A Candidate for Natural Problems in NP-NPC-P under P≠NP*

ZHAO Yun-lei, ZHU Hong

(*Department of Computer Science, Fudan University, Shanghai* 200433, *China*)

E mail: {990314, hzhu}@fudan.edu.cn

Received May 22, 2000; accepted October 16, 2000

Abstract: In 1975, Lander showed that there exist some languages in NP-NPC-P (denoted as NPI) under the assumption P≠NP. But the language constructed there is indeed an unnatural one because the construction needs to run all polynomial time Turing machines. So far, no natural problems have been proved to be in NPI under P≠NP and finding a natural problem in NP-NPC-P is indeed an important open problem. In this paper this long open problem is partially solved. A $2+f(m)$-HSAT model is defined. Based on this model, a candidate for natural problems in ((NP-NPC)-P), denoted as NPI, under the assumption P≠NP is given, and the authors have proven that it is not in NP-Complete under P≠NP. Actually, it indeed is in NPI under some stronger but plausible assumption. In comparison, similar results for the two other candidates, GI and Factoring, are not known.

Key words: NP-complete; Karp-reduction; SAT; NPI

To understand the onset of exponential complexity that occurs when going from a problem in P(2-SAT) to a problem that is NP-Complete (3-SAT), Ref. [1] introduced the $2+p$-SAT model, where p is a constant and $0\leqslant p\leqslant 1$. It's a formula with m clauses, of which $(1\text{-}p)m$ contains two variables (2-clauses) and pm contains three variables (3-clauses). This '$2+p$-SAT' model smoothly interpolates between 2-SAT ($p=0$) and 3-SAT ($p=1$). It can be easily shown that $2+p$-SAT is in NP-Complete[1,2] for all p, $p>0$.

But what about if p is not a constant but a function of m, where m is the number of clauses in the formula?

1 $2+(\log(m))^k$-Hybrid-SAT

Definition 1. Hybrid-SAT (HSAT).

Hybrid-SAT is a combination of 2-SAT and 3-SAT. Any instance of HSAT, say a formula Φ, has the form $\Phi=\Phi_2\wedge\Phi_3$, where Φ_2 is an instance of 2-SAT with $m_2(m_2\geqslant 1)$ 2-clauses and Φ_3 is an instance of 3-SAT with m_3 $(m_3\geqslant 1)$ 3-clauses but the variables which appear in Φ_2 do not appear in Φ_3, and vice versa.

We denote $|\Phi|$ as the number of clauses in Φ.

Definition 2. $2+(\log(m))^k$-HSAT, $k\geqslant 6$.

Let $\Phi=\Phi_2\wedge\Phi_3$. It's a formula with m clauses and n variables, where Φ_3 is an instance of 3-SAT which contains $(\log(m))^k (k\geqslant 6)$ 3-clauses and n_3 variables and Φ_2 is an instance of 2-SAT with $(m-(\log(m))^k)$ 2-clauses and n_2 variables, and the variables which appear in Φ_2 do not appear in Φ_3, and vice versa.

* Supported by the National Natural Science Foundation of China under Grant No.69973013 (国家自然科学基金)

ZHAO Yun-lei was born in 1975. He is a Ph.D. candidate at the Department of Computer Science, Fudan University. His main research interests are algorithm, computational complexity, cryptography. **ZHU Hong** was born in 1939. He is a professor and doctoral supervisor of the Department of Computer Science, Fudan University. His main research interests are algorithm, computational complexity, cryptography.

2 A Candidate for Natural Problems in NP-NPC-P under P≠NP

In 1975, Lander showed that there exist some languages in NP-NPC-P (denoted as NPI) under the assumption P≠NP. But the language constructed there is indeed an unnatural one because the construction needs to run all polynomial time Turing machines. So far, no natural problems have been proved to be in NPI under P≠NP and the problem GI (Graph Isomorphism) is regarded as a most likely candidate[3,4]. Now, we give another candidate for natural problems in NPI under P≠NP. We will prove that it is not in NP-Complete under the assumption P≠NP. Actually, it indeed is in NPI under some stronger but plausible assumption and thus we can partially solve this long open problem.

Theorem 1. $2+(\log(m))^k$-HSAT is not in NP-C under the assumption P≠NP, $k\geqslant 6$.

Proof. Clearly this problem is in NP. We prove this theorem by showing that 3-SAT cannot be reduced to $2(\log(m))^k$-HSAT by Karp reduction.

Assume that there exists a Karp reduction (denoted as F) from 3-SAT to $2-(\log(m))^k$-HSAT. It means that for any instance of 3 SAT, a formula Φ_0 which contains n_0 variables and $m_0$3 clauses, we can construct the $F(\Phi_0)$ which is an instance of $2+(\log(m))^k$-HSAT in polynomial time of n_0, and $F(\Phi_0)$ is satisfiable if and only if Φ_0 is satisfiable.

Let $F(\Phi_0)=\Phi_2\wedge\Phi_3$, where $|\Phi_3|=(\log(|\Phi_3|+|\Phi_2|))^k$. We consider the relation between $|\Phi_3|$ and $|\Phi_0|$. There are two cases:

Case 1. $|\Phi_3|\geqslant|\Phi_0|=m_0$.

Claim 1. m cannot be expressed as a polynomial of $\log(m)^k$, $k\geqslant 1$.

Proof. (of Claim 1). It can be easily proven since for any k', $k'>0$, there exits an m', which makes $m>((\log(m))^k)^{k'}$ when $m>m'$. □

According to Claim 1, in case 1, we can get the fact that $(|\Phi_2|+|\Phi_3|)$ cannot be expressed as a polynomial of $|\Phi_3$, and since $|\Phi_3|\geqslant m_0$, so $|(\Phi_2|+|\Phi_3|)$ also can not be expressed as a polynomial of m_0. Note that $n_0\leqslant 3m_0$ and $m_0\leqslant 8n_0^3$, then $(|\Phi_2|+|\Phi_3|)$ also can not be expressed as a polynomial of n_0. It's absurd since the Karp reduction $F(\Phi_0)$ must be done in polynomial time of n_0.

Case 2. $\Phi_3|<|\Phi_0|$.

Since we assume $F(\Phi_0)$ can be constructed in polynomial time of n_0, then $|\Phi_2|$ can certainly be expressed as $P(n_0)$, where $P(\cdot)$ is a polynomial. So, if $|\Phi_3|<|\Phi_0|$, it means that we can decrease the 3-clause number in Φ_0 by adding $P(n_0)$ 2-clauses (by imposing F on Φ_0). However, note that the variables which appear in Φ_2 do not appear in Φ_3, and vice versa, then we can impose F on Φ_3, and so on, and repeat the above process at most m_0 times we can eliminate all 3-clauses in $F(\Phi_0)$ to get a formula Φ' and guarantee that Φ' is satisfiable if and only if $F(\Phi_0)$ is satisfiable if and only if Φ_0 is satisfiable, where Φ' contains only 2-clauses and $|\Phi'|$ is at most $m_0P(n_0)$, or at most $8n_0^3\cdot P(n_0)$, another polynomial of n_0. It means that there exists a Karp reduction from 3-SAT to 2-SAT which contradicts our assumption P≠NP.

So, from the arguments above, we can conclude that $2+\log(m)^k$-HSAT is not in NP-Complete under the assumption P≠NP. □

3 Can the Candidate Be in *P*?

In practice, the time complexity of the fastest algorithm for 3-SAT is 1.334^{n}[6], where n is the variable number.

In theory, 3-SAT is in SNP-Complete under SERF reductions[5] (definitions of SNP and SERF can be found in

Ref. [5]). That is if 3-SAT has sub-exponential time algorithm then all problems in SNP have sub-exponential time algorithm[5]. It means that we assume the fact that 3-SAT does not have sub-exponential time algorithm is plausible. In light of this, Ref. [7] introduced the following hypothesis:

Definition 3. Define s to be the infimum of $\{\delta$: there exists an $O(2^{\delta n})$ algorithm for solving 3-SAT$\}$. Define ETH (Exponential-Time Hypothesis) for 3-SAT to be that: $s>0$. In other words, 3-SAT does not have sub-exponential time algorithm.

Note that the ETH for 3-SAT is stronger than NP≠P but plausible according to the above both practical and theoretic arguments. And under thus assumption we get that:

Theorem 2. The $2+\log(m)^k$-HSAT is indeed a natural problem in NPI under ETH for 3-SAT, $k\geqslant 6$.

Proof. For an instance of $2+\log(m)^k$-HSAT, $\Phi=\Phi_2\wedge\Phi_3$, where Φ contains m clauses and n variables and Φ_3 is an instance of 3-SAT which contains $m_3=(\log(m))^k$3-clauses and n_3 variables and Φ_2 is an instance of 2-SAT. Note that $m_3\leqslant 8n_3^3$, $n\leqslant 3m$, that is $n_3\geqslant\frac{1}{2}(\log(m))^2\geqslant\frac{1}{2}\left(\log\left(\frac{1}{3}n\right)\right)^2$. Then Φ_3 can not be solved in polynomial time of either m or n under ETH for 3-SAT. It is the same for $\Phi=\Phi_2\wedge\Phi_3$, since the variables which appear in Φ_2 do not appear in Φ_3, and vice versa. It means that $2+\log(m)^k$-HSAT is not in P under ETH for 3-SAT.

Then according to Theorem 1, the theorem does hold. □

The more general case of $2+\log(m)^k$-HSAT($k\geqslant 6$), where the variables which appear in Φ_2 may appear in Φ_3, and vice versa, is currently under investigation.

References:

[1] Monasson, R., Zecchina, R., Kirkpatrick, S., *et al.* Determining computational complexity from characteristic 'phase transitions'. Nature, 1999,400:133～137.

[2] Anderson, W. Solving problems in finite time. Nature, 1999,400:115～116.

[3] Papadimitriou, H. Computational Complexity. New York: Addison Wesley Publishing Company, 1994. 329～332.

[4] Goldreich, O. Introduction to complexity. 1999, 23～25. http://theory.lcs.mit.edu/～oded/.

[5] Russell Impagliazzo, Ramamohan Paturi. Which problems have strongly exponential complexity? In: Rajeev Motwani ed. Proceedings of the Symposium on Foundations of Computer Science'98. Palo Alto, California: IEEE Computer Society, 1998. 653～664.

[6] Uwe Schoning. A probabilistic algorithm for *k* SAT and constraint satisfaction problems. In: Paul Beame ed. Proceedings of the Symposium on Foundations of Computer Science'99. New York: IEEE Computer Society, 1999. 410～420.

[7] Russell Impagliazzo, Ramamohan Paturi. Complexity of *k* SAT. 2000. http://dimacs.rutgers.edu/dieter/Seminar/index.html.

P≠NP 假设下 NP-NPC-P 中自然问题的一个候选者

赵运磊， 朱 洪

(复旦大学 计算机科学系，上海 200433)

摘要：1975年，Lander 证明在 P≠NP 假设下存在一个语言属于 NP-NPC-P(NPI). 但 Lander 给出语言并不是一个自然的语言因在该语言的构造中需运行所有多项式时间的图灵机. 迄今为止，还没有自然的语言被证明在 P≠NP 假设下属于 NPI，并且在 P≠NP 假设下寻找一个属于 NPI 的自然语言是一个重要的未解决问题. 作者部分解决了此长期未解决的问题. 定义了 $2+f(m)$-HAST 模型. 基于该模型，给出了在 P≠NP 假设下 NP NPC-P 中自然问题的一个候选者. 已证明在 P≠NP 假设下它不属于 NPC 并且在更强但合理的假设下它的确属于 NPI.

关键词：NP-Complete; Karp-归约;SAT;NPI

中图法分类号：TP301 **文献标识码**：A