A PROOF OF NP ≠ P

Viktor V. Ivanov

Abstract. *A proof* (*on about* 7 *pages*) *is based on better estimates of lower bounds on time complexity that hold for all solution algorithms. Almost no special knowledge other than logical and combinatorial efforts is needed to understand the proof. The main steps and ideas of the whole proof can be seen in the introduction.*

Keywords: *NP-complete problem*, *Time complexity*, *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 its applications (see, for example, [1]-[7]). We dwell on a typical NP-complete problem of INDEPENDENT SET (M. Garey, D. Johnson, [2], p. 194): INSTANCE: graph $G = (V, E)$, positive integer K. QUESTION: Does G contain an independent set of size K or more, i.e., a subset $V' \subseteq V$ such that $|V'| \geq K$ and such that no two vertices in V' are joined by an edge in E? We denote this problem as P_S.

All NP-complete problems are in 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 the problems solvable on classical computers, e.g. deterministic Turing machines in a polynomial time. Below is a proof that NP ≠ P.

The necessary subsidiary notions can be found in details in [2, 4, and 5]. To help the reader to understand the proof, a short review of its main steps and ideas is given here.

The problem P_S, under the representation of the graph G in the form of the list of binary words, the total size n^2 in order, is restated, as the problem P_V, verifying if a certain set transforms of given sets is empty or not (Lemmas 1 and 1+).

Lemmas 2, 3, and 4 are auxiliary for the proof of Lemma 5. Lemmas 2 and 3 are rather simple-combinatorial. Lemmas 4 and 5 are in the case, when input of P_S has the size n^2 in order and $K = t$ is independent of n. Lemma 4 is almost trivial, since it means that the time for computing n^t distinct objects for any n on any classical computer is not less than n^t.

Lemma 5 is the most important and difficult to prove. Its proof shows existence of exponential number 'hard' instances for which the time of solutions of the problem P_V is near the longest and not less than n^{t-1} in order, since for those instances any solution algorithm has to compute not less than order n^{t-1} intermediate distinct objects. The main ideas here are introducing exponential number instances for which a certain property of the respective sub-problems can be valid or not unless we check it, and showing that any solution algorithm has to solve all those sub-problems.

A proof of the main theorem on NP ≠ P follows easily from Lemmas 1+ and 5. As an implication of the main theorem, we have also a negative result on capability of quantum computers with respect to NP-hard problems.

2. Preliminaries

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

$$I_j = (a_{1j} \dots a_{j-1j}),\ j = 2, \dots, n; \qquad a_{ij} = 0 \vee 1,\ 1 \le i < j \le n; \tag{1}$$

where $a_{ij} = 1$, iff there is the 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}. The respective input for the problem P_S has the size order $n(n-1)/2 + \log_2 K$, $1 \le K \le n$.

We introduce also the binary words

$$A_t = (a_{k,1k,2}\, a_{k,1k,3} \dots a_{k,1k,t}\, a_{k,2k,3}\, a_{k,2k,4} \dots a_{k,2k,t} \dots a_{k,t-2k,t-1}\, a_{k,t-2k,t}\, a_{k,t-1k,t}),$$

$$]A_t[= a_{k,1k,2} + a_{k,1k,3} + \dots + a_{k,1k,t} + a_{k,2k,3} + a_{k,2k,4} + \dots + a_{k,2k,t} + \dots + a_{k,t-1k,t}, \tag{2}$$

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_t.

Let the problem P_1 be P_S with the representation (3), $K = t > 1$, and P_2 with the same representation be the problem if there exists or not a word A_t with $]A_t[= 0$ for any n and t. Since $]A_t[= 0$, iff $(k,1, \dots, k,t)$ is independent set, the problems P_1 and P_2 are restatements of each other.

Denoting by L_n all 2^n binary words, let for any word $I \in L_n$, I' be the ordered set consisting of zero numbers in I ($]I[= n - |I'|$ = number of 1-digits in I) and $\underline{I}$ be complement of I to $\mathbf{0} = (0 \dots 0)$. And let for any words I_1 and I_2 from L_n the intersection $I_1 \cap I_2$ (the union $I_1 \cup I_2$) be a word from 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 $I'_1 = (1,3)$, $I'_2 = (2,3)$, and hence $I'_1 \cap I'_2 = (3)$ and $I'_1 \cup I'_2 = (1,2,3)$.

Lemma 1. *The following relations are valid*:

$$]A_t[\ne 0,\ \forall\ k,1, \dots, k,t:\ 1 \le k,1 < \dots < k,t \le n,\ t > 1, \tag{3}$$

iff

$$R_{n,t-1} = \cup\, (I_{k,2} \cap I_{k,3} \cap \dots \cap I_{k,t-1} \cap I_{k,t})\ (\forall\ k,2, \dots, k,t:\ k,t \in (t, \dots, n),$$

$$k,t-1 \in I'_{k,t}, \dots, k,2 \in I'_{k,3} \cap \dots \cap I'_{k,t}) = \mathbf{1} = (1, \dots, 1), \tag{4}$$

where I'_j are ordered sets of zero-numbers in the binary words $I_j = (a_{1j} \dots a_{j-1j})$.

Besides, if $R_{n,t-1} = \mathbf{1}$, then $R_{n,s} = \mathbf{1}$, $t - 1 < s \le n$, and if $R_{n,t-1} \ne \mathbf{1}$, then there exists an independent set

$$V_t = (k,1, \dots, k,t),\ |V_t| = t. \tag{5}$$

Proof. It is based on a simple fact that the intersections $I_{k,r} \cap \dots \cap I_{k,t} \ne \mathbf{1}$, $r = 2, \dots, t$, iff there exist $k,r-1 \in I'_{k,r} \cap \dots \cap I'_{k,t}$ such that $a_{k,r-1k,r} + \dots + a_{k,r-1k,t} = 0$, $r = 2, \dots, t$, and hence their sum is also equal to 0. Therefore, if $R_{n,t-1} \ne \mathbf{1}$, then there exist $k,2, \dots, k,t$:

(6) $$k,t \in (t, \ldots, n);\ k,t-1 \in I'_{k,t};\ \ldots;\ k,2 \in I'_{k,3} \cap \ldots \cap I'_{k,t},$$

such that the word $I_{k,2} \cap \ldots \cap I_{k,t} \neq \mathbf{1}$, and hence there exists $k,1 \in I'_{k,2} \cap \ldots \cap I'_{k,t}$, for which we have $]A_t[= 0$, and backward: if $]A_t[= 0$, then all $a_{k,rk,s} = 0$, $r = 1, \ldots, t-1$; $s = r+1, \ldots, t$, from where it follows that $I_{k,2} \cap \ldots \cap I_{k,t} \neq \mathbf{1}$ under the restriction (4), and hence $R_{n,t} \neq \mathbf{1}$. If $R_{n,t-1} = \mathbf{1}$, then $R_{n,s} = \mathbf{1}$, $t-1 < s \leq n$, since $R'_{n,s} \subseteq R'_{n\,t-1}$, $t-1 < s$, but $R'_{n,t-1} = \emptyset$. If $R_{n,t-1} \neq \mathbf{1}$, then (5) is valid. ∎

Given the same input as for the problems P_1, let P_V be the problem of verification of

(7) $$R_{n,t-1} =? \mathbf{1},\ 1 < t \leq n,$$

for any n and t. It is clear that the problems P_1, P_2, and P_V are restatements of each other.

Let $P = P(I, R)$ be a problem with a finite input domain I and output domain R, and let A be an algorithm for the solution of P on classical computer C. Note that for decision problems $R = 0 \vee 1$.

The time complexities of the solution of P on C are given by

(8) $$T_p(P, A) = \sup t(x, y, A)\ (\forall x: |x| = p, x \in I),\ T(P) = T_p(P) = \inf T_p(P, A),$$

where $t(x, y, A) = t(x, A) = t(y, A)$ is the time for A on input $x \in I$, with output $y \in R$, and inf is taken over all solution algorithms A. We use also $T(y)$ instead of $T(P)$, when y is the explicit presentation of the problem P.

A problem $P \in \mathrm{P}$ if there exist an algorithm A and a constant c such that $T_p(P, A) < p^c$ for all p.

As an implication of the definitions above as well as Lemmas 1, we have

Lemma 1+. $T(P_1) = T(P_2) = T(P_V) = T(R_{n,t-1} =? \mathbf{1})$.

Proof. Having a solution of P_1, we have also a solution of P_2 and P_V, and backwards. ∎

3. Auxiliary lemmas

Lemma 2. *While X_k, $k = 1, 2, \ldots, t$, generate independently all possible 2^l words from L_l,*

(9) $$I_t = \cap X_k\ (k = 1, \ldots, t),\ U_t = \cup X_k\ (k = 1, \ldots, t),\ 1 < t < l + 1,$$

generate all $\binom{l}{r}$ values with $]I_t[$, $]U_t[= r$ respectively $(2^t - 1)^r$, $(2^t - 1)^{l-r}$ times, , $r = 1, \ldots, l$.

Proof. In the case of $t = 2$, we have

(10) $$3^l = (2 + 1)^l = \sum \binom{l}{s} 2^s\ (s = 0, \ldots, l),$$

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

(11) $$(2^{s+1} - 1)^l = \sum \binom{l}{k} (2^{s+1} - 2)^k\ (k = 0, \ldots, l) = \sum \binom{l}{k} 2^k (2^s - 1)^k\ (k = 0, \ldots, l),$$

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 $l - k$ 1-digits corresponding to 0-digits of I_s.

The result $(2^t - 1)^{l-r}$ follows from above due to $\underline{U_t} = \cup \underline{X_k}\ (k = 1, \ldots, t)$. ∎

Lemma 3. *Let* X_k, $k = 1, \ldots, m$, *be words from* L_m *with the property*

(12) $$X'_k = (.\,, k, .),\ k = 1, \ldots, m,$$

where $(.\,, k, .)$ *is arbitrary subset from* $(1, \ldots, m)$, *containing element* k. *Then it is possible that*

(13) $$U'_{k,1,\ldots,k,t} = \cup\, X'_k\ (k = k,1, \ldots, k,t) = (.\,, k,t, .\,, k,t-1, .\,, \ldots, .\,, k,1, .)$$

is arbitrary set, *containing* $k,t, k,t-1, \ldots, k,1$, t *is independent of* m.

Besides, *the subsets* $(.\,, k, .)$ *can be selected in such a way that for even* m

(14) $$|U'_{k,1,\ldots,k,t}| = m/2,$$

and all $U'_{k,1,\ldots,k,t}$ *in* (13) *and* (14) *can be different*.

Proof. It follows from Lemma 4, since the union of arbitrary sets can be also arbitrary. Besides, (14) is valid, since all $\binom{m}{m/2}$ combinations from $(1, \ldots, m)$ contain all $\binom{m}{t}$ combinations, $t < m/2$. ∎

Lemma 4. *We have*

(15) $\mathrm{T}\,[(k,1, \ldots, k,t), k,t \in K; k,t\text{-}1 \in K\backslash(k,t); \ldots; k,1 \in K\backslash(k,2, \ldots, k,t)] > \Theta(m^t)$, $K = (1, 2, \ldots, m)$,

where T *means the time of computing all* $(k,1, \ldots, k,t)$ *for any* t *and* m, t *is independent of* m.

Proof. It is trivial, since the left side of (15) is the time for computing $\Theta(m^t)$ distinct objects. ∎

4. Main lemma

Let us return to the problem P_V if $R_{n,t-1} = \mathbf{1}$ or not for any n and t, and rewrite $R_{n,t}$ in the form

(16) $$R_{n,t} = \cup\, (I_{k,1} \cap \ldots \cap I_{k,t})\ (k,t = t+1, \ldots, n; k,t\text{-}1 \in I'_{k,t}; \quad \ldots; k,1 \in I'_{k,2} \cap \ldots \cap I'_{k,t}),$$

where $I_{k,s}$ are arbitrary words from $L_{k,s-1}$, the elements of the sets $I'_{k,s}$ are zero-numbers in $I_{k,s}$, and the sign '$r \in S$' for any S means (here and later on) 'for all r in set S'.

Lemma 5. *The time* $\mathrm{T}\ (R_{n,t} =?\ \mathbf{1}) > \Theta(n^{t-1})$, *where a positive integer* t *is independent of* n.

Proof. It consists of the following main parts:

1. Introducing a weaker problem $V_{m,t} =?\ \mathbf{1}$, $n = (t+1)m$, when the matrix of input data $\{a_{ij}\}$ has a form of step-matrix with $t + 1$ steps, so words on each step can be independent of the other words.
2. Considering a special structure of the weaker problem, consisting of t levels, the initial problem $V_{m,t} =?\ \mathbf{1}$ is on the first upper level.
3. Proving the existence of exponential number instances on which the respective sub-problems for each level may have contradictory solutions.
4. Proving that for any solution algorithm to solve all the sub-problems for each level, we must solve all the sub-problems of the next level.
5. Proving the desired estimate.

Thus,

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

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

Under the condition (17), we find that

$$R'_{n,t} = R'_{n,t}|m = U'_{m,t} = \cup (I'_{k,1} \cap \dots \cap I'_{k,t})|m\ (k,t = tm+1, \dots, n;\ k,t\text{-}1 \in (t-1)m|I'_{k,t}|tm; \dots;$$

(18) $\quad k,1 \in m|I'_{k,2} \cap \dots \cap I'_{k,t}|2m).$

To check validity of (18), note that due to (17), $k,t-1 \in I'_{k,t}$ implies $k,t-1 \le tm$; $k,t-2 \in I'_{k,t-1} \cap I'_{k,t}$ implies $k,t-2 \le (t-1)m$; … ; and $k,1 \in I'_{k,2} \cap \dots \cap I'_{k,t}$ implies $k,1 \le 2m$. If $k,1 \le m$, then $I'_{k,1} = \emptyset$. Thus, $m < k,1 \le 2m$, which implies $m|I'_{k,1} = \emptyset$ and $m|(I'_{k,1} \cap \dots \cap I'_{k,t}) = \emptyset$, and hence $m|R'_{n,t} = \emptyset$. If $k,2 \le 2m$, then $k,1 \le m$, $I'_{k,1} = \emptyset$, and hence $2m < k,2 \le 3m$, and so on. If $k,t \le tm$, then $k,1 \le m$, and hence $k,t = tm+1, \dots, n$. We rewrite (18) in the form

$$U_{m,t} = \cup (X,1_{k,1} \cap \dots \cap X,1_{k,t})\ \ (k,t = tm+1, \dots, n;\ k,t\text{-}1 \in X,t'_{k,t}; \dots;\ k,1 \in X,2'_{k,2} \cap \dots \cap X,2'_{k,t}),$$

(19) $\quad X,r'_{k,s} = (r-1)m|I'_{k,s}|rm,\ s = r, \dots, t;\ r = 1, \dots, t,$

where all $X,r_{k,s} \in L_m$ are independent, since they have non-crossing subscripts k,s for each r. Denoting them $X,r,s_{k,s}$, $k,s = 1, \dots, m$; $s = r, \dots, t$; $r = 2, \dots, t$, and $X,r_{k,r}$, $r = 1, \dots, t$, we have

(20) $\quad U_{m,t} = \cup (X,1_{k,1} \cap \dots \cap X,t_{k,t})\ (k,t = 1, \dots, m;\ \dots;\ k,1 \in X,2,2'_{k,2} \cap \dots \cap X,2,t'_{k,t}).$

Assuming that the words $X,r_{k,r}$, $r = 2, \dots, t$, are arbitrary from L_m, letting $K = (1, \dots, m)$, and

(21) $X,r,s'_{k,s} = K\backslash(k,s)$, $k,s = 1, \dots, m$; $s = r, \dots, t$; $r = 2, \dots, t$; $X,1'_{k,1} = (.\ , k,1, .\ )$, $k,1 = 1, \dots, m$,

where $(.\ , k,1, .\ )$ have the same sense as in Lemma 3, we find the particular value of $U_{m,t}$ as

(22) $\quad V_{m,t} = \cup (X,1_{k,1} \cap \dots \cap X,t_{k,t})\ \ (k,t \in K;\ k,t\text{-}1 \in K\backslash(k,t);\ \dots;\ k,1 \in K\backslash(k,2, \dots, k,t)).$

We accept the following notation for the input domain I_V of $V_{m,t}$:

(23) $\quad I_V = (Q_1, Q_2, \dots, Q_t),\ Q_1 = \{X,1_{k,1}\},\ Q_2 = \{X,2_{k,2}\},\ \dots,\ Q_t = \{X,t_{k,t}\},$

and we use (22) in the form

$$V_{m,t} = \cup (S_{k,t} \cap X,t_{k,t})\ (k,t \in K), \quad S_{k,t} = \cup (S_{k,t-1,\,k,t} \cap X,t\text{-}1_{k,t-1})\ (k,t\text{-}1 \in K\backslash(k,t)), \dots,$$

$$S_{k,3,\,\dots\,,k,t} = \cup (S_{k,2,\,\dots\,,k,t} \cap X,2_{k,2})\ (k,2 \in K\backslash(k,3, \dots, k,t)),\ S'_{k,2,\,\dots\,,k,t} = \cup X,1'_{k,1}\ (k,1 \in .$$

(24) $\quad K\backslash(k,2, \dots, k,t)) = K\backslash(X,1'_{k,2} \cup \dots \cup X,1'_{k,t}) = K\backslash(.\ , k,2, .\ , \dots, .\ , k,t, .)\ ,$

assuming that due to Lemma 5 all sets $S'_{k,2,\,\dots\,,k,t}$ are different and $|(.\ , k,2, .\ , \dots, .\ , k,t, .)| = m/2$.

On the strength of Lemma 1, we have

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

2. Let us consider in more detail the structure of the relations (24).

It is clear that the size $|K\backslash(k,r, \dots, k,t)| > m - t = \Theta(m)$, and $|S'_{k,2, \dots, k,t}| = m/2$. Besides,

$$S'_{k,r, \dots, k,t} \subseteq \cup X'_{k,r-1}\ (k,r-1 \in K\backslash(k,r, \dots, k,t)),\ S'_{k,r, \dots, k,t} \subseteq \cup S_{k,r-1, \dots, k,t}\ (k,r-1 \in K\backslash(k,r, \dots, k,t))$$

$$(26) \qquad \subseteq \dots \subseteq \cup X,1'_{k,1}\ (k,1 \in K\backslash(k,r, \dots, k,t)) = K\backslash(., k,r, ., \dots, ., k,t, .),\ r = 3, \dots, t.$$

It follows from (26) that $S'_{k,r, \dots, k,t}$ can be all different, $|S'_{k,r, \dots, k,t}| \le m/2+r-2$, and $|S'_{k,r, \dots, k,t}|$ can be from $m/2+r-2$ to 0, while $|X,p'_{k,p}|$ are changing from m to 0, $2 \le p \le r-1$, $r = 3, \dots, t$.

Note that in the expression $V_{m,t} = \cup (S_{k,t} \cap X,t_{k,t})$ $(k,t \in K)$, $S_{k,t}$ does not depend on $\{X,t_{k,t}\}$, $k,t \in K$. Similarly, in the expression $S_{k,t} = \cup (S_{k,t-1,\ k,t} \cap X,t-1_{k,t-1})$ $(k,t-1 \in K\backslash(k,t))$, $S_{k,t-1,\ k,t}$ does not depend on $\{X,t_{k,t}, X,t-1_{k,t-1}\}$, k,t, $k,t-1 = 1, \dots, m$, and so on.

3. Let us show that there are exponential number instances from Q_t, for which $S_{k,t} \cap X,t_{k,t} = \mathbf{1}$ or not is possible for all $k,t \in K$.

The desired instances are all $^{-}X,t_{k,t}$, and $^{+}X,t_{k,t}$ such that

$$(27) \qquad S_{k,t} \cap {}^{-}X,t_{k,t} = \mathbf{1};\ S_{k,t} \cap {}^{+}X,t_{k,t} \ne \mathbf{1},\ k,t \in K.$$

Since due to (26), $r = t$, we have $|S'_{k,t}| < m/2+t$, the set $\{^{-}X,t_{k,t}\}$ can be the size more than $2^{m/2-t}$, and the set $\{^{+}X,t_{k,t}\}$, including all $X,t_{k,t} \supseteq S_{k,t}$, can be the size not less than 2^{m-1}, if $S_{k,t} \ne \mathbf{1}$.

Among those instances, there is at least one such that to find it, an exponential time is required. Otherwise, we can find exponential number instances for a polynomial time, which is impossible. Not counting this instance, we can conclude existence of another similar one, and so on.

Reasoning this way, we can conclude the existence of exponential number instances from Q_t such that for each of them to find it, an exponential time is required.

Similarly, there are exponential number instances from Q_{t-1}, for which $S_{k,t-1,\ k,t} \cap X,t-1_{k,t-1} \cap X,t_{k,t} = \mathbf{1}$ or not is possible for all $k,t \in K$; $k,t-1 \in K\backslash(k,t)$.

The desired instances are all $^{-}X,t-1^{(k,t)}_{k,t-1}$ and $^{+}X,t-1^{(k,t)}_{k,t-1}$ such that

$$(27') \qquad S_{k,t-1,\ k,t} \cap {}^{-}X,t-1^{(k,t)}_{k,t-1} = \mathbf{1};\ S_{k,t-1,\ k,t} \cap {}^{+}X,t-1^{(k,t)}_{k,t-1} \ne \mathbf{1},\ k,t \in K;\ k,t-1 \in K\backslash(k,t).$$

Since due to (26), we have $|S'_{k,t-1,\ k,t}| < m/2+t$, the sets $\{^{-}X,t-1^{(k,t)}_{k,t-1}\}$ can be the size more than $2^{m/2-t}$, and the sets $\{^{+}X,t-1^{(k,t)}_{k,t-1}\}$ can be the size not less than 2^{m-1}.

Among those instances, there is at least one such that to find it, an exponential time is required. Otherwise, we can find exponential number instances for a polynomial time, which is not possible. Not counting this instance, we can conclude existence of another similar one, and so on, until we come to the existence of exponential number instances from Q_{t-1} such that for each of them to find it, an exponential time is required.

Continuing this reasoning, we can conclude the existence of an exponential number of instances $^{-}X,2^{(k,3, \dots, k,t)}_{k,2}$ and $^{+}X,2^{(k,3, \dots, k,t)}_{k,2}$ from Q_2 such that

$$S_{k,2, \dots, k,t} \cap {}^{-}X,2^{(k,3, \dots, k,t)}_{k,2} = \mathbf{1};\ S_{k,2, \dots, k,t} \cap {}^{+}X,2^{(k,3, \dots, k,t)}_{k,2} \ne \mathbf{1},$$

$$(27'') \qquad k,t \in K.;\ k,t-1 \in K\backslash(k,t);\ \dots;\ k,2 \in K\backslash(k,3, \dots, k,t),$$

and for each of them to find it, an exponential time is required.

But to check for each of instances from Q_2, if $S_{k,2, \dots, k,t} \cap X,2^{(k,3, \dots, k,t)}_{k,2} = \mathbf{1}$ or not, the time less than exponent of m is required.

4. Let A be any solution algorithm for the problem $V_{m,t} =? \mathbf{1}$. Then A has to solve this problem for any particular inputs from the domain I_V. So, for the inputs (27), A has to solve all the problems $V_{m,t} = S_{k,t} \cap X,t_{k,t} =? \mathbf{1}$, $S_{k,t} \neq \mathbf{1}$, $k,t \in K$, and all the problems $V_{m,t} = S_{k,t} =? \mathbf{1}$, in the cases $X,t_{k,t} = S_{k,t}$, $k,t \in K$. If one of them is missed, the problem $V_{m,t} =? \mathbf{1}$ cannot be solved for all inputs.

It means that for the inputs (27), the notations t ($S_{k,t} \cap X,t_{k,t} =? \mathbf{1}, A$) and t ($S_{k,t} =? \mathbf{1}, A$) are valid for all $k,t \in K$ and the same algorithm A as in the case t ($V_{m,t} =? \mathbf{1}, A$).

Similarly, let A be any solution algorithm for the problems $S_{k,t} =? \mathbf{1}$, $k,t \in K$. Then A has to solve these problems for any particular inputs from the domain $I_V \backslash Q_t$. So, for the inputs with the property (27'), A has to solve all the problems $S_{k,t-1,\,k,t} \cap X,t-1_{k,t-1} =? \mathbf{1}$ as well as all the problems $S_{k,t-1,\,k,t} =? \mathbf{1}$, $(k,t-1,\,k,t)$: $k,t \in K$, $k,t-1 \in K\backslash(k,t)$, and we can use the notations t ($S_{k,t-1,\,k,t} \cap X,t-1_{k,t-1} =? \mathbf{1}, A$) and t ($S_{k,t-1,\,k,t} =? \mathbf{1}, A$).

Continuing this downward reasoning, we conclude that for any solution algorithm A, we can use the notations t ($S_{k,r,\,\ldots\,,\,k,t} \cap X,r_{k,r} =? \mathbf{1}, A$), $r = 2, \ldots, t$, for the respective inputs of (27'')-types.

To estimates the times, we undertake the following upward reasoning. Since

$$\text{t}\,(S_{k,3,\,\ldots\,,\,k,t} =? \mathbf{1}, A) = \text{t}\,[\cup\,(S_{k,2,\,\ldots\,,\,k,t} \cap X,2_{k,2})\;(k,2 \in K\backslash(k,3,\,\ldots\,,\,k,t)) =? \mathbf{1}, A],$$

$$k,t \in K;\; k,t-1 \in K\backslash(k,t);\; \ldots\;;\; k,3 \in K\backslash(k,4,\,\ldots\,,\,k,t), \tag{28}$$

we can conclude that, based on (27''), there exist exponential number instances for which the algorithm A has to use all $k,2 \in K\backslash(k,3,\,\ldots\,,\,k,t)$, in order to solve all the problems $S_{k,2,\,\ldots\,,\,k,t} \cap X,2_{k,2} =? \mathbf{1}$. Indeed, we know that $S_{k,2,\,\ldots\,,\,k,t} \neq \mathbf{1}$, and all cases of $S_{k,2,\,\ldots\,,\,k,t} \cap X,2_{k,2} =, \neq \mathbf{1}$ are possible, in particular, the case of $S_{k,2,\,\ldots\,,\,k,t} \cap X,2_{k,2} = \mathbf{1}$ for all $k,2 \in K\backslash(k,3,\,\ldots\,,\,k,t)$, except perhaps the last one (independently of their ordering). Hence, due to Lemma 4,

$$\text{t}\,(S_{k,3,\,\ldots\,,\,k,t} =? \mathbf{1}, A) \geq \sum \text{t}\,(S_{k,2,\,\ldots\,,\,k,t} \cap X,2_{k,2} =? \mathbf{1}, A)\;(k,2 \in K\backslash(k,3,\,\ldots\,,\,k,t)) \geq$$

$$\text{T}\,[k,2,\,k,2 \in K\backslash(k,3,\,\ldots\,,\,k,t)] > \Theta(m),\; k,t \in K;\; k,t-1 \in K\backslash(k,t);\; \ldots\;;\; k,3 \in K\backslash(k,4,\,\ldots\,,\,k,t). \tag{29}$$

Besides, we can assume that all $S_{k,3,\,\ldots\,,\,k,t} \neq \mathbf{1}$.

Similarly, since

$$\text{t}\,(S_{k,4,\,\ldots\,,\,k,t} =? \mathbf{1}, A) = \text{t}\,[\cup\,(S_{k,3,\,\ldots\,,\,k,t} \cap X,3_{k,3})\;(k,3 \in K\backslash(k,4,\,\ldots\,,\,k,t)) =? \mathbf{1}, A],$$

$$k,t \in K;\; k,t-1 \in K\backslash(k,t);\; \ldots\;;\; k,4 \in K\backslash(k,5,\,\ldots\,,\,k,t), \tag{28'}$$

we can conclude that there exist exponential number instances for which the algorithm A has to use all $k,3 \in K\backslash(k,4,\,\ldots\,,\,k,t)$, in order to solve all the problems $S_{k,3,\,\ldots\,,\,k,t} \cap X,3_{k,3} =? \mathbf{1}$, $k,t \in K$; $k,t-1 \in K\backslash(k,t)$;$\ldots$; $k,3 \in K\backslash(k,4,\,\ldots\,,\,k,t)$. Indeed, since $S_{k,3,\,\ldots\,,\,k,t} \neq \mathbf{1}$, all cases of $S_{k,3,\,\ldots\,,\,k,t} \cap X,3_{k,3} =, \neq \mathbf{1}$ are possible, in particular, the cases of $S_{k,3,\,\ldots\,,\,k,t} \cap X,3_{k,3} = \mathbf{1}$ for all $k,3 \in K\backslash(k,4,\,\ldots\,,\,k,t)$, except perhaps the last ones (independently of their ordering). Hence, due to (29) and Lemma 4,

$$\text{t}\,(S_{k,4,\,\ldots\,,\,k,t} =? \mathbf{1}, A) \geq \sum \text{t}\,(S_{k,3,\,\ldots\,,\,k,t} \cap X,3_{k,3} =? \mathbf{1}, A)\;(k,3 \in K\backslash(k,4,\,\ldots\,,\,k,t)) \geq$$

$$\sum \text{t}\,(S_{k,3,\,\ldots\,,\,k,t} =? \mathbf{1}, A)\;(k,3 \in K\backslash(k,4,\,\ldots\,,\,k,t)) \geq \text{T}\,[(k,2,\,k,3),\,k,2 \in K\backslash(k,3,\,\ldots\,,\,k,t);$$

$$k,3 \in K\backslash(k,4,\,\ldots\,,\,k,t)] > \Theta(m^2),\; k,t \in K;\; k,t-1 \in K\backslash(k,t);\; \ldots\;;\; k,4 \in K\backslash(k,5,\,\ldots\,,\,k,t). \tag{29'}$$

Besides, we can assume that all $S_{k,4,\,\ldots\,,\,k,t} \neq \mathbf{1}$.

Continuing this reasoning, we can conclude that, based on (27'),

$$t\,(S_{k,t} =?\ \mathbf{1}, A) \geq \sum t\,(S_{k,3,\,\ldots\,,k,t} =?\ \mathbf{1}, A)\ (k,t-1 \in K\backslash(k,t);\ \ldots\ ;\ k,3 \in K\backslash(k,4,\ \ldots\ ,k,t)) \geq$$

(29'') $$T\,[(k,2,\ \ldots\ ,k,t-1),\ k,2 \in K\backslash(k,3,\ \ldots\ ,k,t);\ \ldots\ ;\ k,t-1 \in K\backslash(k,t)] > \Theta(m^{t-2}),\ k,t \in K,$$

and, based on (27),

$$T\,(V_{m,t} =?\ \mathbf{1}, A) \geq \sup\,[t\,(V_{m,t} =?\ \mathbf{1}, A)]\ (I_V) \geq \sup\,[\sum t\,(S_{k,t} =?\ \mathbf{1}, A)\ (k,t \in K)]\ (I_V) \geq$$

$$\sup\,[\sum t\,(S_{k,3,\,\ldots\,,k,t} =?\ \mathbf{1}, A)\ (k,t \in K;\ k,t-1 \in K\backslash(k,t);\ \ldots\ ;\ k,2 \in K\backslash(k,3,\ \ldots\ ,k,t))]\ (I_V) \geq$$

(30) $$T\,[(k,2,\ \ldots\ ,k,t),\ k,2 \in K\backslash(k,3,\ \ldots\ ,k,t);\ \ldots\ ;\ k,t-1 \in K\backslash(k,t)] > \Theta(m^{t-1}).$$

5. The end of the proof follows from (30) and (25) now. ∎

5. Main result

Theorem. NP ≠ P.

Proof. Let P = NP.

Then the problem P_1, i.e., the problem of independent set with the representation (3) of the graph G, is also in P. Therefore, there exists a constant C, independent of n and t, for which $T\,(P_1) = T\,(P_V) = T\,(R_{n,t-1} =?\ \mathbf{1}) < n^C$, on the strength of Lemma 1+. But verification of $R_{n,t-1} =?\ \mathbf{1}$ can require the time more than n^C in order for any DTM due to Lemma 5 in the case of $t > C + 2$.

This contradiction means that one of NP-complete problems and hence all of them do not belong to P. But they all belong to NP.

Thus, NP is not equal to P. ∎

Consequence. *A solution of NP-hard problem on any quantum computer requires more than a polynomial time*.

Proof. It follows from the results of K. Grover, [3] and C.H. Bennett et al., [1].

The paper [3] shows that there is a quantum computer, which is capable to compute n^t objects for any n for the time $n^{t/2}$. And the paper [1] proves that the order of this time $n^{t/2}$ cannot be improved. ∎

6. Discussion

In one of his critical remarks to the author, Dr. D. S. Johnson 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 reconsider briefly the above estimates whether they are valid for *all* solution algorithms. If the time $T\,(P, A) > T^*$, where T^* is independent of algorithms A, then this estimate is valid for any A, and hence the time $T\,(P)$ is also not less than T^*.

Therefore, the time T ($V_{m,t}$ =? **1**) has order n^{t-1}, since to know if $V_{m,t}$ = **1** or not, we must know not less than order n^{t-1} distinct intermediate results, independent of all solution algorithms A.

Some doubts could arise on using a particular case (3) of possible graph representations.

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

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

A proof of (31) follows rather easily from definitions (8). Thus, using any other representation of the graph, requiring polynomial time, cannot reduce our estimates.

7. On the gap between low and upper bounds

We have T $(R_{n,1} =? \mathbf{1}) = \Theta(n^2)$, and T $(R_{n,1}) = \Theta(n^3)$.

The first relation is obvious. The second can be shown by the mathematical induction based on the fact that $\cup I_s$ (s = 2, … , n) can be considered as the union of the mutually independent sets $I_r \cup I_{r+1}$, r = 2, 4, … .

We have T $(R_{n,2} =? \mathbf{1})$, T $(R_{n,2}) = \Theta(n^4)$.

It follows from a particular case $I'_{k,2} = (k,2, k,2 + n/2)$, n is even, when the words $J_{k,2} = I_{k,2} \cup I_{k,2+n/2}$, $k,2$ = 2, …, $n/2$, are independent, and T $(J_{k,2} =? \mathbf{1})$, T $(J_{k,2}) = \Theta(n^3)$, since a distance between $I_{k,2}$ and $I_{k,2+n/2}$ has the order n^2.

There is a foundation that T $(R_{n,t} =? \mathbf{1})$, T $(R_{n,t}) = \Theta(n^{t+2})$, where t is independent of n, $t > 1$.

This follows likely from the possibility that the estimate in Lemma 5 could be improved from $\Theta(n^{t-1})$ to $\Theta(n^{t+2})$ with regard to T $(R_{n,1} =? \mathbf{1}) = \Theta(n^3)$.

Thus, simple algorithms of computation $R_{n,t}$ via successive unions of intersections according to the relations (16) could be proven to be optimal by the time in order n.

8. Open problems

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

7.1. What is the time for verification of $R_{n,t}$ =? **1**, when t is dependent of n?

That time will be probably also $(n/t)^{t+2}$ in order for all t until $n/2$; n is even.

7.2. Could the estimate $2^{\Theta(m)}$ hard instances in Lemma 5, for which the time for verification of $V_{m,t}$ =? **1** is not less than m^{t-1} in order, be essentially strengthened?

Note that although the estimate in Lemma 5 is valid for exponential number instances, the result wanted is almost all instances, like in the case of the problem recognition of palindromes. For that result, the average time also cannot be polynomial. Note more that for many applications of NP-hard problems, it is very important to know the sets of non-hard instances or their sizes.

7.3. Is it possible that a non-algorithmic-type computer for effective solution of NP-hard problems could be constructed? If yes, then how?

Remarks

The author started the investigation of the problem NP vs. P long time ago (in connection with his investigation of optimal algorithms [V. Ivanov, 6, 7]). He tried to prove NP = P, using n-dimensional integrals [6]. After failing in this direction, he started to prove NP ≠ P. He has his results submitted to J. of Algorithms [V. Ivanov, 8], and left the proof to the reader here.

References

[1] C. H. Bennett, E. Bernstein, G. Brassard, U.V. Vazirani, *Strengths and Weaknesses of Quantum Computing*, SIAM J. on Computing, 26 (1997), 1510-1523
[2] M. R. Garey, D. S. Johnson, *Computers and Intractability*. *A Guide to the Theory of* NP-*Completeness*, W.H. Freeman and Co. (1979)
[3] L. K. Grover, *Quantum Mechanics Helps in Searching for a Needle in a Haystack*, Physical Review Letters, 79, N 2 (1997), 325-328
[4] *Handbook of Theoretical Computer Science*, Edited by J. van Leeuwen, Elsevier Science Publishers B. V. (1990)
[5] *Handbook of Discrete and Combinatorial Mathematics*, Editor-in-Chief Ken Rosen, CRS Press (2000)
[6] V. V. Ivanov, *Methods of Computation on Computers*. *Guidance*, Kiev, Naukova dumka, (1986) (in Russian)
[7] V. V. Ivanov, *Model Development and Optimization*, KAP (1999)
[8] V. V. Ivanov, *A Proof of Cook's Conjecture*, submitted to J. of Algorithms, the last version in March 2005

Dr. Viktor V. Ivanov

Address: 12713 English Hills CT Apt D
Tampa, FL 33617-1321 USA

Phone: (813) 985-9014

E-mail: ivanvvn6@cs.com