y is

$$\exp(\log^{c'} m) = \exp(\log^{c'}(\exp(\log^c n))) = \exp(\log^{c \cdot c'} n)$$

which is again quasi-polynomial in n. Therefore, we can decide $x \in L$ by applying padding and running M' all in quasi-polynomial time.

The second step is to show that also $\forall\mathbf{QP} \subseteq \mathbf{QP}$. By definition, given a language $L \in \forall\mathbf{QP}$, there exists a polynomial p and a quasi-polynomial time verifier M such that

$$x \in L \iff \forall y \in \{0,1\}^{p(|x|)}\ M(x,y) = 1\,.$$

However, the right hand side is equivalent to the negation of $(\exists y \in \{0,1\}^{p(|x|)}\ M(x,y) = 0)$, a statement which can itself be decided in quasi-polynomial time by the above argument. Since all quasi-polynomial time TMs are also deterministic, we can also negate the output efficiently. It follows that $L \in \mathbf{QP}$.

By an inductive argument, we see that $\mathbf{PH}$ collapses into a subset of $\mathbf{QP}$. □

Theorem 2 (Theorem 15)**.** *If* MCSP *is* $\mathbf{NP}$*-hard under a natural reduction from* SAT*, then*

1. $\mathbf{E}$ *contains a family of Boolean functions* f_n *not in* $\mathbf{P}_{/\mathbf{poly}}$ *(i.o.), and*
2. $\mathbf{E}$ *contains a family of Boolean functions* f_n *of circuit complexity* $2^{\Omega(n)}$ *(i.o.), unless* $\mathbf{NP} \subseteq \mathbf{SUBEXP}$

Proof. - Statement 1: consider two cases

+ Case 1: $\mathbf{NP} \subseteq \mathbf{QP}$

  It is obvious to see that $\mathbf{NP} \subseteq \mathbf{QP} \implies \mathbf{PH} \subseteq \mathbf{QP}$ (think of this as a generalisation of $\mathbf{NP} \subseteq \mathbf{P} \implies \mathbf{PH} \subseteq \mathbf{P}$). Also, we make an observation that $\mathbf{QP}^{\Sigma_k^p}$, for some $k \in \mathbb{N}$, contains a language of superpolynomial circuit complexity. The proof of this claim is as follow: (help!!!!)

  Combining these two results, we get the following $\mathbf{NP} \subseteq \mathbf{QP} \implies \mathbf{QP}^{\mathbf{PH}} \subseteq \mathbf{QP}^{\mathbf{QP}} \subseteq \mathbf{QP} \subset \mathbf{E}$ contains a family of function not in $\mathbf{P}_{/\mathbf{poly}}$.

  Hence, $\mathbf{E} \not\subseteq \mathbf{P}_{/\mathbf{poly}}$

[1] The class $\exists\mathbf{QP}$ is defined to consist of all languages L such that there exists a quasi-polynomial TM M and a polynomial p with $x \in L \iff \exists y \in \{0,1\}^{p(|x|)}\ M(x,y) = 1$

+ Case 2: $\mathbf{NP} \not\subseteq \mathbf{QP}$

  A given natural reduction R from SAT to MCSP maps every family of boolean formulas of size n to truth tables of Boolean functions on $k = \theta(\log n)$ variables (is the symbol θ significant? more elaboration on $\theta(\log n)$, like why is that?) and a parameter s_n.

  Since the reduction is natural, s_n is some function of n only and thus it could be some fixed polynomial $\log^c n$. Since there are at most [...] (fill in the gap, I don't really understand why they said there are $n^{\text{polylog(n)}}$ circuits) that many different circuits on k inputs given $\log^c n$ gates. Thus, all such instances of MCSP where $s_n = \log^c n$, for some constant c, are solvable in quasi-polynomial time. This implies that $\mathsf{SAT} \in \mathbf{QP}$ which is a contradiction to our assumption since SAT is $\mathbf{NP}$-complete. (should we say something about the relationship between k and $\log n$? My understanding is that what we have shown here implies that the circuit size $s_n \neq \log^c n$ which is related to something like k^c, i.e. can't get a circuit of poly(k) size which leads to the thing below)

  Therefore, we can obtain the desired k-variable functions not in $\mathbf{P}_{/\mathbf{poly}}$ by applying the reduction R to any trivial family of unsatisfiable formulas. Clearly, this family of hard functions is computable in time $2^{O(k)}$. (maybe more elaboration on this too...).

- Statement 2: the proof is similar (you want to do it too?)

□