Algorithm of Sub Exponential Complexity for the SAT Problem Solution

S.V. Listrovoy*
Ukrainian State Academy of Railway Transport,
Kharkov, Ukrainian
Email: *om1 {at} yandex.ru

V.M. Butenko
Ukrainian State Academy of Railway Transport,
Kharkov, Ukrainian

Abstract— **It is shown that through the use of merge operations of variables belonging to different disjunct can construct an algorithm for solving SAT-problem with time complexity $O(m\log_2(n)n^{\log_2(n)+4})$, where m is the number of clauses in the SAT-problem, and n is the number of variables in a Boolean function SAT-problem, ie . to construct an algorithm of subexponential solutions for SAT-task.**

***Keywords*:** SAT problem, subexponential complexity

I. INTRODUCTION

. The *SAT* problem is a task of satisfiability of a Boolean formula [1]. The formula is considered satisfiable if it has a set of variables to satisfy it, i.e. a set of values for all variables which makes it true. In Russian it is known as the problems 'выполнимость' (ВЫП). The problem finds practical application: in hard-and software verification of modern computers, in designing *FPGA* [2-6], in solving automated proving problems, linked with checking inconsistency of sets of disjunctions in propositional calculus. For example, complex rail infrastructure control processes including passenger traffic control, power supply of various objects, logic control of objects at stations and railway spans, and complex information transfer telecommunications subsystems between infrastructure objects require efficient solution to diagnose such objects, which is possible due to improvements in the mathematical apparatus used in fault detection systems part of which are methods to solve *SAT* problems. A *SAT* problem also finds wide application in cryptographic analysis [6], since encoding algorithms can be considered in terms of conjunctive normal form (*CNF*) and the problem of cryptographic analysis can be interpreted as the problem to find the solving set, in which the solving set is the secret key. A *SAT* problem is of importance in automated proving systems, in which a formula is a set of clauses considered as disjunction of some literals, namely, variables X and $\overline{X}$. The problem is of great value for circuit-satisfiability problems (*CIRCUIT-SAT*). There are a lot of exponential algorithms of its solution and heuristic approaches of polynomial complexity. Among them one should mention the Monien and Shpikermayer algorithm (1985), in which for a *3-SAT* problem a simple search is used: alternate substitution of each variable with 1 or 0, with consequent recursive solution of a smaller problem, having temporal complexity $O(1{,}84^n)$. Generally, it is possible to select two basic types of algorithms to solve *SAT* problems: local search algorithms, beginning with a set of values (though, it doesn't satisfy the whole formula), and then it is modified with successive approach to the satisfiable set; and so-called the *DPLL* algorithms (after the inventors Davis, Putnam, Logemann, Loveland; their description of the basic operational principles for the method dates back to 1968), which evade a tree of possible sets and make a depth-first search. A local search, as a rule, is probabilistic, because one should start with a set, which is taken at random, and it can impact many things. It should be mentioned that the *DPLL*-like algorithms are more deterministic, mostly due to the theory developed by Oliver Kullmann and Horst Luckhardt, linking these estimations with recurrent equation solution. Their idea proved to be fruitful, which made it possible to design programmes automatically proving new higher complexity estimations for algorithms based on these principles. Thus, algorithms, based on local search, win in practice, and the *DPLL*-like algorithms win in theory, for them it is possible to prove far stronger higher estimations. Sizes of the problems, being solved now by industrial solvers, amount to hundreds and thousands of variables, demonstrating high efficiency, but their basic algorithm is exponential anyway. The *SAT* problem dimensions in using modern technologies of the *FPGA* design is rapidly growing, therefore it is important to develop effective algorithms to solve *SAT* problems, i.e. algorithms of low temporal complexity which always give possibility to say if a Boolean function is satisfiable; and if it is satisfiable, to indicate a set of variables for which it is satisfiable. All known deterministic algorithms for *SAT* problem solution have exponential complexity, therefore the aim of the article is to demonstrate that the given task can be solved in subexponential time (growth rates $n^{\log n}$ exceeding any polynomial, but less than $2^{n^{\varepsilon}}$ for any ε>0, are called subexponential [7]).

II. FORMALIZATION OF A SAT PROBLEM AND ITS SOLUTION PROBLEM

Consider Boolean function $f(x_1,x_2,..,x_n)$ in conjunctive form $f(x_1,x_2,..,x_n)=(x_1^{\sigma_{11}}\vee x_2^{\sigma_{12}}\vee...\vee x_n^{\sigma_{1n}})\wedge...\wedge(x_1^{\sigma_{m1}}\vee x_2^{\sigma_{m2}}\vee...\vee x_n^{\sigma_{mn}})$,

where $x_i^{\sigma}=\begin{cases}x_i, -npu\ \sigma=1\\ \overline{x_i}-npu\ \sigma=0\end{cases}$.

The $\vee$, $\wedge$ operations are Boolean and are modeling simple

logic statements: ∨– «OR»; ∧– «AND». For any binary set $x=(x_1,x_2,..,x_n)$ the function takes on one of two possible values: one or zero. The task of satisfiability lies in an answer to the question whether there exists a set of variables $\tilde{o}=(x_1,x_2,..,x_n)$, which turns function *f* into one.

As [8] shows, a *SAT* problem can be considered as a problem of coverage, to do this, take a Boolean function to chart Boolean matrix *B*, in which the columns correspond to variables $(\tilde{O}_1,\tilde{O}_2,...,X_n)$ and $(\overline{\tilde{O}}_1,\overline{\tilde{O}}_2,...,\overline{X}_n)$, and the rows correspond to the disjunctions of the Boolean function. Generally, the number of columns in matrix *B* equals to *2n*, and the number of rows equals to the number of disjunctions *m* in the Boolean function.

For example, for Boolean function

$$F=(X_1\vee X_2\vee\tilde{O}_3)(\overline{X}_1\vee\overline{X}_2\vee\overline{\tilde{O}}_3)\cdot$$
$$\cdot(X_1\vee\overline{\tilde{O}}_3)(X_3\vee\overline{\tilde{O}}_1)(\tilde{O}_1\vee\overline{\tilde{O}}_2),$$

renumber the disjunctions of the Boolean functio (Table 1).

Numeration of disjunctions Table 1

1-$(X_1\vee X_2\vee\tilde{O}_3)$	2-$(\overline{X}_1\vee\overline{X}_2\vee\overline{\tilde{O}}_3)$	
3-$(X_1\vee\overline{\tilde{O}}_3)$	4-$(X_3\vee\overline{\tilde{O}}_1)$	5-$(\tilde{O}_1\vee\overline{\tilde{O}}_2)$,

Whereas, matrix *B* looks like

$$B=\begin{array}{c|cccccc} & X_1 & X_2 & X_3 & \overline{X}_1 & \overline{X}_2 & \overline{X}_3 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 1 & 1 \\ 3 & 1 & 0 & 0 & 0 & 0 & 1 \\ 4 & 0 & 0 & 1 & 1 & 0 & 0 \\ 5 & 1 & 0 & 0 & 0 & 1 & 0 \end{array}$$

The columns corresponding to variables X_i and $\overline{X}_i$ in matrix *B* are called inverse. If matrix *B* covers rows with 'one', belonging to non-inverse columns, it means that function *f* is satisfiable, if such coverage does not exist, it is unsatisfiable. Each variable X_i^{σ} in matrix *B*, in general, with *m* rows, i.e. corresponding to a Boolean function with *m* disjunctions, will be characterized by vector $H^{(i)}(h_1, h_2,\ldots, h_m)$, where $h_i = i$, if variable X_i^{σ} covers the *i*-row in matrix *B* and h_i =0, otherwise. In its turn, assign weighted estimator p_i for each vector, which equals the number component h_i different from zero. If we consider subset of two variables X_i^{σ} and X_j^{σ} or more, thus such a set is characterized by joint vector $H^{(i,j...)}$ combining components of the same name according to the rule

$$i\bigcup i=i;\ i\bigcup 0=i;0\bigcup i=i;0\bigcup 0=0. \quad (1)$$

For example, a given Boolean function is

$$f=(x_2\vee\overline{x}_1\vee\overline{x}_3)\,(x_4\vee\overline{x}_2\vee\overline{x}_3)\,(x_2\vee x_3\vee\overline{x}_4)$$
$$(x_2\vee\overline{x}_1\vee\overline{x}_4)\,(x_1\vee x_4\vee\overline{x}_2)\,(x_1\vee x_2\vee x_3)\,(x_1\vee x_3\vee\overline{x}_4)$$
$$(x_1\vee\overline{x}_2\vee\overline{x}_3)\,(\overline{x}_1\vee\overline{x}_2\vee\overline{x}_3)\,(x_2\vee x_3\vee\overline{x}_1)\,(x_3\vee\overline{x}_2\vee\overline{x}_4)$$
$$(x_1\vee\overline{x}_3\vee\overline{x}_4) \quad (2)$$

Renumber disjunctions

$$1\text{-}(x_2\vee\overline{x}_1\vee\overline{x}_3);2\text{-}(x_4\vee\overline{x}_2\vee\overline{x}_3)$$
$$3\text{-}(x_2\vee x_3\vee\overline{x}_4);4\text{-}(x_2\vee\overline{x}_1\vee\overline{x}_4);$$
$$5\text{-}(x_1\vee x_4\vee\overline{x}_2)\ 6\text{-}(x_1\vee x_2\vee x_3);$$
$$7\text{-}(x_1\vee x_3\vee\overline{x}_4);8\text{-}(x_1\vee\overline{x}_2\vee\overline{x}_3);$$
$$9\text{-}(\overline{x}_1\vee\overline{x}_2\vee\overline{x}_3);10\text{-}(x_2\vee x_3\vee\overline{x}_1);$$
$$11\text{-}(x_3\vee\overline{x}_2\vee\overline{x}_4);12\text{-}(x_1\vee\overline{x}_3\vee\overline{x}_4).$$

Write down their vectors $H^{(i)}$ with weighted estimators, determined according to the given rules (Table 2).

Vectors $H^{(i)}$ with weighted estimators

Table 2

1	x_1	H^1(0,0,0,0,5,6,7,8,0,0,0,12)	p_1=4
2	x_2	H^2(1,0,3,4,0,6,0,0,0,10,0,0,0)	p_2 =5
3	x_3	H^3(0,0,3,0,5,6,7,0,0,1,0,0,0)	p_3 =5
4	x_4	H^4(0,2,0,0,5,0,0,0,0,0,0,0,0)	p_4 =2
5	$\overline{x}_1$	$H^{\bar{1}}$(0,0,0,4,0,0,0,0,9,10,0,0)	$p_{\bar{1}}$ =3
6	$\overline{x}_2$	$H^{\bar{2}}$(0,2,0,0,5,0,0,8,9,0,11,0)	$p_{\bar{2}}$ =5
7	$\overline{x}_3$	$H^{\bar{3}}$(0,2,0,0,0,0,0,8,9,0,0,12)	$p_{\bar{3}}$ =4
8	$\overline{x}_4$	$H^{\bar{4}}$(0,0,3,4,0,0,7,0,0,0,11,12)	$p_{\bar{4}}$ =5

All pairs X_i and $\overline{X}_i$, all possible non-overlapping sets without inverse vertices based on the given pairs with their joint vectors and their weighted estimators, determined according to proportion (1) are given in Table 3, where vectors are presented in brackets, and weighted estimators of the vectors are in bold.

*Stages of procedure **A***

Table 3

1	2	3
All possible pairs X_i and $\overline{X}_i$ and their characteristics	All possible associations of all non-inverse pairs X_i and $\overline{X}_i$ and their characteristics	Number of subsets with an odd number of variables, which wasn't formed by procedure ***A***
$x_1\overline{x_2}$ (0,2,0,0,0,6,7,8,9,0,0,12) **7**	$x_1\overline{x_2}\ x_3\overline{x_4}$ (1,2,0,4,5,6,0,8,9,10,0,0) **8** $x_1\overline{x_2}\ x_4\overline{x_3}$ (0,2,3,4,5,6,7,8,0,0,0,0) **8**	$C_4^3=4$ $C_4^3=4$
$x_1\overline{x_3}$ (0,2,0,0,0,6,7,8,9,0,0,12) **6**	$x_1\overline{x_3}\ x_2\overline{x_4}$ (1,2,3,4,5,6,7,8,9,10,11,12) **12*** $x_1\overline{x_3}\ x_4\overline{x_2}$ (1,2,0,0,5,6,7,8,9,0,11,12) **9**	$C_4^3=4$ $C_4^3=4$
$x_1\overline{x_4}$ (0,0,3,4,0,6,7,8,0,0,11,12) **6**	$x_1\overline{x_4}\ x_2\overline{x_3}$ (1,2, 3,4,5,6,7,8, 9,0,11,12) **11** $x_1\overline{x_4}\ x_3\overline{x_2}$ (0,2,3,4,5,6,7,8,9,10,11,12) **11**	$C_4^3=4$ $C_4^3=4$
$x_2\overline{x_1}$ (1,0,3,4,0,6,0,0,9,10,0,12) **7**	$x_2\overline{x_1}\ x_4\overline{x_3}$ (1,2,3,4, 5,6,7,8, 9,0,11,12) **11** $x_2\overline{x_1}\ x_3\overline{x_4}$ (1,2,3,4,5,6,7,0,9,10,11,12) **11**	$C_4^3=4$ $C_4^3=4$
$x_2\overline{x_3}$ (1,2,0,0,0,6,7,8,9,0,0,12) **7**	$x_2\overline{x_3}\ x_1\overline{x_4}$ (1,2, 3,4,0,6,7,8,9,0, 11,12) **10** $x_2\overline{x_3}\ x_4\overline{x_1}$ (1,2,0,4,5,6,7,8,9,10,0,12) **10**	$C_4^3=4$ $C_4^3=4$
$x_2\overline{x_4}$ (1,0,3,4,0,6,7,8,0,0,11,12) **7**	$x_2\overline{x_4}\ x_3\overline{x_1}$ (1,0,3,4,5,6,7, 8,9,10,11,12) **11** $x_2\overline{x_4}\ x_1\overline{x_3}$ (1,2,3,4,0,6,7,8,9,0,11,12) **10**	$C_4^3=4$ $C_4^3=4$
$x_3\overline{x_1}$ (1,3,4,5,6,7,9,10,11) **9**	$x_3\overline{x_1}\ x_2\overline{x_4}$ (1, 2,3,4,5,6,7, 8,9,10,11,0) **11** $x_3\overline{x_1}\ x_4\overline{x_2}$ (1,2,3,4,5,6,7,8,9,10,11,0) **11**	$C_4^3=4$ $C_4^3=4$
$x_3\overline{x_2}$ (0,2,3,0,5,6,7,8,9,10,11,0) **9**	$x_3\overline{x_2}\ x_4\overline{x_1}$ (1,2,3,4,5,6,7,8,9,10,11,0) **11** $x_3\overline{x_2}\ x_1\overline{x_4}$ (0,2,3,4,5,6,7,8,9,10,11,12) **11**	$C_4^3=4$ $C_4^3=4$
$x_3\overline{x_4}$ (0,0,3,4,5,6,7,0,0,10,11,12) **8**	$x_3\overline{x_4}\ x_1\overline{x_2}$ (0,2,3,4,5,6,7,8,9,10,11,12)**11** $x_3\overline{x_4}\ x_2\overline{x_1}$ (1,0,3,4,5,6,7,9,10,11,12) **11**	$C_4^3=4$ $C_4^3=4$
$x_4\overline{x_1}$ (1,2,0,4,5,0,0,0,9,10,0,0) **6**	$x_4\overline{x_1}\ x_2\overline{x_3}$ (1,2,0,4,5, 6,7,8,9,10,12) **11** $x_4\overline{x_1}\ x_3\overline{x_2}$ (1,2,3,4,5,6,7,8,9,10,11,0) **11**	$C_4^3=4$ $C_4^3=4$
$x_4\overline{x_2}$ (0,2,0,0,5,0,0,8,9,0,11,0) **5**	$x_4\overline{x_2}\ x_1\overline{x_3}$ (0,2,0,0,0,5, 6,7,8,9,0,11,12) **8** $x_4\overline{x_2}\ x_3\overline{x_1}$ (1,2,3,4,5,6,7,8,9,10,11,0) **11**	$C_4^3=4$ $C_4^3=4$
$x_4\overline{x_3}$ (1,2,0,0,5,0,0,8,9,0,0,12) **6**	$x_4\overline{x_3}\ x_1\overline{x_2}$ (1,2,0,0,5,6,7,8,9,0,0,12) **8** $x_2\overline{x_1}$ (1,2,3,4,5,6,0,8,9,10,0,12) **10**	$C_4^3=4$ $C_4^3=4$ 2^7=128 subsets were formed implicitly with an odd number of variables

From Table 3 we can see that there exists subset $x_1\overline{x_3}\ x_2\overline{x_4}$ (1,2,3,4,5,6,7,8,9,10,11,12) p=**12**=m forming the coverage, and it is the satisfiable set for a reference Boolean function, in Table 3 it is marked with asterisk. Therefore, if we have an arbitrary set of non-inverse variables $\{X_i^{\sigma}\}$, for which weighted estimator $p_{i,\ldots,\kappa} = m$ of joint vector $H^{(i,\ldots,\kappa)}$, then it means that the set has a subset providing satisfiability of the Boolean function, as it covers all rows with 'one' in matrix B. Consider possibilities to form maximum sets of non-inverse variables $\{X_i^{\sigma}\}$ for an arbitrary Boolean function with n variables. Divide the set of variables of the Boolean function into two types of maximum subsets of variables, without inverse vertices. The first one corresponds to the sets, which can be classified as maximum, they are

$$\{(x_1x_2x_3\ldots x_n);(\overline{x_1}\,\overline{x_2}\,\overline{x_3}\ldots\overline{x_n})\};\{(x_1\overline{x_2}\,\overline{x_3}\ldots\overline{x_n});$$
$$(\overline{x_2}x_1\overline{x_3}\ldots\overline{x_n})\ldots(\overline{x_n}\,\overline{x_1}\,\overline{x_2}\ldots\overline{x_{n-1}})\}, \quad (3)$$
$$\{(\overline{x_1}x_2x_3\ldots x_n);(\overline{x_2}x_1x_3\ldots x_n);\ldots..(\overline{x_n}x_1x_2\ldots x_{n-1})\}$$

there are only $2n$+2 of them. The second type corresponds to all maximum subsets which can be formed on the base of unions by different methods of sets of variables X_i and $\overline{X_i}$, they can be presented as a two-partite graph, in which non-inverse variables are connected along the ribs (Fig.1), and every pair is characterized by its vector and weighted estimator. For a function of four variables all such pairs are given in Column 1 of Table 3.

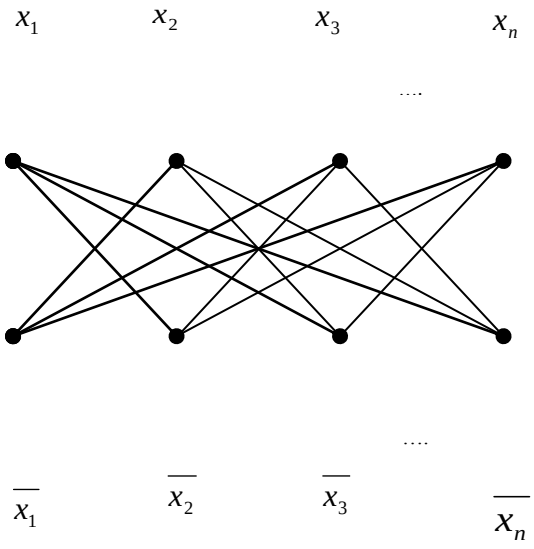

Fig.1 Graph of pairs of non-inverse variables

Consider procedure ***A*** of forming maximum sets without inverse vertices.

Procedure A

Step 1. Form all set of variables of a Boolean function of the first type, their vectors *H (i) (h1, h2, ..., hm)* and determine their weighted estimators $p_{i.}$.

Step 2. Check whether there are sets with p_i=m, if not, proceed to the next step, otherwise the procedure finishes the work, as the analyzed Boolean function is satisfiable.

Step 3. Form all possible sets from pairs X_i and $\overline{X_i}$ with double number of variables without non-inverse vertices and check among the resulting sets whether there are sets of weighted estimators p_i =m, if so, the Boolean function is satisfiable, and the procedure finishes, if not, proceed to the next step.

Step 4. Of all current sets, unite only those non-intersecting by elements of subsets and those without inverse vertices and again receive subsets of double number of variables without inverse vertices. Check out of the resulting sets whether there are sets of weighted estimator p_i =m, if so, the Boolean function is satisfiable, and the procedure finishes, if not, proceed to the next step.

Step 5. Check the cardinality of generated sets, whether it has reached value n, if n is even, or *(n-1,)* if n is odd, if not, proceed to step 4, otherwise proceed to the next step.

Step 6. Check out of the resulting sets whether there are sets of weighted estimator p_i =m, if so, the Boolean function is satisfiable, if not, the Boolean function is not satisfiable.

In fact, in procedure ***A*** the doubling of sets repeats until, on the base of resulting subsets, a subsequent union is not possible, due to inverse vertices or a further union does not change the resulting subsets. It is clear, that such a situation occurs at step k of procedure ***A*** and when the cardinality of generated subsets reaches *(n)*, if n is even, or *(n-1)* if n is odd. A feature of the procedure is that maximal sets without an odd number of variables will always be as subsets within subsets with an even number of variables, but greater by one. The number of subsets that were formed implicitly by procedure ***A***, for each union is given in Column 2 of Table 3.

The number of subsets formed at the first step of the procedure is equal to *n (n-1)*, at the second one is *n (n-1) (n-2)*, etc. at the next steps will be summed, which is shown by ratio (4)

$$n(n\text{-}1)+ n(n\text{-}1)\ (n\text{-}2)+ n(n\text{-}1)\ (n\text{-}2)\ (n\text{-}4)+ n(n\text{-}1)\ (n\text{-}2)\ (n\text{-}4)\ (n\text{-}8)\ldots +\ldots, \quad (4)$$

Rewrite (4) in the form

$$n(n\text{-}1)[1+(n\text{-}2^1)+ (n\text{-}2^1)\ (n\text{-}2^2)+\ldots+ (n\text{-}2^1)\ (n\text{-}2^2)\ (n\text{-}2^3)\ldots\ \ldots(n\text{-}2^k)] = n(n\text{-}1)b. \quad (5)$$

The process of summing in (5) should stop at step *k-th* when reaches $2k = n$, i.e. and $k = log2\ (n)$, where in the last summand the number of multipliers equals *log2 (n)*. As follows from (5) the inequality is true

$$b<n^0+n^1+n^2+n^3+\ldots+n^{\log_2 n} \quad (6)$$

Suppose in (6) all $n^i = n^{\log_2 n}$ one can write inequality (7)

$$n^0+n^1+n^2+n^3+..+n^{\log_2 n} < \log_2(n)n^{\log_2 n+1}, \qquad (7)$$

it follows from (5) and (7) that the number of sets which has to be built in procedure ***A*** does not exceed $\log_2(n)n^{\log_2 n+3}$, and the number of operations to form these sets does not exceed $m\log_2(n)n^{\log_2(n)+4}$. Considering the formation of the first type sets, the total complexity of maximal sets formation without inverse vertices equals $O(m\log_2(n)n^{\log_2(n)+4}+mn(2n+2))\approx O(m\log_2(n)n^{\log_2(n)+4})$. Procedure ***A*** for example (2) can be seen in Table 3, where the first column contains all pairs of non-inverse variables, excluding ones corresponding to type 1. The process of set formation with four variables can be displayed as a graph (see Figure 2).

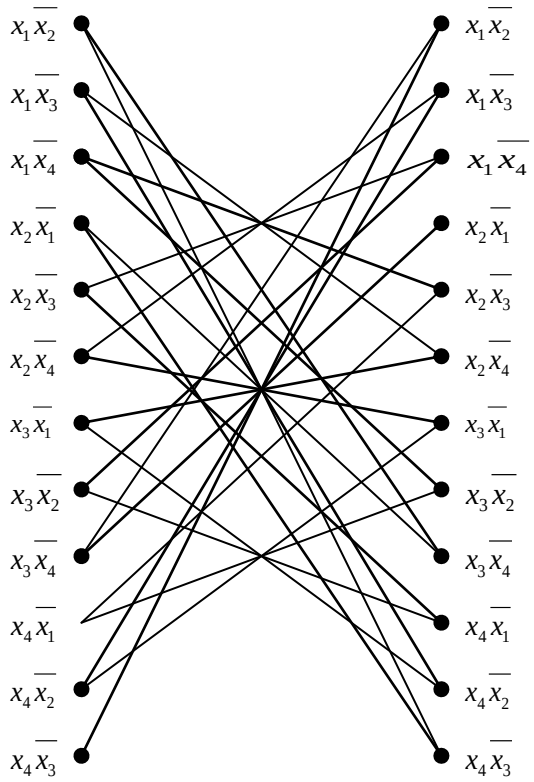

Figure 2 Graph *G* for searching maximal sets of non-inverse vertices

are connected by ribs if subsets do not overlap each other and without non-inverse variables. In general, orders of vertices in graph *G* (Fig. 2) are equal to *n*-2, and for the considered example 4-2=2. The results of unions are shown in the second column of Table 3. The third column of Table 3 shows the number of variables unions by three, which were formed implicitly by the procedure, their total number being 2^7=128. As shown in Table 3 we have for example (2) performing a single set of variables of the second type (1,2,3,4,5,6,7,8,9,10,11,12) with weight *p*=12=*m*, and a subset of the first type with weight *p*=12=*m*.

It should be noted that if graph G is built for an arbitrary Boolean function of *n* variables, the initial degree of vertices is *n*-2, in further unions, in 2, in 4, in 8, etc., the order of vertices in graphs, in which these subsets corresponding to the graph vertices, will decrease exponentially, the sets with an odd number of variables being listed automatically, and the procedure does not spent time on it, because they are part of those listed in procedure *A*. It is seen from proportion (5), which explains the fact that the number of maximal sets, consisting of non-inverse variables are subexponential and that procedure *A* lists these maximal sets of non-inverse variables in subexponential time.

III. Conclusion

Thus, for a *SAT* problem it has been proposed an algorithm of subexponential complexity, if $\varepsilon = 1$. Clearly, the complexity of procedure ***A*** is rather high, but if we take ratio $\frac{2^n}{n^{\log_2 n}}$, for example, where *n* = 100 and *n* 1000, we get respectively $1{,}3\cdot10^{16}$ and $1{,}1\cdot10^{271}$, i.e. time gain could be potentially significant. It should be noted that procedure *A* for unsatisfiable functions enumerates all maximal sets of non-inverse variables and makes, at least, the number of steps defined by proportion (8), i.e. it can list in subexponential time all sets of variables at which the Boolean function is true.

The above procedure ***A*** makes it possible to list all sets of variables for which the analyzed Boolean function is satisfiable in subexponential time, that is, to solve the problem of ”full satisfiability “.

References

1. Geri M., Dzhonson D. Vychislitelnye mashiny i trudnoreshaemye zadachi. – M.: Mir, 1982. – 336 s.

2. A.V. Skatov, A.V. Borisevich Apparatnoe uskorenie reshenija zadach vypolnimosti dlja postroenija testov cifrovyh shem // Informatika, elektronika, svjaz sbornik nauchnyh trudov – Sevastopol: Izd-vo Sev. NTU, 2008, S. 9-15.

3. Cheremisinova L., Novikov D. *SAT*-Based Approach to Verification of Logical Descriptions with Functional Indeterminacy // 8th International Workshop on Boolean Problems. Freiberg: September 18–19, 2008. P. 59–66

4. V. Kheterpal, V. V. Rovner, T. G. Hersan, D. Motiani, Y. Takegawa, A. J. Strojwas, and L. Pileggi. Design Methodology for IC Manufacturability Based on Regular Logic Bricks. In Proceedings of the 42nd Conference on Design Automation, pages 353–358, 2005.

5. B. Taylor, L. Pileggi. Exact Combinatorial Optimization Methods for Physical Design of Regular Logic Bricks. In Proceedings of the 44th Conference on Design Automation, pages 344–349, 2007.

6. V.I. Dulkejt, R.T. Fajzullin, I.G. Hnykin Nepreryvnye approksimacii reshenija zadachi "vypolnimost'" primenitelno k kriptograficheskomu analizu asimmetrichnyh

shifrov. // Kompjuternaja optika №1, T33, 2009, s. 86-90 Omskij gosudarstvennyj universitet.

7. H. Papadimitriu, K. Stajglic Kombinatornaja optimizacija Algoritmy i sozhnost Per. S angl. – M., MIR, 1985. – 512 s.

8. Listrovoy S.V. «On Correlation of P And NP Classes» // I.J. Modern Education and Computer Science,2012,3,21-27.

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.

Butenko Vladimir Mikhaylovich, Ph.D., Associate Professor of specialized systems chair Ukrainian State Academy of Railway Transport. Graduated in 1994 year Kharkov State Academy of Railway Transport and traineeship with it. Circle of scientific interests: microprocessor control system and measurement, quality, standardization and certification on the railway transport.