Electronic Colloquium on Computational Complexity, Report No. 4 (2006)

Finding small OBDDs for incompletely specified truth tables is hard

Jesper Torp Kristensen* and Peter Bro Miltersen*

January 6, 2006

Abstract

We present an efficient reduction mapping undirected graphs G with $n = 2^k$ vertices for integers k to tables of partially specified Boolean functions $g : \{0,1\}^{4k+1} \to \{0,1,\perp\}$ so that for any integer m, G has a vertex colouring using m colours if and only if g has a consistent ordered binary decision diagram with at most $(2m+2)n^2+4n$ decision nodes. From this it follows that the problem of finding a minimum-sized consistent OBDD for an incompletely specified truth table is **NP**-hard and also hard to approximate.

1 Introduction

In this paper we consider the following problem: *Given a partially defined Boolean function* $f : \{0,1\}^k \to \{0,1,\perp\}$ (with $\perp$ being interpreted as "don't care"), *find or approximate the minimum representation of* f *as an Ordered Binary Decision Diagram* (OBDD). For details about OBDDs, see the comprehensive monograph by Wegener [13]. Throughout the paper, we consider OBDDs with a *fixed* variable ordering. For concreteness and simplicity, we assume the ordering to be $x_1 < x_2 < \ldots < x_k$ for Boolean functions on k variables and always define the functions we use with its arguments in the same order, i.e., the i'th argument of a function $g : \{0,1\}^k \to \{0,1\}$ is assigned to the variable x_i. The *size* of an OBDD is the number of its decision nodes. We say that an OBDD D *represents* or *is consistent with* $f : \{0,1\}^k \to \{0,1,\perp\}$ when the fully defined Boolean function $g_D : \{0,1\}^k \to \{0,1\}$ defined by the diagram is consistent with f, i.e., satisfies $g_D(x) = f(x)$ whenever $f(x) \neq \perp$.

The corresponding minimization problem for *fully* defined Boolean functions was shown to be in **P** in the original papers introducing OBDDs by Bryant [2, 3]. Indeed, his efficient algorithm for minimizing OBDD size is one of the main attractions of using OBDD representation for Boolean functions. The minimum-size OBDD problem for partially defined Boolean functions was considered previously in two almost simultaneous papers [11, 7], both showing **NP**-hardness for versions of the problem. The hardness results of the two papers differ mainly by the way the partially defined Boolean function is to be represented.

More precisely, Sauerhoff and Wegener [11] showed the following decision problem D_1 to be **NP**-complete.

D_1: Given two OBDDs representing two Boolean functions $g_1, g_2 : \{0,1\}^k \to \{0,1\}$ and an integer s, does the partially defined Boolean function f given by $f(x) = \perp$ for those x for which $g_1(x) = 0$ and $f(x) = g_2(x)$ for those x for which $g_1(x) = 1$ have an OBDD of size less than s?

Hirata, Shimozono and Shonohara [7, 12] showed the following decision problem D_2 to be **NP**-complete.

D_2: Given two explicitly listed sets $S_0, S_1 \subseteq \{0,1\}^k$ and an integer s, does the partially defined Boolean function f given by $f(x) = 0$ for $x \in S_0$, $f(x) = 1$ for $x \in S_1$ and $f(x) = \perp$ otherwise have an OBDD of size less than s?

To compare the strengths of the two results, we observe that it is immediate that the problem D_2 polynomial-time many-one reduces to D_1: Given two sets S_0 and S_1 we can easily construct small OBDDs

*Department of Computer Science, University of Aarhus. Supported by **BRICS**, a center of the Danish National Research Foundation and by a grant from the Danish National Science Council.

ISSN 1433-8092

representing functions g_1 and g_2 so that $g_1(x) = 1$ if and only if $x \in S_0 \cup S_1$ and $g_2(x) = 1$ if and only if $x \in S_1$. On the other hand, conversion from representation of the input as two OBDDs to representation as two explicitly given sets in general incurs an exponential blowup in size and is hence not a polynomial-time reduction. Hence, the **NP**-hardness result of Hirata, Shimozono and Shonohara is stronger than the one of Sauerhoff and Wegener.

In this paper we look at a *third* input representation and consider the following decision problem.

D_3: Given an explicitly given *table* of $f : \{0,1\}^k \to \{0,1,\perp\}$ (as a string of length 2^k over $\{0,1,\perp\}$) and an integer s, does f have an OBDD of size less than s?

The main result of the present paper is that D_3 is **NP**-complete. To be precise, we establish the following reduction.

Theorem 1. *There is a polynomial time computable reduction mapping undirected graphs G with $n = 2^k$ vertices for integers k to tables of partially specified Boolean functions $g : \{0,1\}^{4k+1} \to \{0,1,\perp\}$ so that for any integer K, G has a vertex colouring using K colours if and only if g has a consistent ordered binary decision diagram with at most $(2K+2)n^2 + 4n$ decision nodes.*

Then, **NP**-hardness of D_3 follows from the **NP**-hardness of graph colouring (see, e.g., Garey and Johnson [6]).

To compare the strength of our result to the result of Hirata, Shimozono and Shonohara, we observe that it is immediate that the problem D_3 polynomial-time many-one reduces to D_2: Given a table of f, we can certainly efficiently list the sets $S_0 := \{x | f(x) = 0\}$ and $S_1 := \{x | f(x) = 1\}$. On the other hand, conversion from representation as two sets S_1, S_2 to a full table on the domain $\{0,1\}^n$ may incur an exponential blowup in size. This happens when the sets S_0 and S_1 are small (i.e., when f is undefined on most of the domain $\{0,1\}^n$). Hence, our **NP**-hardness result is stronger than the **NP**-hardness result of Hirata, Shimozono and Shonohara. Also, the proof of Hirata, Shimozono and Shonohara uses functions undefined everywhere on $\{0,1\}^k$ except on a subset of size $k^{O(1)}$, so their proof does not tell us anything about the hardness of the problem in a situation where the functions considered *are* defined on a non-negligible fraction of the domain $\{0,1\}^k$ and it does not yield our hardness result.

We find our stronger result well-motivated, as we'll explain next: A practical relevance of concrete **NP**-hardness results are their *redirection of attention* from the construction of efficient algorithms towards the construction of good heuristics for the problems at hand. This point is made explicitly by Sauerhoff and Wegener who cite several studies in the VLSI verification domain where the problem of finding minimum size OBDDs for given partial Boolean functions arise. For these applications, the input mode of Sauerhoff and Wegener is indeed the relevant one: The Boolean functions arising when formally verifying correctness of VLSI chips have truth tables so huge that representing them explicitly is out of the question, so typically, they are defined by OBDDs to begin with, as assumed by Sauerhoff and Wegener. Thus, for these applications our result provides no new "redirection signal".

However, there are other natural applications of using OBDDs for partially defined functions where the function to be encoded *is* given explicitly as a table. An application studied in the master's thesis [10] of the first author is the compression of *endgame tables* for chess. Such an endgame table may provide, for any chess position with a given set of pieces (say, a King and a Queen for White and a King and a Rook for Black) a Boolean value indicating whether the player with material advantage has a winning strategy. Given an encoding of chess positions as Boolean vectors, we may think of the table as a Boolean function $f : \{0,1\}^n \to \{0,1\}$ where $f(x)$ is the value of the chess position with Boolean encoding x. One may vary the way chess positions are represented as Boolean vectors, but any natural and efficiently computable encoding will have many Boolean vectors not representing any position. The values assigned to such vectors are inconsequential, so we may think of them as *undefined* values and hence of the table as defining a partially defined Boolean function. The potential usefulness of endgame tables for chess playing software is obvious. However, to be actually useful for such applications, an endgame table must support fast lookup and thus it should, preferably, reside in fast memory. For most endgame tables, this means that some *compression scheme* has to be applied on the table. Unfortunately, most state-of-the-art lossless compression schemes do *not* support efficient retrieval of individual bits of the compressed table (i.e., efficient table lookup). Here, representing the table by an OBDD seems to be an attractive alternative. From a theoretical point of view, Kieffer, Flajolet and Yang [9] showed that representation by OBDDs has the important *universality* property: The compression rate achieved asymptotically (i.e., for long inputs, and up to a low-order additive term) matches the *block entropy* of the string to be compressed (for any constant block size). At the same time, by construction, a table represented by an OBDD supports fairly fast lookups (we may lookup an entry in the table by following a path from the root to a leaf in the OBDD). In his master's thesis [10], the first author obtained encouraging practical results on using OBDDs to compress endgame tables for chess while preserving efficient lookup. To achieve this, heuristics had to be used to minimize the OBDDs. The hardness result of the present paper indicates that such heuristics cannot be replaced with efficient algorithms.

We finally note that we may combine our reduction with known non-approximability results for graph colouring to show that the minimum consistent OBDD problem is also hard to approximate. In particular Feige and Killian [5] showed the following theorem[1]. Recall that **ZPP** is the class of decision problems which can be solved in expected polynomial time by a randomized algorithm.

Theorem 2 (Feige and Killian). *For any $\epsilon > 0$, if* $\mathbf{NP} \neq \mathbf{ZPP}$*, no polynomial time algorithm distinguishes between the following two classes of graphs:*

- *Graphs $G = (V, E)$ with chromatic number less than $|V|^{\epsilon}$.*
- *Graphs $G = (V, E)$ with chromatic number bigger than $|V|^{1-\epsilon}$.*

Combining Theorem 1 with Theorem 2, noticing that we in Theorem 2 without loss of generality can assume that the graphs considered have $n = 2^k$ vertices for an integer k, we immediately obtain:

Corollary 3. *Let $\epsilon > 0$ be an arbitrary constant. If* $\mathbf{NP} \neq \mathbf{ZPP}$*, no polynomial time algorithm distinguishes between the following two classes of incompletely specified truth tables $f : \{0,1\}^k \to \{0,1,\perp\}$:*

- *Truth tables for which a consistent OBDD of size less than $2^{(0.5+\epsilon)k}$ exists.*
- *Truth tables for which all consistent OBDDs have size more than $2^{(0.75-\epsilon)k}$*

In particular, unless **NP** equals **ZPP**, no efficient approximation algorithm for the minimum consistent OBDD problem has an approximation factor of $2^{(0.25-\epsilon)k}$, for any constant $\epsilon > 0$. Somewhat weaker non-approximability results for chromatic number assuming only $\mathbf{NP} \neq \mathbf{P}$ are known [1]; these may be combined with our reduction to show similarly weaker non-approximability results for our minimum consistent OBDD problem. We omit the details.

2 The Reduction

We consider an auxiliary problem. Given a family (s_i) of truth tables $s_i : \{0,1\}^k \to \{0,1,\perp\}$ of *partially* defined Boolean functions and a family (g_i) of truth tables $g_j : \{0,1\}^k \to \{0,1\}$ of *fully* defined Boolean functions, we say that the family (g_j) *covers* the family (s_i) if for every s_i there is some g_j consistent with s_i. The *minimum truth table cover* problem is the following optimization problem: Given a family (s_i) of $n = 2^k$ truth tables of partially defined Boolean functions (represented as a collection of n strings of length 2^k over $\{0,1,\perp\}$), find the smallest family (g_j) that covers (s_i).

We present a reduction from the graph colouring problem to the minimum truth table cover problem:

Lemma 4. *There is a polynomial time computable reduction mapping undirected graphs G with $n = 2^k$ vertices for integers k to a collection of n tables of partially specified Boolean functions $s_i : \{0,1\}^k \to \{0,1,\perp\}, i = 1, \ldots, n$ so that for any integer K, G has a vertex colouring using K colours if and only if (s_i) has a truth table cover of size K.*

[1] Subsequently, the theorem was refined by Khot [8] and Engebretsen and Holmerin [4] who replaced the constant ϵ in Theorem 2 with specific subconstant functions. However, when combining inapproximability results for chromatic number with our reduction, such improvements are more or less irrelevant.

Proof. Given a graph $G = (V, E)$ with $V = \{0, \dots, n-1\}$, we define

$$s_i(j) = \begin{cases} 0 & \text{if } i \neq j \wedge (i,j) \in E; \\ 1 & \text{if } i = j; \\ \perp & \text{otherwise.} \end{cases}$$

Note that we in the definition of s_i identify the integer j with its binary representation. We shall do so in the following as well. It is an easy observation that the reduction has the desired property. $\square$

In the rest of the section, we reduce the minimum truth table cover problem to the minimum consistent OBDD problem, thus completing the proof of Theorem 1.

We need in our reduction an auxiliary family of functions $g_j^{p,m} : \{0,1\}^p \to \{0,1\}$ where p is an arbitrary non-negative integer, $1 \leq m \leq 2^{2^p}$ and $0 \leq j \leq m-1$. The family must have the following properties.

1. For fixed p, m, the functions $g_j^{p,m}, j \in \{0, \dots, m-1\}$ are all different.
2. The truth table for $g_j^{p,m}$ can be generated in time polynomial in 2^p (given the parameters p, m, j),
3. For fixed p, m, the family $(g_j^{p,m}), j \in \{0, \dots, m-1\}$ is computed by a multi-source OBDD (an OBDD with m sources, one for each member of the family) of size at most $m + 2\sqrt{m} + 3p$.

Note that the third property makes the construction of the family a bit tricky: The sources of the desired multi-source OBDD use almost its entire "node budget". We give an inductively defined construction. For $p = 0$, the construction is trivial as we must have $m = 1$ or $m = 2$. For $p > 0$ we let $q = \lceil \sqrt{m} \rceil$. Note that $q \leq 2^{2^{p-1}}$ since $\sqrt{2^{2^p}} = 2^{2^{p-1}}$ is an integer. We define for integers $i, j \in \{0, \dots, q-1\}$:

$$g_{jq+i}^{p,m}(x_1 x_2 \dots x_p) = \begin{cases} g_i^{p-1,q}(x_2 \dots x_p) & \text{if } x_1 = 0; \\ g_j^{p-1,q}(x_2 \dots x_p) & \text{if } x_1 = 1. \end{cases}$$

The construction clearly satisfies properties 1 and 2. Also, if we let $B^{p,m}$ be the size of a multi-source OBDD computing the family $(g_j^{p,m})$ we have by induction that $B^{p,m} = B^{p-1,q} + m \leq q + 2\sqrt{q} + 3(p-1) + m \leq m + 2\sqrt{m} + 3p$, so it also satisfies property 3.

We consider the values $k \geq 5$ and $n = 2^k$ fixed in the discussion to follow. For $j \in \{0, \dots, n^2 - 1\}$ we let $b_j = g_j^{k,n^2}$. By property 3 of the family (g_j^{k,n^2}), the family (b_j) is computed by a multi-source OBDD of size at most $n^2 + 3n$.

Our reduction from minimum truth table cover to the minimum consistent OBDD problem is then defined as follows. It maps the minimum truth table cover instance $\{s_i\}_{i=1,\dots,n}$, $s_i : \{0,1\}^k \to \{0,1,\perp\}$ to the truth table of the partial function $g : \{0,1\}^k \times \{0,1\}^{2k} \times \{0,1\} \times \{0,1\}^k \to \{0,1,\perp\}$ defined by:

$$g(i,j,t,z) = \begin{cases} b_j(z) & \text{if } t = 0; \\ s_i(z) & \text{if } t = 1. \end{cases} \tag{1}$$

(where we again identify integers with their binary notation). By property 2 of the family (g_j^{k,n^2}) the reduction is polynomial time computable. In the remainder of this section, we show that the composition of the reduction with the reduction of Lemma 4 has the property claimed in Theorem 1.

Lemma 5. *For any integer K, if (s_i) has a cover of size K, then g has a consistent OBDD of size at most $(2K+2)n^2 + 4n$.*

Proof. We can assume $K \leq n$. Let T be the cover. Let s_i' be a total function in T consistent with s_i. Then, a total function h consistent with g is

$$h(i,j,t,z) = \begin{cases} b_j(z) & \text{if } t = 0; \\ s_i'(z) & \text{if } t = 1. \end{cases}$$

Let us give an upper bound for the size of an OBDD computing h. For each truth table $s \in T$, there is an OBDD of size at most $n-1$ computing s (as $n-1$ is the number of decision nodes in a complete decision tree on $k = \log n$ Boolean variables). There is a multi-source OBDD computing all functions b_j of size $n^2 + 3n$ by construction. The number of different subfunctions of h of the form $(j, t, z) \to h(i_0, j, t, z)$ (for some i_0) is K, the size of the cover. Each of these subfunctions can be computed by an OBDD with an additional $2^{2k+1} - 1 = 2n^2 - 1$ nodes above the OBDDs for (s'_i) and (b_j). Having constructed OBDDs for all these subfunctions, an OBDD for h needs at most an additional $n-1$ nodes to read the first k input bits to decide which subfunction to use. Thus, h can be computed by an OBDD of size at most $(n-1) + K(2n^2-1) + K(n-1) + n^2 + 3n \leq (2K+2)n^2 + 4n$. □

Lemma 6. *Let a minimum-sized OBDD G consistent with g be given. Viewing G as a graph, the subgraph of G induced by nodes reading variables $x_{k+1}, \ldots, x_{3k}, x_{3k+1}$ (i.e. nodes reading the Boolean variables defining arguments j and t in equation (1)) forms a forest of disjoint complete binary trees (each tree containing $2^{2k+1} - 1 = 2n^2 - 1$ nodes).*

Proof. Let a minimum-sized OBDD G consistent with g be given, computing a function h. First note that since all the functions b_j are different, any OBDD consistent with g must read all variables $x_{k+1}, \ldots, x_{3k}$ on all paths through the OBDD. For the same reason, the left and right son of any node reading any variable $x_{k+1}, \ldots, x_{3k}$ must be different. Thus, the subgraph of G induced by nodes reading variables $x_{k+1}, \ldots, x_{3k}, x_{3k+1}$ is a union of complete binary trees. To prove the lemma, we just have to prove that they are disjoint. This follows if we show that any two nodes v and v' both reading a variable x_{k+m}, $m \in \{1, \ldots, 2k\}$ cannot share a son u. Assume to the contrary that they do and without loss of generality that u is a left son of v (corresponding to reading $x_{k+m} = 0$ in v).

The node v corresponds to a subfunction of h of the form $(x, t, z) \to h(a_1, c_1 \cdot x, t, z)$ for constants $a_1 \in \{0,1\}^k$ and $c_1 \in \{0,1\}^{m-1}$ and variables $x \in \{0,1\}^{2k-m+1}$, $t \in \{0,1\}$, $z \in \{0,1\}^k$. Here $c_1 \cdot x$ denotes concatenation of the bit-strings c_1 and x.

The node v' corresponds to a subfunction of h of the form $(x, t, z) \to h(a_2, c_2 \cdot x, t, z)$ for constants $a_2 \in \{0,1\}^k$ and $c_2 \in \{0,1\}^{m-1}$ and variables $x \in \{0,1\}^{2k-m+1}$, $t \in \{0,1\}$, $z \in \{0,1\}^k$.

Since u is a son of v as well as v' and all the b_j's are different we must have that $c_1 = c_2$ and that u is a *left* son of v'. Also, we must have the partial truth tables s_{a_1} and s_{a_2} are consistent, i.e., that they agree on inputs where neither has value $\perp$. Thus, we can get a smaller OBDD than G also consistent with g by removing the node v' and redirecting any incoming arc to v' to v. This contradicts G being minimum-sized. □

Lemma 7. *Assume $n > 3$. For any integer K, if g has a consistent OBDD of size at most $(2K+2)n^2 + 4n$, then (s_i) has a cover of size at most K.*

Proof. We can assume $K \leq n$. Let a minimum-sized OBDD consistent with g of size at most $(2K+2)n^2+4n$ be given, computing a function h. According to Lemma 6, the nodes reading variables $x_{k+1}, ...x_{3k+1}$ induces a collection of disjoint complete binary trees. There must be at most K trees in this collection: Otherwise the contribution of nodes from the trees would amount to at least $(2n^2-1)(K+1)$ nodes. Also, all members of the family (b_i) are subfunctions of g and since they are distinct and fully defined, each must be computed at a distinct node in the diagram, yielding n^2 additional nodes. In total, there would be at least $(2n^2-1)(K+1) + n^2$ nodes which is strictly more than $(2K+2)n^2 + 4n$ nodes.

Let (v_i) be the roots of the trees. The corresponding subfunctions of h are $(x, t, y) \to h(a_i, x, t, y)$ for constants a_i. The functions $j \to h(a_i, 0, 1, j)$ then form a cover for the family (s_i) of size at most K. □

Combining Lemma 4, Lemma 5 and Lemma 7, we have proved Theorem 1 and are done.

Acknowledgements

We would like to thank Martin Sauerhoff for detailed comments and corrections.

References

[1] Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, PCPs, and nonapproximability—towards tight results. *SIAM J. Comput.*, 27(3):804–915, 1998.

[2] R. E. Bryant. Symbolic manipulation of boolean functions using a graphical representation. In *Proceedings of the 22nd ACM/IEEE Design Automation Conference*, Los Alamitos, Ca., USA, 1985. IEEE Computer Society Press.

[3] Randal E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*, 35(8):677–691, 1986.

[4] Lars Engebretsen and Jonas Holmerin. Towards optimal lower bounds for clique and chromatic number. *Theoret. Comput. Sci.*, 299(1-3):537–584, 2003.

[5] Uriel Feige and Joe Kilian. Zero knowledge and the chromatic number. *J. Comput. System Sci.*, 57(2):187–199, 1998.

[6] Michael R. Garey and David S. Johnson. *Computers and intractability.* W. H. Freeman and Co., San Francisco, Calif., 1979.

[7] Kouichi Hirata, Shinichi Shimozono, and Ayumi Shinohara. On the hardness of approximating the minimum consistent OBDD problem. In *Algorithm theory—SWAT '96 (Reykjavík, 1996)*, volume 1097 of *Lecture Notes in Comput. Sci.*, pages 112–123. Springer, Berlin, 1996.

[8] Subhash Khot. Improved inapproximability results for MaxClique, chromatic number and approximate graph coloring. In *42nd IEEE Symposium on Foundations of Computer Science (Las Vegas, NV, 2001)*, pages 600–609. IEEE Computer Soc., 2001.

[9] J.C. Kiefer, P. Flajolet, and E.-H Yang. Data compression via binary decision diagrams. In *Proc. of the 2000 IEEE Intern. Symp. Inform. Theory, Sorrento, Italy, June 25–30*, page 296. 2000.

[10] Jesper Torp Kristensen. Generation and compression of endgame tables in chess with fast random access using OBDDs. Master's thesis, University of Aarhus, Department of Computer Science, 2005.

[11] Martin Sauerhoff and Ingo Wegener. On the complexity of minimizing the OBDD size for incompletely specified functions. *IEEE Transactions on Computer-Aided Design*, 15:435–1437, 1996.

[12] Shinichi Shimozono, Kouichi Hirata, and Ayumi Shinohara. On the hardness of approximating the minimum consistent acyclic DFA and decision diagram. *Inform. Process. Lett.*, 66(4):165–170, 1998.

[13] Ingo Wegener. *Branching programs and binary decision diagrams.* SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics (SIAM), 2000.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'