

Problem NP versus P: A proof that NP is not equaled to P

VIKTOR IVANOV

Abstract. This proof of $P \neq NP$ is based on better estimates of lower bounds on the time complexity that hold for all solution algorithms. Almost no special knowledge is needed to understand the proof. A short review of the main steps and ideas of the proof can be seen in the Introduction below.

Keywords: Algorithm time complexity; NP-complete problem; Turing machine; Word transform.

2000 Mathematics Subject Classification: Primary 05D99, 68Q25.

1. Introduction

There are hundreds of important so-called NP-complete and NP-hard problems from various areas of mathematics and their applications (see, for example, Sipser [1994] and Johnson [1981-2006]). All NP-complete problems belong to the class NP, i.e., they are solvable on non-deterministic Turing machines in a polynomial time. However, it is unknown whether they are in the class P of problems, solvable on deterministic Turing machines (DTM) in a polynomial time. Most computer scientists believe that $P \neq NP$.

“That’s the most mind-boggling problem facing theoretical computer science, and maybe all of science at the moment.” (Knuth [2002]).

Below is a proof that $P \neq NP$. To help the reader to understand that proof, a short review of its main steps and ideas is given here.

We consider the proof using a typical NP-complete problem of INDEPENDENT SET (IS) (Garey & Johnson [1979]): INSTANCE (or WITNESS): graph $G = (V, E)$, positive integer K . QUESTION: Does G contain an independent set of size K , i.e., a subset $V' \subseteq V$ such that $|V'| = K$ and such that no two vertices in V' are joined by an edge in E ? We denote this problem as P_{IS} .

The problem P_I , which is P_{IS} under representation of the graph G in an ordered list of binary words (the size n^2 in order), is restated as the problem P_Q . This problem verifies whether a word transform $Q_{n,K}$ consisting of a certain union of intersections of given binary words is $\mathbf{1} = (1 \dots 1)$ (Lemmas 1, 2). The apparatus of word transforms was introduced in the paper by Ivanov [1999].

Under given natural representation of graphs, the author proved that there exists exponential size number of the initial words, for which any solution algorithm for (IS) requires a super-polynomial time, because any solution algorithm has to use not only each of those order n^2 words, but also order n^{K-2} connections among them, where arbitrary integer K (the size of possible IS) is independent of n . It is proved by the contradiction (‘perhaps the most prevalent of mathematical proofs’, Wikipedia). In particular, it is proven that ‘no one of those words can be missed, since otherwise a solution ... might be missed.’ It is similar to the case of palindrome recognition on one-tape DTM, for which there also exists exponential size data (almost all possible words), each of which requires the time order n^2 , since any solution algorithm has to use not only each of those word, but also order n^2 connections between one and another half of word. However, there is a principal difference between palindrome recognition and IS: on multi-tape DTM the first problem requires linear time, but the IS requires super polynomial time again, since $K/2$ is arbitrary if K is arbitrary.

The results of the author are based on a novel general approach that can be described on qualitative level as consisting of 3 main parts.

The first part consists of construction of a combinatorial lemma for a problem on the whole input domain. A sense of that lemma is a rather detailed spectrum of witnesses for the answer (1, true) and/or (0, false). The results of that lemma are independent of any solution algorithm of a problem.

Having arbitrary exponential number witnesses from the input domain of a problem for both answers (1) and (0), it is not difficult to prove existence of an exponential number of ‘depersonalized witnesses’, for which any polynomial solution algorithm does not know the answer 1 or 0 in advance and hence it has to be realized for each of those witnesses to find the answer.

The third part consists of estimation of a low bound for any solution algorithm, proving (on the basis of a combinatorial lemma and a logical structure of a problem) which of its operations are necessary.

Results of application of this approach depend on many properties of a problem in question. Its application to the problems similar to the palindrome recognition gives results by far simpler reasoning than by the well-known method crossing sequences (Hennie, 1965). However, in the case of more complicated problems, results of its application depend heavily on both combinatorial and logical structure of those problems.

Preliminary Lemmas 1, 2, and supporting results 1-7 are subsidiary for the proof of Lemma 3.

The main Lemma 3 is the most important and difficult to prove. Its proof shows existence of ‘hard’ witnesses for which the time on any DTM is near the longest and at least n^{K-2} in order. This is because any DTM for those witnesses has to compute order n^{K-2} intermediate distinct objects.

As it turns out, the proof combines two possibilities anticipated by Cook [1971]: diagonalization with reduction (parts 1 and 2 of the main Lemma 3 proof, up to the relation (26)), and super-polynomial lower bounds for some special weaker problem (parts 3, 4, and 5).

Realizing the first possibility, we have consequent lower bounds for any fixed solution algorithm since each step signifies further contraction of the input domain.

Let us consider briefly the second possibility in the case of $K = 4$. For this case, a weaker problem: $V_{m,3} = \cup (S_k \cap Z_k) (k = 1, \dots, m) =? \mathbf{1}$, $S_k = \cup (S_{ik} \cap Y_j) (j = 1, \dots, m)$, $S_{jk} = \cup X_i (i \in P'_j \cap Q'_k)$, where X_i, Y_i, Z_i, P_i, Q_i are arbitrary binary words of the size $m = n/2$, I' is a set of zero-numbers in any word I .

It is proven that arbitrary situated parts of words S_k are arbitrary words of the size $2\log (2\log m)$. In turn, arbitrary situated parts of words S_{jk} and $P_j \cap Q_k$ are arbitrary words of the size $2\log m$. No one of those words can be missed not to miss a solution. They all have to be computed for determining $V_{m,3} =? \mathbf{1}$ or the solution requires more time than m^2 .

Similar reasons exist in the case of any K independent of n . In that case a weaker problem: $V_{m,t} = \cup (S_{k_t} \cap X_{tk_t}) (k_t = 1, \dots, m) =? \mathbf{1}$, $S_{k_t} = \cup (S_{k_{t-1}k_t} \cap X_{t-1k_{t-1}}) (k_{t-1} = 1, \dots, m), \dots, S_{k_3 \dots k_t} = \cup (S_{k_2 \dots k_t} \cap X_{2k_2}) (k_2 = 1, \dots, m)$, $S_{k_2 \dots k_t} = \cup X_i (i \in Y'_{2k_2} \cap \dots \cap Y'_{tk_t})$, where $X_i, X_{2k_2}, \dots, X_{tk_t}, Y_{2k_2}, \dots, Y_{tk_t}$ are arbitrary binary words of the size m , $m = n/K$, $t = K-1$.

For determining $V_{m,t} =? \mathbf{1}$, we have to compute S_{k_t} , for which we have to compute $S_{k_{t-1}k_t}$, and so on. So, the required time will be not less than m^{t-1} .

The main reason for $NP \neq P$ is a hard problem of contacting all words given among each other. A proof of the theorem on $P \neq NP$ follows from Lemmas 2 and 3 after reducing general case of inputs to the case of inputs for the problem P_1 or P_Q .

2. Preliminaries

Let P be a problem with input domain $I \subseteq \Sigma^*$ and output domain $R \subseteq \Sigma^*$, where Σ^* is a set of all finite strings from the alphabet $\Sigma = (0, 1)$, and let A be a solution algorithm of P on a DTM with a finite set of states S . A specifies transition function of DTM (detailed definition of a DTM, see e.g., in Hopcroft, Motwani, Ullman [2006]).

In general, input of a DTM $x \in L_r$, where L_r is all 2^r binary words of length r . For decision problems $R = \mathbf{1} = (1 \dots 1)$ or not, i.e., $R = ? \mathbf{1}$.

The time complexity of the solution of P on a DTM is given by

$$T(P) = T(I, R) = T_r(P, A) = T_r(P, A, S) = \sup t(x, y, A, S) (\forall x: |x| = r, x \in I), \quad (1)$$

where $t(x, y, A, S)$ is the time for A on input $x \in I$ with output $y \in R$ on a DTM with states S . The space complexity is given by

$$S(P) = S_r(P, A) = S_r(P, A, S) = \sup s(x, y, A, S) (\forall x: |x| = r, x \in I), \quad (2)$$

where $s(x, y, A, S)$ is the number of necessary tape cells for A on input x with output y on a DTM. A problem $P \in \mathbf{P}$ iff there are S, A , and a constant c such that $T_r(P, A) < r^c$ for all $r \in \mathbf{Z}^+$.

Note that different notations in (1) or (2) mean the same magnitudes. We omit by default certain terms to simplify the notations later. We use $T(y)$ or $T(y = ? \mathbf{1})$ instead of $T(P)$ when y is explicit presentation of the problem P . For all the problems below we use $r = n^2$ in order.

Let the representation of the graph $G = (V, E)$ be an ordered list of binary words

$$I_j = (a_{1j} \dots a_{j-1j}), j = 2, \dots, n; a_{ij} = 0 \vee 1, 1 \leq i < j \leq n, \quad (3)$$

where $a_{ij} = 1$, iff there is an edge from the node i to the node j , $i < j$. This representation is convenient for the query of both nodes $1, \dots, n$ and edges a_{ij} [J. van Leeuwen, 1990]. The respective input for this initial problem has the size order $n(n-1)/2 + \log_2 K$, $1 \leq K \leq n$.

We also introduce the binary words

$$A_{k_1 \dots k_t} = (a_{k_1 k_2} a_{k_1 k_3} \dots a_{k_1 k_t} a_{k_2 k_3} a_{k_2 k_4} \dots a_{k_2 k_t} \dots a_{k_{t-2} k_{t-1}} a_{k_{t-2} k_t} a_{k_{t-1} k_t}), \quad (4)$$

where (k_1, \dots, k_t) are arbitrary ordered combinations from $(1, \dots, n)$ by t . Under fixed t , there are $\binom{n}{t} = n! / [(n-t)! t!]$ different words $A_{k_1 \dots k_t}$.

Let the problem P_1 be $P_{I, S}$ with the inputs (3), $n \geq K = t > 1$, and let P_2 with the same inputs (3) be the problem whether there exists a word $A_{k_1 \dots k_t} = \mathbf{0} = (0 \dots 0)$ for any n and t , $n \geq t > 1$. Since $A_{k_1 \dots k_t} = \mathbf{0}$, iff (k_1, \dots, k_t) is an independent set, the problems P_1 and P_2 are restatements of each other.

Denoting all 2^n binary words by L_n , for any word $I \in L_n$, let I' be the ordered set consisting of zero numbers in I ($n - |I'| =$ number of 1-digits in I) and \bar{I} be complement of I to $\mathbf{0} = (0 \dots 0) \in L_n$. For any words I_1 and I_2 from L_m and L_n , $1 \leq m \leq n$, let the intersection $I_1 \cap I_2$ (the union $I_1 \cup I_2$) be a word from L_m (L_n) combining all common 0-digits (all 0-digits) of I_1 and I_2 . This definition of intersection and union for binary words corresponds to the ordinary definition of intersection and union for the respective sets I'_1 and I'_2 .

If, for example, $I_1 = (010)$, $I_2 = (100)$, then $I_1 \cap I_2 = (110)$ and $I_1 \cup I_2 = (000)$, since $I'_1 = (1, 3)$, $I'_2 = (2, 3)$, then $I'_1 \cap I'_2 = (3)$ and $I'_1 \cup I'_2 = (1, 2, 3)$.

LEMMA 1. *The following relations are valid:*

$$A_{k_1 \dots k_t} \neq 0, \forall k_1, \dots, k_t: 1 \leq k_1 < \dots < k_t \leq n, n \geq t > 1, \quad (5)$$

iff

$$Q_{n,t} = \cup (I_{k_2} \cap \dots \cap I_{k_t}) (\forall k_2, \dots, k_t: k_t \in (t, \dots, n),$$

$$k_{t-1} \in I'_{k_t}, k_{t-2} \in I'_{k_{t-1}} \cap I'_{k_t}, \dots, k_2 \in I'_{k_3} \cap \dots \cap I'_{k_t}) = \mathbf{1}, \quad (6)$$

where I'_j are ordered sets of zero-numbers in the binary words $I_j = (a_{1j} \dots a_{ij-1j})$. In addition, if $Q_{n,t} = \mathbf{1}$, then $Q_{n,s} = \mathbf{1}$, $t < s \leq n$, and if $Q_{n,t} \neq \mathbf{1}$, then there exists an independent set

$$V_t = (k_1, \dots, k_t), |V_t| = t. \quad (7)$$

PROOF. It is based on a simple fact that $k_{r-1} \in I'_{k_r}$, iff $a_{k_{r-1}k_r} = 0$, and hence the intersections $I_{k_r} \cap \dots \cap I_{k_t} \neq \mathbf{1}$, $r = 2, \dots, t$, iff there exist $k_{r-1} \in I'_{k_r} \cap \dots \cap I'_{k_t}$ such that $a_{k_{r-1}k_r} + \dots + a_{k_{r-1}k_t} = 0$, $r = 2, \dots, t$, and hence their sum, one of $A_{k_1 \dots k_t}$ in (4), is also equal to $\mathbf{0}$. Thus, if $Q_{n,t} \neq \mathbf{1}$, then there exists (k_2, \dots, k_t) , $k_2 \in I'_{k_3} \cap \dots \cap I'_{k_t}; \dots; k_{t-1} \in I'_{k_t}; k_t \in (t, \dots, n)$, such that in (6) $I_{k_2} \cap \dots \cap I_{k_t} \neq \mathbf{1}$, and hence there exists $k_1 \in I'_{k_2} \cap \dots \cap I'_{k_t}$, for which $A_{k_1 \dots k_t} = \mathbf{0}$. If $A_{k_1 \dots k_t} = \mathbf{0}$, then all $a_{k_r k_s} = 0$, $r = 1, \dots, t-1; s = r+1, \dots, t$, and hence at least one $I_{k_2} \cap \dots \cap I_{k_t} \neq \mathbf{1}$, and hence $Q_{n,t} \neq \mathbf{1}$. If $Q_{n,t} = \mathbf{1}$, then $Q_{n,s} = \mathbf{1}$, $t < s \leq n$, since $Q'_{n,s} \subseteq Q'_{n,t}$, $t < s$, but $Q'_{n,t} = \emptyset$. And if $Q_{n,t} \neq \mathbf{1}$, then $A_{k_1 \dots k_t} = \mathbf{0}$, and hence (7) is valid.

Given the same input domain as for the problems P_1 or P_2 , let P_Q be the problem of verification of $Q_{n,t} = ? \mathbf{1}$, $1 < t \leq n$, for any n and t .

LEMMA 2. *We have $T(P_1) = T(P_2) = T(P_Q)$.*

PROOF. The problems P_1 , P_2 , and P_Q have the same input domain, and if for any input the output of one of these problems is 'yes', then the outputs for the other problems are also 'yes'.

3. Supporting results

RESULT 1. *While each of X_k , $k = 1, 2, \dots, q$, runs independently all 2^m words from L_m ,*

$$I_q = \cap_{k=1}^q X_k, U_q = \cup_{k=1}^q X_k, 1 \leq q \leq m, \quad (8)$$

run each of all $\binom{m}{r}$ words with r 1-digits respectively $(2^q - 1)^r$, $(2^q - 1)^{m-r}$ times. In particular,

$$|\{I_q = \mathbf{1}\}| = (2^q - 1)^m, |\{I_q \neq \mathbf{1}\}| = 2^{qm} - (2^q - 1)^m, |\{U_q = \mathbf{1}\}| = 1, |\{U_q \neq \mathbf{1}\}| = 2^{qm} - 1, \quad (9)$$

where for any relation R of inputs $X = (X_k, k = 1, 2, \dots, q)$, the notation $\{R\}$ means the sets of all X , for which R is valid.

PROOF. In the case of I_q , $q = 2$,

$$(2^2-1)^r = 3^r = (2+1)^r = \sum_{s=0}^r \binom{r}{s} 2^s, r = 0, 1, \dots, m, \quad (10)$$

since for each of $\binom{r}{s}$ words X_1 with s 1-digits, X_2 can generate all possible 2^s words with fixed $r-s$ 1-digits corresponding to 0-digits of X_1 . For the other q , the proof can be obtained by the mathematical induction. Indeed, using the equality $I_{s+1} = I_s \cap X_{s+1}$, $1 < s < q$, we have

$$(2^{s+1}-1)^r = \sum_{k=0}^r \binom{r}{k} (2^{s+1}-2)^k = \sum_{k=0}^r \binom{r}{k} 2^k (2^s-1)^k, r = 0, 1, \dots, m, \quad (11)$$

where the right side means that for each k 1-digits of I_s (each is running $(2^s-1)^k$ times by premise), X_{s+1} can generate all possible 2^k words with fixed $r-k$ 1-digits corresponding to 0-digits of I_s .

The term $(2^q-1)^{m-r}$ follows from the fact that $\underline{U}_q = \cap_{k=1}^q \underline{X}_k$. The terms in (9) follow directly from the proof above.

RESULT 2. *Let $M = (1, \dots, m)$ and let $X_i, P_i, Q_i, i \in M$, be arbitrary words from L_m . Then the numbers of different sets of different words $P_j \cap Q_k$ and different respective words*

$$S_{jk} = \cup X_i (i \in P'_j \cap Q'_k), j, k \in M, \quad (12)$$

can be super polynomial and such that missing at least one word $P_j \cap Q_k$ results in missing at least one of respective words S_{jk} .

PROOF. Putting

$$P_j = (\alpha, b, \mathbf{1}), Q_k = (a, \beta, \mathbf{1}), P_j \cap Q_k = (\alpha, \beta, \mathbf{1}), j, k \in M, \quad (13)$$

where (asymptotically by m) α and β are arbitrary independent words from $L_{\log m}$, $\mathbf{1}$ is a unit word from $L_{m-2\log m}$, we have $3^{2\log m}$ possible different witnesses $\{P_j, Q_k, j, k \in M\}$, when $P_j \cap Q_k = (\alpha, \beta, \mathbf{1}), j, k \in M$ (for each of $\binom{\log m}{s}$ values of α with s 0-digits, the respective s digits in a can be arbitrary, and the same independent possibility for β and b instead of α and a).

With regard to order $m/(2\log m)$ independent dispositions of (α, β) , we find the total order

$$N_2 = m/(2\log m) \binom{m}{2\log m} \quad (14)$$

cases, when the sets of different words $P_j \cap Q_k, j, k \in M$, can be independent.

In addition, due to the result 1, for each of $3^{2\log m-y}$ sets of different words

$$P_j \cap Q_k, |P'_j \cap Q'_k| = y, 0 \leq y \leq 2\log m, j, k \in M, \quad (15)$$

we can have different set of 2^y different respective words $S_{jk}, 1 \leq y \leq 2\log m, j, k \in M$. So, if one of those $P_j \cap Q_k$ words is missed, at least one of different respective words S_{jk} is also missed. With regard to all independent sets, a number of different sets of words $P_j \cap Q_k$ such that missing at least one of the words results in missing at least one of the respective words S_{jk} is super polynomial, not less than order N_2 .

RESULT 3. Let $X_i, Y_{rk_r}, k_r \in M, r = 1, \dots, t$, be arbitrary words from L_m . Then the number of sets of different words $I_{m,t} = Y_{1k_1} \cap \dots \cap Y_{tk_t}, k_r \in M, r = 1, \dots, t$, and respective words

$$S_{k_1 \dots k_t} = \cup X_i (i \in Y'_{1k_1} \cap \dots \cap Y'_{tk_t}), k_r \in M, r = 1, \dots, t, \quad (16)$$

can be super polynomial and such that missing at least one word $I_{m,t}$ results in missing at least one of respective words $S_{k_1 \dots k_t}$.

PROOF. It follows actually from the same proof as in the result 2. Now, we find that the total

$$N_t = m/(t \log m) \binom{m}{t \log m} \quad (17)$$

in order cases are possible, when the sets of different words $Y_{1k_1} \cap \dots \cap Y_{tk_t}, k_r \in M, r = 1, \dots, t$, can be independent. In addition, for each of $3^{t \log m - y}$ sets of different words

$$Y_{1k_1} \cap \dots \cap Y_{tk_t}, |Y'_{1k_1} \cap \dots \cap Y'_{tk_t}| = y, 0 \leq y \leq t \log m, k_r \in M, r = 1, \dots, t, \quad (18)$$

we can have different set of 2^y different respective words $S_{k_1 \dots k_t}, 1 \leq y \leq t \log m, k_r \in M, r = 1, \dots, t$. So, a number of sets of words $I_{m,t}$ such that missing at least one of those $I_{m,t}$ results in missing at least one of the respective words $S_{k_1 \dots k_t}$ is also super polynomial, not less than N_t .

RESULT 4. Let X_k, Y_k be arbitrary from L_m . Then the word $\cup_{k=1}^m (X_k \cap Y_k)$ can have an arbitrary situated inside it arbitrary part from $L_{2 \log m}$.

PROOF. It is consequence of the result 1 and the result 2 proof, because arbitrary situated parts of $X_k \cap Y_k, k \in M$, can be arbitrary from $L_{2 \log m}$.

RESULT 5. For any DTM and almost all X, Y from L_m , the time for determining each of $I = X \cap Y, U = X \cup Y$, and solution of the problem $I = X \cap Y =? \mathbf{1}$, is not less than $\Theta(m)$.

PROOF. This result is known; it can be proven by ‘crossing sequence’ argument (Hennie [1965]). However, we can prove the result 4 much simpler using the result 1. Indeed, there are exponential number witnesses $|\{I = \mathbf{1}\}| = 3^m$ and $|\{I \neq \mathbf{1}\}| = 2^{2m} - 3^m$. Among all those witnesses, there exists at least one such that to find its output without checking it, an exponential time is required. Otherwise, we can use an exponential number of different witnesses for a polynomial time, which is impossible. Not counting this witness, we can conclude the existence of another one, and so on, until we come to the existence of exponential number depersonalized witnesses. Any algorithm A has to compare all digits of X and Y , since otherwise the desired solution might be missed. So, the time for any DTM will be not less than order m . The case of $U = X \cup Y$ can be reduced to the previous one. Considering next pairs of X, Y , and so on, we complete a proof.

RESULT 6. For any DTM, the time

$$T(\cup_{k=1}^m (X_k \cap Y_k) =? \mathbf{1}), T(\cup_{k=1}^m (X_k \cap Y_k)) \geq T(X_k, k \in M), \quad (19)$$

where X_k are unknown in advance words from L_m , having potentially super polynomial cases of arbitrary situated arbitrary parts of unbounded size, and Y_k are arbitrary from L_m .

PROOF. It is a consequence from the results 1, 2, 4, and 5 proofs. Indeed, no one of words $X_k \cap Y_k, k \in M$, can be missed, since otherwise solutions of these problems might be missed. If we compute only certain parts of X_k , then we can miss computing the required parts. And if we do not compute X_k , then the solution time can be more than the right side in (19), since there are super polynomial cases of X_k , which all cannot be missed.

RESULT 7. *There is a super polynomial number of set of words $Y_{rk_r}, k_r \in M, r = 1, 2, \dots, t$, from L_m such that for any DTM, the time*

$$T(Y_{1k_1} \cap Y_{2k_2} \cap \dots \cap Y_{tk_t}, k_r \in M, r = 1, 2, \dots, t) \geq m^t. \quad (20)$$

PROOF. It follows from the result 3 and the fact that any DTM has to compute m^t different objects, and hence its time is not less than m^t (see Li, Vitanyi [1997]).

Let us return to the problem P_Q whether $Q_{n,t} = \mathbf{1}$ for any n and t , and rewrite $Q_{n,t}$ in the form

$$Q_{n,t} = \cup (I_{k_1} \cap \dots \cap I_{k_t}) (k_1 \in I'_{k_2} \cap \dots \cap I'_{k_{t-1}} \cap I'_{k_t}, k_2 \in I'_{k_3} \cap \dots \cap I'_{k_{t-1}}, \dots, k_{t-2} \in I'_{k_{t-1}} \cap I'_{k_t}, k_{t-1} \in I'_{k_t}, k_t = t+1, \dots, n), \quad (21)$$

where I_{k_s} are arbitrary words from L_{k_s-1} , disposed in fixed natural ordering on the tape of DTM, the elements of the sets I'_{k_s} are zero-numbers in the words I_{k_s} , the sign $k \in I'$ means $\forall k, k \in I'$. Note that in the case of (21), a possible size of an independent set is equal to $t+1$, not t .

LEMMA 3. *On any DTM, the time $T(Q_{n,t} =? \mathbf{1}) \geq \Theta(n^{t-1})$, where an arbitrary integer $t > 1$ is independent of n .*

PROOF. It consists of the following main parts: 1. Introducing a weaker problem $U_{m,t} =? \mathbf{1}, n = (t+1)m$, where the matrix $\{a_{ij}\}$ has a form of step-matrix with t steps, so that the subscripts or the arguments of the word transform are independent of the words in the union (in contrast to (21)).

2. Introducing a particular case of $U_{m,t} =? \mathbf{1}$ (a problem $V_{m,t} =? \mathbf{1}$) so that its structure can be easily examined. 3. Proving Lemma for $t = 2$, using the result 5. 4. Proving Lemma for $t = 3$, using the results 2, 6, and 7. 5. Proving Lemma in a general case, using the results 3, 6, and 7.

1. Denoting the subsets of any ordered set S by $q|S$ and $S|q$, respectively, on the right of q and on the left of q , including q , let us put

$$n = (t+1)m, t < m; (r-1)m | I'_k | rm = \emptyset, k = (r-1)m+1, \dots, rm; r = 1, \dots, t+1. \quad (22)$$

Under condition (22),

$$Q'_{n,t} = U'_{m,t} = \cup (I'_{k_1} \cap \dots \cap I'_{k_t}) | m (k_1 \in m | I'_{k_2} \cap \dots \cap I'_{k_t} | 2m; \dots; k_{t-1} \in (t-1)m | I'_{k_t} | tm; k_t = tm+1, \dots, n). \quad (23)$$

To check the validity of (23), note that due to (22), $k_{t-1} \in I' k_t$ implies $k_{t-1} \leq tm$; $k_{t-2} \in I' k_{t-1} \cap I' k_t$ implies $k_{t-2} \leq (t-1)m$; ... ; and $k_1 \in I' k_2 \cap \dots \cap I' k_t$ implies $k_1 \leq 2m$.

If $k_1 \leq m$, then $I' k_1 = \emptyset$. Thus, $m < k_1 \leq 2m$, which implies $m \mid I' k_1 = \emptyset$ and $m \mid (I' k_1 \cap \dots \cap I' k_t) = \emptyset$, and therefore $m \mid R'_{n,t} = \emptyset$. If $k_2 \leq 2m$, then $k_1 \leq m$, $I' k_1 = \emptyset$, and hence $2m < k_2 \leq 3m$, and so on. If $k_t \leq tm$, then $k_1 \leq m$, and therefore $k_t = tm+1, \dots, n$.

Let us rewrite (23) in the form

$$U_{m,t} = \cup (X_{1k_1} \cap \dots \cap X_{tk_t}) (k_1 \in X'_{2k_2} \cap \dots \cap X'_{2k_t}; k_2 \in X'_{3k_3} \cap \dots \cap X'_{3k_t}; \dots ;$$

$$k_{t-1} \in X'_{tk_t}; k_t = tm+1, \dots, n), X'_{rk_s} = (r-1)m \mid I'_{k_s} \mid rm, s = r, \dots, t; r = 1, \dots, t, \quad (24)$$

where all $X_{rk_s} \in L_m$ are independent, since they have non-crossing subscripts k_s for each r . Designating them $X_{rsk_s}, k_s = 1, \dots, m; s = r, \dots, t; r = 2, \dots, t$, and $X_{rk_r}, r = 1, \dots, t$, we find

$$U_{m,t} = \cup (X_{1k_1} \cap \dots \cap X_{tk_t}) (k_1 \in X'_{22k_2} \cap \dots \cap X'_{2tk_t}; k_2 \in X'_{33k_3} \cap \dots \cap X'_{3tk_t}; \dots ;$$

$$k_{t-1} \in X'_{tk_t}; k_t = 1, \dots, m). \quad (25)$$

2. Assuming that the words $X_{sk_s}, k_s = 1, \dots, m; s = 1, \dots, t$, and $X_{2sk_s} = Y_{sk_s}, k_s = 1, \dots, m; s = 2, \dots, t$, are arbitrary from L_m and putting $X'_{rrk_r} \cap \dots \cap X'_{rtk_t} = M, r = 3, \dots, t$, we find a particular value of $U_{m,t}$ as $V_{m,t}$, which we use in the form

$$V_{m,t} = \cup (S_{k_t} \cap X_{tk_t}) (k_t \in M), S_{k_t} = \cup (S_{k_{t-1}k_t} \cap X_{t-1k_{t-1}}) (k_{t-1} \in M), \dots, S_{k_3 \dots k_t} = \cup (S_{k_2 \dots k_t}$$

$$\cap X_{2k_2}) (k_2 \in M), X_{1k_1} = X_i, S_{k_2 \dots k_t} = \cup X_i (i \in Y'_{2k_2} \cap \dots \cap Y'_{tk_t}), k_2, \dots, k_t \in M. \quad (26)$$

Since $U_{m,t}$ is a particular case of $R_{m,t}$ and $V_{m,t}$ is a particular case of $U_{m,t}$, we have

$$T(R_{n,t} = ? \mathbf{1}) \geq T(U_{m,t} = ? \mathbf{1}) \geq T(V_{m,t} = ? \mathbf{1}). \quad (27)$$

3. In the case $t = 2$, we have

$$V_{m,2} = \cup (S_j \cap Y_j) (j = 1, \dots, m), S_j = \cup X_i (i \in Z'_j). \quad (28)$$

Putting $Z'_j = j$, we find $S_j = X_j$. So, due to the result 5 and independence of $X_j \cap Y_j$ from each other, $T(V_{m,2} = ? \mathbf{1}) \geq \Theta(m^2)$.

4. In the case $t = 3$, we have the following word composition

$$V_{m,3} = \cup (S_k \cap Z_k) (k \in M), S_k = \cup (S_{jk} \cap Y_j) (j \in M),$$

$$S_{jk} = \cup X_i (i \in P'_j \cap Q'_k), j, k \in M. \quad (29)$$

We can obtain the estimate $T(V_{m,3} = ? \mathbf{1}) \geq \Theta(m^2)$, since the case $t = 2$ can be considered as a particular case of $t = 3$. We consider here another approach, which can be generalized later on.

On the basis of the result 2, parts of S_{jk} and $P_j \cap Q_k$ can be arbitrary situated arbitrary words from $L_{2\log m}$, and parts of S_k can be arbitrary situated arbitrary words from $L_{2\log(2\log m)}$. No one of those words can be missed, since otherwise a solution of the problem $V_{m,2}=? \mathbf{1}$ might be missed. Indeed, if one of those words missed, then in the case when the respective missed word $S_{k^*} = \mathbf{1}$, but the other words $S_k \neq \mathbf{1}$, we would miss a solution. So, on the basis of the result 6, the time for a problem $\cup (S_k \cap Z_k) (k \in M) =? \mathbf{1}$ is not less than the time for computing all words $S_k, k \in M$. In turn, the time for computing $S_k, k \in M$, is not less than the time for computing all words S_{jk} , which is not less than the time for computing all m^2 different words $P_j \cap Q_k$, corresponding those parts of words S_{jk} (by the same reason as in the case of the result 6).

Thus, it follows from the result 7 that the time

$$T(V_{m,3}=? \mathbf{1}) \geq m^2. \quad (30)$$

5. In the general case, we have the word composition (26), and similarly to the case $t = 3$, due to the results 3, parts of words $Y_{2k_2} \cap \dots \cap Y_{tk_t}$ and $S_{k_2 \dots k_t}$ can be arbitrary situated arbitrary words from $L_{(t-1)\log m}$, parts of words $S_{k_3 \dots k_t}$ can be arbitrary situated arbitrary words from $L_{2\log((t-1)\log m)}$, and so on, at last parts of words S_{k_t} can be arbitrary situated arbitrary words from $L_{2\log(2\log \dots 2\log((t-1)\log m) \dots)}$, where $2\log$ are repeated $t-2$ times.

Therefore, on the basis of the result 6, the time for a problem $\cup (S_{k_t} \cap X_{tk_t}) (k_t \in M) =? \mathbf{1}$ is not less than the time for computing all unknown words $S_{k_t}, k_t \in M$. In turn, the time for computing all words $S_{k_t}, k_t \in M$, is not less than the time for computing all unknown words $S_{k_{t-1}k_t}, k_{t-1}, k_t \in M$, which by recursion is not less than the time for computing all unknown different words $S_{k_2 \dots k_t}, k_2, \dots, k_t \in M$, which is not less than the time for computing all m^{t-1} different words $Y_{2k_2} \cap \dots \cap Y_{tk_t}, k_2, \dots, k_t \in M$, corresponding to those parts of the words $S_{k_2 \dots k_t}$. Since there exist super-polynomial number of different sets of all intermediate words, including intersections $Y_{2k_2} \cap \dots \cap Y_{tk_t}, k_2, \dots, k_t \in M$ (no one of them can be missed), if we do not compute each of their realization or any part of them, then we have to store and use them on DTM, which requires in the total more time than direct computing of those intersections.

Thus, on the strength of the results 6 and 7, the time

$$T(V_{m,t}=? \mathbf{1}) \geq m^{t-1}. \quad (31)$$

5. Main results

THEOREM 1. $P \neq NP$.

PROOF. The problem P_1 , i.e., the problem of independent set with the representation (3) of the graph G , is NP-complete, since the NP-complete problem of an independent set can be transformed to P_1 for a polynomial time. Let $P = NP$. Then the problem P_1 is also in P . Therefore, there exists a constant C , independent of n and t , for which $T(P_1) = T(Q_{n,t}=? \mathbf{1}) < n^C$, on the basis of Lemma 2. But verification of $Q_{n,t}=? \mathbf{1}$ can require the time more than n^C in order for any DTM due to Lemma 3 in the case of $t > C+1$. That contradiction means that one of the NP-complete problems and hence all of them do not belong to P . But they all belong to NP .

The result of Lemma 3 is also a principal one for another reason (see, van Melkebeek, 2007), because the size of the input in (21) is order n^2 , and the required time is not less than $\Theta[(n^2)^{t/2-1/2}]$, where t is arbitrary independent of n .

In a manner of van Melkebeek [2007], our result can be formulated as the following theorem.

THEOREM 2. *Any NP complete problem cannot be solved by any DTM that runs in times n^t and space n^τ , where t and τ are any positive numbers independent of n .*

Up to poly-logarithmic factors, our result as the results in van Melkebeek [2007], is likely also robust with the respect to the details of each of the basic models of computation: deterministic, randomized, and quantum machines.

6. Discussion

Many computer scientists share the point of view of Dr. D. S. Johnson, who noted,

“A common failing in P vs. NP proofs is the step in which the author says (without proof), ‘any algorithm for solving this problem must do it in the following way.’”

Let us dwell on the question whether the above estimates are valid for *all* solution algorithms.

In Lemma 3, the case $t = 3$, the respective problem can be stated as: Given arbitrary words $X_i, Y_j, Z_k, P_j, Q_k, i, j, k \in M$, from L_m , decide $V_{m,3} =? \mathbf{1}$, where $V_{m,3}$ is given, in particular, by the formula (29). The properties of $V_{m,3}$, following from that formula, are remained for any other representation of $V_{m,3}$. So, the same problem can be stated as: Given words $S_k, Z_k, k \in M$, from L_m , where S_k (off-line or on-line) can have arbitrary situated arbitrary parts from $L_{2 \log (2 \log m)}$, decide $V_{m,3} = \cup (S_k \cap Z_k) (k \in M) =? \mathbf{1}$. It follows from this that $V_{m,3} = \mathbf{1}$, iff $S_k \cap Z_k = \mathbf{1}, k \in M$. And $V_{m,3} \neq \mathbf{1}$, iff at least one of $S_k \cap Z_k \neq \mathbf{1}$. Since the unknown in advance case is possible that all $S_k \cap Z_k = \mathbf{1}, k \in M$, the required time can be not less than $T(S_k, k \in M)$. In turn, S_k have properties of $\cup (S_{jk} \cap Y_j) (j \in M)$, where words S_{jk} (off-line or on-line) can have arbitrary situated arbitrary parts from $L_{2 \log m}$, and hence $T(S_k, k \in M)$ can be not less than $T(S_{jk}, j, k \in M)$, which, by the same reason, can be not less than $T(P_j \cap Q_k, j, k \in M) \geq m^2$, where $P_j \cap Q_k$ can be computed off-line or can be obtained on-line (in the last case the total time can be more than in the former case). It is clear that the above reasoning is also valid in the general case of t and the formula (26).

The main Lemma 3 seems to be in contradiction with the result, for example, in the paper by Woeginger [2005]: ‘ $3k$ -CLIQUE problem for a p -vertex graph can be solved in $O(p^{\omega k})$ time and $O(p^{2k})$ space,’ where $\omega < 2.376$ is the square matrix multiplication exponent.’

However, the formulation above omitted that the parameter k is fixed (see Downey & Fellows [1999]). It should be emphasized that the time not less than order p^{3k-1} due to Lemma 3 is obtained for any DTM under arbitrary k independent of p .

There are much more traps for the author to avoid.

A well-known example of such a trap is the so-called ‘natural proof’ (see Razborov & Rudich [1999]). Possible answers to this example: there is an important detail of the proof in question that is absent in any natural proof.

Indeed, there exists an exponential number of instances for which the solution time on any DTM is not less than n^{t-1} in order, but we cannot indicate *naturally* (constructively) any one of them (this evidence is connected with Kolmogorov complexity and investigated in detail by Li & Vitani [1997]).

One more common trap is the relativization of the proofs that can be done for all known natural proofs, i.e., those proofs remain valid if one uses, in addition, any oracle (see Davis, Sigal & Weyuker [1994]). However, in the case of P versus NP problem, there exist oracles A and B such that $P^A = NP^A$, but $P^B \neq NP^B$ (Garey & Johnson [1979]).

So, any proof that P is not NP cannot be relativized and hence cannot be natural. A possible answer to this example is that our proof does not belong to the class of ordinary proofs, which can be relativized. In addition, as explained in Johnson [1990], a possible answer is that individual problems do not relativize.

Let us, in conclusion, dwell on a debatable question, whether the proof of the main Lemma implies that namely the independent set problem requires the time not less than n^{t-1} in order, because the graph $G = (V, E)$ can be represented by many other ways.

Let a word J instead of I be given, $|J'| = |I'|$. Besides, there are computable functions: $J \rightarrow V_{m,t} =? \mathbf{1}$, and $I \rightarrow J$, $T(I, J) < n^a$, where a is any fixed positive number, independent of n . Then

$$T(J, V_{m,t} =? \mathbf{1}) \geq T(I, V_{m,t} =? \mathbf{1}) - T(I, J) > \Theta(n^{t-1}) - n^a > \Theta(n^{t-1}), t > a + 1. \quad (*)$$

A proof of (*) follows rather easily from definitions (1). Thus, using any other representation of the graph, requiring polynomial time, cannot reduce our estimates.

7. Open problems

It seems that in order of increasing importance, we have at least the following open problems:

- 7.1. Is co-NP equal to NP?
- 7.2. Is P^{NP} equal to NP?
- 7.3. Do one-way functions exist?
- 7.4. What is the time for verification of $Q_{n,t} =? \mathbf{1}$, when t is arbitrary?
- 7.5. Can the estimate of number hard witnesses in Lemma 3 be essentially strengthened?
- 7.6. Is it possible that a non-algorithmic-type computer for an effective solution of NP-hard problems could be constructed?
- 7.7. If yes, then how?

A certain result with respect to the last problem can be seen in Ivanov & Ivanova [2006] and Ivanov [2009]. It is shown that the probability of a desired solution for a hard problem of global minimization of function of n variables can be very close to 1, so that in the nature, for example, in the case of proteins folding, it can be found instantly.

In this connection, also see Braverman & Cook [2006].

Remarks: The author started the investigation of the problem NP versus P many years ago (in connection with his investigation of optimal algorithms (Ivanov books [1986, 1999])). He published (Ivanov paper [1999]) an attempt to prove $NP \neq P$ based on lemmas 1, 2-types and the result 1 above and certain postulates similar to some of the postulates in Li & Vitani [1997]. He left a similar (by logic) proof to the reader on the WEB in March, 2005 (see [www.P](#) versus NP problem/P-versus-NP page).

REFERENCES

- BRAVERMAN M., COOK S. 2006. Computing over the reals: foundations for scientific computing, *Notices of the AMS*, 53, # 3, 318-329
- COOK, S.A. 1971. The complexity of theorem-proving procedures, In *Proceeding of 3rd ACM Annual Symposium on the Theory of Computing*, 151-158.
- DAVIS M.D., SIGAL R., AND WEYUKER E.J. 1994. *Computability, Complexity, and Languages. Fundamental of Theoretical Computer Sciences*, Academic Press.
- DOWNEY R.G., FELLOWS M.R. 1999. *Parameterized Complexity*, Springer.
- GAREY, M.R., AND JOHNSON, D.S. 1979. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. W.H. Freeman and Co., New York.
- HENNIE, F.C. 1965. One-Tape, Off-Line Turing Machine Computations, *Information and Control*, 8, 553-578.
- HOPCROFT, J.E., MOTWANI, R., ULLMAN, J.D. 2006. *Introduction to Automata Theory, Languages, and Computations*, Addison Wesley.
- IVANOV, V.V. 1986. *Methods of Computation. Guidance*. Naukova dumka, Kiev. (In Russian).
- IVANOV, V.V. 1999. *Model Development and Optimization*. Kluwer Academic Publishers.
- IVANOV, V.V. 1999. On $NP \neq P$. In *Theory of Computation*, Glushkov Institute of Cybernetics, Kiev, 184-188. (In Russian).
- IVANOV, V.V. 2009. Global minimum formula and its implications. *Proceeding of International Symposium 'The Issues of Calculation Optimization (ISCOPT-XXXV)*, Glushkov Institute of Cybernetics, Kiev, vol. I, 273-278.
- IVANOV, V.V., AND IVANOVA, N.V. 2006. *Mathematical Models of the Cell and Cell Associated Objects*, Elsevier B.V.
- JOHNSON, D.S. 1990. A Catalog of Complexity Classes, In *Handbook of the Theoretical Computer Science*. 1990., Elsevier Science Publishers B.V., 66-150.
- JOHNSON, D.S. 1981-2006. The NP-completeness column: an ongoing guide. *J. Algorithms*, starting with the first edition in 2, 393-405, and continuing in *ACM Transactions on Algorithms*.
- KNUTH, D.E. 2002. All questions answered, *Notices of the AMS*, 49, # 3, 318-324.
- LEEUEWEN, J. 1990. Graph algorithms, In *Handbook of the Theoretical Computer Science*. 1990., Elsevier Science Publishers B.V., 525-631.
- LI, M., AND VITANYI, P.M.B. 1997. *An Introduction to Kolmogorov Complexity and Its Applications*, Springer.
- RASBOROV A.A., AND RUDICH S. 1999. Natural Proofs. In *Proceeding of 26th Annual Symposium on the Theory of Computing*, 204-213.
- SIPSER, M. 1992. The history and status of the P versus NP question, In *Proceeding of 24th Annual Symposium on the Theory of Computing*, 603-619.
- van MELKEBEEK D. 2007. A survey of lower bounds for satisfiability and related problems, *University of Wisconsin-Madison, ECCC Report TR-97-099*, 61 pp.
- WÖEGINGER, G.J. 2005. Open problems around exact algorithms, gwoegi@win.tue.nl, I-14.

Dr. Viktor V. Ivanov

Address: 831 Grove St. North,
St. Petersburg, FL 33701 USA
Phone: (617) 888 0318
E-mail: ivanvvn@hotmail.com