I.J.Modern Education and Computer Science, **2012, 3, 21-27**
Published Online April 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2012.03.03

On Correlation of *P* And *NP* Classes

Listrovoy Sergey Vladimirovich

Ukrainian State Academy of Railway Transport, Kharkov, Ukraine
Email: om1@yandex.ru

Abstract — It is shown an incorrectness of introduction of a class *of NP-complete problem*s, which reason is that Cook's S.A. theorem on that the "satisfiability" problem is the universal *NP*-complete problem, is not true and, therefore, the issue on existence of at least one *NP*-complete problem remains open, that explains failures of attempts to estimate correlations between *P* and *NP* classes.

Index Terms— Theory to difficulties algorithm, Digital Systems.

I. Introduction

Whether the issue on those NP-complete problems is really hard to solve, is considered now as one of the key open issues of modern mathematics and theoretical cybernetics. Contrary to readiness of the majority of experts to consider that all *NP*-complete problems are hard to solve, there is no progress both in proof, and in a refutation of this proposal. Therefore the purpose of the work is an attempt to explain why there was the situation. The theory *of NP-complete problem*s is constructed for problems of recognition of properties. The problem of recognition *L* can be considered as consisting of two sets: D_i and Y_∂, where D_i – set of all single problems, and Y_∂ – set of problems with the answer "yes", thus $Y_\partial \subset D_i$. The form of these problems consists of two parts. In the first part the exposition of conditions of the problem in terms of various components is given: sets, networks, numbers etc. In the second part the question assuming one of two answers "yes" or "no" is formulated. Informally, class *of NP-complete problem*s is defined by means of concept of nondeterministic algorithm. Such algorithm consists of two stages: guessing and check. At first, under the set single problem *I* a guessing of structure *S* takes place, and further, taking into account statements of problem *I,* check by the determined algorithm which is ended either by the answer "yes" or "no" is carried out. As it is shown in [1], nondeterministic algorithm solves the problem of recognition *L*, if for any single problem $I \in D_i$ two following conditions are met:

1. If $I \in Y_\partial$ there is such structure *S* which guessing leads to that the check stage will be completed by the answer "yes".
2. If $I \notin Y_\partial$ there is no such structure *S* which guessing for *I* will lead to that the check stage will be completed by the answer «no».

The concept of polynomial "checkability" [1] allows actual selection of a class *of NP-complete problem*s, and in addition, "checkability" for polynomial time does not attract decidability of the problem of recognition for polynomial time. ***The problem of recognition L is called NP - complete if L ∈ NP and any other problem L′ from this class is reduced to L polynomially.***

In his study [2] titled «Complexity of procedures of a conclusion of theorems», Cook has proven that one specific problem from class *NP* named "satisfiability" possesses the property that any other problem from class *NP* can be reduced to it for polynomial time. Let's prove, that the statement is incorrect.

II. FORMALISING OF THE "SATISFIABILITY" PROBLEM

Let's consider a Boolean function $f(x_1, x_2, .., x_n)$ in the conjunctive notation $f(x_1,x_2,..,x_n)=(x_1^{\sigma_{11}} \vee x_2^{\sigma_{12}} \vee \ldots \vee x_n^{\sigma_{1n}}) \wedge \ldots \wedge (x_1^{\sigma_{m1}} \vee x_2^{\sigma_{m2}} \vee \ldots \vee x_n^{\sigma_{mn}})$,

Where $x_i^\sigma = \begin{cases} x_i, at - \sigma = 1 \\ \overline{x_i} \; at - \sigma = 0 \end{cases}$.

Operations $\vee$ and $\wedge$ are Boolean and model elementary logical expressions: $\vee$– "OR"; $\wedge$– "AND". For any binary gang $x = (x_1, x_2, .., x_n)$ the function accepts one of two possible values: unit or zero. The "satisfiability" problem consists in the answer to the question: whether there is a gang of values of the variables $x = (x_1, x_2, .., x_n)$ converting function f to unit.

A. Problem setting and solution

For estimation of satisfiability or an unsatisfiability of Boolean functions let's, at first, consider properties of unsatisfiable functions and methodes of their obtaining.

B. Properties of unsatisfiable Boolean functions

Let's introduce concept of minimum conjunctive form $F^{min}{}_f$ of an arbitrary Boolean function *f* of two variables with a minimum number of disjuncts *r* at which $F^{min}{}_f$ accepts value "false" for all possible gangs of the function. For this purpose let's consider a bipartite graph G_{22} of a view shown in Figure 1, which one part is formed by variables (*a, b*), and the second is formed by variables $(\overline{a}, \overline{b})$. In the graph those nodes are connected by edges which do not form inverse pairs of $(a, \overline{a})$ type.

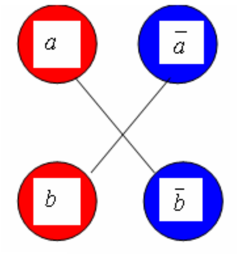

Figure .1.Graph G_{22}

For inconsistency origin in a Boolean function, it is necessary that both variables (*a*,*b*), and $(\overline{a},\overline{b})$ were present. The perfect matching in graph G_{22} provides all possible additions of combinations of variables which can be connected among themselves by an edge as these nodes in the graph are not inverse. Addition of disjuncts comprising variables (*a*,*b*) and $(\overline{a},\overline{b})$ by disjuncts which comprise the variables stipulated by a perfect matching in the graph G_{22}, will allow the minimum form to receive:

$$F_{2f}^{\min} = (a \vee b)(\overline{a} \vee \overline{b})(a \vee \overline{b})(b \vee \overline{a}) \, . (1)$$

Let's present a Boolean function *f* in the form of sliced graph *G* where each variable in disjuncts corresponds to graph node, and each slice or a tier correspond to disjunct in which nodes are not linked among themselves.

Connections between nodes of tiers are carried out as follows. The node *i*, corresponding to variable X_i of an arbitrary tier, is connected to all nodes of a following tier, except nodes i^*, corresponding to a variable $\overline{X}_i$.

Let's name nodes *i* and i^* as inverse nodes of the graph. As it will be visible further, the arrangement order of tiers in graph *G* for the solved problem is of no concern. Let's introduce also two dummy nodes *s* and *t* which are linked to all nodes of the first and last tier of graph *G,* accordingly. Then the graph of the function (1,) looks like in fig. 2.

Clearly, that if in graph *G* of an arbitrary Boolean function *f* there is a path from node *s* to *t* which does not include inverse nodes, there is a gang of variables on which function *f* accepts value "true". Let's assume, it is obtained the path from *s* to *t* in the graph *G* which does not contain inverse nodes (*s, i, j, ... k, t*). A variable either X_i, or $\overline{X}_i$ corresponds to each node there, i.e. we can put in correspondence paths (*s, i, j, ... k, t*) to a gang of variables ($X_i, \overline{X}_j, \ldots X_k$) among which there are no contradictory, that ensures absence of inverse nodes in the path. Therefore, we can easily specify a gang of variables at which all gang will consist of units. However, as variables in the gang were used from each tier of the graph, it means that there will be unit at every disjunct and function *f* accepts value «true» for this gang of variables.

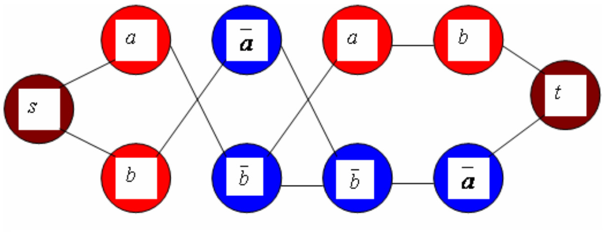

Figure.2.Graph G_{2f}

Thus, if function *f* accepts value "false" for all gangs, then there is no path from node *s* to *t* in the graph of function *G* which does not include inverse nodes, that is clear on the graph presented on fig.2. Considered conjunctive forms possess the important property, ν, consisting in that the modification of a sign of inversion in an arbitrary literal leads to that changed function F_f, becomes satisfiable. It is easily possible to be convinced of that directly parsing graph G_{2f} of the function $F^{min}{}_{2f}$ from which it is clear, that the modifications defined by property, ν , will lead to occurrence of the paths in the graph which are not containing inverse nodes and, therefore, function *f* will become satisfiable. If we consider functions F_{nf} with arbitrary number of variables and with number of variables in disjunct not less two, it is possible to present such functions in the form of association of two graphs of bipartite graph G_{nf} , see fig.3, in which nodes of one part correspond to variables $(\tilde{O}_1, \tilde{O}_2, \ldots, X_n)$, and for the second part - $(\overline{X}_1, \overline{X}_2, \ldots, \overline{X}_n)$ (subsets of binary graphs G'_{nf} in which links between nodes corresponding to variables $(X_1, X_2, \ldots, X_n)$ and $(\overline{X}_1, \overline{X}_2, \ldots, \overline{X}_n)$ are defined by perfect matchings in graph G_{nf}) .

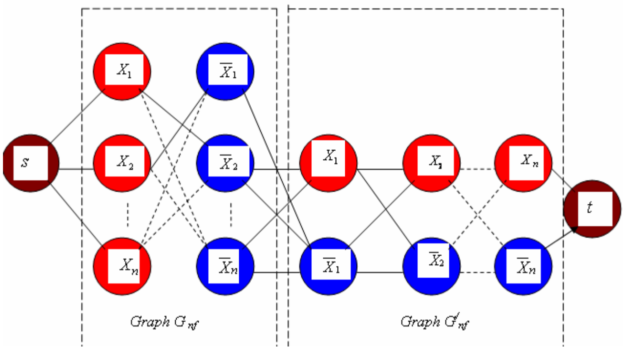

Figure.3. The graph which generates functions F_{nf}

Let's show, that conjunction of disjuncts defined by binary graphs $\{G'_{nf}\}$, which lead to an unsatisfiability of functions, is generally equal to $\tilde{O}_1 \tilde{O}_2 \ldots X_n \vee \overline{X}_1 \overline{X}_2 \ldots \overline{X}_n$ and it is possible to present arbitrary function $F^{min}{}_{nf}$ in the form

$$F_{nf}^{\min}=\left(\alpha_1 X_1 \vee \alpha_2 X_2 \vee \ldots \vee \alpha_n X_n\right)\left(\beta_1 \overline{X}_1 \vee \beta_2 \overline{X}_2 \vee \ldots \beta_n \overline{X}_n\right)\left(X_1 X_2 \ldots X_n \vee \overline{X}_1 \overline{X}_2 \ldots \overline{X}_n\right)=$$
$$=X_1 X_2 \ldots X_n\left(\beta_1 \overline{X}_1 \vee \beta_2 \overline{X}_2 \vee \ldots \vee \beta_n \overline{X}_n\right) \vee \overline{X}_1 \overline{X}_2 \ldots \overline{X}_n\left(\alpha_1 X_1 \vee \alpha_2 X_2 \vee \ldots \vee \alpha_n X_n\right)=$$
$$=\prod_{i=1}^{n} x_i f(\beta_i \overline{x}_i) \vee \prod_{i=1}^{n} \overline{x}_i f(\alpha_i x_i), \text{ỳπ□} \alpha_i \in \{0,1\}; \beta_i \in \{0,1\}; f(\beta_i, \overline{x}_i)=\left(\beta_1 \overline{X}_1 \vee \beta_2 \overline{X}_2 \vee \ldots \vee \beta_n \overline{X}_n\right);$$
$$f(\alpha_i x_i)=\left(\alpha_1 X_1 \vee \alpha_2 X_2 \vee \ldots \vee \alpha_n X_n\right). \qquad (2)$$

As the “satisfiability” problem can be considered as the cover problem on a Boolean, then using a Boolean function let’s construct a matrix B in which variables $\left(X_1, X_2, \ldots, X_n\right)$ and $\left(\overline{X}_1, \overline{X}_2, \ldots, \overline{X}_n\right)$ there correspond to columns, and disjuncts of a Boolean function correspond to lines. Generally, the number of columns in a matrix *B* equal to 2*n*, and a number of lines is equal to number of disjuncts *m* in a Boolean function.

For example, for a Boolean function $F_{nf} = \left(X_1 \vee X_2 \vee X_3 \vee X_4\right)\left(\overline{X}_1 \vee \overline{X}_2 \vee \overline{X}_3 \vee \overline{X}_4\right) \cdot$ $\cdot \left(X_1 \vee \overline{X}_3\right)\left(X_2 \vee \overline{X}_4\right)\left(X_3 \vee \overline{X}_1\right)\left(X_1 \vee \overline{X}_2\right)$,

The matrix *B* will look like

$$B = \begin{array}{r|cccccccc|}
 & X_1 & X_2 & X_3 & X_4 & \overline{X}_1 & \overline{X}_2 & \overline{X}_3 & \overline{X}_4 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
3 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
4 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
5 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
6 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{array}.$$

Let’s name the columns corresponding to variables X_i and $\overline{X}_i$ in a matrix *B* as inverse. If in a matrix *B* there is a cover of lines by units which belongs not to inverse columns, it means that function *f* is satisfiable, and if such cover is not present it is unsatisfiable.

And so, let the Boolean matrix *B* is set with 2*n* columns and *m* lines where *m* corresponds to number of disjuncts in a Boolean function *f*. Let’s define columns using a vector $\left(X_1, X_2, \ldots, X_n, \overline{X}_1, \overline{X}_2, \ldots, \overline{X}_n\right)$, and lines – using *the* vector $M = \{\mu_1, \mu_2, \ldots, \mu_m\}$. Let’s call as cover *Q* of lines *M* such set of graphs B which covers all lines of *M* with units. *For* definition of all covers of the matrix we will use an algebraic method of obtaining the reduced systems of simple implicants of Boolean functions using table of prime implicants. If to consider each column from a population $\left(X_1, X_2, \ldots, X_n, \overline{X}_1, \overline{X}_2, \ldots, \overline{X}_n\right)$ as a «prime implicant» cover population of lines $M = \{\mu_1, \mu_2, \ldots, \mu_m\}$, and each line μ_I as a gang of the variables covered with prime implicants, then it is possible to present a matrix *B* as table of prime implicants of Boolean function *f*. At such interpretation of a matrix *B*, it is possible to note for each line μ_i a disjunction of columns b_i, cover considered line, as follows:

$$d\mu_I = (x_1 \text{v}\ \overline{\tilde{o}}_{\hat{e}}\ \text{v} \ldots \ldots)$$
$$\ldots \ldots \ldots \ldots \ldots \ldots \ldots. \quad (3)$$
$$d\mu_m = (x_P \text{v}\ \overline{x}_t\ \text{v} \ldots \ldots).$$

Conjunction of disjunctions (3) for all lines μ_1, μ_2, …, μ_m *of the* matrix *B* forms conjunctive representation of the matrix *B* comprising all covers of a population of lines *M*:

$$k(M)=d\mu_1 \cdot d\mu_2 \ldots d\mu_r=(x_1 \text{v}\ \overline{x}_{\kappa}\ \text{v} \ldots)\ldots(x_P \text{v}\ \overline{x}_t\ \text{v} \ldots). \quad (4)$$

Removing the parentheses according to distributivity laws, we receive disjunctive representation of the matrix *B* forming the list of all possible covers of a population of lines $M = \{\mu_1, \mu_2, \ldots, \mu_m\}$. For clearing up of existence of a cover, let’s take a principle of superposition in the Boolean algebra based on following equalities:

$$f\left(X_1^{\sigma 1}, X_2^{\sigma 2}, \ldots, X_n^{\sigma n}\right)=f\left(X_1^{\sigma 1}=1, X_2^{\sigma 2}, \ldots, X_n^{\sigma n}\right) \vee f\left(X_1^{\delta 1}, X_2^{\sigma 2}=1, \ldots, X_n^{\sigma n}\right) \vee \ldots \vee$$
$$\vee f\left(X_1^{\sigma 1}, X_2^{\sigma 2}, \ldots, X_n^{\sigma n}=1\right);$$

$$f^{/}\left(X_1^{\sigma 1}, X_2^{\sigma 2}, \ldots, X_n^{\sigma n}\right)=f^{/}\left(X_1^{\sigma 1}=0, X_2^{\sigma 2}, \ldots, X_n^{\sigma n}\right) \vee f^{/}\left(X_1^{\delta 1}, X_2^{\sigma 2}=0, \ldots, X_n^{\sigma n}\right) \vee \ldots \vee$$
$$\vee f^{/}\left(X_1^{\sigma 1}, X_2^{\sigma 2}, \ldots, X_n^{\sigma n}=0\right). \qquad (5)$$

Without breaking the principle of superposition (5), it is possible to present an initial Boolean function in the form $f\left(X_1^{\sigma 1}, \ldots, X_n^{\sigma n}\right) = f_1\left(X_1^{\sigma 1}, \ldots, X_n^{\sigma n}\right) \vee f_2\left(X_1^{\sigma 1}, \ldots, X_n^{\sigma n}\right) \vee \ldots \vee$ $f_n\left(X_1^{\sigma 1}, \ldots, X_n^{\sigma n}\right) \vee f_{n+1}^{/}\left(X_1^{\sigma 1}, \ldots, X_n^{\sigma n}\right) \vee f_{n+2}^{/}\left(X_1^{\sigma 1}, \ldots, X_n^{\sigma n}\right) \vee \ldots \vee \vee$ $f_{2n}^{/}\left(X_1^{\sigma 1}, \ldots, X_n^{\sigma n}\right)$, (6)

Where $f_i\left(X_1^{\sigma 1}, \ldots, X_n^{\sigma n}\right) = X_i \quad f\left(X_1^{\sigma}, \ldots, X_n^{\sigma n}\right)$; $f_i^{/}\left(X_1^{\sigma=1}, \ldots, X_n^{\sigma n}\right) = \overline{\tilde{O}}_i \ f\left(X_1^{\sigma}, \ldots, X_n^{\sigma n}\right)$.

Let's rewrite a relation (6) in the form

$$f\left(X_1^{\sigma 1}, \ldots, X_n^{\sigma n}\right)=(X_1 \vee X_2 \vee \ldots \vee X_n) f(X_1^{\sigma 1}, \ldots, X_n^{\sigma n}) \vee \ldots. \quad (7)$$
$$\vee (\overline{X}_1 \vee \overline{X}_2 \vee \ldots \vee \overline{X}_n) f^{/}(X_1^{\sigma 1}, \ldots, X_n^{\sigma n})$$

As follows from (7), in order that this Boolean function was identically equal 0 for all gangs of variables, it is necessary that functions $f_i\left(X_1^{\sigma 1}, \ldots, X_n^{\sigma n}\right)$ and $f_i^{/}\left(X_1^{\sigma=1}, \ldots, X_n^{\sigma n}\right)$ were equal to $(\overline{X}_1 \cdot \overline{X}_2 \cdot \ldots \cdot \overline{X}_n)$ and $\left(X_1 \cdot X_2 \cdot \ldots \cdot X_n\right)$, accordingly and, therefore, the relation (7) is degenerated into a relation (2), as was to be proved.

In a disjunctive normal form, expression (2) looks like

$$F_{fn}^{\min} = X_1 \overline{X}_1 \overline{X}_2 \ldots \overline{X}_n \vee X_2 \overline{X}_1 \overline{X}_2 \ldots \overline{X}_n \vee \ldots \vee X_n \overline{X}_1 \overline{X}_2 \ldots \overline{X}_n \vee$$
$$\vee \overline{X}_1 X_1 X_2 \ldots X_n \vee \overline{X}_2 X_1 X_2 \ldots X_n \vee \ldots \vee \overline{X}_n X_1 X_2 \ldots X_n. \qquad (8)$$

Let's designate set of single “satisfiability” problems S. It is possible to divide the set into two subsets S^+ - a subset of satisfiable single problems and S^- - a subset of unsatisfiable single problems, when $S = S^+ \cup S^-$.

Any unsatisfiable Boolean function should comprise, in explicit or implicit form, $F^{min}{}_{nл}$ and if it comprises $F^{min}{}_{nf}$ implicitly, it can be always transformed to the form (2) or (8). For an arbitrary Boolean function with *n* variables it is possible to construct some minimum false forms, which number is actually defined by number of perfect combinations in graph $G^{/}{}_{nf}$ and as they can be present at a Boolean function and in various combinations it is possible to make a conclusion that the potency of set of unsatisfiable single problems S^- is exponential-fold than potencies of a set of satisfiable single problems S^+ i.e. the inequality is true

$$|S^-| >> |S^+| \quad (9)$$

As shown in studies [1-6], polynomial reducibility of the problem of recognition I_1 to the problem of recognition I_2 means availability of function f which transforms a subset of problems D_{i1} into a subset of problems D_{i2} ($D_{i1} \rightarrow D_{i2}$), on the basis of some rule Π_i and, thus, satisfies to two conditions:

1. f – is calculated by a polynomial algorithm;

2. For all $I \in D_i$ $I \in Y_{\partial 1}$, when and only when $f(I)$ $Y \in_{\partial 2}$.

Let's consider three subsets of problems $\{I_i\}$; $\{Z_i\}$; $\{C_i\}$. Let the problem I-NP-is complete and represents the universal problem, and problems Z and C are also *NP*-complete then, according to that as the class *of NP-complete problem*s is introduced, they should be reduced polynomially one to another and, thus, if the polynomial algorithm for one of them is discovered, there should be polynomial algorithms for all single problems $\{I_i\}$; $\{Z_i\}$; $\{C_i\}$. As the universal problem can appear any of *NP*-complete problems, all following reductions should be true

$$\{I_i\} \rightarrow \{Z_i\} \rightarrow \{C_i\}; (10) \ \{I_i\} \rightarrow \{C_i\} \rightarrow \{Z_i\}; (11)$$
$$\{C_i\} \rightarrow \{I_i\} \rightarrow \{Z_i\}; (12) \ \{C_i\} \rightarrow \{I_i\} \rightarrow \{Z_i\}; (13)$$
$$\{Z_i\} \rightarrow \{C_i\} \rightarrow \{I_i\}; (14) \ \{Z_i\} \rightarrow \{I_i\} \rightarrow \{C_i\}. (15)$$

In addition, there are rules Π_{iz} and Π_{zc} which permit to reduce problems $I_p \rightarrow Z_p$ and, thus, $\{I_p\} \in Y_{\partial i}$ and problems $Z_p \rightarrow C_p$ and, in this respect, $\{Z_p\} \in_{\partial z}$, i.e. transformation rules Π_{iz} and Π_{zc} satisfy to conditions of polynomial reducibility 1 and 2. Let's consider a case similar to (9) when structures S are such that they generate set of single problems $\{Z\}$ which by its potency exceeds set of single problems $\{I\}$. If the subset $\{I\}$ contains n single problems, and sets $\{Z\}$ and $\{C\}$ contain $n+k$ single problems each, then for some subset of problems $\{Z_{n+1}, Z_{n+2}, \ldots, Z_k\}$ we cannot put in correspondence any problem from $\{I_i\}$. Therefore, the reductions (10) and (11) are possible for all problems, and reductions (12), (13), (14) and (15) are possible not for all problems, they are not possible for problems $\{C_{n+1}, C_{n+2}, \ldots, C_k\}$ and $\{Z_{n+1}, Z_{n+2}, \ldots, Z_k\}$, and, it means in this case, that the statement about all *NP*-complete problems are polynomially reduced to each other, is not fulfilled. Thus, the concept *of the NP-complete problem* requires an improvement. In order that *the NP-complete problem* was universal and reduced in any directions within a class, it is necessary that there was a one-to-one correspondence between all single problems $\{I_i\}$; $\{Z_i\}$; $\{C_i\}$, i.e. for any pair of single problems there should be the direct and inverse polynomial reduction defined by conditions 1 and 2.

Thus, if we have subsets of problems $\{I_i\}$; $\{Z_i\}$; $\{C_i\}$, and the potency of set of single problems $\{I_i\}$ differs from a potency of sets of problems $\{Z_i\}$ and $\{C_i\}$, then to prove that some problem I is *NP*-complete, it is not enough to show that any single problem $\{I_i\}$ is polynomially reduced to set of problems $\{Z_i\}$ and $\{C_i\}$, i.e. conditions 1 and 2 are satisfied as it was made in the proof *of NP-completeness* of the "satisfiability" problem in Cook's and Carp's studies, but thus it is necessary to show, that there are also problems $\{I_{n+1}, I_{n+2}, \ldots, I_k\}$, which are polynomially reduced to problems $\{C_{n+1}, C_{n+2}, \ldots, C_k\}$ and $\{Z_{n+1}, Z_{n+2}, \ldots, Z_k\}$, and "checkability" of these recognition problems should remain possible for polynomial time.

Let's show, that the "satisfiability" problem is not universal. So, Cook has proved universality of the "satisfiability" problem at first. After one *NP*-complete problem has become known, process of the proof *of NP-completeness* of the problem $\boldsymbol{A}$ becomes simpler. To prove *NP-completeness* of problem $A \in NP$ it is enough to show, that any known *NP*-complete problem $\boldsymbol{A}'$ can be reduced to $\boldsymbol{A}$ as property of polynomial reducibility is transitive i.e. if the problem $\boldsymbol{A}$ for polynomial time is transformed to the problem B and if B is transformed to $\boldsymbol{C}$ for polynomial time**,** then $\boldsymbol{A}$ is transformed to $\boldsymbol{C}$ for polynomial time**.** First**,** under the circuit *NP*-completeness of six primal problems has been proved: «three-dimensional combination»; "partition"; «vertex cover", "Hamilton cycle";"complete subgraph». As they were first problems which have been introduced into a class of *NP*-complete ones after the "satisfiability" problem, the proof of their *NP-completeness* was reduced to introduction of rule $\boldsymbol{\Pi}$ on which basis for some arbitrary "satisfiability" problem $y \in Y$ structure $\boldsymbol{S}$ possessing property $\boldsymbol{v}$ if and only if y accepts value "true". For example, for a «vertex cover» problem, graph $\boldsymbol{G}$ has appeared as structure $\boldsymbol{S}$, and property $\boldsymbol{v}$ consisted in that graph $\boldsymbol{G}$ has a vertex cover with number of elements no more K, if and only if a gang of disjunctions, $C = \{c_1, c_2, \ldots, c_m\}$ is fulfilled defining the arbitrary single "3-satisfiability" problem. Generally, the «satisfiability» problem represents some set Y of single problems defined by various modes of set up of logical function. It should be noted that during proof *of NP-completeness* of all six enumerated problems the arbitrary single problem is selected at first $y \in Y$ and then under the problem graph G is built for polynomial time. That graph should contain interesting structure which possess the necessary property, v, only if the logical function corresponding to the single problem, accepts value «true». As graph G can be arbitrary, we will solve some single problem $z \in Z$ where Z is the set of problems generated by usage of various types of graphs G.

During solution of the arbitrary *NP*-complete problem of graph theory there is an inverse problem: arbitrary graph G is set, and then it is required to determine, whether the graph G possesses structure with property $\boldsymbol{V}$**,** or not.

This brings up the questions: what single problem (y*)* from set of single "satisfiability" problems Y corresponds to the problem $z \in Z$ generated by graph G, and whether for all problems Z there is an inverse correspondence, and if such correspondence exists, how to construct it under the initial graph and whether will be this construction polynomial or not?

In the theory *of NP-complete problem*s [1-5] there are no answers to these problems yet, and property of inverse polynomial reducibility of single problems *y↔z* is by default transferred to sets *Y* and *Z*.

Let's show that if consider «satisfiability» problem as universal the circuit according to which all list of NP-complete problems is actually obtained, does not ensure existence of correspondence of problems y↔z and their polynomial reducibility to each other. Let’s consider correspondence of problems y↔z, in which we should assign a corresponding «satisfiability» problem to structure S possessing property ν, and the problem accepts value “true” when and only when structure S possesses property ν.

Let rule ***Π*** is introduced, on which basis under some arbitrary «satisfiability» problem $y \in Y$ structure ***S*** has been constructed which possesses property ***v*** in then and only then when *y* accepts value «true». It should be noted that this approach has been used for proof *of NP-completeness* of all six principal *NP*-complete problems.

Generally, in order to justify a possibility of that correspondences *y↔z* at the proposal that the «satisfiability» problem is *NP*- complete, cannot take place and thus the problem *z* cannot be transformed to the initial problem *y* for polynomial time it is enough to show it by the example of one of six principal *NP*-complete problems. Let’s consider this by the example of a «vertex cover» problem.

Let's mention the rule ***Π*** which was used for the proof *of NP-completeness* of the «vertex cover» problem in the study [1]. Let $U = \{u_1, u_2, ..., u_n\}$ and $C = \{c_1, c_2, ..., c_m\}$ define the arbitrary single "3-satisfiability" problem. Let’s specify graph *G* (*V*, *E*) and a positive integer $K \le |V|$ such that *G* has a vertex cover with number of units no more *K* when and only when a gang of disjunctions *C* is fulfilled. *For* each variable $u_i \in U$ there is a component $T_i = (V_i, E_i)$ of truth values gang (where $V_i = \{u_i, \overline{u_i}\}$, $E_i = \{\{u_i, \overline{u_i}\}\}$), i.e. T_i is pair of the nodes connected by an edge, thus any vertex cover should cover an edge from E_i, therefore it should contain at least one node u_i or $\overline{u_i}$. For each disjunction, $c_j \in C$ there is a component of satisfiability check $S_j = (V_j^|, E_j^|)$ consisting of three nodes and three edges connecting them and forming a triangle:

$$V_j^| = \{a_1[j], a_2[j], a_3[j]\},$$

$$E_j^| = \{\{a_1[j], a_2[j]\}, \{a_1[j], a_3[j]\}, \{a_2[j], a_3[j]\}\}.$$

Any vertex cover should contain at least two nodes from $V_j^|$ to cover all three edges from $E_j^|$. Linking edges are the single part of a construction depending on what literals enter into disjunctions. We put in correspondence three literals x_j, l_j, p_j to each disjunction $c_j \in C$ containing those literals. Then the linking edges which are starting from S_j are set as follows:

$$E_j^{||} = \{a_1[j], x_j\}, \{a_2[j], l_j\}, \{a_3[j], p_j\}\}.$$

Construction of the single «vertex cover» problem is ended if to suppose $K=n+2m$ and $G = (V, E)$, where

$$V = (\bigcup_{i=1}^{n} V_i) \bigcup (\bigcup_{j=1}^{m} V_j^|); E = (\bigcup_{i=1}^{n} E_i) \bigcup (\bigcup_{j=1}^{m} E_j^|) \bigcup (\bigcup_{j=1}^{m} E_j^{||}).$$

Let's consider a Boolean function

$$f = (u_1 \vee u_2 \vee u_3)(\overline{u_1} \vee \overline{u_2} \vee u_3)(u_1 \vee \overline{u_2} \vee u_3)(u_2 \vee \overline{u_1} \vee u_3) \cdot \quad (16)$$

$$\cdot (\overline{u_3} \vee u_4 \vee u_5)(\overline{u_3} \vee u_4 \vee \overline{u_5})(\overline{u_3} \vee \overline{u_4} \vee u_5)(\overline{u_3} \vee \overline{u_4} \vee \overline{u_5}).$$

According to described procedure ***Π***, graph $G_1 = (V_1, E_1)$

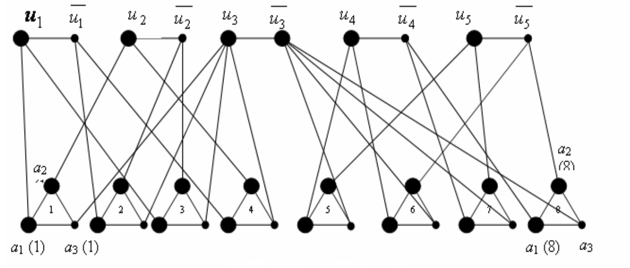

Figure.4. Graph $G_1 = (V_1, E_1)$

for the function will be of the form shown on fig.4. Thus the number of the nodes forming the minimum cover in the graph should not exceed $K = n+2m=5+16=21$. Let’s show that function (16) is unsatisfiable on any gang of variables. For this purpose let’s consider function of two variables

$$f = (u_1 \vee u_2)(\overline{u_1} \vee \overline{u_2})(u_1 \vee \overline{u_2})(u_2 \vee \overline{u_1}) \quad (17)$$

It is easy to check up, that function (17) is unsatisfiable on any gang of variables. Let’s transform (16), having added into each parenthesis a variable u_3 and having increased function (17) by a multiplier $(\overline{u_3} \vee u_4 \vee u_5)(\overline{u_3} \vee u_4 \vee \overline{u_5})(\overline{u_3} \vee \overline{u_4} \vee u_5)(\overline{u_3} \vee \overline{u_4} \vee \overline{u_5})$. It is easy to see, that thus we will receive function (16). Such adding of variables leads to that the variable u_3 will accept value "false" in any gang of the validity which fulfills conditions (16). Therefore disjuncts containing by two variables in (17) will be equivalent to disjuncts substituted them, therefore, function (16) is satisfiable when and only when function (17) is satisfiable. Considering that (17) is unsatisfiable on any gang of variables, the function (16) is unsatisfiable on any gang of variables, too. From fig. 4 it is clear that there is no the cover in graph G_1 consisting of 21 nodes that is well agree with that there is no gang of variables on which function (16) would be satisfiable. But, it is easy to specify a cover on graph $G_1 = (V_1, E_1)$ consisting of 22 nodes; these nodes are shown on fig. 4 as boldfaced, thus it is easy to check up, that the current cover in graph G_1 is minimum. There is a question, whether it is possible to construct such Boolean function for graph G_1, adhering to rule ***Π***, that it would be satisfiable only in that case when the minimum cover in graph G_1 did not exceed 22 nodes. It is obvious, that such function should contain function (16), and this part of function will ensure, that there is no cover less than 22 in graph G_1, and a number of disjuncts which is necessary to have in its structure in order that rule ***П*** was fulfilled, should be equal to (22-5)/2=8,5. That is impossible, as such function, at first, will be unsatisfiable for all possible gangs as it includes function (16), and, secondly, the number of disjuncts in function cannot be fractional. If trying to realize reconversion for an arbitrary graph considering rule ***П,*** we should have the number of disjuncts in the single "satisfiability" problem equal to $m = (K\text{-}n)/2$, i.e. if $K\text{-}n$ is odd, the number of disjuncts will be fractional number. Thus, there is no such «satisfiability» problem which function constructed on the basis of rule ***П,*** considered in the study [1], would accept value «true» when and only when the number of nodes in a cover of graph G_1 does not exceed 22. It is possible to construct so much such examples, how many unsatisfiable functions exist, i.e. exponentially large set. It is necessary also to note, that even if a conversion exists, after performance of the conversion the number of true propositions can appear exponentially large, and only one true proposition possessing some propertyγ will satisfy to property ν of initial graph. So, it is shown in the study [6], that if *f is* the Boolean function constructed by graph $G = (V, E)$ in the form of product of disjuncts $(V_i \vee \mathrm{V}_j)$ where $\{V_i\} \in 0,1\}$, $i = \overline{1,n}$ $j = \overline{1,n}$ $i \neq \mathrm{j}$ and in the same time every disjunct $(V_i \vee \mathrm{V}_j)$ corresponds to an edge (V_i, V_j) all gangs of variables $\{V_i, V_j\}$ for which it accepts value «true» correspond to vertex covers in graph $G = (V, E)$. It follows that for enumeration of all vertex covers of graph $\mathrm{G} = (V, E)$, it is necessary to define those systems of values of variables $\{V_{i,}\ V_j\}$, at which expression is true

$$f(V_1,V_2\ldots V_n)=1, \quad (18)$$

In order to determine these systems of values of variables $\{v_i,v_j\}$, it is necessary to reduce the left part (18) to minimum DNF (a disjunctive normal form) removing the parenthesis and using the law of absorption. Such form is unique in view of logical negations in (18) are absent. Let's demonstrate this by an example of graph *G* presented in fig.5.

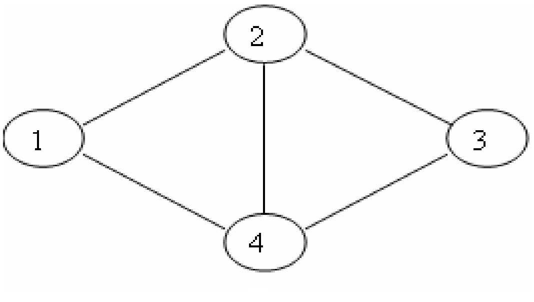

Figure.5. Graph *G*

Boolean function for the graph will look like:

$$f = (V_1 \vee V_2\ )\ (V_2 \vee \mathrm{V}_3)\ (V_3 \vee \mathrm{V}_4)\ (\mathrm{V}_2 \vee V_4)\ (V_1 \vee V_4)$$

$$= (V_2 \vee V_1\, V_3)\ (V_4 \vee \mathrm{V}_1 V_2 V_3) = V_2 V_4 \vee V_1 V_2 V_3 \vee V_1 V_3 V_4\ .(19)$$

Apparently from (19), as a result of disclosure of parenthesis and collecting terms, we have obtained the complete enumeration of vertex covers of graph *G* (fig.5). They are subsets of nodes: {2, 4}; {1, 2, 3}; {1, 3, 4}. Generally, function (18) contains exponential number of true propositions as it does not contain logical negations and, thus, if we find the minimum cover, the expression should satisfy to the property γ consisting that disjunct which corresponds to the minimum vertex cover, should be of minimum length, i.e. contain the least number of variables $\{V_i\}$. The conversion can be easily applied to graph G_1, but at the same time for definition of property of the graph ν, performance of exponential number of steps is required. If to assume nevertheless, that in this case there is a polynomial algorithm for the conversion, it would mean, that the «vertex cover» problem is solvable for polynomial time that contradicts the proposal that *NP*-complete problems are insoluble for polynomial time. Therefore, if direct and inverse conversions are exist, reconversion can be exponential, instead of polynomial.

It means, that if to assume in example (10-15), that *I* is the «satisfiability» problem, then we basically can put in correspondence for problems $\{C_{n+1},\ C_{n+2},\ \ldots,\ C_k\}$ and $\{Z_{n+1},\ Z_{n+2},\ \ldots,\ Z_k\}$ the problem $\{I_{n+1},\ I_{n+2},\ \ldots,\ I_k\}$ such that all reducibilities (10-15) are satisfiable for polynomial time, but upon that, "checkability" of the problem of recognition $I_{p>n}$ can require exponential number of steps.

And so, if to select the «satisfiability» problem as the universal problem we can face a situation of impossibility of polynomial reducibility of some subset of problems in a class *of NP-complete problem*s. It happens because Cook's proof is carried out for set of single problems of satisfiable Boolean functions S^+ for which polynomial reducibility is simply enough justified using a Turing machine as the calculator model, and upon that the set of single problems S^- of unsatisfiable Boolean functions is completely eliminated from the analysis, which potency essentially exceeds a potency of set S^+.

III. Inference

So, the class *of NP-complete problem*s is introduced improperly as the «satisfiability» problem cannot claim for a role of the universal *NP*-complete problem. Also, the issue on existence of at least one *NP*-complete problem remains open as universality of any problem from *NP* class, except the «satisfiability» problem, has not been proved, and they have been included in list of *NP*-complete problems only on the basis of reducibility of the «satisfiability» problem to them.

Upon that, it is actually possible to divide all known set of problems which are called *NP*-complete into subsets within which polynomial reducibility between all problems of the subset is possible, for example: vertex cover, maximum independent set and a complete subgraph in the graph. However, most likely, reducibility between subsets can take place only for separate single problems [7]. The problem of polynomial reducibility between subsets can appear algorithmically insoluble problem. Therefore, while if we cannot answer a question whether there is at least one universal *NP*-complete problem then the question if *P=NP* or not, is simply premature and has no sense yet. It is necessary to note also, that all conclusions made in the theory of problems from class *NP* on the basis of the proposal on "total" polynomial reducibility in a class *of NP-complete problem*s, can appear incorrect if the problem of existence of a universal problem appears insoluble.

REFERENCES

[1] Gary M., Johnson D. Computing machines and hard problems. – M: Mir, 1982. – 336 p.

[2] Cook S.A. Complexity of procedures of a conclusion of theorems. - Cybernetic collection of a new series, vol.12. - Moscow.: Mir, 1975.-P 5 - 15.

[3] Michael R. Garey and David S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freman, 1979.

[4] Karp R.M. Reducibility of combinational problems. - Cybernetic collection of a new series, vol. 12. - Moscow: Mir, 1975.-P16-38.

[5] S. Arora and S. Safra. Probabilistic checking of proofs: a new characterization of NP. To appear Journal of the ACM. Preliminary version in Proceedings of the Thirty Third Annual Symposium on the Foundations of Computer Science, IEEE,1992.

[6] Listrovoy S.V., Yablochkov S.V. Problem-solving procedure for the minimum vertex covers and maximum independent sets//Electronic modeling 2003, vol.25, No.2, P. 31-40.

[7] Listrovoy S.V. «About polynomial reducibility in class NP» Ukrainian Mathematical Congress −2009, Algebra and Number Theory.
http://www.imath.kiev.ua/

Listrovoy Sergey Vladimirovich, doctor of technical sciences, professor of Ukrainian State Academy of Railway Transport, Kharkov. In 1972 has finished high military command engineering school in Kharkov. The Area of the scientific studies of the problem to discrete optimization and graph theory and their use to analysis of the computing systems and networks.