Linear Programming Formulation of kSAT

Algirdas Antano Maknickas

***Abstract*—With using of multi-nary logic analytic formula and linear programming proposition that *kSAT is in P and could be solved in linear time* was proved.**

***Index Terms*—Linear Programming, kSAT, multivalued logic.**

I. Introduction

THE Boolean satisfiability (SAT) problem [1] is defined as follows: Given a Boolean formula, check whether an assignment of Boolean values to the propositional variables in the formula exists, such that the formula evaluates to true. If such an assignment exists, the formula is said to be satisfiable; otherwise, it is unsatisfiable. For a formula with m variables, there are 2m possible truth assignments. The conjunctive normal form (CNF)

$$(X_1 \vee X_2) \wedge (X_3 \vee X_4) \wedge \cdots \wedge (X_{n-1} \vee X_n) \quad (1)$$

is most the frequently used for representing Boolean formulas, where $\neg\forall X_i$ are independent. In CNF, the variables of the formula appear in literals (e.g., x) or their negation (e.g., ¬x (logical NOT ¬)). Literals are grouped into clauses, which represent a disjunction (logical OR ∨) of the literals they contain. A single literal can appear in any number of clauses. The conjunction (logical AND ∧) of all clauses represents a formula.

Several algorithms are known for solving the 2 - satisfiability problem; the most efficient of them take linear time [2], [3], [4]. Instances of the 2-satisfiability or 2SAT problem are typically expressed as 2-CNF or Krom formulas [2]

SAT was the first known NP-complete problem, as proved by Cook and Levin in 1971 [1] [5]. Until that time, the concept of an NP-complete problem did not even exist. The problem remains NP-complete even if all expressions are written in conjunctive normal form with 3 variables per clause (3-CNF), yielding the 3SAT problem. This means the expression has the form:

$$\begin{aligned}(X_1 \vee X_2 \vee X_3) \wedge (X_4 \vee X_5 \vee X_6) \wedge \ldots \\ \wedge (X_{n-2} \vee X_{n-1} \vee X_n)\end{aligned} \quad (2)$$

NP-complete and it is used as a starting point for proving that other problems are also NP-hard. This is done by polynomial-time reduction from 3-SAT to the other problem.

In 2010, Moustapha Diaby provided two further proofs for P=NP. His papers Linear programming formulation of the vertex colouring problem and Linear programming formulation of the set partitioning problem give linear programming formulations for two well-known NP-hard problems [6], [7].

The goal of this paper is proof of proposition that kSAT is in P using multi logic formula of discrete second order logic proposed first in [8], [9] and could be solved in $O(m)$.

Manuscript received December 01, 2012; revised December 19, 2012.

A.A. Maknickas is with the Department of Information Technologies, Vilnius Gediminas Technical University, Vilnius, Lithuania LT-06227 e-mail: algirdas.maknickas@vgtu.lt.

II. 2SAT is in P

THEOREM 1 If all variables are unique, equation

$$\max \beta_2 (X_1, X_2, ..., X_{n-1}, X_n) \quad (3)$$

where

$$\begin{aligned}&\beta_2 (X_1, X_2, ..., X_{n-1}, X_n) = \\ &\mu (X_1 + X_2, \mu (X_3 + X_4, ..., \\ &\ \mu (X_{n-3} + X_{n-2}, X_{n-1} + X_n)))\end{aligned}$$

$$\mu(a,b) = \begin{array}{c|ccc} \lfloor a \rfloor \backslash \lfloor b \rfloor & 0 & 1 & 2 \\ \hline 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 2 & 0 & 1 & 1 \end{array} \quad (4)$$

and $+$ *is algebraic summation could be solved for* $\forall X_i \in \{0,1\}$ *in* $O(m)$.

Proof. Let start from investigation of

$$f(x_1, x_2, ..., x_m) = \prod_{i=1}^{m} x_i, \ \forall x_i \in \mathbb{R} \quad (5)$$

in hyper-cube of sides $[0,1]$. This function is convex, because

$$\prod_{i=1}^{m} x_i \le \sum_{i=1}^{m} x_i \alpha^i \quad (6)$$

$$1 = \sum_{i=1}^{m} \alpha^i \quad (7)$$

It is obvious, on right side of (6) we have hyper-plane which goes through vertexes $(0,0,0,..,0)$ and $(1,1,1,..,1)$ of investigating hyper-cube. This hyper-plane has global maximum like a function $f(x_1, x_2, ..., x_m)$ and it equals 1. So we could resume that finding of global maximum of $f(x_1, x_2, ..., x_m)$ could be replaced by finding of global maximum of this hyper plane or

$$\max \prod_{i=1}^{m} x_i = \frac{1}{n} \max \sum_{i=1}^{n} x_i \quad (8)$$

Let start to investigate (3) $\forall X_i \in \mathbb{R}, i \in \{1, 2, ..., n\}$. Function β_2 could be calculated within $O(m)$ [10]. According to (4) equation (3) could be rewritten as follow

$$\begin{aligned}&\max \mu (X_1 + X_2, \mu (X_3 + X_4, ..., \\ &\quad \mu (X_{n-3} + X_{n-2}, X_{n-1} + X_n))) = \\ &\max \mu (X_1 + X_2, \max \mu (X_3 + X_4, ..., \\ &\quad \max \mu (X_{n-3} + X_{n-2}, X_{n-1} + X_n)))\end{aligned} \quad (9)$$

So, m local partial maximums must satisfy equalities $\max \mu(X_{k-1} + X_k) = 1$ to avoid 0 result of global maximum. This leads to inequalities $1 \le (X_{k-1} + X_k) \le 2$ but if we want to use (8) we must leave $(X_{k-1} + X_k) = 1$ Now we could start to solve satisfiability problem as global maximum of (3). If we have all variables unique, (9) could be solved repeating solving of system of LP equations

$$\begin{cases} \max \sum_{i=k-1}^{k} X_i \\ X_k + X_{k-1} = 1 \\ 0 \le X_{k-1} \le 1 \\ 0 \le X_k \le 1 \end{cases} \quad (10)$$

Each of them has not zero max and could be solved using best known algorithm of linear programming [11] in $O\left(n^{3.5}\right)$ or in $O(2^{3.5})$. So all clauses should be optimized in $O\left(2^{3.5}m\right)$ ◯

A. Special cases

THEOREM 1.1 If some of unique variables are negations $\neg X_i$, equation

$$\max \beta_2 \left(X_1, X_2, ..., X_{n-1}, X_n\right) \quad (11)$$

could be solved for $\forall X_i \in \{0,1\}$ in $O\left(m\right)$.

Proof. If we have all variables unique, let replace all negations $\neg X_k$ with $X_k^{'}$ and all others just renamed by $X_i^{'}$. Now (11) could be solved repeating solving of system of LP equations

$$\begin{cases} \max \sum_{i=k-1}^{k} X_i^{'} \\ X_k^{'} + X_{k-1}^{'} = 1 \\ 0 \le X_{k-1}^{'} \le 1 \\ 0 \le X_k^{'} \le 1 \end{cases} \quad (12)$$

Each of them has not zero max and could be solved in $O(2^{3.5})$. Going back to old variables do not change complexity of each solution. So all clauses should be optimized in $O\left(2^{3.5}m\right)$ ◯

THEOREM 1.2 If some of variables in different clauses are not unique, equation

$$\max \beta_2 \left(X_1, X_2, ..., X_{n-1}, X_n\right) \quad (13)$$

could be solved for $\forall X_i \in \{0,1\}$ in $O\left(m\right)$.

Proof. If X_n, X_{n-1} are unique, let start to solve (13) starting from a system of LP equations

$$\begin{cases} \max \sum_{i=n-1}^{n} X_i \\ X_n + X_{n-1} = 1 \\ 0 \le X_{n-1} \le 1 \\ 0 \le X_n \le 1 \end{cases} \quad (14)$$

Now we know two unique variables. So maximum is reached and system of equation is solved in $O(2^{3.5})$. If $X_n = X_{n-1}$, $X_n = 1 \wedge X_{n-1} = 1$. All other sub-equation could be solved as follow: if solving sub-equation has one variable with earlier found value, LP equations

$$\begin{cases} \max \sum_{i=k-1}^{k} X_i \\ 1 \le X_k + X_{k-1} \le 2 \\ 0 \le X_{k-1} \le 1 \\ 0 \le X_k \le 1 \end{cases} \quad (15)$$

could be solved by reducing of (15) within inserting of earlier found variable value resigned to 1 (lower and upper bound could be increased in case the broken plane result to 1 earlier and than is flat in 2D space); if all variables of solving sub-equations (15) aren't unique, these variables could be reassigned to 1 and solving repeated within next clause; if solving sub-equation has two unique variables, LP equations

$$\begin{cases} \max \sum_{i=k-1}^{k} X_i \\ X_k + X_{k-1} = 1 \\ 0 \le X_{k-1} \le 1 \\ 0 \le X_k \le 1 \end{cases} \quad (16)$$

could be solved. So all clauses should be optimized in $O\left(2^{3.5}m\right)$ ◯

THEOREM 1.3 If some of variables in different clauses are not unique and are negation each other, equation

$$\max \beta_2 \left(X_1, X_2, ..., X_{n-1}, X_n\right) \quad (17)$$

could be solved for $\forall X_i \in \{0,1\}$ in $O\left(m\right)$.

Proof. Let mark all variables they are unique or not starting from the end of CNF. Now cycle through n variables must be repeated to rename them to $X_k^{'}$ so that new replaced variables marked as not unique will be negations which must be replaced with $1 - X_k^{'}$, if they found second time. In kind first time found not unique variable could be assigned to 0. This lead to value of 1 for negation. It could be done in $O\left(2n\right)$. Now we start solving process for last clause. If $X_n^{'} = \neg X_{n-1}^{'}$, $X_n^{'} = 1$. If $X_n^{'}, X_{n-1}^{'}$ are unique, let start to solve (17) from a system of LP equations

$$\begin{cases} \max \sum_{i=n-1}^{n} X_i^{'} \\ X_n^{'} + X_{n-1}^{'} = 1 \\ 0 \le X_{n-1}^{'} \le 1 \\ 0 \le X_n^{'} \le 1 \end{cases} \quad (18)$$

Now we know two or one unique variables. All other sub-equation could be solved as follow: if solving sub-equation has one variable with earlier found value, LP equations

$$\begin{cases} \max \sum_{i=k-1}^{k} X_i \\ 1 \le X_k + X_{k-1} \le 2 \\ 0 \le X_{k-1} \le 1 \\ 0 \le X_k \le 1 \end{cases} \quad (19)$$

could be solved by reducing of (19) within inserting of earlier found variable value reassigned to 1 (lower and upper bound could be increased in case the broken plane result to 1 earlier and than is flat in 2D space); if all variables of solving sub-equations aren't unique and are negations of earlier found values in a different clauses, they values could be assigned to 0 and values of residual variables leads to 1; if all variables of solving sub-equations aren't unique and are negations of earlier found values in the same clause, one of variables must be assigned to 1 and other to 0; if at least two clauses $\exists$, where $X_i \wedge \neg X_i$, CNF is not satisfiable ;if solving sub-equation has two unique variables, LP equations

$$\begin{cases} \max \sum_{i=k-1}^{k} X_i \\ X_k + X_{k-1} = 1 \\ 0 \le X_{k-1} \le 1 \\ 0 \le X_k \le 1 \end{cases} \quad (20)$$

could be solved. Each sub-system of equation is solved in $O(2^{3.5})$ to reach global maximum 1. Finally, all clauses should be optimized m times or in $O\left(2^{3.5}m + 2n\right)$. In case $m \ge n$ we have $O\left(m\right)$ algorithm complexity ◯

ISBN: 978-988-19252-6-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

III. 3SAT IS IN P

THEOREM 2 If all variables are unique, equation

$$\max \beta_3 (X_1, X_2, ..., X_{n-1}, X_n) \quad (21)$$

where

$$\begin{aligned} &\beta_3 (X_1, X_2, ..., X_{n-1}, X_n) = \\ &\mu (X_1 + X_2 + X_3, \mu (X_4 + X_5 + X_6, ..., \\ &\mu (X_{n-5} + X_{n-4} + X_{n-3}, X_{n-2} + X_{n-1} + X_n))) \end{aligned} \quad (22)$$

$$\mu (a, b) = \begin{array}{c|cccc} a\backslash b & 0 & 1 & 2 & 3 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 2 & 0 & 1 & 1 & 1 \\ 3 & 0 & 1 & 1 & 1 \end{array} \quad (23)$$

and + is algebraic summation could be solved for $\forall X_i \in \{0, 1\}$ in $O(m)$.

Proof. Let start to investigate (21) when $\forall X_i \in \mathbb{R}, i \in \{1, 2, ..., n\}$. Function β_3 could be calculated within $O(m)$ [10]. According to (22) equation (21) could be rewritten as follow

$$\begin{aligned} &\max \mu (X_1 + X_2 + X_3, \mu (X_4 + X_5 + X_6, ..., \\ &\mu (X_{n-5} + X_{n-4} + X_{n-3}, X_{n-2} + X_{n-1} + X_n))) \end{aligned} \quad (24)$$

So, m local partial maximums must satisfy equalities $\max \mu(X_{k-2} + (X_{k-1} + X_k) = 1$ to avoid 0 result of global maximum. This leads to inequalities $1 \leq (X_{k-2} + X_{k-1} + X_k) \leq 3$ but if we want to use (8) we must leave $(X_{k-2} + X_{k-1} + X_k) = 1$

If we have all variables unique, (21) could be solved repeating solving of system of LP equations

$$\begin{cases} \max \sum_{i=k-2}^{k} X_i \\ X_k + X_{k-1} + X_{k-2} = 1 \\ 0 \leq X_{k-2} \leq 1 \\ 0 \leq X_{k-1} \leq 1 \\ 0 \leq X_k \leq 1 \end{cases} \quad (25)$$

Each of them has not zero max and could be solved in $O(3^{3.5})$. So all clauses should be optimized in $O\left(3^{3.5} m\right)$ ◯

A. Special cases

THEOREM 2.1 If some of unique variables are negations $\neg X_i$, equation

$$\max \beta_3 (X_1, X_2, ..., X_{n-1}, X_n) \quad (26)$$

could be solved for $\forall X_i \in \{0, 1\}$ in $O(m)$.

Proof. If we have all variables unique, let replace all negations $\neg X_k$ with $X_k^{'}$ and all others just renamed by $X_i^{'}$. Now (26) could be solved repeating solving of system of LP equations

$$\begin{cases} \max \sum_{i=k-2}^{k} X_i^{'} \\ X_k^{'} + X_{k-1}^{'} + X_{k-1}^{'} = 1 \\ 0 \leq X_{k-2}^{'} \leq 1 \\ 0 \leq X_{k-1}^{'} \leq 1 \\ 0 \leq X_k^{'} \leq 1 \end{cases} \quad (27)$$

Each of them has not zero max and could be solved in $O(3^{3.5})$. Going back to old variables do not change complexity of each solution. So all clauses should be optimized in $O\left(3^{3.5} m\right)$ ◯

THEOREM 2.2 If some of variables in different clauses are not unique, equation

$$\max \beta_3 (X_1, X_2, ..., X_{n-1}, X_n) \quad (28)$$

could be solved for $\forall X_i \in \{0, 1\}$ in $O(m)$.

Proof. If X_n, X_{n-1}, X_{n-2} are unique, let start to solve (28) starting from a system of LP equations

$$\begin{cases} \max \sum_{i=n-2}^{n} X_i \\ X_n + X_{n-1} + X_{n-2} = 1 \\ 0 \leq X_{n-2} \leq 1 \\ 0 \leq X_{n-1} \leq 1 \\ 0 \leq X_n \leq 1 \end{cases} \quad (29)$$

Now we know three unique variables. Maximum is reached and system of equation is solved in $O(3^{3.5})$. If $X_n = X_{n-1}$, $X_n = 1 \wedge X_{n-1} = 1 \wedge X_{n-2} = 1$. All other sub-equation could be solved as follow: if solving sub-equation has one or two variables with earlier found value, LP equations

$$\begin{cases} \max \sum_{i=k-2}^{k} X_i \\ 1 \leq X_k + X_{k-1} + X_{k-2} \leq 3 \\ 0 \leq X_{k-2} \leq 1 \\ 0 \leq X_{k-1} \leq 1 \\ 0 \leq X_k \leq 1 \end{cases} \quad (30)$$

could be solved by reducing of (30) with inserting of earlier found variables value reassigned to 1; if solving sub-equation three unique variables, LP equations

$$\begin{cases} \max \sum_{i=k-2}^{k} X_i \\ X_k + X_{k-1} + X_{k-2} = 1 \\ 0 \leq X_{k-2} \leq 1 \\ 0 \leq X_{k-1} \leq 1 \\ 0 \leq X_k \leq 1 \end{cases} \quad (31)$$

could be found. So all clauses should be optimized in $O\left(3^{3.5} m\right)$ ◯

THEOREM 2.3 If some of variables in different clauses are not unique and are negation each other, equation

$$\max \beta_3 (X_1, X_2, ..., X_{n-1}, X_n) \quad (32)$$

could be solved for $\forall X_i \in \{0, 1\}$ in $O(m)$.

Proof. Let mark all variables they are unique or not starting from the end of CNF. Now cycle through n variables must be repeated to rename them to $X_k^{'}$ so that new replaced variables marked as not unique will be negations which must be replaced with $1 - X_k^{'}$, if they found second time. In kind first time found not unique variable could be assigned to 0. This lead to value of 1 for negation. It could be done in $O(2n)$. Now we start solving process for last clause. If not all variables are unique in the first clause, after sorting of variables so that third one will be negation and second one will be unique, LP equations

$$\begin{cases} \max X_n^{'} + X_{n-1}^{'} + 1 - X_n^{'} \\ X_n^{'} + X_{n-1}^{'} + 1 - X_{n-2}^{'} = 1 \\ 0 \leq X_{n-2}^{'} \leq 1 \\ 0 \leq X_{n-1}^{'} \leq 1 \\ 0 \leq X_n^{'} \leq 1 \end{cases} \quad (33)$$

ISBN: 978-988-19252-6-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

could be solved. If $X_n^{'}, X_{n-1}^{'}, X_{n-2}^{'}$ are unique, let start to solve (22) from a system of LP equations

$$\begin{cases} \max \sum_{i=n-2}^{n} X_i^{'} \\ X_n^{'} + X_{n-1}^{'} + X_{n-2}^{'} = 1 \\ 0 \le X_{n-2}^{'} \le 1 \\ 0 \le X_{n-1}^{'} \le 1 \\ 0 \le X_n^{'} \le 1 \end{cases} \tag{34}$$

Now we know three or two unique variables. All other sub-equation could be solved as follow: if solving sub-equation has one or two variable with earlier found value, LP equations

$$\begin{cases} \max \sum_{i=k-2}^{k} X_i^{`} \\ 1 \le X_k^{`} + X_{k-1}^{`} + X_{k-2}^{`} \le 3 \\ 0 \le X_{k-2}^{`} \le 1 \\ 0 \le X_{k-1}^{`} \le 1 \\ 0 \le X_k^{`} \le 1 \end{cases} \tag{35}$$

where one or two of three variables $X_{k-2}^{`}, X_{k-1}^{`}, X_k^{`}$ are equal $1 - X_q^{'}$ could be solved by reducing of (35) within inserting of earlier found variable value reassigned to 1; if all variables of solving sub-equations aren't unique and are negations of earlier found values in a different clauses, they values could be assigned to 0 and values of residual variables leads to 1; if all variables of solving sub-equations aren't unique and are negations of earlier found values in the same clause, one of variables must be assigned to 1 and other to 0; if solving sub-equation has three unique variables, LP equations

$$\begin{cases} \max \sum_{i=k-2}^{k} X_i \\ X_k + X_{k-1} + X_{k-2} = 1 \\ 0 \le X_{k-2} \le 1 \\ 0 \le X_{k-1} \le 1 \\ 0 \le X_k \le 1 \end{cases} \tag{36}$$

could be solved. Each sub-system of equation is solved in $O(3^{3.5})$ to reach global maximum 1. Finally, all clauses should be optimized m times or in $O\left(3^{3.5}m + 2n\right)$. In case $m \ge n$ we have $O\left(m\right)$ algorithm complexity ◯

IV. kSAT is in P

THEOREM 3 If all variables are unique, equation

$$\max \beta_k \left(X_1, X_2, ..., X_{n-1}, X_n\right) \tag{37}$$

where

$$\begin{gathered} \beta_k \left(X_1, X_2, ..., X_{n-1}, X_n\right) = \\ \mu\left(\sum_{i=1}^{k} X_i, \mu\left(\sum_{i=k+1}^{2k} X_i, ..., \right.\right. \\ \left.\left.\mu\left(\sum_{i=n-2k+1}^{n-k} X_i, \sum_{i=n-k+1}^{n} X_i\right)\right)\right) \end{gathered} \tag{38}$$

$$\mu\left(a,b\right) = \begin{array}{c|cccccc} a\backslash b & 0 & 1 & 2 & \dots & n-1 \\ \hline 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 1 & 1 & \dots & 1 \\ 2 & 0 & 1 & 1 & \dots & 1 \\ \dots & & & & & \\ n-1 & 0 & 1 & 1 & \dots & 1 \end{array} \tag{39}$$

where + is algebraic summation could be solved for $\forall X_i \in \{0, 1\}$ *in* $O\left(k^{3.5}m\right)$.

Proof. Let start to investigate 37 when $\forall X_i \in \mathbb{R}, i \in \{1, 2, ..., n\}$. Function β_k could be calculated within $O\left(m\right)$ [10]. According to 4 equation 37 could be rewritten as follow

$$\begin{gathered} \max \mu\left(\sum_{i=1}^{k} X_i, \mu\left(\sum_{i=k+1}^{2k} X_i, ..., \right.\right. \\ \left.\left.\mu\left(\sum_{i=n-2k+1}^{n-k} X_i, \sum_{i=n-k+1}^{n} X_i\right)\right)\right) \end{gathered} \tag{40}$$

If we have all variables unique, (38) could be solved repeating solving of system of LP equations

$$\begin{cases} \max \sum_{i=k+1}^{2k} X_i \\ \sum_{i=k+1}^{2k} X_i = 1 \\ 0 \le X_i \le 1 \ \wedge \forall i \in \{1, 2, ..., n\} \end{cases} \tag{41}$$

Each of them has not zero max and could be solved in $O(k^{3.5})$. So all clauses should be optimized in $O\left(k^{3.5}m\right)$ ◯

A. Special cases

THEOREM 3.1 If some of unique variables are negations $\neg X_i$*, equation*

$$\max \beta_k \left(X_1, X_2, ..., X_{n-1}, X_n\right) \tag{42}$$

could be solved for $\forall X_i \in \{0, 1\}$ *in* $O\left(k^{3.5}m\right)$.

Proof. If we have all variables unique, let replace all negations $\neg X_k$ with Y_k and all others just renamed by Y_i. Now (42) could be solved repeating solving of system of LP equations

$$\begin{cases} \max \sum_{i=k+1}^{2k} Y_i \\ \sum_{i=k+1}^{2k} Y_i = 1 \\ 0 \le Y_i \le 1 \ \wedge \forall i \in \{1, 2, ..., n\} \end{cases} \tag{43}$$

Each of them has not zero max and could be solved in $O(k^{3.5})$. Going back to old variables do not change complexity of each solution. So all clauses should be optimized in $O\left(k^{3.5}m\right)$ ◯

THEOREM 3.2 If some of variables in different clauses are not unique, equation

$$\max \beta_k \left(X_1, X_2, ..., X_{n-1}, X_n\right) \tag{44}$$

could be solved for $\forall X_i \in \{0, 1\}$ *in* $O\left(k^{3.5}m\right)$.

Proof. If $X_n, X_{n-1}, ..., X_{n-k+1}$ are unique, let start to solve (44) starting from a system of LP equations

$$\begin{cases} \max \sum_{i=n-k+1}^{n} X_i \\ \sum_{i=n-k+1}^{n} X_i = 1 \\ 0 \le X_i \le 1 \ \wedge \forall i \in \{1, 2, ..., n\} \end{cases} \tag{45}$$

Now we know k unique variables. Maximum is reached and system of equation is solved in $O(k^{3.5})$. If $X_n = X_{n-1}$, $X_n = 1 \wedge X_{n-2} = 1, ...$ All other sub-equation could be solved as follow: if solving sub-equation has at least one variable with earlier found value, LP equations

$$\begin{cases} \max \sum_{i=n-k+1}^{n} X_i \\ 1 \le \sum_{i=n-k+1}^{k} X_i \le k \\ 0 \le X_i \le 1 \ \wedge \forall i \in \{1, 2, ..., n\} \end{cases} \tag{46}$$

ISBN: 978-988-19252-6-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

could be solved by reducing of (46) with inserting of earlier found variables value reassigned to 1; if solving sub-equation k unique variables, LP equations

$$\begin{cases} \max \sum_{i=n-k+1}^{k} X_i \\ \sum_{i=n-k+1}^{k} X_i = 1 \\ 0 \le X_i \le 1 \ \wedge \forall i \in \{1, 2, ..., n\} \end{cases} \quad (47)$$

could be solved. So all clauses should be optimized in $O\left(k^{3.5}m\right)$ ◯

THEOREM 3.3 If some of variables in different clauses are not unique and are negation each other, equation

$$\max \beta_k \left(X_1, X_2, ..., X_{n-1}, X_n\right) \quad (48)$$

could be solved for $\forall X_i \in \{0, 1\}$ *in* $O\left(k^{3.5}m\right)$.

Proof. Let mark all variables they are unique or not starting from the end of CNF. Now cycle through n variables must be repeated to rename them to $X_k^{'}$ so that new replaced variables marked as not unique will be negations which must be replaced with $1 - X_k^{'}$, if they found second time. In kind first time found not unique variable could be assigned to 0. This lead to value of 1 for negation. It could be done in $O\left(2n\right)$. Now we start solving process for last clause. If not all variables are unique in the first clause, after sorting of variables so that negations occurs at the end of the list, LP equations

$$\begin{cases} \max \sum_{i=n-k+1}^{l} X_i^{'} + \sum_{i=l+1}^{n} \left(1 - X_i^{'}\right) \\ \sum_{i=n-k+1}^{l} X_i^{'} + \sum_{i=l+1}^{n} \left(1 - X_i^{'}\right) = 1 \\ 0 \le X_i^{'} \le 1 \ \wedge \forall i \in \{1, 2, ..., n\} \end{cases} \quad (49)$$

could be solved. If $X_n^{'}, X_{n-1}^{'}, ..., X_{n-k+1}^{'}$ are unique, let start to solve (48) from a system of LP equations

$$\begin{cases} \max \sum_{i=n-k+1}^{n} X_i^{'} \\ \sum_{i=n-k+1}^{n} X_i^{'} = 1 \\ 0 \le X_i^{'} \le 1 \ \wedge \forall i \in \{1, 2, ..., n\} \end{cases} \quad (50)$$

Now we know at most k unique variables. All other sub-equation could be solved as follow: if solving sub-equation has at least one variable with earlier found value, LP equations

$$\begin{cases} \max \sum_{i=k+1}^{l} X_i^{'} + \sum_{i=l+1}^{2k} \left(1 - X_i^{'}\right) \\ 1 \le \sum_{i=k+1}^{l} X_i^{'} + \sum_{i=l+1}^{2k} \left(1 - X_i^{'}\right) \le k \\ 0 \le X_i^{'} \le 1 \ \wedge \forall i \in \{1, 2, ..., n\} \end{cases} \quad (51)$$

could be solved by reducing of (51) within inserting of earlier found variables values reassigned to 1; if all variables of solving sub-equations aren't unique and are negations of earlier found values in a different clauses, they values could be assigned to 0 and values of residual variables leads to 1; if all variables of solving sub-equations aren't unique and are negations of earlier found values in the same clause, one of variables must be assigned to 1 and other to 0; if solving sub-equation has k unique variables, LP equations

$$\begin{cases} \max \sum_{i=k+1}^{2k} X_i^{'} \\ \sum_{i=k+1}^{2k} X_i^{'} = 1 \\ 0 \le X_i^{'} \le 1 \ \wedge \forall i \in \{1, 2, ..., n\} \end{cases} \quad (52)$$

could be solved. Each sub-system of equation is solved in $O(k^{3.5})$ to reach global maximum 1. Finally, all clauses should be optimized m times or in $O\left(k^{3.5}m\right)$ ◯

V. Conclusion

Every NP mathematical problem is solvable in linear time if exist full, appropriate and correct knowledge basis for it and the time to get each item of knowledge basis is match less than calculation time on this items.

References

[1] Cook, Stephen (1971). "The complexity of theorem proving procedures". Proceedings of the Third Annual ACM Symposium on Theory of Computing. pp. 151158.

[2] Krom, Melven R. (1967), "The Decision Problem for a Class of First-Order Formulas in Which all Disjunctions are Binary", Zeitschrift fr Mathematische Logik und Grundlagen der Mathematik 13: 1520, doi:10.1002/malq.19670130104.

[3] Aspvall, Bengt; Plass, Michael F.; Tarjan, Robert E. (1979), "A linear-time algorithm for testing the truth of certain quantified boolean formulas", Information Processing Letters 8 (3): 121123, doi:10.1016/0020-0190(79)90002-4.

[4] Even, S.; Itai, A.; Shamir, A. (1976), "On the complexity of time table and multi-commodity flow problems", SIAM Journal on Computing 5 (4): 691703, doi:10.1137/0205048.

[5] Levin, Leonid (1973). "Universal search problems . Problems of Information Transmission 9 (3): 265266. (Russian), translated into English by Trakhtenbrot, B. A. (1984). "A survey of Russian approaches to perebor (brute-force searches) algorithms". Annals of the History of Computing 6 (4): 384400. doi:10.1109/MAHC.1984.10036.

[6] Diaby M., Linear programming formulation of the vertex colouring problem, Int. J. of Mathematics in Operational Research, 2010 Vol.2, No.3, p. 259 - 289

[7] Diaby M., Linear programming formulation of the set partitioning problem, Int. J. of Operational Research, 2010 Vol.8, No.4, p. 399 - 427

[8] Maknickas Algirdas, A., (2010). Finding of k in Fagin's R. Theorem 24, arXiv:1012.5804v1.

[9] Maknickas Algirdas, A., (2012). How to solve kSAT in polinomial time, arXiv:1203.6020v2.

[10] Maknickas Algirdas, A., (2012). How to solve kSAT in polinomial time, arXiv:1203.6020v1.

[11] Ilan Adler, Narendra Karmarkar, Mauricio G.C. Resende and Geraldo Veiga (1989). "An Implementation of Karmarkar's Algorithm for Linear Programming". Mathematical Programming, Vol 44, p. 297335.

ISBN: 978-988-19252-6-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)