アルゴリズム 41－2
（1994. 9. 21）

2分決定グラフ，ガウスの消去法，グラフ理論

今井 浩　　谷 誠一郎

東京大学大学院理学系研究科情報科学専攻

2分決定グラフは，論理関数を効率よく表現することができ，広く用いられている．本論文では，グラフや計算幾何の問題の解を特性関数を用いて論理関数で表してその論理関数を2分決定グラフで表現するとき，その2分決定グラフの大きさを小さくする問題が，グラフ理論で確立されているグラフとガウスの消去法の関連を用いることにより，高速かつ高性能で解けることを示す．2分決定グラフで表現するグラフの性質としては，全域木，マッチング，クリーク，安定集合を考え，その2分決定グラフをよい変数順序を選んで最小化するために，平面分離定理などのグラフ理論の成果を適用する．これにより，たとえば，任意のn点単純平面グラフに対して，その全域木を表す$O(n)$変数の論理関数は$O(4^{\sqrt{n}})$の大きさの2分決定木で表現することができ，その安定集合を表すn変数の論理関数は$O(n2^{\sqrt{n}})$の大きさの2分決定木で表現することができることを示す．また，OBDDパッケージを用いてこれらの論理関数を構成した計算機実験の結果についても述べる．本論文で扱う論理関数はグラフに関連したもので特殊なクラスではあるが，一般の論理関数を表す2分決定グラフに関しても有用な知見が得られる．

Ordered Binary Decision Diagrams, Gaussian Elimination and Graph Theory

Hiroshi Imai and Seiichiro Tani

Department of Information Science, University of Tokyo
Tokyo 113, Japan

Ordered binary decision diagrams (OBDDs in short) have been shown as a powerful paradigm in handling Boolean functions and have been applied to many fields such as VLSI CAD, AI, combinatorics, etc. In this paper, we consider OBDDs of Boolean functions representing some concepts in graph theory such as spanning trees, matchings, cliques, etc., as well as concepts in computational geometry such as planar triangulations. We demonstrate that the problem of finding good variable orderings that make the sizes of these OBDDs smaller is strongly related to Gaussian elimination of graphs.

Our results have many implications. From the viewpoint of OBDD research, the results give much more insight to the variable ordering problem to minimize the size of OBDDs. From the viewpoint of graphs as well as computational geometry, we provide a new way of solving graph problems by the existing OBDD packages and improve the efficiency of this approach greatly. In fact, we also demonstrate by computational experiments that many can be done by the existing OBDD packages using bipartite matchings and planar triangulations as examples.

1 Introduction

Efficient Boolean function manipulation is needed to solve many problems in various areas such as digital logic design, artificial intelligence and combinatorics. For that purpose, binary decision diagrams (BDD), or ordered binary decision diagrams (OBDD) [1, 6], have been shown to be a powerful tool (e.g., see [7]). There have been developed several BDD packages [5, 13], and further enhancements in its time and space/size complexity are required.

A main issue in handling OBDDs for Boolean functions is to find a good ordering of variables to make the size of OBDDs small, since the size varies very much, from polynomial to exponential in some cases, by the orderings. However, the intrinsic intractability of finding a best ordering of variables that minimizes the size of OBDDs for general functions has been shown recently. In [17], it was shown that the problem for the shared binary decision diagram is NP-complete. The NP-hardness of the problem for OBDDs when the function is given in terms of CNF, DNF, logical circuits, etc., in [12]. The NP-completeness of the problem for OBDDs was shown in [4]. Thus, the problem of finding a not best but better ordering of variables to make OBDDs smaller has been a challenging problem. There have been proposed many heuristics (e.g., [13, 8]), while there is a theoretical framework which connects the OBDD minimization problem with the register allocation problem [3].

In this paper, we consider Boolean functions representing many concepts in graph theory such as spanning trees, matchings, cliques and stable sets, and demonstrate that the problem of finding a good variable ordering for these functions is strongly related to Gaussian elimination of graphs. Triangulations of a planar point set in computational geometry are also investigated. This generalizes a framework of applying OBDDs to combinatorics of graphs first done in [16] and further extended in [18]. Using many fertile graph properties underlying the Boolean functions, we show that many graph-theoretic techniques, especially those for Gaussian elimination, can be applied to find a good ordering, which may be regarded as an extension of the above-mentioned work by Berman.

Specifically, this paper shows the following.

- The existing theorems on extremal problems of graphs can be used to obtain a better nontrivial bound on the size of OBDDs. For example, the size of any OBDD representing a Boolean function whose support is a set of characteristic vectors of spanning trees of a graph with n vertices and m edges is shown to be $O(\min\{mn^{n-2}, m\binom{m}{n-1}\})$.
- We relate Gaussian elimination of graphs to OBDDs. In this framework, we obtain many useful results using separator theorems, etc., for graphs. For example, for a planar graph with n vertices, there is a variable ordering such that the size of the corresponding OBDD, whose support is a set of characteristic vectors of spanning trees, is $O(n2^{\sqrt{n}})$.
- We also give bounds on OBDDs related to matchings, perfect graphs and triangulations of a planar point set, together with some computational experiments. Through the experiments, we demonstrate how the existing BDD packages can be applied to bipartite matchings and planar triangulations as examples to illustrate and elucidate the power of this approach.

All the computational experiments shown in this paper were made by using the BDD package utilizing edge attributes developed in [13].

2 Preliminaries on OBDDs

An ordered binary decision diagram (OBDD in short) is a directed acyclic graph which represents a Boolean function [6]. An OBDD which represents a Boolean function $f(x_1, x_2, ..., x_l)$ is defined as a 5-tuple $(X, N, root, label, edge)$:

$X = \{x_1, x_2, ..., x_l\}$ is a totally ordered set of variables,

$N = N_V \cup N_C$ $(N_V \cap N_C = \phi)$ is a set of nodes, where N_V is a set of *variable nodes*, $N_C = \{c_0, c_1\}$ is a set of *constant terminal nodes*,

$root \in N$ is an *root node*,

$label : N \rightarrow (X \cup \{0, 1\})$,

$edge : N_V \times \{0, 1\} \rightarrow (N_V \cup N_C)$ is a set of *edges*, such that, for all $v \in N_V$ with $label(v) \in X$, $label(edge(v, b)) \in \{0, 1\}$ or $label(v) < label(edge(v, b))$ $(b \in \{0, 1\})$.

In OBDDs, nodes on any path from the rootnode to a terminal node are labeled differently, and that the ordering of variables on each path is consistent with one another. We denote the ith variable over X by $\pi[i]$, and call $\pi = (\pi[1], \pi[2], \ldots, \pi[n])$ *variable ordering*. An OBDD with a root node v represents a Boolean function $F(v)$ such that if v is a constant node, then $F(v) = label(v)$ and that if v is a variable node, then $F(v) = \overline{label(v)}F(edge(v,0)) + label(v)F(edge(v,1))$.

The size of an OBDD is defined to be $|N_V|$. The size of an OBDD can be reduced by applying repeatedly the following *reduction rules*: If there exists a variable node v where $edge(v,0) = edge(v,1)$ (such a node v is called a *redundant node*), then eliminate v and redirect all incoming edges to $edge(v,0)$. If there exist *equivalent nodes* u, v, then eliminate v and redirect all edges into u to v. Two nodes u, v are equivalent if one of the following holds:
(1) If both u and v are constant nodes, $label(u) = label(v)$.
(2) If both u and v are variable nodes, $label(u) = label(v)$, $edge(u,0) = edge(v,0)$ and $edge(u,1) = edge(v,1)$.

Maximally reduced OBDD is called a reduced OBDD (ROBDD). For a fixed variable ordering, an ROBDD is canonical, i.e., uniquely determined. For an ROBDD the *width of level* i is defined as the number of nodes v in that level plus the number of edges which pass the level. The *width* of an ROBDD is defined as the maximum value of the width of level i for $i = 1, 2, ..., |X|$. In the following, an ROBDD will be called an *OBDD* simply.

3 Bounding the size of OBDDs by combinatorics

Let f be a Boolean function of l logical variables. We call a set of all truth assignments to the variables which make $f = 1$ the *support* of f. Since the size of truth assignments is 2^l, the size of support is bounded by 2^l. But, for some functions, it may be much less than 2^l.

Let $S = \{1, 2, \ldots, l\}$ be a finite set with $|S| = l$, and $\mathcal{S}$ be a collection of subsets of S that satisfy some specified properties. A subset S' of S can be represented by a characteristic vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_l)$ with $x_i = 1$ for $i \in S'$ and $x_i = 0$ for $i \notin S'$. This vector can be regarded as a truth assignment $\boldsymbol{x}$ for l logical variables. With this family $\mathcal{S}$, we can naturally associate a Boolean function $f_{\mathcal{S}}$ of l variables x_i $(i = 1, \ldots, l)$ defined by

$$f_{\mathcal{S}}(\boldsymbol{x}) = \begin{cases} 1 & \boldsymbol{x}\text{: characteristic vector of } S' \in \mathcal{S} \\ 0 & \text{otherwise} \end{cases}$$

In this way, for any family of subsets, there is the uniquely determined Boolean function, and we can apply OBDDs to the function to solve problems concerning the family of subsets. Once the OBDD for $f_{\mathcal{S}}$ is at hand, the cardinality $|\mathcal{S}|$ of $\mathcal{S}$ and an optimal set in $\mathcal{S}$ (maximum/minimum-weight set, bottleneck optimal set, etc.) can be found in time linear to the size of the OBDD, thus solving many counting and optimization problems concerning $\mathcal{S}$. This framework was shown to be useful to solve many counting problems in combinatorics [16]. The framework was extended in [18] algorithmically so that it can be applied to a wider class of problems mainly related to graphs.

When this framework is used for combinatorial problems whose combinatorial structure is partly known, we may have nontrivial bounds on the size of OBDD of relevant functions as follows.

Theorem 3.1 *The size of any OBDD representing a Boolean function f of l variables is bounded by the size of support of f times the number l of variables.*

The proof of this theorem is omitted in this version due to the space limitation. In the following, we describe a few applications of this theorem.

Let G be a simple graph with n vertices and m edges. Let f_{tree} be a Boolean function of m logical variables corresponding to the edges such that $f_{\text{tree}} = 1$ iff its truth assignment corresponds to a spanning tree of G. Since this Boolean function have m logical variables, its size is bounded by $O(2^m/m)$ [9]. But, for complete graphs, $m = n(n-1)/2$, and this bound becomes $O(2^{\binom{n}{2}}/\binom{n}{2})$ which is quite large.

Due to the structure of graphs, the OBDD cannot be so large, which can be seen as follows. First of all, the number of spanning trees is trivially bounded by $\binom{m}{n-1}$. It is known that the number of spanning trees of a complete graph of n vertices is n^{n-2}, and hence that for G is bounded by $\min\{n^{n-2}, \binom{m}{n-1}\}$. We thus obtain the following corollary.

Corollary 3.1 *The size of any OBDD representing f_{tree} is $O(m \cdot \min\{n^{n-2}, \binom{m}{n-1}\})$.*

This is a better bound than the trivial bound mentioned above. Thus, the OBDD representing spanning trees is much smaller than the general bound of $2^m/m$. However, if we regard that the vertices play a central role here, the above bound is rather natural. In section 5.2, we show that for

n-vertex planar graph, there is a variable ordering by which the size of OBDD is $O(n2^{\sqrt{n}})$. This is better than $O(2^n/n)$ bound even if n is regarded as a parameter determining the complexity of the problem.

For some of the problems related to perfect graphs, we can have nontrivial results. Here, an example concerning cliques is mentioned. Later, in section 5.2, we investigate the stable set problem for planar graphs. For a graph $G = (V, E)$, let f_{clique} be a Boolean function whose logical variables correspond to the vertex set V and it becomes 1 iff its truth assignment corresponds to a maximal clique of the graph. A chordal graph (or, triangulated graph) with n vertices only have at most n maximal cliques, and from this we have the following.

Corollary 3.2 *The size of any OBDD representing f_{clique} of a chordal graph of n vertices is at most n^2.*

The chordal graph has strong connection with Gaussian elimination of graphs, and we will come back to this issue in section 5.2.

Next, we consider the problem for triangulations of a planar point set. This problem can be formulated as that related to stable sets of an intersection graph of all line segments generated by the point set, and of course this is related to cliques for the complement graph. For n points in the plane, let f_{tri} be a Boolean function of $\binom{n}{2}$ variables corresponding to line segments connecting all pairs of points such that $f_{\text{tri}} = 1$ iff the truth assignment corresponds to a triangulation of the points. Since the number of triangulations is known to be exponential in n, we have the following.

Corollary 3.3 *The size of any OBDD representing f_{tri} of n points in the plane is $O(n^2c^n)$ for some constant c.*

We thus have another example of a Boolean function of $\binom{n}{2}$ variables such that the OBDD for any variable ordering can have nodes singly exponential in n.

Using this problem as an example, we show some computational results indicating how the variable ordering affects the OBDD size and also the merit of the theorem and corollaries so far obtained. In Table 1, the size of OBDDs of f_{tri} for $n-4$ points uniformly distributed in the unit square plus 4 corner points of the square is shown. We tested three variable orderings:

Table 1: Computational results for f_{tri} representing triangulations of n points generating $m = \binom{n}{2}$ line segments

		U1	U2	U3
n	m	|OBDD|	|OBDD|	|OBDD|
8	28	253	145	357
9	36	415	903	1187
10	45	1028	3035	2519
11	55	2028	4851	8210
12	66	6544	7021	46710
13	78	16904	—	—
14	91	29762	—	—
15	105	76788	—	—
16	120	—	—	—

—: not applicable
|OBDD|: the size of OBDD

U1: Sort n points in the increasing order of their x-coordinates from p_1 to p_n, and order line segments $\overline{p_ip_j}$ by the following program: **for** $i :=$ 2 **to** n **do for** $j := 1$ **to** $i-1$ **do** add $\overline{p_ip_j}$;

U2: Order line segments in the increasing order of their degrees (the number of other line segments intersecting it).

U3: Random ordering.

From the results in Table 1, it is observed that for all variable orderings the size of OBDDs grows singly exponentially. However, there is a big difference concerning the relative sizes among U1, U2 and U3. U1 is the best ordering among these. This ordering has connection with a good ordering for the convex case mentioned in the next paragraph. U2 is a kind of greedy heuristic, and however the package we used could not compute the OBDD for $n = 13$. This would be because of a problem of the traditional algorithms used in the package, i.e., constructing OBDDs from small ones by applying Boolean operations to them, which may cause unnecessary combinatorial explosion during the computation even if the final OBDD itself is small (e.g., see [14]). Such a difficulty can be overcome by using the output-size sensitive algorithm proposed in [18].

It should be noted that, once the OBDD is constructed for f_{tri}, a minimum-length triangulation can be found in time linear to the size of that OBDD. The problem of finding a minimum-length triangulation has been a big open problem, and the OBDD-approach to combinatorial optimization can solve this problem in polynomial time in case there is a variable ordering by which the OBDD size is bounded polynomially in n and it can be found in polynomial time. Note that, by using the result concerning Catalan number in

[16], we can show that there exists a polynomial-size OBDD for f_{tri} when n points are all on the boundary of their convex hull.

4 Perfect matchings of a bipartite graph

In this section, we consider the problem of finding all perfect matchings or counting the number of perfect matchings for a given bipartite graph $G = (U, V; E)$ with left vertex set U, right vertex set V and edge set $E \subseteq U \times V$. We assign a variable x_i $(i = 1, 2, ..., |E|)$ to each edge of G. Furthermore we define a function $f_{\mathrm{match}} = 1$ iff the truth assignment corresponds to a perfect matching. We are here interested in the size of OBDDs for f_{match}, and obtain the following.

Theorem 4.1 *Suppose that the bandwidth of the associated matrix of a given bipartite graph with $|U| = |V| = n$ vertices is k. Then, by ordering vertices according to the row ordering of the matrix attaining the bandwidth and ordering edges naturally induced by it, the corresponding OBDD consists of $O(kn2^k)$ nodes.*

Here, the *bandwidth* of a matrix $A = (a_{ij})$ is defined to be $2\max\{\, |i-j| \mid a_{ij} \neq 0 \,\} + 1$.

Theorem 4.2 *For a random bipartite graph, its OBDD size is exponential for any variable ordering with high probability.*

These theorems imply that we should investigate the structure of graphs. Instead of giving rigorous proofs, we here give some computational results to give nice flavor of practical aspects of OBDDs besides results for triangulations of a planar point set.

4.1 Complete bipartite graph $K_{n,n}$

Let $K_{n,n}$ be a complete bipartite graph with $U = \{u_1, \ldots, u_n\}$, $V = \{v_1, \ldots, v_n\}$ and $|E| = n^2$. For this graph, we tested the following three variable orderings:

V1: Arrange edges (u_i, v_j) in the lexicographically increasing order of (i, j) $(1 \leq i, j \leq n)$. (For $K_{3,3}$, (u_1, v_1), (u_1, v_2), (u_1, v_3), (u_2, v_1), (u_2, v_2), (u_2, v_3), (u_3, v_1), (u_3, v_2), (u_3, v_3).)

V2: Recursively arrange edges of $K_{n-1,n-1}$ formed by $u_1, \cdots, u_{n-1}$ and $v_1, \cdots, v_{n-1}$ and then add edges (u_i, v_n) $(i = 1, \ldots, n-1)$ and (u_n, v_j) $(j = 1, \ldots, n)$ in this order. (For $K_{3,3}$, (u_1, v_1), (u_1, v_2), (u_2, v_1), (u_2, v_2), (u_1, v_3), (u_2, v_3), (u_3, v_1), (u_3, v_2), (u_3, v_3).)

V3: Random order.

Table 2: Computational results for $K_{n,n}$

—	V1	V2	V3
n	|OBDD|	|OBDD|	|OBDD|
2	6	6	6
3	23	25	29
4	72	90	135
5	201	296	711
6	522	923	3679
7	1291	2780	14291
8	3084	8169	104713
9	7181	23579	440583
10	16398	67145	—
11	36879	189208	—
12	81936	—	—
13	180241	—	—
14	393234	—	—
15	—	—	—

Computational results are shown in Table 2.

As is seen from the table, the sizes of OBDDs are small in the order of V1<V2<V3. In all cases, their sizes exponentially blow up like $(c_i)^n$ for each $i = 1, 2, 3$, and $c_1 < c_2 < c_3$. As stated in Theorem 4.1, by the ordering of V1, the size of OBDDs is bounded by $O(n^2 2^n)$.

4.2 Random bipartite graph

Next, consider a random bipartite graph $R_{n,p}$ such that each edge of $K_{n,n}$ exists in the graph with probability p $(0 < p < 1)$. Theoretically, for $p = \Omega(\log n/n)$, $R_{n,p}$ is connected and has a perfect matching. For the variable ordering, V1 for the complete graph was modified for this case. For the results, see Table 3.

Table 3: Computational results for $R_{n,p}$

$p = 0.1$		
n	$\|E\|$	|BDD|
50	236	—

$p = 0.2$		
n	$\|E\|$	|BDD|
20	87	25351
22	104	160211
23	112	—

$p = 0.5$		
n	$\|E\|$	|BDD|
14	101	71512
16	130	560788
17	151	—

$p = 0.9$		
n	$\|E\|$	|BDD|
10	89	13388
11	108	31912
12	128	70516
13	152	161844
14	178	347093
15	207	—

It is easy to show that the size of the OBDD for $R_{n,p}$ is smaller than that of $K_{n,n}$ if we use the consistent variable ordering V1. From the computational results, the size for the OBDD for $R_{n,p}$

is rather large compared with that of the OBDD for $K_{n',n'}$ consisting of roughly same number of edges under the variable ordering V1.

4.3 Bipartite graphs with a constant bandwidth

Consider a bipartite graph $BB_{n,3} = (U, V; E)$ with U, V as above and $E = \{(u_i, v_j) \mid |i-j| \leq 3\}$. The associated matrix of this bipartite graph has a bandwidth of 7. As for the variable orderings, V1, V2 and V3 are modified for this case in an obvious way. Computational results are shown in Table 4.

Table 4: Computational results for $BB_{n,3}$

n	$\lvert E\rvert$	V1 $\lvert$OBDD$\rvert$	V2 $\lvert$OBDD$\rvert$	V3 $\lvert$OBDD$\rvert$
10	58	870	911	75009
11	65	1010	1051	259868
12	72	1150	1191	—
15	93	1570	1611	—
20	128	2270	2311	—
30	198	3670	3711	—
40	268	5070	5111	—
100	688	13470	13511	—

It is observed that the sizes of OBDDs are linear for V1 and V2, whereas that for V3 exponentially blows up. This would be rather apparent from the structure of the bipartite graph.

Summarizing observations obtained so far, we have the following

- Sparsity of graphs does not necessarily imply that the size of OBDDs is small. Instead, the structure of graphs greatly affect the size of OBDDs.
- In fact, we have also tested a very sparse bipartite graph whose associate matrix is constructed in connection with the u-resultant in symbolic computation. The computational results was very poor although the graph was really sparse and the variable ordering used seemed to be appropriate. In this case, by removing several elements taking account of the structure of the u-resultant and treat them separately, the size of OBDDs is greatly reduced. This may provide a fast way of computing the u-resultant.
- It is natural to represent the structure of CNF and DNF by bipartite graphs, and the results so far also apply to logic functions of such types. In this way, by representing the internal structure of Boolean functions appropriately by graphs, we may obtain a good variable ordering for OBDDs. This observation has strong connection with the results in [3].

5 Elimination schemes and the ordering of variables in OBDDs

5.1 Decomposable cases

As observed in computational experiments, we should shed light on the structure of graphs to find a good variable ordering. The decomposition of original graphs may be such a structure, and the following theorems are obtained in [18].

Theorem 5.1 *For a bipartite graph such that every DM irreducible components in its DM decomposition are of a constant size, by using an ordering induced among the partial order among DM components, the size of OBDD representing perfect matchings is linear in the graph size.*

Theorem 5.2 *The OBDD representing all the spanning trees of a series-parallel graph is linear when an appropriate variable ordering is used.*

Theorem 5.3 *For a graph the triconnected components of which are all of a constant size, there is a linear-size OBDD representing all the spanning trees of the graph.*

These theorems indicate that if there is a decomposition of original graphs into constant size components a variable ordering consistent with the decomposition gives linear-size OBDDs. In other words, the decomposition corresponds to direct sums of OBDDs.

5.2 Separators

The example for bipartite graphs with bounded bandwidth in section 4.3 illustrates that the bandwidth has connection with good orderings for OBDDs. The bandwidth is a useful concept and has been applied to many problems. One of its most remarkable applications is Gaussian elimination of graphs. For a graph with small bandwidth, there exists a good elimination ordering in Gaussian elimination of a matrix defined by the graph. However, the converse is not necessarily the case, that is, even if the bandwidth is not small, there is a case having a better elimination ordering than an elimination ordering for the bandwidth. Rather, the existence of a good elimination ordering has strong connection with the existence of a small separators for the graph [10].

In fact, the planar separator theorem was applied in [11] to the pebbling problem strongly related to the register allocation problem. Also, from the viewpoint of OBDDs representing some properties of graphs, they gave many graph applications of the theorem. For example, they showed that a maximum stable set of a planar graph can be obtained in $O(2^{\sqrt{n}})$ time.

Thus, the problem of finding a good ordering for OBDDs for graphs is closely related to separators of graphs. Before applying this observation to OBDDs, we summarize some properties concerning separators.

For a graph $G = (V, E)$, an elimination scheme is an ordering of vertices. For $i = 1, \ldots, n$ ($n = |V|$), the i-th elimination front is a set of vertices numbered not smaller than i which are incident to some vertex numbered less than i. For $i = 1$, the elimination front is regarded as a set of the first-numbered vertex. Let $\mathcal{G}_\alpha$ be a class of graphs which are $O(n^\alpha)$-separable for $0 \le \alpha \le 1$. The planar separator theorem [11] states that any planar graph with n vertices is $O(\sqrt{n})$-separable, that is in $\mathcal{G}_{1/2}$. Then, the following lemma holds.

Lemma 5.1 *For an $O(n^\alpha)$-separable graph in $\mathcal{G}_\alpha$ with $0 \le \alpha < 1$, there exists an ordering of vertices such that the size of elimination front at any stage is $O(n^\alpha)$ if $0 < \alpha < 1$ and $O(\log n)$ if $\alpha = 0$.*

This lemma is used implicitly in many ways in [11]. Based on this observation and using the lemma, we obtain the following theorem, where $f_{\text{stable}} = 1$ iff the truth assignment corresponds to a stable set.

Theorem 5.4 *With the ordering corresponding to the elimination scheme based on separators for a graph in $\mathcal{G}_\alpha$ with ($0 < \alpha < 1$), the size of an OBDD representing f_{stable} for the graph is $O(n2^{n^\alpha})$.*

Outline of Proof We use the ordering in Lemma 5.1. The width of level i of the OBDD is equal to the number of the subproblems generated by determining whether the jth vertex ($j = 1, 2, ..., i$) in the ordering is adopted as an element of a stable set or not. Since the number of such subproblems is at most the size of elimination front power of 2, the width of the OBDD is $2^{O(n^\alpha)}$. Since the number of variables is n, the theorem follows. □

Corollary 5.1 *For a planar graph, there is a variable ordering such that the size of an OBDD representing f_{stable} for the graph is $O(n2^{\sqrt{n}})$.*

Once an OBDD is obtained, a max-weight stable set and the number of stable sets can be computed in time linear to the size of OBDD, which generalizes the results of [11, 15].

We next consider f_{tree} for planar graphs and a class of $O(n^\alpha)$-separable graphs, where f_{tree} is a function of variables corresponding to edges such that $f_{\text{tree}} = 1$ iff the truth assignment corresponds to a spanning tree. For planar graphs, we have the following theorem.

Theorem 5.5 *With the ordering corresponding to the elimination scheme based on separators, the size of an OBDD representing f_{tree} of a planar graph is $O(4^{\sqrt{n}})$.*

In this extended abstract, due to the space limitation, we omit the proof of this theorem. The proof is very similar to the following theorem for a class of $O(n^\alpha)$-separable graphs and to obtain the above theorem we have only to show that the Bell number in the following proof can be replaced with the Catalan number $(\frac{1}{n}\binom{2n-2}{n-1})$ by virtue of the planarity. For a class of $O(n^\alpha)$-separable graphs, we have the following.

Theorem 5.6 *Based on the elimination scheme based on separators, the size of an OBDD representing f_{tree} of an $O(n^\alpha)$-separable graph ($0 < \alpha < 1$) is $O(n^{\alpha n^\alpha + 2})$.*

Outline of Proof We use the same ordering with that in the proof of Theorem 5.4. Then, we can show that the width of each level is bounded by B_{n^α}, where B_k is Bell number denoting the number of partitions of a set of k objects. Since $B_k \le k^k$, we obtain the theorem. □

In the above, we have also mentioned graph properties related to perfect graphs. Among perfect graphs, chordal graphs admit an elimination scheme without any fill-in. Based on this observation, we obtain the following.

Theorem 5.7 *Let G be a chordal graph whose maximum clique size is k. With the ordering generated from a perfect elimination with using the tree structure of maximal cliques, the size of an OBDD representing f_{tree} of the graph is $O(n^{k+2})$.*

Outline of Proof Maximal cliques of a chordal graph form a tree structure. Since tree is $O(1)$-separable, we can further apply the decomposition technique based on separators to this tree structure. Then, the size of any elimination front can be shown to be $k \log n$, and the width is $O(2^{k \log n}) = O(n^k)$. The theorem follows. □

6 Concluding Remarks

This paper has demonstrated that, for OBDDs of Boolean functions representing some graph properties, a good variable ordering for OBDDs can be obtained on the analogy of Gaussian elimination for the graphs. In Gaussian elimination for graphs, separators play a key role, and besides planar graphs there are a class of graphs having good separators (e.g., see [2]). For such a wider class of graphs, we can apply the paradigm presented in this paper to OBDDs concerning those graphs.

Originally, OBDDs are applied to represent a logic circuit compactly. The structure of a logic circuit can be represented as a directed acyclic graph (DAG), and the problem of finding a good ordering for the circuit is related to the register allocation problem for this DAG [3]. From the viewpoint of complexity theory, OBDDs form a restricted class of branching program, and the size of OBDDs has connection with the space complexity. In this respect, relating the OBDD minimization problem to the register allocation problem is natural. It will be required to treat these related problems, especially Gaussian elimination of graphs to solve graph problems, in a unified fashion for further practical and new applications.

Acknowledgment

The work of the first author was supported in part by the Grant-in-Aid for Scientific Research of the Ministry of Education, Science and Culture of Japan.

References

[1] S. B. Akers: Binary decision diagrams. *IEEE Trans. Comput.*, C-27 (1978), pp.509–516.

[2] N. Alon, P. Seymour and R. Thomas: A separator theorem for graphs with an excluded minor and its applications. *Proc. 22nd ACM Symp. on Theory of Computing*, 1990, pp.293-299.

[3] C. L. Berman: Circuit width, register allocation, and ordered binary decision diagram. *IEEE Trans. Computer-Aided Design*, 10, 8 (1991), pp.1059–1066.

[4] B. Bollig and I. Wegener: Improving the bariable ordering of OBDDs is NP-complete. Preprint, 1994.

[5] K. S. Brace, R. L. Rudell and R. Bryant: Efficient implementation of a BDD package, *Proc. 27th Design Automation Conference* (1990), pp.40–45.

[6] R. E. Bryant: Graph based algorithms for Boolean function manipulation, *IEEE Trans. Comput.*, C-35 (1986), pp.677–691.

[7] J. R. Burch, E. M. Clarke, K. L. McMillan, D. L. Dill, and J. Hwang: Symbolic model checking: 10^{20} states and beyond. *Proc. 5th Symp. on Logic in Computer Science*, (1990), pp.428–439.

[8] M. Fujita, H. Fujisawa and Y. Matsunaga: Variable ordering algorithms for ordered binary decision diagrams and their evaluation. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 12, 1 (1993), pp.6–12.

[9] H.-T. Liaw and C.-S. Lin: On the OBDD-representation of general Boolean functions. *IEEE Trans. Comput.*, 41, 6 (1992), pp.661–664.

[10] R. J. Lipton, D. J. Rose and R. E. Tarjan: Generalized nested dissection. *SIAM J. Num. Anal.*, 16, 2 (1979), pp.346–358.

[11] R. J. Lipton and R. E. Tarjan: Applications of a planar separator theorem. *SIAM J. Comput.*, 9, 3 (1980), pp.615–627.

[12] Ch. Meinel and A. Slobodová: On the complexity of constructing optimal OBDD's. *Proc. 19th Int. Symp. on Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science, 1994, to appear.

[13] S. Minato, N. Ishiura and S. Yajima: Shared binary decision diagram with attributed edges for efficient Boolean function manipulation, *Proc. 27th Design Automation Conference* (1990), pp.52–57.

[14] H. Okuno: Reducing combinatorial expressions in solving search-type combinatorial problems with binary decision diagrams (in Japanese). *Trans. IPS Japan*, 35, 5 (1994), pp.739–753.

[15] S. S. Ravi, H. B. Hunt, III: An application of the planar separator theorem to counting problems. *Inf. Process. Lett.*, 25, 5 (1987), pp.317–321.

[16] I. Semba and S. Yajima: Combinatorial algorithms by boolean processing. *Trans. IPS Japan*, 35, 9 (1994), in press.

[17] S. Tani, K. Hamaguchi and S. Yajima: The complexity of the optimal variable ordering problems of shared binary decision diagrams. *Proc. Int. Symp. on Algorithms and Computation (ISAAC'93)*, Lecture Notes in Computer Science, 762 (1993), pp.389–398.

[18] S. Tani and H. Imai: A reordering operation for an ordered binary decision diagram and an extended framework for combinatorics of graphs. *Proc. Int. Symp. on Algorithms and Computation (ISAAC'94)*, Lecture Notes in Computer Science, (1994), to appear.