A formula for systems of Boolean polynomial equations and applications to computational complexity

Tomoya Machide∗

Abstract

It is known a method for converting a system of Boolean polynomial equations to a single Boolean polynomial equation with less variables. In this paper, we show a formula for systems of Boolean polynomial equations which is based on the method. The formula has a structure of binary tree, and conforms to De Morgan’s duality. Using the formula, we prove a computational complexity result with a parameter for solving systems. The parameter is the bandwidth in matrix and graph theories: to be precise, the definition follows convention in matrix and the value depends on the order of variables. We also apply the result to the NP-complete problems, SAT and graph list-coloring, to show that these problems are fixed parameter tractable by bandwidth.

1 Introduction

The finite field $\mathbb{F}_2 = \{0, 1\}$ with two elements, which is also called the Galois field $\mathbb{GF}(2)$ in his honor, plays fundamental roles in mathematics and computer science. It is the smallest finite field and its algebraic rules are determined by a few equations involving the addition “+” and multiplication “·”. One of the outstanding facts of $\mathbb{F}_2$ is a structural relation to the two-element Boolean algebra $\mathbb{B} = \{\text{False, True}\}$ under the identifications of False = 0 and True = 1. That is, for any pair $(\alpha, \beta)$ of elements,

$$
\alpha \land \beta = \alpha \cdot \beta, \quad \alpha \lor \beta = (\alpha + 1) \cdot (\beta + 1) + 1, \quad \alpha \oplus \beta = \alpha + \beta,
$$

(1.1)

where $\land$, $\lor$, and $\oplus$ stand for the binary operations of conjunction, disjunction, and exclusive disjunction in $\mathbb{B}$, respectively. The unary operation $\neg$ of negation is expressed as $\neg \alpha = \alpha + 1$.

A Boolean polynomial, which is also called a Boolean expression in algebraic normal form [8], Reed-Muller expansion [30, 31], and Zhegalkin polynomial [18], naturally arises when we transform a Boolean expression to a polynomial using (1.1). The polynomial is a congruence class of the polynomial ring $\mathbb{F}_2[x_1, \ldots, x_n]$ in $n$ variables, and identified with a Boolean function from $\mathbb{F}_2^n$ to $\mathbb{F}_2$. (Details will be introduced in Section 2.) The Boolean polynomials and the ring consisting of them are important subjects in various areas: e.g., algebraic geometry [3, 12, 25], Boolean ideal and variety [28, 32], circuit theory [35], coding theory [20, 29], cryptography [8, 23], and Gröbner basis [7, 10, 33]. Although the contexts differ depending on the areas, solving a system of Boolean polynomial equations is a common problem.

Recently, Lokshtanov et al. [27] used several techniques developed from circuit complexity to construct algorithms for the problem, which beat brute force search without relying on any

*Global Research Center for Big Data Mathematics, National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan

e-mail: machide@nii.ac.jp

MSC-class: 03D15, 03G05, 08A40 (Primary); 06E30, 13P15, 68W30 (Secondary)

Key words: binary tree, Boolean polynomial, computational complexity, formula, system of polynomial equations
heuristic conjectures. (They actually studied for not only $\mathbb{F}_2$ but also any finite fields.) In this paper, we focus on the two basic techniques in [27]: (T1) transform a system of Boolean polynomial equations to a single Boolean polynomial equation; and (T2) transform a single Boolean polynomial equation to one with less variables. It may be worth noting that (T1) is a classical fact in algebraic geometry; for example, see Exercise 3 in [25, Chapter I, Section 1]. Combining (T1) and (T2), we can convert a system of Boolean polynomials to a single Boolean polynomial equation with less variables.

The aims of this paper are to show a formula which is based on the converting method, and to give applications to computational complexity. Considering the method from a sequential viewpoint, we construct a formula for systems of Boolean polynomial equations (Theorem 3.1). Using the formula, we prove a parameterized complexity result for solving systems (Theorem 3.2). Then we apply the complexity result to NP-complete problems: SAT and graph list-coloring (Corollaries 3.3 and 3.4).

Sketches of our results are following. The details will be stated in Section 3.

The formula of Theorem 3.1 possesses both operations of conjunction and disjunction recursively, with a structure of binary tree. By the recursiveness and structure, the formula conforms to De Morgan’s duality. The distributivity of the operations plays a fundamental role in the proof.

Theorem 3.2 follows from the fact that it is possible to reduce leaf nodes on the binary tree in some cases. Let $O$ be the big O notation, and let $O^*$ denote the notation which omits polynomial factors in $O$. Theorem 3.2 implies that the satisfiability of a system is decidable in time $O^*(2^B)$, where $B$ is the bandwidth in matrix and graph theories. The definition of bandwidth in this paper will adopt convention in matrix; that is, the value is not minimum but depends on the order of variables. The annihilator and identity laws for disjunction are crucial in the proof. We note that polynomial factors are not omitted in the actual statement.

Corollaries 3.3 and 3.4 are consequences of the fact that systems of Boolean polynomial equations can express the NP-complete problems, CNF-SAT, BMQ-SAT and graph list-coloring. The CNF-SAT problem, the Boolean satisfiability problem in conjunctive normal form, is the first NP-complete problem [11, 26]. CNF-SAT has many applications in the real world [4]. The BMQ-SAT problem, for which the algorithm beating brute force search was presented in [27], is the satisfiability problem of a Boolean multivariate quadratic system (or a system of Boolean polynomial equations of degree 2). BMQ-SAT is significant in cryptography to generate secure ciphers [1, 2]. We mean by SAT either one of both. The graph list-coloring problem is a generalization of the original coloring problem: in addition to the proper condition such that no two adjacent vertices receive the same color, a list of allowed colors is imposed for each vertex. Graph coloring is a central problem as SAT is, in theoretical, practical and historical aspects [19].

A problem of input size $n$ with a parameter $k$ is called fixed parameter tractable (or FPT for short) if it can be solved in time $f(k)n^{O(1)}$, where $f$ is a function only depending on $k$. Parameterized complexity theory is a two dimensional analog of the classical framework of P versus NP, and class FPT corresponds to class P (see [15] and references therein for details). Our complexity results show that SAT and list-coloring are FPT by bandwidth $B$, where, in the latter problem, the total number $l$ of allowed colors is considered to be constant and independent to the size $n$.

It appears that our complexity result of list-coloring problem is especially interesting, because the problem is known to be W[1]-hard for both parameters of treewidth and vertex cover [15, 16, 17], where W[1] is the class corresponding to NP. That is, the bandwidth is a different type parameter in list-coloring. The function $f$ is roughly $(2l)^B$. Our complexity result of CNF-SAT is already known (in a sense), because CNF-SAT is FPT by treewidth (of incidence graph) [34] and treewidth is more general than bandwidth. However our result
has an advantage: \( f \) is expressed as \( 2^B \) and it is concrete; in contrast, \( f \) in [34] is abstract (see Theorem 4 and Corollary 1 in the paper). Our complexity result of BMQ-SAT seems to be new.

The problem of finding bandwidth is NP-hard and its decisional version is NP-complete. However there are many heuristic algorithms including the Cuthill-McKee algorithm, and polynomial-time algorithms for spacial classes of graphs. (See [9, 13] and references therein for details.) Randomized approximate algorithms for general graphs, which run in polynomial or nearly linear time and have polylogarithmic factors of optimal, are also known [6, 14]. Thanks to those algorithms, our complexity results are practical if the bandwidth \( B \) is small.

The paper is organized as follows. In Section 2, we quickly review the Boolean polynomials and their basic properties. Rigid statements of our results are given in Section 3. We prove Theorem 3.1 in Section 4, and Theorem 3.2 in Section 5. Section 6 is devoted to the proofs of Corollaries 3.3 and 3.4.

2 Review of the Boolean polynomials

The finite field \( \mathbb{F}_2 \) is commutative, and its algebraic rules are determined by the equations involving the addition and multiplication:

\[
0 + 0 = 1 + 1 = 0 \cdot 0 = 0 \cdot 1 = 0, \quad 0 + 1 = 1 \cdot 1 = 1.
\]

The subtraction and division are unnecessary, because the subtraction is identical to the addition and no invertible elements except 1 exist.

The Boolean polynomial ring \( \mathbb{BP}_n = \mathbb{BP}[x_1, \ldots, x_n] \) is defined by the quotient ring

\[
\mathbb{BP}_n := \mathbb{F}_2[x_1, \ldots, x_n]/\mathcal{I}_n,
\]

where

\[
\mathcal{I}_n := \{p_1(x_1^2 + x_1) + \cdots + p_n(x_n^2 + x_n) \mid p_i \in \mathbb{F}_2[x_1, \ldots, x_n]\}.
\]

A Boolean polynomial \( p = p(x_1, \ldots, x_n) \) is a congruence class in \( \mathbb{BP}_n \). In the ring, the variables are idempotent (i.e., \( x_j^2 = x_j \)), and the number of monomials is \( 2^n \). Because the monomials are independent, \( p \) is uniquely expressed as

\[
p = \sum_{e_1, \ldots, e_n \in \{0, 1\}} a_{e_1, \ldots, e_n} x_{e_1}^1 \cdots x_{e_n}^n \quad (a_{e_1, \ldots, e_n} \in \mathbb{F}_2).
\]

We thus have \(|\mathbb{BP}_n| = 2^{2^n}\), where we mean by \(|A|\) the number of elements of a set \( A \).

Let \( BF_n \) be the ring of Boolean functions of \( n \) variables, or the ring of \( \mathbb{F}_2 \)-valued functions with the domain \( \mathbb{F}_2^n \). For a Boolean polynomial \( p = p(x_1, \ldots, x_n) \), we denote by \( \hat{p} \) the polynomial function of \( p \), which is defined by

\[
\hat{p} : \mathbb{F}_2^n \to \mathbb{F}_2 \quad \cup \quad \cup
\]

\[
(\alpha_1, \ldots, \alpha_n) \mapsto p(\alpha_1, \ldots, \alpha_n).
\]

This induces a well-defined homomorphism from \( \mathbb{BP}_n \) to \( BF_n \), since \( 0^2 + 0 = 1^2 + 1 = 0 \) and \( \hat{p} \) is the zero function if \( p \) is in \( \mathcal{I}_n \). The homomorphism is isomorphic,\(^1\) and \( \mathbb{BP}_n \) can be

\(^1\)For the injectivity, we may show that \( \hat{p} \neq 0 \) for a non-zero Boolean polynomial \( p \), which follows from the unique expression in (2.2). For the surjectivity, we may show that the numbers of elements in both rings are equal, or \(|BF_n| = 2^{2^n}\), which follows from \(|\mathbb{F}_2| = 2^n\) and \(|\mathbb{F}_2| = 2\).
identified with $\mathcal{BF}_n$:

$$\begin{align*}
\mathbb{BP}_n & \simeq \mathcal{BF}_n \\
p & \leftrightarrow \hat{p}.
\end{align*}$$  \hfill (2.3)

We see from (2.3) that $\mathbb{BP}_n$ has the same calculation rules as the codomain of $\mathcal{BF}_n$, or each of $\mathbb{F}_2$ and $\mathbb{B}$. Thus the identities in (1.1) hold on $\mathbb{BP}_n$. In addition, we have modular arithmetic properties

$$p + p = 0, \quad p^2 = p,$$

and annihilator and identity laws

$$0 \land p = 0, \quad 1 \lor p = 1, \quad 1 \land p = 0 \lor p = p.$$  \hfill (2.4)

Generalizing the second equation in (1.1) to $m$ elements, we also have

$$p_1 \lor \cdots \lor p_m = (p_1 + 1) \cdots (p_m + 1) + 1,$$

where $p_1, \ldots, p_m$ are Boolean polynomials. (For (2.6), see, e.g., [35, Section 3].)

### 3 Statement of results

We begin with preparing notations and terminologies.

For a pair $(S, T)$ of systems, we say that $S \approx T$ if either both systems are satisfiable or both are not. It is easily seen that $\approx$ is an equivalence relation. We call a system including two or more equations a multiple system; in contrast, we call a system including only one equation a single system. A system means either one of both. Let $i$ be a positive integer at most $n + 1$. For a positive integer $j$ less than $n + 1$, we denote by $\mathbb{BP}_{i,j}$ the subring $\mathbb{BP}_{\phi}$ of $\mathbb{BP}_n$, where $\mathbb{BP}_{i,j} = \mathbb{BP}_{\phi} = \mathbb{F}_2$ if $i > j$. It holds that

$$p|x_i=\alpha \in \mathbb{BP}_{i+1,j}$$  \hfill (3.1)

when $p \in \mathbb{BP}_{i,j}$ and $\alpha \in \mathbb{F}_2$.

We will describe the two basic techniques in [27], which enable us to convert a multiple system to a single system with less variables. Let $S = S(p_1, \ldots, p_m)$ be a system of Boolean polynomials $p_1, \ldots, p_m$. We define a Boolean polynomial by

$$P_S := p_1 \lor \cdots \lor p_m = (p_1 + 1) \cdots (p_m + 1) + 1.$$  \hfill (3.2)

The annihilator and identity laws for $\lor$ imply that $P_S = 0$ if and only if $p_1 = \cdots = p_m = 0$. Hence $S$ is equivalent to the single system consisting of $P_S$ under $\approx$, and we can apply $P_S$ to solve $S$. This is one of the techniques, which enable us to transform a multiple system to a single system. We put $P_1 = P_S$. For an integer $j$ from 2 to $n + 1$, we recursively define a Boolean polynomial in $\mathbb{BP}_{j,n}$ by

$$P_j := \prod_{\alpha \in \mathbb{F}_2} (P_{j-1}|_{x_{j-1}=\alpha}).$$  \hfill (3.3)

The number of variables in $P_j$ is at most $n + 1 - j$, and it decreases as $j$ increases. Let $S_j$ be a single system consisting of $P_j$ for each $j$. Obviously, $S_{j-1} \approx S_j$, and we can apply $P_j$ to solve $P_1 = P_S$. This is another technique to reduce variables. Combining these techniques, we can use $P_j \in \mathbb{BP}_{j,n}$ to solve $S$. 


More notations will be required to state Theorem 3.1. We will first introduce the definition of the CNF-SAT problem, next define notations on systems which involve CNF-SAT, and then mention the others.

Let $x$ be a variable. To distinguish $x$ and $\neg x = x + 1$, we call the former a positive literal and the latter a negative literal. A literal means either one of both. The CNF-SAT problem is the problem of deciding if there exists an assignment of variables which satisfies a conjunction of clauses, where a clause means a disjunction of literals. For instance, a CNF-SAT problem is solving

$$x_1 \land (\neg x_1 \lor x_2) = \text{True},$$

which is satisfiable because $(x_1, x_2) = (\text{True}, \text{True})$ is a solution. It is easily seen from the annihilator and identity laws for $\land$ that (3.4) is equivalent to the system of Boolean equations,

$$\begin{cases}
x_1 = 1, \\
\neg x_1 \lor x_2 = 1.
\end{cases}$$

We define a subspace in $\mathbb{B}^n$ by

$$\mathcal{CL}_n := \mathbb{F}_2 \cup \{l_1 \cdots l_k \mid k \geq 1, l_i \text{ are literals}\},$$

and its extension by

$$\mathcal{CL}_n^{(j)} := \mathcal{CL}_n + \cdots + \mathcal{CL}_n = \{c_1 + \cdots + c_j \mid c_1, \ldots, c_j \in \mathcal{CL}_n\}.$$  

We have $\mathcal{CL}_n^{(2^n)} = \mathbb{B}^n$ since $\mathcal{CL}_n$ includes all monomials in $\mathbb{B}^n$. Let $c = l_1 \cdots l_k$ be a non-constant Boolean polynomial in $\mathcal{CL}_n$. When $l_i = l_j$, $l_il_j = l_i = l_j$ by (2.4) and we can remove either $l_i$ or $l_j$ from $c$. When $l_i = l_i + 1$, $l_il_j = 0$ by (2.4) and $c$ is the zero polynomial, which contradicts the non-constant. Therefore, in this paper, we will assume that the literals $l_1, \ldots, l_k$ appearing in a polynomial of $\mathcal{CL}_n$ satisfy

$$l_i \notin \{l_j, l_j + 1\} \quad \text{for} \quad i \neq j.$$  

For each literal $l_i$, let $y_i$ and $\alpha_i$ denote a variable in $\{x_1, \ldots, x_n\}$ and a value in $\mathbb{F}_2$, respectively, such that $l_i = y_i + \alpha_i$. We have the following correspondence between equations of a polynomial and a clause:\footnote{The following equivalences hold by (1.1) and De Morgan’s duality: $(y_1 + \alpha_1) \cdots (y_k + \alpha_k) = 0 \iff (y_1 + \alpha_1) \land \cdots \land (y_k + \alpha_k) = \text{False} \iff (y_1 + \alpha_1 + 1) \lor \cdots \lor (y_k + \alpha_k + 1) = \text{True} \iff \bullet y_1 \lor \cdots \lor \bullet y_k = \text{True}.$}

$$(y_1 + \alpha_1) \cdots (y_k + \alpha_k) = 0 \iff \bullet y_1 \lor \cdots \lor \bullet y_k = \text{True},$$

where $\bullet$ stands for the negation ‘$\neg$’ if $\alpha_i = 0$ and the empty letter if $\alpha_i = 1$. For instans, $y_1 + 1 = 0$ and $y_1(y_2 + 1) = 0$ correspond to $y_1 = \text{True}$ and $\neg y_1 \lor y_2 = \text{True}$, respectively. Therefore we call an element of $\mathcal{CL}_n$ a clause polynomial, or simply a clause. Because of (3.9) and the correspondence between (3.4) and (3.5), the set of CNF-SAT problems in $n$ variables is equivalent to

$$\text{Sat}_n := \{S(c_1, \ldots, c_m) \mid c_1, \ldots, c_m \in \mathcal{CL}_n\}.$$  

As an extension of (3.10), we define

$$\text{Sat}_n^{(l)} := \{S(c_1, \ldots, c_m) \mid c_1, \ldots, c_m \in \mathcal{CL}_n^{(l)}\}.$$
where \( l \) is a positive integer. Since \( \mathbb{C}_m^{(2^n)} = \mathbb{B}_n^{(2^n)} \), \( \text{Sat}_n^{(2^n)} \) covers all systems of Boolean polynomial equations.

For a system \( S = S(p_1, \ldots, p_m) \), we call \( k = \max \deg p_i \) the degree of \( S \); the system is usually called a \( k \)-CNF-SAT problem if \( S \) belongs to \( \text{Sat}_n \). We order the variables according to their subscripts: i.e., \( x_i < x_j \) if \( i < j \). We denote by \( \pi_{\min}(p) \) the subscript of the minimum variable in a Boolean polynomial \( p \), where \( \pi_{\min}(p) = n + 1 \) if \( p \) is constant.\(^3\) Replacing \( P \) by \( F \), we apply (3.2) to a subset \( \mathcal{P} \) in \( \mathbb{B}_n \) such that
\[
F_P := \bigvee_{p \in \mathcal{P}} p = \prod_{p \in \mathcal{P}} (p + 1) + 1, \tag{3.12}
\]
where \( F_P = 0 \) if \( \mathcal{P} = \emptyset \). Obviously, \( P_S = F_{\{p_1, \ldots, p_m\}} \). We define a map from the power set of \( \mathbb{B}_n \) to itself by
\[
\mathcal{N}(\mathcal{P}) := \begin{cases} 
\{1\} & \text{if } 1 \in \mathcal{P}, \\
\mathcal{P} \setminus \{0\} & \text{otherwise.}
\end{cases} \tag{3.13}
\]
It holds that \( \mathcal{N}^2 = \mathcal{N} \), \( \mathcal{N}(\mathcal{P}) \subseteq \mathcal{P} \) and
\[
F_{\mathcal{N}(\mathcal{P})} = F_P. \tag{3.14}
\]
The operations used in \( \mathcal{N} \) are only search of 1 and delete of 0. Hence the computation time of \( \mathcal{N} \) is considered to be \( O(1) \) by means of hashing technique (see, e.g., [24, Section 6.4] for the idea of hash).

We are in a position to state Theorem 3.1.

**THEOREM 3.1.** Let \( S = S(p_1, \ldots, p_m) \) be a system in \( \text{Sat}_n^{(l)} \).
We put \( \mathcal{P}_0 = \mathcal{N}(\{p_1, \ldots, p_m\}) \), and divide \( \mathcal{P}_0 \) into
\[
\mathcal{P}_j = \{p \in \mathcal{P}_0 \mid \pi_{\min}(p) = j\} \quad (j = 1, \ldots, n + 1). \tag{3.15}
\]
For an integer \( j \) from 1 to \( n + 1 \), we recursively define a family
\[
\mathbf{P}_j = \{F_{\mathcal{P}_j^{\alpha_1 \cdots \alpha_{j-1}}} \mid 1 \leq i \leq j - 1, (\alpha_i, \cdots, \alpha_{j-1}) \in \mathbb{F}_2^{j-1}\} \tag{3.16}
\]
whose elements are subsets in \( \mathbb{B}_n \), as follows. First, set \( \mathbf{P}_1 = \emptyset \). Suppose \( \mathbf{P}_{j-1} \) is determined. From the elements of \( \mathbf{P}_{j-1} \), we construct those of \( \mathbf{P}_j \) such that
\[
\mathcal{P}_j^{\alpha_1 \cdots \alpha_{j-2} \alpha_{j-1}} = \mathcal{N}(\{p_{x_{j-1} = \alpha_{j-1}} \mid p \in \mathcal{P}_{j-1}^{\alpha_1 \cdots \alpha_{j-2}}\}), \tag{3.17}
\]
where \( \mathcal{P}_{j-1}^{\alpha_1 \cdots \alpha_{j-2}} = \mathcal{P}_{j-1} \) if \( i = j - 1 \).
Then, for the Boolean polynomials \( P_j \) in (3.3) with \( P_1 = P_S \), we have
\[
P_j = \left( \bigwedge_{\alpha_{j-1} \in \mathbb{F}_2} \left( \cdots \left( \bigwedge_{\alpha_2 \in \mathbb{F}_2} \left( \bigwedge_{\alpha_1 \in \mathbb{F}_2} F_{\mathcal{P}_j^{\alpha_1 \alpha_{j-1}}} \right) \vee F_{\mathcal{P}_j^{\alpha_2 \cdots \alpha_{j-1}}} \right) \cdots \right) \vee F_{\mathcal{P}_j^{\alpha_{j-1}}} \right) \vee F_{\mathcal{P}_j^{\cup \cdots \cup \mathcal{P}_n^{\alpha_{j+1}}}}. \tag{3.18}
\]
We also have the following properties of families \( \mathbf{P}_j \).

\( (A) \) \( |\mathbf{P}_j| = 2^j - 2 \).

\( (B) \) \( |\mathcal{P}_j^{\alpha_1 \cdots \alpha_{j-1}}| \leq |\mathcal{P}_{j-1}^{\alpha_1 \cdots \alpha_{j-2}}| \) for \( j \geq 2 \).

\(^3\)For instance, \( \pi_{\min}(x_1x_3) = 1 \), \( \pi_{\min}(x_2x_4 + x_3 + 1) = 2 \), and \( \pi_{\min}(1) = n + 1 \).
(C) The computing time of (3.17) for all elements of $P_j$ is bounded by
\[
O\left(\sum_{i=1}^{j-1} \left| \sum_{(\alpha_i, \ldots, \alpha_{j-2}) \in F_2^{j-1}} \right| \right).
\]

The formula (3.18) reads as
\[
P_1 = F_{P_1}^{P_2 \cup \ldots \cup P_{n+1}},
\]
\[
P_2 = (F_{P_2} \land F_{P_2}) \lor F_{P_2}^{P_2 \cup \ldots \cup P_{n+1}},
\]
\[
P_3 = (((F_{P_3} \lor F_{P_3}) \lor (F_{P_3} \land F_{P_3})) \lor F_{P_3}^{P_2 \cup \ldots \cup P_{n+1}}),
\]
and so on. Both operations of conjunction and disjunction appear recursively. By De Morgan’s duality, the equations of (3.19) are equivalent to
\[
\neg P_1 = \neg F_{P_1}^{P_2 \cup \ldots \cup P_{n+1}},
\]
\[
\neg P_2 = (\neg F_{P_2} \lor \neg F_{P_2}) \land \neg F_{P_2}^{P_2 \cup \ldots \cup P_{n+1}},
\]
\[
\neg P_3 = (((\neg F_{P_3} \lor \neg F_{P_3}) \land \neg F_{P_3}) \lor ((\neg F_{P_3} \lor \neg F_{P_3}) \land \neg F_{P_3})) \lor \neg F_{P_3}^{P_2 \cup \ldots \cup P_{n+1}}.
\]
The conjunction and disjunction are replaced each other, and the negation is appended to each factor. The dual of (3.18) is thus
\[
\neg P_j = \left( \lor_{\alpha_{j-1} \in F_2} \left( \cdots \left( \lor_{\alpha_{2} \in F_2} \left( \lor_{\alpha_{1} \in F_2} - F_{P_j}^{\alpha_1 \cdots \alpha_{j-1}} \right) \land - F_{P_j}^{\alpha_2 \cdots \alpha_{j-1}} \right) \cdots \right) \land - F_{P_j}^{\alpha_{j-1}} \right)
\]
\[
\land - F_{P_j}^{P_2 \cup \ldots \cup P_{n+1}},
\]
We can see from (3.19) that the formula (3.18) has expressions of binary tree as Figure 1, in which the cases of $P_2$ and $P_3$ are demonstrated. The same applies to (3.21) with dual replacements of symbols.

Figure 1: The left and right trees express $P_2$ and $P_3$ in (3.18), respectively, where $F_2^{\circ} = F_{P_2}^{P_2 \cup \ldots \cup P_{n+1}}$ and $F_3^{\circ} = F_{P_3}^{P_2 \cup \ldots \cup P_{n+1}}$. 

![Diagram](diagram.png)
Let \( p \) be a Boolean polynomial in \( \mathbb{B}^n \). We mean by \( \pi_{\max}(p) \) the subscript of the maximum variable in \( p \), where \( \pi_{\max}(p) = n+1 \) if \( p \) is constant. For a system \( S = (p_1, \ldots, p_m) \), we define the bandwidth by

\[
B_{S, \pi_S} := \max_i \{ \pi_{\max}(p_i) - \pi_{\min}(p_i) \},
\]

where \( \pi_S \) denotes the layout of the variable order in \( S \), i.e., \( \pi_S \) is the map from the set of variables to \( \{1, \ldots, n\} \) defined by \( \pi_S(x_j) = j \). The values of \( \pi_{\min}(p) \) and \( \pi_{\max}(p) \) are changed in general when variables \( x_1, \ldots, x_n \) are rearranged, and so \( B_{S, \pi_S} \) depends on \( \pi_S \). Instead of \( B_{S, \pi_S} \), we will use \( B \) for short.

We will state Theorem 3.2. The theorem comes from the fact that, for any \( P_j \) in (3.18), we can remove leaf factors as the tree depth is at most \( B \) while keeping the satisfiability.

**Theorem 3.2.** Let \( S = (p_1, \ldots, p_m) \) be a system in \( \text{Sat}^{(l)}_n \), and let \( B \) denote the bandwidth on a variable order. The satisfiability of \( S \) is decidable in time \( O(2^{Bl}(m+n)) \).

We require a bit of notations on graph to state the corollaries. Let \( G \) be a graph, and let \( V = V(G) \) and \( E = E(G) \) denote the vertex and edge sets, respectively. In this paper, we always assume that \( G \) is simple and undirected, and that the vertices are ordered as \( v_1, \ldots, v_n \), where \( v_i < v_j \) if \( i < j \). Let \( \pi_G \) denote the layout of the vertex order in \( G \). We define the bandwidth of \( G \) on \( \pi_G \) by

\[
B_{G, \pi_G} := \max \{i - j\}.
\]

Note that \( B_{G, \pi_G} = B_{S, \pi_S} \) if \( S \) is a linear system corresponding to the adjacency matrix of \( G \) under the identifications of \( x_i = v_i \). We consider \( [l] = \{0, 1, \ldots, l-1\} \) to be \( l \) distinct colors, and we call \( L \) a \([l]\)-list if \( L \subset [l] \).

**Corollary 3.3.** Let \( S \) be a SAT problem of \( n \) variables and \( m \) equations, and let \( B \) denote the bandwidth on a variable order.

(i) If \( S \) is BMQ, we can decide the satisfiability in time \( O(2^B(m+n)n^2) \).

(ii) If \( S \) is CNF, we can decide the satisfiability in time \( O(2^B(m+n)) \).

**Corollary 3.4.** Let \( G \) be a graph of \( n \) vertices and \( m \) edges, and let \( B \) denote the bandwidth on a vertex order. For a tuple \( (L_v)_{v \in V} \) of \([l]\)-lists, we can decide the list-colorability in time \( O((2l)^{B+2}(m+ln)) \).

Our complexity results give examples of FPT by bandwidth. Our algorithms for the results are deterministic as we will see in their proofs. In practice, the parameters \( l \) and \( m \) are equal to \( O(n^c) \) for some constants \( c \). Particularly, in the list-coloring problem, \( m \) is bounded by \( \binom{n}{2} \), and \( l \) is usually considered \( O(1) \).

An advantage of the results is that \( f \) is either \( 2^B \) or \( (2l)^B+2 \) and it is not abstract, where \( f \) stands for the function in the definition of FPT which is used to describe tractableness. Therefore we can compare complexity results related to ours by ignoring differences in polynomial factors.

Let \( B \ll g(n) \) mean that \( B \) is sufficiently smaller than the number \( g(n) \) such that log values of polynomial factors in \( n \) for base 2 have no influence. In [27], a randomized algorithm for BMQ-SAT is presented, whose computation time is bounded by \( O^*(2^{0.8765n}) \). Our algorithm for BMQ is faster when \( B \ll 0.8765n \). In [21], randomized algorithms for 3-CNF-SAT and 4-CNF-SAT are presented, whose computation times are bounded by \( O^*(1.30704^n) \) and \( O^*(1.46899^n) \), respectively. Our algorithm for CNF is faster when \( B \ll 0.38630n \) and
Lemma 4.2. Let $p$ be a Boolean polynomial in $\mathcal{CL}_n^{(t)}$, let $x_h$ be a variable, and let $\alpha$ be a value in $\mathbb{F}_2$. The computing time of $p|_{x_h=\alpha}$ is in $O(1)$.

We will first prove Theorem 3.1 dividing into two parts: one is devoted to the formula (3.18) and the other is devoted to the properties (A), (B) and (C). We will then prove Lemmas 4.1 and 4.2.

Proof of the formula (3.18). We will use induction on $j$. The case of $P_1$ is obvious, because

$$P_1 = P_{\mathcal{S}(p_1,\ldots,p_m)} \stackrel{(3.2)}{=} F_{\{p_1,\ldots,p_m\}} \stackrel{(3.14)}{=} F_{\mathcal{N}(\{p_1,\ldots,p_m\})} \stackrel{(3.15)}{=} F_{P_1^\alpha \cup \cdots \cup P_{n+1}^\alpha}.$$ 

Let $j > 1$, and suppose that (3.18) is true in the case of $P_{j-1}$. Let $\alpha_{j-1} \in \mathbb{F}_2$. We have

$$p|_{x_{j-1}=\alpha_{j-1}} = p$$

for a Boolean polynomial $p$ with $\pi_{\min}(p) > j - 1$, and so

$$\{p|_{x_{j-1}=\alpha_{j-1}} \mid p \in P_{j-1}^\alpha \cup \cdots \cup P_{n+1}^\alpha\} = \{p|_{x_{j-1}=\alpha_{j-1}} \mid p \in P_{j-1}^\alpha\} \cup (P_{j-1}^\alpha \cup \cdots \cup P_{n+1}^\alpha),$$

where $\cup$ means the disjoint union. Hence, by (3.12),

$$F_{P_{j-1}^\alpha \cup \cdots \cup P_{n+1}^\alpha} \big|_{x_{j-1}=\alpha_{j-1}} = \bigvee_{p \in P_{j-1}^\alpha \cup \cdots \cup P_{n+1}^\alpha} p \big|_{x_{j-1}=\alpha_{j-1}}$$

$$= \bigvee_{p' \in \{p|_{x_{j-1}=\alpha_{j-1}} \mid p \in P_{j-1}^\alpha \cup \cdots \cup P_{n+1}^\alpha\}} p'$$

$$= \left(\bigvee_{p' \in \{p|_{x_{j-1}=\alpha_{j-1}} \mid p \in P_{j-1}^\alpha\}} p'\right) \vee \left(\bigvee_{p' \in \{p|_{x_{j-1}=\alpha_{j-1}} \mid p \in P_{j-1}^\alpha\}} p'\right)$$

$$= F_{P_{j-1}^\alpha} \vee F_{P_{j-1}^\alpha} \cup F_{P_{j-1}^\alpha},$$

which, together with (3.14) and (3.17), yields

$$F_{P_{j-1}^\alpha \cup \cdots \cup P_{n+1}^\alpha} \big|_{x_{j-1}=\alpha_{j-1}} = F_{P_{j-1}^\alpha} \cup F_{P_{j-1}^\alpha} \cup F_{P_{j-1}^\alpha}.$$
For an element $\mathcal{P}_{j-1}^{\alpha_1\cdots\alpha_j-2}$ in $\mathcal{P}_{j-1}$, combining (3.12), (3.14) and (3.17) also yields

$$F_{\mathcal{P}_{j-1}^{\alpha_1\cdots\alpha_j-2}} \big|_{x_{j-1}=\alpha_{j-1}} = F_{\mathcal{P}_{j-1}^{\alpha_1\cdots\alpha_j-2\alpha_{j-1}}}.$$  \hspace{1cm} (4.3)

By the induction hypothesis, $P_{j-1}$ satisfies (3.18). Therefore, by (4.2) and (4.3),

$$P_{j-1}|_{x_{j-1}=\alpha_{j-1}} = \bigg( \bigwedge_{\alpha_j\in\mathbb{F}_2} \cdots \bigg( \bigwedge_{\alpha_2\in\mathbb{F}_2} \bigg( \bigwedge_{\alpha_1\in\mathbb{F}_2} F_{\mathcal{P}_{j-1}^{\alpha_1\cdots\alpha_j-2}} \vee F_{\mathcal{P}_{j-1}^{\alpha_2\cdots\alpha_j-2}} \bigg) \cdots \bigg) \vee F_{\mathcal{P}_{j}^{\alpha_j-2}} \bigg|_{x_{j-1}=\alpha_{j-1}}$$

$$= G_{j,\alpha_{j-1}} \vee F_{\mathcal{P}_{j}^{\alpha_j-2}} \vee F_{\mathcal{P}_{j}^{\alpha_j-2\alpha_{j-1}}},$$  \hspace{1cm} (4.4)

where

$$G_{j,\alpha_{j-1}} = \bigg( \bigwedge_{\alpha_{j-2}\in\mathbb{F}_2} \cdots \bigg( \bigwedge_{\alpha_{2}\in\mathbb{F}_2} \bigg( \bigwedge_{\alpha_{1}\in\mathbb{F}_2} F_{\mathcal{P}_{j-1}^{\alpha_1\cdots\alpha_{j-2}}} \vee F_{\mathcal{P}_{j-1}^{\alpha_2\cdots\alpha_{j-2}}} \bigg) \cdots \bigg) \vee F_{\mathcal{P}_{j}^{\alpha_{j-2}}} \bigg|_{x_{j-1}=\alpha_{j-1}}.$$  \hspace{1cm} (4.4)

Because of (3.3) and (4.4), $P_j$ is expressed as

$$P_j = (G_{j,0} \vee F_{\mathcal{P}_{j}^{\alpha_j}} \vee F_{\mathcal{P}_{j}^{\alpha_j}}) \wedge (G_{j,1} \vee F_{\mathcal{P}_{j}^{\alpha_j}} \vee F_{\mathcal{P}_{j}^{\alpha_j}}),$$

Using the distributivity of $\vee$ over $\wedge$, we obtain

$$P_j = ((G_{j,0} \vee F_{\mathcal{P}_{j}^{\alpha_j}}) \wedge (G_{j,1} \vee F_{\mathcal{P}_{j}^{\alpha_j}})) \vee F_{\mathcal{P}_{j}^{\alpha_j}} \vee F_{\mathcal{P}_{j}^{\alpha_j}}$$

$$= \bigg( \bigwedge_{\alpha_{j-1}\in\mathbb{F}_2} (G_{j,\alpha_{j-1}} \vee F_{\mathcal{P}_{j}^{\alpha_{j-1}}}) \bigg) \vee F_{\mathcal{P}_{j}^{\alpha_j}} \vee F_{\mathcal{P}_{j}^{\alpha_j}},$$  \hspace{1cm} (4.5)

which shows that (3.18) is true in the case of $P_j$. \hfill \Box

**Proof of the properties (A), (B) and (C).** The property (A) immediately follows from (3.16) and (4.1) with $h = 1$.

For an element $\mathcal{P}_{j}^{\alpha_1\cdots\alpha_j-1}$ in $\mathcal{P}_j$ for $j \geq 2$, we have

$$|\mathcal{P}_{j}^{\alpha_1\cdots\alpha_j-1}| = |\mathcal{N}(|p|_{x_{j-1}=\alpha_{j-1}} \mid p \in \mathcal{P}_{j-1}^{\alpha_1\cdots\alpha_j-2})|$$

$$\leq |p|_{x_{j-1}=\alpha_{j-1}} \mid p \in \mathcal{P}_{j-1}^{\alpha_1\cdots\alpha_j-2}|$$

$$\leq |\mathcal{P}_{j-1}^{\alpha_1\cdots\alpha_j-2}|,$$

which implies (B).

For an element $\mathcal{P}_{j}^{\alpha_1\cdots\alpha_j-1}$, we can calculate (3.17) in time $O(|\mathcal{P}_{j-1}^{\alpha_1\cdots\alpha_j-2}|)$ by Lemma 4.2, where remember that $\mathcal{N}$ is a constant cost map by hashing technique. Thus, by (3.16), the computing time of (3.17) for all elements of $\mathcal{P}_j$ is bounded by

$$O \left( \sum_{i=1}^{j-1} \sum_{(\alpha_i,\ldots,\alpha_{j-1}) \in \mathbb{F}_2^{i-1}} |\mathcal{P}_{j-1}^{\alpha_1\cdots\alpha_{j-2}}| \right) = O \left( 2^i \sum_{i=1}^{j-1} \sum_{(\alpha_i,\ldots,\alpha_{j-2}) \in \mathbb{F}_2^{i-1}} |\mathcal{P}_{j-1}^{\alpha_1\cdots\alpha_{j-2}}| \right),$$
which proves (C).

We will prove Lemma 4.1.

Proof of Lemma 4.1. We have
\[ \sum_{i=h}^{j-1} |F_{2}^{j-i}| = \sum_{i=h}^{j-1} 2^{j-i} = 2(1 + 2 + \cdots + 2^{j-h}) = 2(2^{j-h} - 1), \]
which implies (4.1).

Let \( L_n \) be the set consisting of the literals and the constant 1. We define a map \( \psi \) from \( \mathbb{C}L_n \) to the power set of \( L_n \) by
\[
\psi(c) := \begin{cases} 
\phi & \text{if } c = 0, \\
\{1\} & \text{if } c = 1, \\
\{l_1, \ldots, l_k\} & \text{if } c = l_1 \cdots l_k,
\end{cases} \tag{4.6}
\]
where \( c \in \mathbb{C}L_n \) and \( l_i \) are literals. This map is well-defined and injective by (3.8).

We will prove Lemma 4.2.

Proof of Lemma 4.2. We may show that the computing time of \( c|_{x_h=\alpha} \) is in \( O(1) \) for a non-constant clause \( c \). Let \( k \) denote the degree of \( c \). There exist \( k \) variables \( x_{h_i} \) and \( k \) values \( \beta_i \) in \( \mathbb{F}_2 \) such that
\[
c = \prod_{i=1}^{k} (x_{h_i} + \beta_i), \tag{4.7}
\]
where literals \( x_{h_i} + \beta_i \) satisfy (3.8). Put \( \mathcal{X} = \{x_{h_1}, \ldots, x_{h_k}\} \). If \( x_h \notin \mathcal{X} \),
\[
c|_{x_h=\alpha} = c. \tag{4.8}
\]
If \( x_h \in \mathcal{X} \),
\[
c|_{x_h=\alpha} = (x_h + \beta_h)|_{x_h=\alpha} \prod_{i=1}^{k} (x_{h_i} + \beta_i)_{(i \neq h)}
= (\alpha + \beta_h) \prod_{i=1}^{k} (x_{h_i} + \beta_i)_{(i \neq h)}
= \begin{cases} 
0 & \text{if } \alpha = \beta_h, \\
\prod_{i=1}^{k} (x_{h_i} + \beta_i)_{(i \neq h)} & \text{if } \alpha \neq \beta_h.
\end{cases} \tag{4.9}
\]
We can see from (4.7), (4.8) and (4.9) that evaluating \( \psi(c|_{x_h=\alpha}) \) from \( \psi(c) \) is implemented by the following process:

1. Set \( \omega = \psi(c) \).
2. Return \( \omega \) if \( \omega = \phi \) or \( 1 \in \omega \).

\(^4\)Note that \( \omega = \phi \) if and only if \( c = 0 \). Also note that \( 1 \in \omega \) if and only if \( c = 1 \); the reason is because \( \psi(c) = \{1\} \) if \( c = 1 \) and \( 1 \notin \psi(c) \) otherwise. Hence the return condition of step 2 is equivalent to \( c \in \mathbb{F}_2 \).
3. Search \( x_h \) and \( x_h + 1 \) from \( \omega \). Return \( \omega \) if not exist.

4. Set \( l_h = x_h + \beta_h = (\text{the literal searched in step 3}) \).

5. Delete \( l_h \) from \( \omega \).

6. Return \( \phi \) if \( \alpha = \beta_h \), otherwise return \( \omega \).

The operations used in the process which are not elemental are search and delete. By hashing technique, costs of these operations are constants. Thus the computation time of the process is bounded by \( O(1) \). Since \( \psi \) is an embedding of \( \mathbb{C}^n \), the process implies that the time of computing \( c|_{x_h=\alpha} \) is bounded by \( O(1) \), and we compete the proof. \( \square \)

5 Proof of Theorem 3.2

We will require Proposition 5.1 to prove Theorem 3.2. The proposition is a refinement of Theorem 3.1, which has an additional condition of the bandwidth.

**Proposition 5.1.** Let \( S = S(p_1, \ldots, p_m) \) be a system in \( \text{Sat}^{(l)} \) with the bandwidth \( B \) on a variable order. Let \( \mathcal{P}^\alpha_j \) denote the subsets defined in (3.15), and let \( \mathcal{P}_{B+1} \) denote the \((B+1)-\text{th}\) family determined in (3.16) and (3.17).

We put \( I_j = j - B \) for \( j \geq B + 1 \). For an integer \( j \) from \( B + 1 \) to \( n + 1 \), we recursively define a family

\[
\mathcal{Q}_j = \{ \mathcal{Q}_j^{\alpha_i \cdots \alpha_j} | I_j \leq i \leq j - 1, (\alpha_i, \ldots, \alpha_{j-1}) \in \mathbb{F}_2^{j-1} \} \tag{5.1}
\]

whose elements are subsets in \( \mathbb{F}_{2,n} \), as follows. Firstly, set \( \mathcal{Q}_{B+1} = \mathcal{P}_{B+1} \). Suppose that \( \mathcal{Q}_{j-1} \) is determined. From the elements of \( \mathcal{Q}_{j-1} \), we construct temporal elements such that

\[
\tilde{\mathcal{Q}}_j^{\alpha_i \cdots \alpha_j - 2\alpha_{j-1}} = \mathcal{N}(\{ p|_{x_{j-1}=\alpha_{j-1}} | p \in \mathcal{Q}_{j-1}^{\alpha_i \cdots \alpha_{j-2}} \}), \tag{5.2}
\]

where \( \mathcal{Q}_{j-1}^{\alpha_i \cdots \alpha_{j-2}} = \mathcal{P}_{j-1}^\alpha \) if \( i = j - 1 \). Then we define the elements of \( \mathcal{Q}_j \) by

\[
\mathcal{Q}_j^{\alpha_i \cdots \alpha_j} := \begin{cases} 
\{1\} & \text{if } i = I_j \text{ and } \mathcal{F} \mathcal{Q}_j^{\alpha_j \cdots \alpha_{j-1}} | \mathcal{Q}_j^{\alpha_j \cdots \alpha_{j-1}} > 0, \\
\tilde{\mathcal{Q}}_j^{\alpha_i \cdots \alpha_j} & \text{if } i > I_j \text{ or } \mathcal{F} \mathcal{Q}_j^{\alpha_j \cdots \alpha_{j-1}} = 0. \tag{5.3}
\end{cases}
\]

Let \( j \geq B + 1 \). Then, for the Boolean polynomials \( P_j \) in (3.3) with \( P_1 = P_S \), we have

\[
P_j = \bigg( \land_{\alpha_{j-1} \in \mathbb{F}_2} \cdots \land_{\alpha_{j+1} \in \mathbb{F}_2} \left( \land_{\alpha_{j} \in \mathbb{F}_2} \mathcal{F} \mathcal{Q}_j^{\alpha_j \cdots \alpha_{j-1}} \bigg) \bigg) \lor \mathcal{F} \mathcal{Q}_j^{\alpha_j \cdots \alpha_{j-1}} \bigg) \cdots \lor \mathcal{F} \mathcal{Q}_j^{\alpha_{j-1}} \bigg) \land_{\alpha_{j} \in \mathbb{F}_2} \cdots \land_{\alpha_{j-1} \in \mathbb{F}_2} \bigg).
\tag{5.4}
\]

We also have the following properties of families \( \mathcal{Q}_j \).

\[\text{(A)} \quad |\mathcal{Q}_j| = 2^{B+1} - 2.\]

\(^5\)Note that \([x, x+1] \cap \psi(c) \mid \leq 1 \) by (3.8), and \( l_h \) in step 4 is uniquely determined.

\(^6\)We can replace \( != 0 \) with \( != 1 \) in (5.3), because \( \mathcal{F} \mathcal{Q}_j^{\alpha_j \cdots \alpha_{j-1}} \subset \{1\} \) by (5.17) and the number of elements in \( \mathcal{F} \mathcal{Q}_j^{\alpha_j \cdots \alpha_{j-1}} \) is one or zero.
(B) If \( i > I_j \), \( |Q_j^{\alpha_i-\alpha_j-2}| \leq |Q_j^{\alpha_i-\alpha_j-1}| \). If \( i = I_j \), \( |Q_j^{\alpha_i-\alpha_j-1}| \leq 2 \).

(C) The computing time of (5.2) and (5.3) for all elements of \( Q_j \) is bounded by

\[
O \left( 2B + l \sum_{i=I_j-1}^{j-1} \sum_{(\alpha_i, \ldots, \alpha_j-2) \in \mathbb{F}_2^{j-1-i}} |Q_j^{\alpha_i-\alpha_j-2}| \right).
\]

We will prove Theorem 3.2. Then we will prove Proposition 5.1.

**Proof of Theorem 3.2.** Let \( P_1, \ldots, P_{B+1} \) denote the first \((B+1)\) families determined in (3.16) and (3.17). Let \( j \in \{2, \ldots, B+1\} \). Using (B) in Theorem 3.1 repeatedly, we obtain

\[
|P_{j-1}^{\alpha_1-\alpha_j-2}| \leq |P_{j-2}^{\alpha_1-\alpha_j-3}| \leq \cdots \leq |P_{i+1}^{\alpha_1}| \leq |P_i^O|
\]

for any \( P_{j-1}^{\alpha_1-\alpha_j-2} \in P_{j-1} \). Since \( |\mathbb{F}_2^h| = 2^h \) for \( h \geq 0 \), we have

\[
\sum_{i=1}^{j-1} \sum_{(\alpha_i, \ldots, \alpha_j-2) \in \mathbb{F}_2^{j-1-i}} |P_{j-1}^{\alpha_1-\alpha_j-2}| \leq \sum_{i=1}^{j-1} 2^{j-1-i} |P_i^O|,
\]

which, together with (C) in Theorem 3.1, shows that the total time to calculate the families \( P_2, \ldots, P_{B+1} \) is bounded by

\[
O \left( l \sum_{j=2}^{B+1} \sum_{i=1}^{j-1} 2^{j-1-i} |P_i^O| \right).
\]

(5.6)

Note that \( P_1 = \emptyset \) by the initial condition and no calculation is required for \( P_1 \).

Let \( Q_{B+2}, \ldots, Q_{n+1} \) denote the families determined in (5.1), (5.2) and (5.3) with the initial condition \( Q_{B+1} = P_{B+1} \). Let \( j \in \{B+2, \ldots, n+1\} \). Similarly to (5.5), it follows from \( I_{j-1} = I_j - 1 = j - 1 - B \) and (B) in Proposition 5.1 that

\[
\sum_{i=I_{j-1} - 1}^{j-1} \sum_{(\alpha_i, \ldots, \alpha_j-2) \in \mathbb{F}_2^{j-1-i}} |Q_j^{\alpha_i-\alpha_j-2}| = \sum_{(\alpha_{I_{j-1} - 1}, \ldots, \alpha_j-2) \in \mathbb{F}_2^{j-B}} |Q_j^{\alpha_{I_{j-1} - 1}-\alpha_j-2}| + \sum_{i=I_j}^{j-1} \sum_{(\alpha_i, \ldots, \alpha_j-2) \in \mathbb{F}_2^{j-1-i}} |Q_j^{\alpha_i-\alpha_j-2}|
\]

\[
\leq 2 \cdot 2^B + \sum_{i=j-B}^{j-1} 2^{j-1-i} |P_i^O|,
\]

which, together with (C) in Proposition 5.1, shows that the total time to calculate the families \( Q_{B+2}, \ldots, Q_{n+1} \) is bounded by

\[
O \left( l \sum_{j=I_{B+2}}^{n+1} \sum_{i=j-B}^{j-1} 2^{j-1-i} |P_i^O| \right).
\]

(5.7)

We define subsets in \( \mathbb{Z}^2 \) as follows:

\[
L_1 := \{(i, j) \in \mathbb{Z}^2 \mid 2 \leq j \leq B + 1, \quad 1 \leq i \leq j - 1\},
\]

\[
L_2 := \{(i, j) \in \mathbb{Z}^2 \mid B + 2 \leq j \leq n + 1, \quad j - B \leq i \leq j - 1\},
\]

\[
L_3 := \{(i, j) \in \mathbb{Z}^2 \mid n + 2 \leq j \leq n + B + 1, \quad j - B \leq i \leq n + 1\}.
\]
Obviously, \( L_a \cap L_b = \emptyset \) for \( a \neq b \). Switching the roles of \( i \)-axis and \( j \)-axis, we obtain
\[
L_1 = \{(i, j) \in \mathbb{Z}^2 | \quad 1 \leq i \leq B, \quad i + 1 \leq j \leq B + 1\},
\]
\[
L_2 = \{(i, j) \in \mathbb{Z}^2 | \quad 2 \leq i \leq B, \quad B + 2 \leq j \leq i + B\}
\cup \{(i, j) \in \mathbb{Z}^2 | \quad B + 1 \leq i \leq n + 1 - B, \quad 1 \leq j \leq i + B\}
\cup \{(i, j) \in \mathbb{Z}^2 | \quad n + 2 - B \leq i \leq B, \quad i + 1 \leq j \leq n + 1\},
\]
\[
L_3 = \{(i, j) \in \mathbb{Z}^2 | \quad n + 2 - B \leq i \leq n + 1, \quad n + 2 \leq j \leq i + B\}.
\]
Hence,
\[
L_1 \cup L_2 \cup L_3 = \{(i, j) \in \mathbb{Z}^2 | \quad 1 \leq i \leq n + 1, \quad i + 1 \leq j \leq i + B\},
\]
and
\[
\sum_{j=2}^{B+1} \sum_{i=1}^{j-1} 2^{j-1-i} |P_i^o| + \sum_{j=B+2}^{n+1} \sum_{i=1}^{j-1} 2^{j-1-i} |P_i^o| = \sum_{(i,j) \in L_1 \cup L_2} 2^{j-1-i} |P_i^o|
\leq \sum_{(i,j) \in L_1 \cup L_2 \cup L_3} 2^{j-1-i} |P_i^o|
= \sum_{i=1}^{n+1} |P_i^o| \sum_{j=i+1}^{i+B} 2^{j-1-i}.
\]
Since
\[
\sum_{i=1}^{n+1} |P_i^o| = |\mathcal{N}(\{p_1, \ldots, p_m\})| \leq m, \quad \sum_{j=i+1}^{i+B} 2^{j-1-i} = \sum_{j=1}^{B} 2^{j-1} < 2^B,
\]
we have
\[
\sum_{j=2}^{B+1} \sum_{i=1}^{j-1} 2^{j-1-i} |P_i^o| + \sum_{j=B+2}^{n+1} \sum_{i=1}^{j-1} 2^{j-1-i} |P_i^o| < m2^B. \tag{5.8}
\]
Therefore, we see from (5.6), (5.7), and (5.8) that the whole time to calculate all families \( P_i \) and \( Q_j \) is bounded by
\[
O(l(m + n)2^B). \tag{5.9}
\]

It is required to compute (3.15) for starting the above procedure to calculate all the families; this costs in \( O(lm) \) since (3.15) is done by dividing \( m \) polynomials consisting of \( l \) clauses into \( (n + 1) \) sets. The solvability of \( S \) is equivalent to \( P_{n+1} = 0 \), and it is also required to confirm whether \( P_{n+1} \) is zero or not for closing the procedure; this costs in \( O(2^B) \), since, by \((A)_Q\), the number of factors in the right-hand side of (5.4) for \( j = n + 1 \) is less than \( 2 \cdot 2^B \).\(^7\) Both computation times for starting and closing are bounded by (5.9), and we prove Theorem 3.2. \(\square\)

We require the following lemmas to show Proposition 5.1.

**Lemma 5.2.** Let \( p \) be a Boolean polynomial, and let \( B \) be a positive integer such that \( \pi_{\max}(p) - \pi_{\min}(p) \leq B \). Put \( i = \pi_{\min}(p) \), and let \( j \) be an integer with \( i < j \leq n + 1 \). Then we have
\[
p|_{x_i=\alpha_i | x_{i+1}=\alpha_{i+1} \cdots | x_{j-1}=\alpha_{j-1}} \in \mathbb{F}_2^{j,i+B} \tag{5.10}
\]
for values \( \alpha_i, \alpha_{i+1}, \ldots, \alpha_{j-1} \) in \( \mathbb{F}_2 \).\(^7\)

\(^7\)Note that the factors belong to \( \mathbb{F}_2 \) because \( P_{n+1} \in \mathbb{F}_2 \), and that binary operations on \( \mathbb{F}_2 \) cost in \( O(1) \).
LEMMA 5.3. Let $\mathcal{P}$ be a subset in $\mathbb{B}\mathbb{F}_n$. Let $i, j$ be integers with $1 \leq i < j \leq n + 1$, and let $\alpha_i, \ldots, \alpha_{j-1}$ be values in $\mathbb{F}_2$. For an integer $h$ from $i + 1$ to $j$, we recursively define a subset $\mathcal{P}^{\alpha_i \cdots \alpha_h}$ as

$$\mathcal{P}^{\alpha_i \cdots \alpha_h} := \mathcal{N} (\{ p | x_{h-1} = \alpha_{h-1} \mid p \in \mathcal{P}^{\alpha_i \cdots \alpha_{h-2}} \})$$

where $\mathcal{P}^{\alpha_i \cdots \alpha_h,2} = \mathcal{P}$ if $h = i + 1$. Then we have

$$\mathcal{P}^{\alpha_i \cdots \alpha_{j-1}} \subseteq \mathcal{N} (\{ p | x_i = \alpha_i, x_{i+1} = \alpha_{i+1}, \ldots, x_{j-1} = \alpha_{j-1} \mid p \in \mathcal{P} \}).$$

(5.11)

The proofs of the lemmas will be given after that of the proposition.

Proof of Proposition 5.1. It immediately follows from (4.1) and (5.1) that

$$|Q_j| = 2^{j+1-I_j} - 2 = 2^{B+1} - 2,$$

which proves (A)$_Q$.

We will show (B)$_Q$, (C)$_Q$ and (5.4) by induction on $j$ from $B + 1$ to $n + 1$. Suppose that $j = B + 1$. We do not need to prove (C)$_Q$ because $Q_{B+1}$ is set to $\mathcal{P}_{B+1}$ by the initial condition and calculation is unnecessary. We can easily verify (5.4) because it is equal to (3.18) in Theorem 3.1. We will prove (B)$_Q$. We may assume $i = 1$, since $I_{B+1} = 1$ and (B)$_Q$ for $i > I_{B+1}$ holds by (B) in Theorem 3.1. From (5.10) with $(i, j) = (1, B + 1)$ and (5.11) with $\mathcal{P} = \mathcal{P}_1^2$, we see that

$$\mathcal{P}_{B+1}^{\alpha_1 \cdots \alpha_B} \subseteq \mathcal{N}(\mathbb{B}\mathbb{F}_{B+1}^{B+1}) = \mathcal{N}(\mathbb{B}\mathbb{F}[x_{B+1}]).$$

(5.12)

We define

$$X_x = \{ \phi, \{1\}, \{x\}, \{-x\}, \{x, -x\} \}$$

for a variable $x$. By (3.13), any subset of $\mathcal{N}(\mathbb{B}\mathbb{F}[x_{B+1}])$ belongs to $X_{x_{B+1}}$, which, together with (5.12) and $Q_{B+1}^{\alpha_1 \cdots \alpha_B} = \mathcal{P}_{B+1}^{\alpha_1 \cdots \alpha_B}$, implies

$$Q_{B+1}^{\alpha_1 \cdots \alpha_B} \in X_{x_{B+1}}.$$  

(5.13)

Thus $|Q_{B+1}^{\alpha_1 \cdots \alpha_B}| \leq 2$, and we prove (B)$_Q$ for $i = 1$.

Suppose that $j > B + 1$, and (B)$_Q$, (C)$_Q$ and (5.4) are true in the case of $j - 1$.

Firstly we will prove (B)$_Q$ for the case of $j$. Let $\tilde{Q}_j$ denote the family consisting of the temporal subsets defined in (5.2):

$$\tilde{Q}_j = \{ Q_{j-i}^{\alpha_1 \cdots \alpha_{j-i-1}} \mid I_{j-i} \leq i \leq j - 1, (\alpha_i, \ldots, \alpha_{j-i}) \in \mathbb{F}_2^{j-i} \}.$$  

(5.14)

The following properties hold.

(B)$_{\tilde{Q}}$ $|Q_{j-1}^{\alpha_1 \cdots \alpha_{j-1}}| \leq |Q_{j-2}^{\alpha_1 \cdots \alpha_{j-2}}|$. 

(C)$_{\tilde{Q}}$ The computing time of (5.2) for all elements of $\tilde{Q}_j$ is bounded by

$$O \left( l \sum_{i=I_{j-1}}^{j-1} \left( \sum_{(\alpha_1, \ldots, \alpha_{j-2}) \in \mathbb{F}_2^{j-i-1}} |Q_{j-1}^{\alpha_1 \cdots \alpha_{j-2}}| \right) \right).$$
These properties can be shown as in the cases of (B) and (C) in Theorem 3.1. We omit their proofs for space limitation.\footnote{We give brief explanations. Both definitions of $P^\alpha_j$ and $Q^\alpha_j$ in (5.17) and (5.18) are almost same as we see from (3.17) and (5.2); only the conditions $p \in P^\alpha_{i-1}$ and $p \in Q^\alpha_{i-1}$ differ. We also see from (3.16) and (5.14) that both definitions of $P_j$ and $Q_j$ are almost same; the conditions $1 \leq i \leq j - 1$ and $I_j - 1 \leq i \leq j - 1$ differ. We can prove (B)$Q_j$ and (C)$Q_j$ in the same ways as (B) and (C), respectively, by commuting the above different places.} We will prove (B)$Q_j$. Let $Q^\alpha_{i-\alpha_j-1} \in Q_j$. If $i > I_j$, we see from (5.3) that
\[
Q^\alpha_{i-\alpha_j-1} = \tilde{Q}^\alpha_{i-\alpha_j-1},
\]
which, together with (B)$\tilde{Q}_j$, yields
\[
\left| Q^\alpha_{i-\alpha_j-1} \right| \leq \left| Q^\alpha_{i-\alpha_j-2} \right| .
\]
Assume $i = I_j$. We also see from (5.3) that
\[
Q^\alpha_{i-\alpha_j-1} = \{ 1 \} \text{ or } \tilde{Q}^\alpha_{i-\alpha_j-1}.
\]
By (5.10) with $i = I_j$ and (5.11) with $P = P^\alpha_{I_j}$, we obtain $\tilde{Q}^\alpha_{i-\alpha_j-1} \subset N(\mathbb{B}P[x_j])$, and $\tilde{Q}^\alpha_{i-\alpha_j-1} \in x_j$. Thus $Q^\alpha_{i-\alpha_j-1} \in x_j$, and
\[
\left| Q^\alpha_{i-\alpha_j-1} \right| \leq 2.
\]
It follows from (5.15) and (5.16) that (B)$Q$ holds in the case of $j$.

Next we will prove (C)$Q$ for the case of $j$. When $i > I_j$, setting $Q^\alpha_{i-\alpha_j} = \tilde{Q}^\alpha_{i-\alpha_j}$ is only required in (5.3). Thus, the computing time of (5.2) and (5.3) for the elements $Q^\alpha_{i-\alpha_j-1}$ with $i > I_j$ is bounded by the time stated in (C)$\tilde{Q}$. Therefore, to prove (C)$Q$, we may show that the computing time of (5.3) for all elements $Q^\alpha_{i-\alpha_j-1}$ with $i = I_j$ is bounded by $O(2^B)$. Let $(\alpha_{I_j}, \ldots, \alpha_{j-1}) \in \mathbb{F}_2^{-I_j}$ and $\alpha = \alpha_{I_j-1} \in \mathbb{F}_2$. By (5.10) with $i = I_j$ and (5.11) with $P = P^\alpha_{I_j}$, we obtain $\tilde{Q}^\alpha_{I_j-1} \subset N(\mathbb{B}P[x_j])$, and
\[
\tilde{Q}^\alpha_{I_j-1} \subset N(\mathbb{B}P[x_j]) \quad \text{and} \quad \tilde{Q}^\alpha_{I_j-1} \subset N(\mathbb{B}P[x_j]) \subset \{ 1 \},
\]
which implies that $\tilde{Q}^\alpha_{I_j-1}$ is either $\phi$ or $\{ 1 \}$, or equivalently, $|\tilde{Q}^\alpha_{I_j-1}|$ is either zero or one. Hence, the time of checking whether $\prod_{\alpha \in \mathbb{F}_2} \left| \tilde{Q}^\alpha_{I_j-1} \right|$ is zero or not is in $O(1)$. By the definition of (5.3), we can calculate the single element $Q^\alpha_{I_j-\alpha_j}$ in time $O(1)$. Since $|\mathbb{F}_2^{-I_j}| = |\mathbb{F}_2^\alpha| = 2^B$, the computing time of (5.3) for all elements $Q^\alpha_{i-\alpha_j-1}$ with $i = I_j$ is bounded by $O(2^B)$. This concludes that (C)$Q$ in the case of $j$ is true.

Finally we will prove (5.4) for the case of $j$. By the induction hypothesis, $P_{j-1}$ satisfies (5.4). Using the distributivity of $\lor$ over $\land$, we can obtain the following equation as in (4.4) and (4.5):
\[
P_j = \bigvee_{\alpha_{j-1} \in \mathbb{F}_2} \left( \bigwedge_{\alpha_{I_j} \in \mathbb{F}_2} \left( \bigwedge_{\alpha_{j-1} \in \mathbb{F}_2} F^\alpha_{j-1-\alpha_j-1} - F^\alpha_{j-1} \right) \cup F^\alpha_{j-1} \right) \bigvee F^\alpha_{j-1}.
\]
Because $\tilde{Q}_j^{\alpha_1 \cdots \alpha_{j-1}} \in \{\phi, \{1\}\}$, we see from (3.12) that

$$F_{\tilde{Q}_j^{\alpha_1 \cdots \alpha_{j-1}}} = |\tilde{Q}_j^{\alpha_1 \cdots \alpha_{j-1}}|,$$

where the values 0 and 1 in $F$ are identified with those in $\mathbb{Z}$. Hence

$$\bigwedge_{\alpha_{j-1} \in \mathbb{F}_2} F_{\tilde{Q}_j^{\alpha_1 \cdots \alpha_{j-1}}} = \prod_{\alpha_{j-1} \in \mathbb{F}_2} |\tilde{Q}_j^{\alpha_1 \cdots \alpha_{j-1}}|,$$

and

$$\left(\bigwedge_{\alpha_{j-1} \in \mathbb{F}_2} F_{\tilde{Q}_j^{\alpha_1 \cdots \alpha_{j-1}}} \right) \lor F_{\tilde{Q}_j^{\alpha_1 \cdots \alpha_{j-1}}} = \begin{cases} 1 \lor F_{\tilde{Q}_j^{\alpha_1 \cdots \alpha_{j-1}}} & \text{if } \prod_{\alpha_{j-1} \in \mathbb{F}_2} |\tilde{Q}_j^{\alpha_1 \cdots \alpha_{j-1}}| = 1, \\
0 \lor F_{\tilde{Q}_j^{\alpha_1 \cdots \alpha_{j-1}}} & \text{if } \prod_{\alpha_{j-1} \in \mathbb{F}_2} |\tilde{Q}_j^{\alpha_1 \cdots \alpha_{j-1}}| = 0. \end{cases}$$

By the annihilator and identity laws for $\lor$,

$$\left(\bigwedge_{\alpha_{j-1} \in \mathbb{F}_2} F_{\tilde{Q}_j^{\alpha_1 \cdots \alpha_{j-1}}} \right) \lor F_{\tilde{Q}_j^{\alpha_1 \cdots \alpha_{j-1}}} = \begin{cases} 1 & \text{if } \prod_{\alpha_{j-1} \in \mathbb{F}_2} |\tilde{Q}_j^{\alpha_1 \cdots \alpha_{j-1}}| = 1, \\
F_{\tilde{Q}_j^{\alpha_1 \cdots \alpha_{j-1}}} & \text{if } \prod_{\alpha_{j-1} \in \mathbb{F}_2} |\tilde{Q}_j^{\alpha_1 \cdots \alpha_{j-1}}| = 0, \end{cases}$$

(5.19)

Since $\tilde{Q}_j^{\alpha_1 \cdots \alpha_j} = Q_j^{\alpha_i \cdots \alpha_j}$ for $i > J_j$, combining (5.18) and (5.19) gives (5.4) in the case of $j$.

We conclude that all of (B)Q, (C)Q and (5.4) are true in the case of $j$, and we complete the induction step. Therefore Proposition 5.1 holds.

Proof of Lemma 5.2. The definition of $i$ implies $p \in \mathcal{BP}_{i,n}$, and that of $B$ implies $p \in \mathcal{BP}_{i,i+B}$. Therefore, by (3.1), we obtain (5.10).

Proof of Lemma 5.3. Obviously, (5.11) with $j = i + 2$ holds by definition. Since $\mathcal{N}(Q) \subset Q$ for any subset $Q$ of $\mathbb{F}_p^n$, (5.11) with $j = i + 2$ is proved by

$$P^{\alpha_i \alpha_{i+1}} = \mathcal{N}(\{p|x_{i+1}=\alpha_{i+1} \mid p \in \mathcal{P}^{\alpha_i}\}) = \mathcal{N}(\{p|x_{i+1}=\alpha_{i+1} \mid p \in \mathcal{N}(\{p|x_i=\alpha_i \mid p \in \mathcal{P}\})\}) \subset \mathcal{N}(\{p|x_{i+1}=\alpha_{i+1} \mid p \in \mathcal{N}(\{p|x_i=\alpha_i \mid p \in \mathcal{P}\})\}) = \mathcal{N}(\{p|x_i=\alpha_i,x_{i+1}=\alpha_{i+1} \mid p \in \mathcal{P}\}).$$

Similarly, we can prove (5.11) for general $j$ using induction on $k = j - i$. We omit the details for space limitation.
6  Proofs of Corollaries 3.3 and 3.4

We will prove Corollary 3.3.

Proof of Corollary 3.3. Suppose that S is BMQ. Then the Boolean polynomials in S are quadratic polynomials, and their degrees are at most 2. The number of monomials of degrees at most 2 is bounded by \( O(n^2) \), and S belongs to \( \text{Sat}_n^{(l)} \) with \( l = O(n^2) \). Thus Theorem 3.2 implies (i).

Suppose that S is CNF. Then the Boolean polynomials in S are clause polynomials, and S belongs to \( \text{Sat}_n^{(l)} \) with \( l = 1 \), which, together with Theorem 3.2, proves (ii).  

We will introduce some notions and facts for graph list-coloring to prove Corollary 3.4.

Let \( S \) be a system in the variables \( x_{v,h} \) which consists of the following equations:

\[
g_{v,c} := \prod_{h=1}^{k} (x_{v,h} + \gamma_h + 1) \in \mathbb{B}[x_{v,1}, \ldots, x_{v,k}],
\]

where \( (\gamma_1, \ldots, \gamma_k) = \varphi^{-1}(c) \). For an edge \( uv \), we also define

\[
g_{uv,c} := \prod_{h=1}^{k} (x_{u,h} + x_{v,h} + \gamma_h + 1) \in \mathbb{B}[x_{u,1}, \ldots, x_{u,k}, x_{v,1}, \ldots, x_{v,k}].
\]

Let \( S_{(G,L)} \) be a system in the variables \( x_{v,h} \) which consists of the following equations:

\[
g_{v,c} = 0 \quad (v \in V, c \in L_v'), \tag{6.1}
\]

\[
g_{uv,0} = 0 \quad (uv \in E), \tag{6.2}
\]

where \( L_v \) are \([l]\)-lists of allowed colors for vertices \( v \). Note that the color 0 is corresponding to the zero tuple, and \( g_{uv,0} = \prod_{h=1}^{k} (x_{u,h} + x_{v,h} + 1) \).

Let \( \alpha_v \in \mathbb{F}_2^k \) and set \( a_v = \varphi(\alpha_v) \) for vertices \( v \). We see that \( a_v \neq c \) if and only if \( g_{v,c}(\alpha_v) = 0 \) for a color \( c \), because zero is an annihilating element for product. We also see that \( g_{uv,0}(\alpha_u, \alpha_v) = g_{u,a_u}(\alpha_u) = g_{v,a_v}(\alpha_v) \) because of the definitions. By these facts we can find the following properties:

(V) For a vertex \( v \), the color \( a_v \) is in \( L_v \) if and only if \( g_{v,c}(\alpha_v) = 0 \) for all \( c \in L_v' \).

(E) For an edge \( uv \), the colors \( a_u \) and \( a_v \) are different if and only if \( g_{uv,0}(\alpha_u, \alpha_v) = 0 \).

We will show that the list-colorability of \( G \) on \( (L_v)_{v \in V} \) is equivalent to the satisfiability of \( S_{(G,L)} \). Suppose that \( G \) is list-colorable. Then there exists a tuple \((a_v)_{v \in V}\) of colors such that (i) \( a_v \in L_v \) for every \( v \in V \); and (ii) \( a_u \neq a_v \) for every \( uv \in E \). It follows from (i) and (V) that \((\varphi^{-1}(a_u))_{u \in V}\) satisfies (6.1), and from (ii) and (E) that \((\varphi^{-1}(a_v))_{v \in V}\) satisfies (6.2). Hence \((\varphi^{-1}(a_v))_{v \in V}\) is a solution, and \( S_{(G,L)} \) is satisfiable. Suppose that \( S_{(G,L)} \) is satisfiable, and \((\alpha_v)_{v \in V}\) is its solution. Similarly to the above, it can be seen that \((\varphi(\alpha_v))_{v \in V}\) is a proper assignment of colors. Thus \( G \) is list-colorable.
We are in a position to prove Corollary 3.4.

Proof of Corollary 3.4. We may assume that $l \geq 2$ and $k \geq 1$, where $2^{k-1} < l \leq 2^k$. Let $S_{(G,L)}$ be the system defined by (6.1) and (6.2). The Boolean polynomials in the system are in $\mathbb{B}^P(x_{v,h})_{v\in V,1\leq h\leq k}$, and $S_{(G,L)} \in \text{Sat}^{(2k)}_{kn}$.

Firstly, we will show $S_{(G,L)} \in \text{Sat}^{(2l)}_{kn}$. 

If $g$ is $g_{v,c}$ in (6.1), then $g$ is a clause and $g \in \text{CL}^{(1)}_{kn} \subset \text{CL}^{(2l)}_{kn}$. Suppose that $g$ is $g_{uv,0}$ in (6.2). Then

$$g = \prod_{h=1}^{k} (x_{u,h} + x_{v,h} + 1) = \sum_{(H_u,H_v) \in \{1,\ldots,k\}^2, H_u \cap H_v = \emptyset} \left( \prod_{h_u \in H_u} x_{u,h_u} \right) \left( \prod_{h_v \in H_v} (x_{v,h_v} + 1) \right) \in \text{CL}^{(2k)}_{kn}.$$ 

Because $2^{k-1} < l$, we have $2^k < 2l$ and $\text{CL}^{(2k)}_{kn} \subset \text{CL}^{(2l)}_{kn}$. Therefore $g \in \text{CL}^{(2l)}_{kn}$. Since $S_{(G,L)}$ consists of Boolean polynomials in (6.1) and (6.2), we obtain (6.3).

Let $v_1,\ldots,v_n$ be vertices whose order give the bandwidth $B$. Referring to the order of vertices, we define that of variables by

$$x_{1,1},\ldots,x_{1,k},x_{2,1},\ldots,x_{2,k},\ldots,x_{n,1},\ldots,x_{n,k},$$

where $x_{i,h} = x_{v_i,h}$. By (3.22) and (3.23), the bandwidth of $S_{(G,L)}$ is $(B+1)k - 1$. With (6.3), Theorem 3.2 implies that the satisfiability of $S_{(G,L)}$ is decidable in time

$$O(2^{(B+1)k-1}2l(m' + kn)),$$

where $m'$ is the number of equations in (6.1) and (6.2). We have

$$2^{(B+1)k} < (2l)^{B+1}, \quad m' \leq 2^kn + m < 2ln + m, \quad kn < ln.$$ 

Thus the time of solving $S_{(G,L)}$ is bounded by $O((2l)^{B+2}(m + ln))$. This completes the proof, because the list-colorability of $G$ on $(L_v)_{v \in V}$ is equivalent to the satisfiability of $S_{(G,L)}$. \qed

Acknowledgement

The author would like to thank Tomohiro Sonobe for his helpful comments.

References

[1] G. V. Bard, N. T. Courtois and C. Jefferson, Efficient methods for conversion and solution of sparse systems of low-degree multivariate polynomials over GF(2) via SAT-solvers, IACR Cryptology ePrint Archive: Report 2007/024, 2007.

[2] M. Bardet, J-C. Faugère, S. Bruno and P-J. Spaenlehauer, On the complexity of solving quadratic Boolean systems, J. Complexity 29 (2013), 53–75.
[3] P. Beame, R. Impagliazzo, J. Krajíček, T. Pitassi and P. Pudlák, *Lower bounds on Hilbert’s Nullstellensatz and propositional proofs*, Proc. London Math. Soc. 73 (1996), 1–26.

[4] A. Biere, M. Heule, H. V. Maaren, and T. Walsh (eds.), *Handbook of satisfiability*, Frontiers in Artificial Intelligence and Applications, Volume 185, IOS Press, Amsterdam, The Netherlands, 2009.

[5] A. Björklund, T. Husfeldt and M. Koivisto, *Set Partitioning via Inclusion-Exclusion*, SIAM J. Comput. 39 (2009), 546–563.

[6] A. Blum, G. Konjevod, R. Ravi and S. Vempala, *Semi-definite relaxations for minimum bandwidth and other vertex-ordering problems*, Proceedings of the thirtieth Annual ACM Symposium on Theory of Computing, 100–105, 1998.

[7] M. Brickenstein and A. Dreyer, *PolyBoRi: a framework for Gröbner-basis computations with Boolean polynomials*, J. Symbolic Comput. 44 (2009), 1326–1345.

[8] C. Carlet and P. Guillot, *A new representation of Boolean functions*, Applied algebra, algebraic algorithms and error-correcting codes, 94–103, Lecture Notes in Comput. Sci., 1719, Springer, Berlin, 1999.

[9] P. Z. Chinn, J. Chvátalová, A. K. Dewdney and N. E. Gibbs, *The bandwidth problem for graphs and matrices - a survey*, J. Graph Theory 6 (1982), 223–254.

[10] M. Clegg, J. Edmonds and R. Impagliazzo, *Using the Groebner basis algorithm to find proofs of unsatisfiability*, Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing, 174–183, 1996.

[11] S. A. Cook, *The complexity of theorem-proving procedures*, Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing, 151–158, 1971.

[12] D. A. Cox, J. Little and D. O’Shea, *Ideals, varieties, and algorithms. An introduction to computational algebraic geometry and commutative algebra. Fourth edition.*, Undergraduate Texts in Mathematics. Springer, Cham, 2015.

[13] J. Díaz, J. Petit and M. Serna, *A survey of graph layout problems*, ACM Comput. Surv. 34 (2002), 313–356.

[14] U. Feige, *Approximating the bandwidth via volume respecting embeddings*, J. Comput. Syst. Sci. 60 (2000), 510–539.

[15] M. R. Fellows, F. V. Fomin, D. Lokshtanov, F. Rosamond, S. Saurabh, S. Szeider and C. Thomassen, *On the complexity of some colorful problems parameterized by treewidth*, Inf. Comput. 209 (2011), 143–153.

[16] M. R. Fellows, D. Lokshtanov, N. Misra, F. A. Rosamond and S. Saurabh, *Graph layout problems parameterized by vertex cover*, In: Hong SH., Nagamochi H., Fukunaga T. (eds) Algorithms and Computation. ISAAC 2008. Lecture Notes in Computer Science, vol 5369. Springer, pp 294-305.

[17] J. Fiala, P. A. Golovach and J. Kratochvíl, *Parameterized complexity of coloring problems: Treewidth versus vertex cover*, Theor. Comput. Sci. 412 (2011), 2513–2523.

[18] J. J. Gégalkine, *Sur le calcul des propositions dans la logique symbolique*, Mat. Sb. 34 (1927), 9–28.
[19] P. A. Golovach, M. Johnson, D. Paulusma and J. Song, A Survey on the Computational Complexity of Coloring Graphs with Forbidden Subgraphs, J. Graph Theory 84 (2017), 331–363.

[20] E. Guerrini, E. Orsini and M. Sala, Computing the distance distribution of systematic nonlinear codes, J. Algebra Appl. 9 (2010), 241–256.

[21] T. Hertli, 3-SAT faster and simpler – Unique-SAT bounds for PPSZ hold in general, SIAM J. Comput. 43 (2014), 718–729.

[22] K. Jansen and P. Scheffler, Generalized coloring for tree-like graphs, Discrete Appl. Math. 75 (1997), 135–155.

[23] A. Joux and V. Vitse, A crossbred algorithm for solving Boolean polynomial systems, Number-theoretic methods in cryptology, 3–21, Lecture Notes in Comput. Sci., 10737, Springer, Cham, 2018.

[24] D. E. Knuth, The art of computer programming. Vol. 3., Sorting and searching. Second edition. Addison-Wesley, Reading, MA, 1998.

[25] E. Kunz, Introduction to commutative algebra and algebraic geometry, Translated from the German by Michael Ackerman. With a preface by David Mumford. Birkhäuser Boston, Inc., Boston, 1985.

[26] L. Levin, Universal search problems (in Russian), Probl. Peredachi Inf. 9 (1973), 115–116. Translated into English by B. A. Trakhtenbrot: A survey of Russian approaches to perebor (brute-force searches) algorithms, Ann. Hist. Comput. 6 (1984), 384–400.

[27] D. Lokshtanov, R. Paturi, S. Tamaki, R. Williams and H. Yu, Beating brute force for systems of polynomial equations over finite fields, Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, 2190–2202, 2017.

[28] S. Lundqvist, Boolean ideals and their varieties, J. Pure Appl. Algebra 219 (2015), 4521–4540.

[29] A. Meneghetti, Vanishing ideals of binary Hamming spheres, preprint; arXiv:1802.02875 [math.AC], 2018.

[30] D. E. Muller, Application of Boolean algebra to switching circuit design and to error detection, Transactions of the I.R.E. Professional Group on Electronic Computers EC-3 (1954), 6–12.

[31] I. S. Reed, A class of multiple-error-correcting codes and the decoding scheme, Transactions of the IRE Professional Group on Information Theory 4 (1954), 38–49.

[32] A. Rostovtsev and A. Mizyukin, On Boolean ideals and varieties with application to algebraic attacks, Nonlinear Phenom. Complex Syst. 17 (2014), 242–252.

[33] Y. Sato, S. Inoue, A. Suzuki, K. Nabeshima and K. Sakai, Boolean Gröbner bases, J. Symbolic Comput. 46 (2011), 622–632.

[34] S. Szeider, On fixed-parameter tractable parameterizations of SAT, In: Giunchiglia E., Tacchella A. (eds) Theory and Applications of Satisfiability Testing. SAT 2003. Lecture Notes in Computer Science, vol 2919, pp 188-202. Springer, Berlin, Heidelberg (2004).
[35] R. Williams, *The polynomial method in circuit complexity applied to algorithm design (invited talk)*, 34th International Conference on Foundation of Software Technology and Theoretical Computer Science, 47–60, 2014.