Exponential Lower Bounds on the Space Complexity of OBDD-Based Graph Algorithms*

Daniel Sawitzki**

University of Dortmund, Computer Science 2, D-44221 Dortmund, Germany
daniel.sawitzki@cs.uni-dortmund.de

Abstract. Ordered Binary Decision Diagrams (OBDDs) are a data structure for Boolean functions which is successfully applied in many areas like Integer Programming, Model Checking, and Relational Algebra. Nevertheless, many basic graph problems like Connectivity, Reachability, Single-Source Shortest-Paths, and Flow Maximization are known to be PSPACE-hard if their input graphs are represented by OBDDs. This holds even for input OBDDs of constant width. We extend these results by concrete exponential lower bounds on the space complexity of OBDD-based algorithms for the Reachability Problem, the Single-Source Shortest-Paths Problem, and the Maximum Flow Problem. This involves the first exponential lower bound on the OBDD size for the highest bit of Integer Multiplication w. r. t. the natural interleaved variable order.

1 Introduction

Algorithms on (weighted) graphs G with node set V and edge set $E \subseteq V^2$ typically work on adjacency lists of size $\Theta(|V| + |E|)$ or on adjacency matrices of size $\Theta(|V|^2)$. But in areas like CAD, Model Checking, and Relational Algebra graphs arise whose size does not allow an explicit enumeration of all their elements. There [2, 10] and in further areas like Algorithmic Learning [7] and Integer Programming [1], the implicit representation of data by Ordered Binary Decision Diagrams (OBDDs) [3, 4, 18] is well-established as a succinct alternative. Their convenient algorithmic properties help to save time and space through solving problems by efficient logical operations. So OBDDs are applied in heuristic methods with hopefully sublinear resource usage.

Though each single OBDD manipulation is always efficient, a short sequence of them may suffice to cause an exponential blow-up in the OBDD size. Most algorithms on OBDD-represented graphs have only been analyzed experimentally [11, 12, 21] or w. r. t. rough measures like the number of OBDD manipulations [8, 9, 13]. Feigenbaum et al. [6] showed that even the very basic problem of s–t-Connectivity is PSPACE-complete on OBDD-represented graphs. That is, the success of OBDD-based approaches has to be explained by means of advantageous properties of real-world instances causing an essentially better behavior

* An extended version of this paper is available at http://ls2-www.cs.uni-dortmund.de/~sawitzki/.

** Supported by DFG grant We 1066/10-2.

J.R. Correa, A. Hevia, and M. Kiwi (Eds.): LATIN 2006, LNCS 3887, pp. 781–792, 2006.

than in the worst case. Recent research tries to build theoretical foundations for the analysis of the over-all runtime of OBDD-based algorithms [15, 16, 20]. This includes the investigation of the parameterized complexity of graph problems when structural properties of input and/or output OBDDs are considered as fixed parameters [17]. So basic graph problems are known to be fixed-parameter intractable w. r. t. a fixed input OBDD width (unless P=PSPACE). (The book of Downey and Fellows [5] gives a comprehensive introduction to the field of parameterized complexity.)

Despite these hardness results, there is no nontrivial lower bound for the complexity of any problem on OBDD-represented instances so far. The challenge is to prove both an upper bound on the input's OBDD size and an exponentially larger lower bound on the size of some OBDD occurring during the computation. We present such bounds for the Single-Source Shortest-Paths Problem and the Maximum Flow Problem. For the Reachability Problem this succeeds only for a certain class of OBDD-based algorithms. However, all existing algorithms known to the author belong to this class. We do not assume a separate output tape because the separation of working space and output size is not reasonable in practical applications.

The paper is organized as follows: After giving foundations on OBDDs in Section 2, we sum up both known and some trivial new results on the complexity of graph problems on OBDD-represented instances in Section 3. In Section 4, we introduce a construction method for functions with constant OBDD width. With these preliminaries we are able to construct pathological instances for all three considered graph problems in Sections 5, 6, and 7 giving us the desired exponential lower bounds. Due to space limitations, the technical lower bound on the OBDD size of the highest bit of multiplication has been shifted into the extended version of the paper. Finally, Section 8 gives conclusions on the work.

2 Ordered Binary Decision Diagrams

For $\mathbb{B} := \{0, 1\}$, let us denote the ith character of a binary string $x \in \mathbb{B}^n$ by x_i and let $|x| := \sum_{i=0}^{n-1} x_i 2^i$ identify its value. The class of Boolean functions $f \colon \{0,1\}^n \to \{0,1\}$ is denoted by B_n. The set of all permutations of n elements is denoted by Σ_n.

A function $f \in B_n$ defined on variables $x_0, \ldots, x_{n-1}$ can be represented by an *Ordered Binary Decision Diagram (OBDD)* [3, 4]. An OBDD $\mathcal{G}$ is a directed acyclic graph consisting of *internal nodes* and *sink nodes*. Each internal node is labeled with a Boolean variable x_i, while each sink node is labeled with a Boolean constant. Each internal node is left by two edges one labeled 0 and the other 1. A *function pointer* p marks a special node that represents f. Moreover, a permutation $\pi \in \Sigma_n$ called *variable order* must be respected by the internal nodes' labels on every path from p to a sink. For a given variable assignment $\alpha \in \mathbb{B}^n$, we compute the function value $f(\alpha)$ by traversing $\mathcal{G}$ from p to a sink labeled $f(\alpha)$ while leaving each node labeled x_i via its α_i-edge. An OBDD with variable order π is called π-OBDD. The minimal-size π-OBDD for a function

$f \in B_n$ is known to be canonical and will be denoted by π-OBDD$[f]$. Its *size* size(π-OBDD$[f]$) is measured by the number of its nodes; its *width* is the maximum number of inner nodes labeled with the same variable. Finding an optimal variable order leading to the minimum size OBDD for a given function is known to be NP-hard. Independent of π, it is size(π-OBDD$[f]$) $\leq \big(2 + o(1)\big)2^n/n$ for any $f \in B_n$. The book of Wegener [18] gives a comprehensive survey on different types of Binary Decision Diagrams.

Efficient Algorithms on OBDDs. Functional operations on OBDD-represented functions can be implemented by efficient algorithms called *OBDD operations* in the following. The satisfiability of f can be decided in time $\mathcal{O}(1)$. The negation $\overline{f}$, the replacement of a variable x_i by some constant c (i. e., $f_{|x_i=c}$), and computing $|f^{-1}(1)|$ are possible in time $\mathcal{O}\big(\text{size}(\pi\text{-OBDD}[f])\big)$. The set $f^{-1}(1)$ of f's minterms can be obtained in time $\mathcal{O}\big(n \cdot |f^{-1}(1)|\big)$. Whether two functions f and g are equivalent (i. e., $f = g$) can be decided in time $\mathcal{O}\big(\text{size}(\pi\text{-OBDD}[f]) + \text{size}(\pi\text{-OBDD}[g])\big)$. The most important OBDD operation is the *binary synthesis* $f \otimes g$ for $f, g \in B_n$, $\otimes \in B_2$ (e. g., $\wedge$, $\vee$); in general, it produces the result π-OBDD$[f \otimes g]$ in time and space $\mathcal{O}\big(\text{size}(\pi\text{-OBDD}[f]) \cdot \text{size}(\pi\text{-OBDD}[g])\big)$. The synthesis is also used to implement *quantifications* $(\mathcal{Q}x_i)f$ for $\mathcal{Q} \in \{\exists, \forall\}$. Hence, computing π-OBDD$\big[(\mathcal{Q}x_i)f\big]$ takes time $\mathcal{O}\big(\text{size}^2(\pi\text{-OBDD}[f])\big)$ in general. All operations produce minimum size π-OBDDs.

Representing Graphs by OBDDs. One canonical way to represent data implicitly by an OBDD is to express it in terms of a subset $S \subseteq \{0, \ldots, N-1\}^k$, where N depends on the data size and k is constant. Assuming w. l. o. g. $N = 2^n$, S can be represented by an OBDD for the *characteristic function* $\chi_S \in B_{kn}$ of S defined by $\chi_S(x^{(1)}, \ldots, x^{(k)}) = 1 :\Leftrightarrow (|x^{(1)}|, \ldots, |x^{(k)}|) \in S$, where $x^{(1)}, \ldots, x^{(k)} \in \mathbb{B}^n$. Correspondingly, a digraph $G = (V, E)$ with nodes $v_0, \ldots, v_{N-1}$ can be represented by $\chi_G \in B_{2n}$ with $\chi_G(x, y) = 1 :\Leftrightarrow (v_{|x|}, v_{|y|}) \in E$ for $x, y \in \mathbb{B}^n$. Undirected graphs are then considered as digraphs with symmetric edges. If G's edges are weighted by $c\colon E \to \{0, \ldots, B\}$ with maximum weight B, we extend the definition to $\chi_G(x, y, a) = 1 :\Leftrightarrow (v_{|x|}, v_{|y|}) \in E \wedge c(v_{|x|}, v_{|y|}) = |a|$. In the context of characteristic functions, one further functional operation is of interest: the argument reordering.

Definition 1. *Let* $\rho \in \Sigma_k$ *and* $f \in B_{kn}$ *be defined on variable vectors* $x^{(1)}, \ldots, x^{(k)} \in \mathbb{B}^n$. *The* argument reordering $\mathcal{R}_\rho(f) \in B_{kn}$ *w. r. t.* ρ *is defined by* $\mathcal{R}_\rho(f)\big(x^{(1)}, \ldots, x^{(k)}\big) = f\big(x^{(\rho(1))}, \ldots, x^{(\rho(k))}\big)$.

In order to enable efficient argument reorderings (see Lemma 3 and Theorem 1), it is common to use k-interleaved variable orders.

Definition 2. *The* k-interleaved *variable order* $\pi^\tau_{k,n} \in \Sigma_{kn}$ *of* k *variable vectors* $x^{(1)}, \ldots, x^{(k)} \in \mathbb{B}^n$ *reads bits of same significance index en bloc:*

$$\pi^\tau_{k,n} := \left(x^{(1)}_{\tau(0)}, \ldots, x^{(k)}_{\tau(0)}, x^{(1)}_{\tau(1)}, \ldots, x^{(k)}_{\tau(1)}, \ldots\ldots, x^{(k)}_{\tau(n-1)}\right),$$

where τ *is the local order of each* $x^{(1)}, \ldots, x^{(k)}$. *The order* $\pi^{\text{id}}_{k,n}$ *is called* natural.

3 Survey of Previous and New Results

Even the very basic problem of deciding whether two nodes s and t are connected in a digraph $G = (V, E)$ is known to be PSPACE-complete if the input graph is represented by π-OBDD$[\chi_G]$ (see [6]). In [17], this result is extended to fixed-parameter intractability results for a variety of fundamental graph problems including Connectivity, Bipartiteness, Planarity, Acyclicity, Single-Source Shortest-Paths, and Flow Maximization. This is for the fixed parameter of the input graph's OBDD width and under the assumption P $\neq$ PSPACE. Unless stated otherwise, we assume in the following that input and, if required, output are represented by OBDDs.

With similar techniques, it can be proved that these problems as well as computing minimum spanning trees on OBDD-represented graphs remain fixed parameter intractable when the maximum of input and output OBDD width is considered as fixed parameter. That is, even if the characteristic functions of input *and* output have OBDDs of constant width and, therefore, size $\mathcal{O}(\log |V|)$, intermediately generated OBDDs may still be superpolynomially larger, unless P=PSPACE. Because these negative results rely on the compactness of OBDD-represented configuration graphs of Turing machines, they directly carry over to more general branching program models. Interestingly, the situation changes when the edge weight zero is forbidden. Then, the All-Pairs Shortest-Paths Problem can be solved in polynomial time if certain width restrictions apply to input and output OBDD [15]. The prefix "PW-" indicates the restriction to positive weights (PW-APSP and PW-SSSP).

On the other hand, all decision problems mentioned so far can be trivially solved in space poly$(\log |V|)$ by a nondeterministic Turing machine using χ_G as oracle. Each oracle request can be implemented by an OBDD evaluation operation. Together with the reasonable assumption size$(\pi$-OBDD$[\chi_G]) \geq \log_2 |V|$ and the fact NPSPACE=PSPACE, we conclude that these problems can be solved in polynomial space w. r. t. size$(\pi$-OBDD$[\chi_G])$. But what about search problems?

If the output OBDD has polynomial size w. r. t. the input size size$(\pi$-OBDD$[\chi_G])$, we can enumerate all potential output OBDDs of polynomial size and verify the result in polynomial space. Without bounding the output OBDD size, this is not possible: We prove in this paper that constant input OBDD width does not suffice for polynomial space complexity of the Single-Source Shortest-Paths Problem, the Maximum Flow Problem, and the restricted reachability problem Reachability*. That is, their space complexity is not fixed-parameter tractable w. r. t. the parameter of input OBDD width.

Table 1 gives an overview of the state of affairs w. r. t. eight complexity classes named α-β-γ for $\alpha \in \{\mathrm{I}, \mathrm{IO}\}$, $\beta \in \{\mathrm{FPT}, \mathrm{P}\}$, and $\gamma \in \{\mathrm{T}, \mathrm{S}\}$. Component α indicates whether the complexity is related to input (I) or input *and* output OBDDs (IO); β separates fixed-parameter (FPT) from polynomial (P) complexities; γ separates time (T) from space (S) complexity. The classes are related as follows:

$$\text{I-}\beta\text{-T} \begin{array}{c} \subseteq \text{IO-}\beta\text{-T} \subsetneq \\ \subseteq \text{I-}\beta\text{-S} \subsetneq \end{array} \text{IO-}\beta\text{-S}, \qquad \alpha\text{-P-}\gamma \subseteq \alpha\text{-FPT-}\gamma\ .$$

Table 1. The complexity of graph problems on OBDD-represented inputs, unless P=PSPACE. The IO case is left out for decision problems. Results from [15, 17] are marked with daggers. The main contributions of this paper are starred.

	I-FPT-T	IO-FPT-T	I-FPT-S	IO-FPT-S	I-P-T	IO-P-T	I-P-S	IO-P-S
MaxFlow	no †	no	no ⋆	yes	no †	no	no ⋆	yes
APSP	no †	no	no ⋆	yes	no †	no	no ⋆	yes
PW-APSP	no †	yes †	no ⋆	yes †	no †	?	no ⋆	yes
SSSP	no †	no	no ⋆	yes	no †	no	no ⋆	yes
PW-SSSP	no †	?	no ⋆	yes	no †	?	no ⋆	yes
Reachability*	no	no	no ⋆	yes	no	no	no ⋆	yes
TransClos*	no	no	no ⋆	yes	no	no	no ⋆	yes
MST	no	no	?	yes	no	no	?	yes
s–t-Conn.	no †	-	yes	-	no †	-	yes	-
Connected	no †	-	yes	-	no †	-	yes	-
Bipartite	no †	-	yes	-	no †	-	yes	-
Acyclic	no †	-	yes	-	no †	-	yes	-
Euler Cycle	no †	-	yes	-	no †	-	yes	-

4 Constructing Functions with Constant OBDD Width

The pathological graph instances constructed in the following sections will have constant OBDD width w. r. t. natural interleaved variable orders. This section supplies a convenient construction method for functions with constant OBDD width. Actually we generate OBDDs with constant complete-OBDD width.

Definition 3. *An OBDD for $f \in B_n$ is called* complete *if every path from its function pointer to a sink has length n.*

That is, complete OBDDs are not allowed to skip variable tests. The minimal-size complete π-OBDD for $f \in B_n$ is also known to be canonical [18] and will be denoted by $\pi\text{-OBDD}_c[f]$ in the following.

Definition 4. *The* complete-OBDD width *of a function $f \in B_n$ w. r. t. a variable order $\pi \in \Sigma_n$ is the width of* $\pi\text{-OBDD}_c[f]$.

So it is $\text{size}(\pi\text{-OBDD}[f]) \leq \text{size}(\pi\text{-OBDD}_c[f]) = \mathcal{O}(nw)$ for any $f \in B_n$ with complete-OBDD width w and variable order π. On the other hand, it is $\text{size}(\pi\text{-OBDD}_c[f]) \leq n \cdot \text{size}(\pi\text{-OBDD}[f])$ (see, e. g., [18]).

The basic building blocks of the construction technique are multivariate threshold functions [20].

Definition 5. *Let $f \in B_{kn}$ be defined on variable vectors $x^{(1)}, \dots, x^{(k)} \in \mathbb{B}^n$. A function f is called k*-variate threshold function *iff there are $W \in \mathbb{N}$, $T \in \mathbb{Z}$, and $\delta_1, \dots, \delta_k \in \{-W, \dots, W\}$ such that*

$$f\left(x^{(1)}, \dots, x^{(k)}\right) = \left(\sum_{i=1}^{k} \delta_i \cdot \left|x^{(i)}\right| \geq T\right) .$$

The corresponding class of functions is denoted by $\mathbb{T}^W_{k,n}$.

Clearly, each of the relations $>$, $\leq$, $<$, and $=$ can be composed by binary syntheses of a constant number of multivariate threshold functions.

Lemma 1 ([20]). *Functions* $f \in \mathbb{T}^W_{k,n}$ *have complete OBDDs of width* $\mathcal{O}(k^2W)$ *using the variable order* $\pi^{\mathrm{id}}_{k,n} \in \Sigma_{kn}$.

That is, for $k, W = \mathcal{O}(1)$ multivariate threshold functions have constant complete-OBDD width. Moreover, both critical OBDD operations that may increase the OBDD size preserve constant complete-OBDD width (proved in the paper's extended version):

Let $f_1, f_2 \in B_n$ be defined on variables $x_0, \ldots, x_{n-1} \in \mathbb{B}$; assume f_1 resp. f_2 has complete-OBDD width w_1 resp. w_2 w.r.t. $\pi \in \Sigma_n$.

Lemma 2. *The binary synthesis result* π-OBDD$[f_1 \otimes f_2]$, $\otimes \in B_2$, *has a complete-OBDD width of at most* $w_1 w_2$.

Let $f_3 \in B_{kn}$ be defined on variable vectors $x^{(1)}, \ldots, x^{(k)} \in \mathbb{B}^n$; assume f_3 has complete-OBDD width w_3 w.r.t. $\pi^{\tau}_{k,n}$, $\tau \in \Sigma_n$. Let $\rho \in \Sigma_k$.

Lemma 3. *The argument reordering result* $\mathcal{R}_\rho(f_3)$ *of* f_3 *w.r.t.* ρ *has a complete-OBDD width of at most* $w_3 3^k$.

We conclude that a constant number of operations increases the complete-OBDD width independently of n.

Theorem 1. *Let* $x^{(1)}, \ldots, x^{(k)} \in \mathbb{B}^n$, *k constant. Let* $\mathcal{S}$ *be a sequence of* $\mathcal{O}(1)$ *operations as introduced in Section 2 applied to functions from* $\mathbb{T}^W_{k,n}$ *defined on* $x^{(1)}, \ldots, x^{(k)}$ *and to intermediate results generated by the current prefix of* $\mathcal{S}$.

Each function generated by $\mathcal{S}$ *has complete-OBDD width* $\beta(W)$ *w.r.t.* $\pi^{\mathrm{id}}_{k,n}$ *for some appropriate function* $\beta\colon \mathbb{N} \to \mathbb{N}$ *independent of* n.

5 Single-Source Shortest-Paths

The previous section has enabled us to construct functions with constant OBDD width starting from simply structured multivariate threshold functions. We now have to use this framework to generate instances for graph problems with an exponential gap between the input and output OBDD size. At first, we consider the *Single-Source Shortest-Paths Problem* on a weighted graph $G = (V, E, c)$ as introduced in Section 2. Let D be the set of all solution pairs $(v, d) \in V \times \mathbb{N}$ such that a shortest s–v-path $(s, \ldots, v) =: P$ has weight $d := \sum_{e \in P} c(e)$. The input for the problem's OBDD-based version consists of π-OBDD$[\chi_G]$ for some $\pi \in \Sigma_{kn}$ and a source node $s \in V$; the output is π-OBDD$[\chi_D]$ for D's characteristic function χ_D. That is, we even restrict ourselves to computing only the *costs* of shortest s–v-paths.

We define a sequence $(G_m)_m$ of pathological graph instances with solution sets D_m. $G_m := (V_m, E_m, c_m)$ consists of 2^m components H_i, $0 \leq i < 2^m$. Each

H_i is a path of 2^m nodes $w_{i,0},\ldots,w_{i,2^m-1}$ with edges $(w_{i,j},w_{i,j+1})$ of weight i for $0 \le i < 2^m$. Moreover, a common source node s is connected to all H_is by edges $(s,w_{i,0})$ of weight 0. So the path $(s,\ldots,w_{i,j})$ has weight ij and it is $(w_{i,j},ij) \in D_m$. We add $2^{2(m+1)} - (2^{2m}+1)$ dummy singletons and number the nodes in $V_m := \{v_0,\ldots,v_{2^{2(m+1)}-1}\}$ by $w_{i,j} := v_{i2^{m+1}+j}$ and $s := v_{2^{2m+1}}$.

Claim. The function χ_{G_m} has a complete $\pi^{\mathrm{id}}_{5,m+1}$-OBDD of size $\mathcal{O}(m)$ with constant width independent of m.

Proof. We express $\chi_{G_m} \in B_{5(m+1)}$ in terms of Theorem 1. For $x \in \mathbb{B}^{2(m+1)}$ let $i(x) := x_{2m+1}\ldots x_{m+1}$, and $j(x) := x_m \ldots x_0$. We begin with a tentative function $\chi^*_{G_m}$.

$$\begin{aligned}\chi^*_{G_m}(x,y,a) := &\left[(|x| = 2^{2m+1} \neq |y|) \wedge (|j(y)| = |a| = 0)\right]\\ &\vee \left[(|x| \neq 2^{2m+1} \neq |y|) \wedge (|i(x)| = |i(y)| = |a|) \wedge (|j(y)| = |j(x)|+1)\right] \quad (1)\end{aligned}$$

for node numbers $x,y \in \mathbb{B}^{2(m+1)}$ and a weight encoding $a \in \mathbb{B}^{m+1}$. This definition does not take care yet of dummy singletons occurring due to the node numbering. Hence, let $\chi_{V_m} \in B_{2(m+1)}$ be defined by

$$\chi_{V_m}(x) := (|x| = 2^{2m+1}) \vee (|i(x)|, |j(x)| < 2^m) \quad (2)$$

which is 1 exactly for all nondummy nodes. We finally have χ_{G_m} with

$$\chi_{G_m}(x,y,a) := \chi_{V_m}(x) \wedge \chi_{V_m}(y) \wedge \chi^*_{G_m}(x,y,a)\ . \quad (3)$$

Each comparison in (1)–(3) can be realized by $\mathcal{O}(1)$ functions from $\mathbb{T}^{\mathcal{O}(1)}_{5,m+1}$. So Theorem 1 applies and $\pi^{\mathrm{id}}_{5,m+1}$-OBDD$_\mathrm{c}[\chi_{G_m}]$ has constant width. □

Having proved that G_m has compact complete OBDDs of constant width, it remains to show that the output χ_{D_m} has exponential OBDD size.

Claim. Every OBDD for χ_{D_m} has exponential size w. r. t. m.

Proof. Assume w. l. o. g. that m is even. Replacing some variables of a Boolean function by constant values does not enlarge the corresponding π-OBDD. So we show a lower bound on a subfunction $f_m \in B_{2m}$ of χ_{D_m} which is obtained by replacing $x_{2m+1},\ldots,x_{m+m/2+1}$ and $x_m,\ldots,x_{m/2}$ by 0. Hence, argument x represents a $w_{i,j}$ node with $i,j < 2^{m/2}$ and it is $f_m(x,a) = 1 \Leftrightarrow |i(x)| \cdot |j(x)| = |a|$, where $|a| < 2^m$. So f_m is the *Graph of m/2-bit Integer Multiplication* whose π-OBDD size is bounded below by $2^{m/1536-1}$ for any variable order π (proved in [16]). □

The claims in this section imply the result on the space complexity of SSSP.

Theorem 2. *The Single-Source Shortest-Paths Problem on OBDD-represented graphs has exponential space complexity, even for instances with constant complete-OBDD width. This implies* SSSP $\notin$ I-FPT-S.

By further variable replacements, the single-source variant can be trivially reduced to the all-pairs variant as defined in [15]. In Section 3, the special role of the edge weight zero has been mentioned. Though G_m contains such edges they can be avoided by constructions which are a little more complicated.

Proposition 1. PW-SSSP, PW-APSP $\notin$ I-FPT-S.

6 Maximum Flow

We continue with an exponential lower bound on the space complexity of the OBDD-based *Maximum Flow Problem.* Again, the input is an OBDD π-OBDD$[\chi_G]$ representing a weighted graph instance G. We want to compute a maximum flow $\phi\colon E \to \{0,\dots,B\}$ from a source $s \in V$ to a terminal $t \in V$. This time, the edge weights represent capacities. The flow ϕ must respect $\phi(e) \le c(e)$ as well as $\sum_{e=(u,v)} \phi(e) = \sum_{e=(v,w)} \phi(e)$ for each $v \in V$. The final output is π-OBDD$[\chi_F]$ for the solution set $F := \{(v,w,d) : \phi(v,w) = d\}$.

We define a sequence $(G_m)_m$ of pathological graph instances with unique maximum flows ϕ_m and solution sets F_m. $G_m := (V_m, E_m, c_m)$ consists of 2^{2m} components $H_{i,j}$, $0 \le i,j < 2^m$. Each $H_{i,j}$ consists of $j+2$ nodes $s_{i,j}$, $t_{i,j}$, and $w_{i,j,\ell}$ for $0 \le \ell < j$. $H_{i,j}$ contains $2j$ edges $(s_{i,j}, w_{i,j,\ell})$ and $(w_{i,j,\ell}, t_{i,j})$ with capacity i. The global source s and terminal t are connected to all components $H_{i,j}$ by edges $(s, s_{i,j})$ and $(t_{i,j}, t)$ of capacity 2^{2m}. Obviously, ϕ_m sends $i \cdot j$ units of flow through each $H_{i,j}$ and it is $(s, s_{i,j}, ij) \in F_m$.

The nodes in $V_m := \{v_0, \dots, v_{2^{3(m+1)}-1}\}$ are numbered by $w_{i,j,\ell} := v_{i2^{2(m+1)}+j2^{m+1}+\ell}$, $s_{i,j} := v_{i2^{2(m+1)}+j2^{m+1}+2^m}$, $t_{i,j} := v_{i2^{2(m+1)}+j2^{m+1}+2^m+1}$, $s := v_{2^{2m+1}}$, and $t := v_{2^{2m+1}+1}$.

Claim. Function χ_{G_m} has a complete $\pi^{\mathrm{id}}_{8,m+1}$-OBDD of size $\mathcal{O}(m)$ with constant width independent of m.

Proof. We express $\chi_{G_m} \in B_{8(m+1)}$ in terms of Theorem 1. For $x \in \mathbb{B}^{3(m+1)}$ let $i(x) := x_{3m+2} \dots x_{2m+2}$, $j(x) := x_{2m+1} \dots x_{m+1}$, and $\ell(x) := x_m \dots x_0$. Again we begin with a tentative function $\chi^*_{G_m}$ reflecting all four types of edges.

$$\begin{aligned} \chi^*_{G_m}(x,y,a,b) := {} & \big[(|x| = 2^{2m+1}) \wedge (v_{|y|} \in \{s_{i,j} : i,j\}) \wedge (|ab| = 2^{2m})\big] \\ & \vee \big[(v_{|x|} \in \{t_{i,j} : i,j\}) \wedge (|y| = 2^{2m+1}+1) \wedge (|ab| = 2^{2m})\big] \\ & \vee \big[(v_{|x|} \in \{s_{i,j}\}_{i,j}) \wedge (v_{|y|} \in \{w_{i,j,\ell}\}_{i,j,\ell}) \\ & \wedge (|i(x)| = |i(y)| = |ab|) \wedge (|\ell(y)| < |j(x)| = |j(y)|)\big] \\ & \vee \big[(v_{|x|} \in \{w_{i,j,\ell}\}_{i,j,\ell}) \wedge (v_{|y|} \in \{t_{i,j}\}_{i,j}) \\ & \wedge (|i(x)| = |i(y)| = |ab|) \wedge (|\ell(x)| < |j(x)| = |j(y)|)\big] \end{aligned} \quad (4)$$

for node numbers $x, y \in \mathbb{B}^{3(m+1)}$ and a weight encoding ab consisting of concatenated components $a, b \in \mathbb{B}^{m+1}$. This definition does not take care yet of dummy singletons occurring due to the node numbering. Hence, let $\chi_{V_m} \in B_{3(m+1)}$ be defined by

$$\chi_{V_m}(x) := (v_{|x|} \in \{s, t, s_{i,j}, w_{i,j,\ell}, t_{i,j} : i, j, \ell\}) \ . \tag{5}$$

We finally have χ_{G_m} with

$$\chi_{G_m}(x, y, a) := \chi_{V_m}(x) \wedge \chi_{V_m}(y) \wedge \chi^*_{G_m}(x, y, a) \ . \tag{6}$$

Each comparison in (4)–(6) can be realized by $\mathcal{O}(1)$ functions from $\mathbb{T}^{\mathcal{O}(1)}_{8,m+1}$. This holds also for type checks like $v_{|x|} \in \{s_{i,j} : i, j\} \Leftrightarrow (|i(x)|, |j(x)| < 2^m) \wedge (|\ell(m)| = 2^m)$. Comparisons with the concatenation $|ab|$ can be broken down to $\mathcal{O}(1)$ comparisons with both parts $|a|$ and $|b|$. So Theorem 1 applies and $\pi^{\text{id}}_{8,m+1}$-OBDD$_\text{c}[\chi_{G_m}]$ has constant width. □

Claim. Every OBDD for χ_{F_m} has exponential size w. r. t. m.

Proof. We show a lower bound on a subfunction $f_m \in B_{4m}$ of the solution χ_{F_m} which is obtained by replacing x by the source number 2^{2m+1}, y_{3m+2} and y_{2m+1} by 0, and $|\ell(y)|$ by 2^m. Hence, argument y represents an $s_{i,j}$ node with $i, j < 2^m$. The maximum flow ϕ sends $i \cdot j$ flow units through $(s, s_{i,j})$ and it is $f_m(y, a, b) = 1 \Leftrightarrow |i(y)| \cdot |j(y)| = |ab|$. So f_m is the Graph of m-bit Integer Multiplication whose π-OBDD size is bounded below by $2^{m/768-1}$ for any variable order π (see [16]). □

The claims in this section imply the result on the space complexity of MaxFlow.

Theorem 3. MaxFlow $\notin$ I-FPT-S.

7 Reachability

Computing the set R of nodes that are reachable from some source $s \in V$ in a digraph $G = (V, E)$ is an important problem in CAD and Model Checking (see, e. g., [18–Chapters 13.2 and 13.3]). Let G be defined as in Section 2. In the OBDD-based setting, we want to compute the characteristic function χ_R of the solution set $R \subseteq V$. There are both BFS-like approaches with $\Omega(|V|)$ OBDD operations [11] as well as iterative squaring methods with $\mathcal{O}(\log^2 |V|)$ operations [14]. All popular algorithms known to the author achieve this by iteratively increasing the length of considered paths. This involves computing intermediate subfunctions χ_{R_p} with $\chi_{R_p}(x) = 1$ iff s and $v_{|x|}$ are connected by a directed path not longer than 2^p for $p \in \{1, \ldots, \lfloor \log_2 |V| \rfloor\}$. So we denote the problem of computing $\{\chi_{R_p}, \chi_R\}_p$ by *Reachability**. Moreover, we assume that the variable order is not changed during the algorithm.

We construct instances $(G_m)_m$ with constant complete-OBDD width whose intermediate result $R_{m,p}$ has exponential OBDD size for some maximum path length 2^p. Each $G_m := (V_m, E_m)$ consists of 2^{2m} components $H_{i,j}$, $0 \le i, j < 2^m$. Each $H_{i,j}$ is the concatenation $P_{i-1} \ldots P_0$ of paths $P_\ell := (w_{i,j,\ell,j-1}, \ldots, w_{i,j,\ell,0})$. P_ℓ is concatenated to $P_{\ell-1}$ by $(w_{i,j,\ell,0}, w_{i,j,\ell-1,j-1})$. Moreover, a common source node s is connected to all $H_{i,j}$s by edges $(s, w_{i,j,i-1,j-1})$. The nodes in $V_m := \{v_0, \ldots, v_{2^{4(m+1)}-1}\}$ are numbered by $w_{i,j,\ell,r} := v_{i2^{3(m+1)}+j2^{2(m+1)}+\ell 2^{m+1}+r}$ and $s := v_{2^{4m+3}}$.

Claim. Function χ_{G_m} has a complete $\pi^{\mathrm{id}}_{8,m+1}$-OBDD of size $\mathcal{O}(m)$ with constant width independent of m.

Proof. We express $\chi_{G_m} \in B_{8(m+1)}$ in terms of Theorem 1. For $x \in \mathbb{B}^{4(m+1)}$ let $i(x) := x_{4m+3} \ldots x_{3m+3}$, $j(x) := x_{3m+2} \ldots x_{2m+2}$, $\ell(x) := x_{2m+1} \ldots x_{m+1}$, and $r(x) := x_m \ldots x_0$. Again we begin with a tentative function $\chi^*_{G_m}$.

$$\begin{aligned}\chi^*_{G_m}(x,y) := &\left[(|x| = 2^{4m+3} \neq |y|) \wedge (|\ell(y)| = |i(y)| - 1) \wedge (|r(y)| = |j(y)| - 1)\right]\\ &\vee \left[(|x| \neq 2^{2m+1} \neq |y|) \wedge (|i(x)| = |i(y)|) \wedge (|j(x)| = |j(y)|)\right.\\ &\left.\wedge\, (|\ell(x)| = |\ell(y)| < |i(x)|) \wedge (|j(x)| > |r(x)| = |r(y)| + 1)\right]\\ &\vee \left[(|x| \neq 2^{2m+1} \neq |y|) \wedge (|i(x)| = |i(y)|) \wedge (|j(x)| = |j(y)|\right.\\ &\left.\wedge\, (|i(x)| > |\ell(x)| = |\ell(y)| + 1) \wedge (|r(x)| = 0) \wedge (|r(y)| = |j(y)| - 1)\right] \qquad (7)\end{aligned}$$

for node numbers $x, y \in \mathbb{B}^{4(m+1)}$. This definition does not take care yet of dummy singletons occurring due to the node numbering. Hence, let $\chi_{V_m} \in B_{4(m+1)}$ be defined by

$$\chi_{V_m}(x) := (v_{|x|} = 2^{4m+3}) \vee (|i(x)|, |j(x)|, |\ell(x)|, |r(x)| < 2^m) \qquad (8)$$

which is 1 exactly for all nondummy nodes. We finally have χ_{G_m} with

$$\chi_{G_m}(x,y,a) := \chi_{V_m}(x) \wedge \chi_{V_m}(y) \wedge \chi^*_{G_m}(x,y,a)\ . \qquad (9)$$

Each comparison in (7)–(9) can be realized by $\mathcal{O}(1)$ functions from $\mathbb{T}^{\mathcal{O}(1)}_{8,m+1}$. So Theorem 1 applies and $\pi^{\mathrm{id}}_{8,m+1}$-OBDD$_\mathrm{c}[\chi_{G_m}]$ has constant width. □

Having proved that G_m has compact complete OBDDs of constant width, it remains to show that for some appropriate p the function $\chi_{R_{m,p}}$ has exponential OBDD size. We first consider the ith bit of n-bit Integer Multiplication.

Definition 6. *Let $x, y \in \mathbb{B}^n$. The ith bit of n-bit Integer Multiplication $\mathrm{MULT}_{n,i} \in B_{2n}$ on variables x, y is defined to be the ith bit of $|x| \cdot |y|$.*

There are well-known exponential lower bounds on the OBDD-size of the middle bit $\mathrm{MULT}_{n,n-1}$ (see [19]). The π-OBDD size of the highest bit $\mathrm{MULT}_{n,2n-1}$ for any nontrivial variable order $\pi \in \Sigma_{2n}$ has been open so far [18–Problem 4.12].

Theorem 4. *The size of $\pi^{\mathrm{id}}_{2,n}$-OBDD$[\mathrm{MULT}_{n,2n-1}]$ is at least $2^{(n-5)/6}$.*

The proof of this theorem uses techniques from analytic number theory; it has been shifted into the extended version of this paper.

Claim. The size of $\pi^{\mathrm{id}}_{4,m+1}$-OBDD$[\chi_{R_{m,2m-1}}]$ is exponential w. r. t. m.

Proof. Replacing some variables of a Boolean function by constant values does not enlarge the corresponding π-OBDD. So we show a lower bound on a subfunction $f_m \in B_{2m}$ of $\chi_{R_{m,2m-1}}$ which is obtained by replacing x_{4m+3}, x_{3m+2}, and $x_{2m+1}, \ldots, x_0$ by 0. Hence, argument x represents a $w_{i,j,0,0}$ node which is reachable from s via at most 2^{2m-1} edges iff the s–$w_{i,j,0,0}$-path length $i \cdot j$ is not larger than 2^{2m-1}. So it is $f_m(x) = 1 \Leftrightarrow |i(x)| \cdot |j(x)| \leq 2^{2m-1}$.

Let $g_m(x) = 1 :\Leftrightarrow |i(x)| \cdot |j(x)| = 2^{2m-1}$. It is easy to see that the $\pi_{2,m}^{\text{id}}$-OBDD size of g_m is $\mathcal{O}(m^2)$. Hence, the $\pi_{2,m}^{\text{id}}$-OBDD for

$$h_m(x) := f_m(x) \wedge \overline{g(x)} = (|i(x)| \cdot |j(x)| < 2^{2m-1})$$

is at most polynomially larger than for f_m. Due to $\text{MULT}_{m,2m-1}(x, y) = 1 :\Leftrightarrow |x| \cdot |y| \geq 2^{2m-1}$ for $x, y \in \mathbb{B}^m$, it is $\overline{h_m} = \text{MULT}_{m,2m-1}$.

Altogether, we showed that the $\pi_{2,m}^{\text{id}}$-OBDD size of $\text{MULT}_{m,2m-1}$ is at most polynomially larger than of $\chi_{R_{m,2m-1}}$ implying the claim's statement. □

Both claims imply the result on the space complexity of Reachability*.

Theorem 5. Reachability* $\notin$ I-FPT-S.

By further variable replacements, the Reachability Problem can be trivially reduced to computing the OBDD of all connected node pairs—the *transitive closure*. It is an important submodule of many OBDD-based graph algorithms [11, 14, 20]. So it follows for the analogous restricted variant *TransClos**:

Proposition 2. TransClos* $\notin$ I-FPT-S.

These results rely on the assumption that the output OBDDs of the starred problem variants use the same variable order as the input OBDDs. In contrast, practical algorithms usually run variable reordering heuristics on intermediate OBDD results in order to minimize their size. However, we conjecture that $\text{MULT}_{n,2n-1}$ has exponential OBDD size for every variable order.

8 Conclusions

None of the graph problems that have been considered on OBDD-represented instances has an FPT algorithm w. r. t. a fixed input OBDD width, unless P=PSPACE. Except restricted shortest paths problems, this holds also if both input and output OBDD width are fixed parameters. On the other hand, a polynomially bounded output OBDD size allows to solve all considered problems in polynomial space. We contributed exponential lower bounds on the space (and, therewith, time) complexity of general OBDD-based shortest paths and maximum flow algorithms and a common class of reachability algorithms. Consequently, a restriction of the input OBDD width does not suffice to guarantee polynomial space for these problems. The analyses include the first nontrivial exponential lower bound on the OBDD size for the highest bit of Integer Multiplication w. r. t. the natural interleaved variable order.

It remains an open question whether the OBDD-based Minimum Spanning Tree Problem is in I-FPT-S.

Acknowledgments. Thanks to Oliver Giel, Martin Sauerhoff, and Ingo Wegener for proofreading and discussions.

References

1. B. Becker, M. Behle, F. Eisenbrand, and R. Wimmer. BDDs in a branch and cut framework. In *WEA 2005*, volume 3503 of *LNCS*, pages 452–463. Springer, 2005.
2. R. Berghammer and F. Neumann. RELVIEW - An OBDD-based computer algebra system for relations. In *CASC 2005*, volume 3718 of *LNCS*, pages 40–51. Springer, 2005.
3. R.E. Bryant. Symbolic manipulation of Boolean functions using a graphical representation. In *DAC 1985*, pages 688–694. ACM Press, 1985.
4. R.E. Bryant. Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers*, 35:677–691, 1986.
5. R.G. Downey and M.R. Fellows. *Parameterized Complexity.* Springer, Berlin Heidelberg New-York, 1999.
6. J. Feigenbaum, S. Kannan, M.Y. Vardi, and M. Viswanathan. Complexity of problems on graphs represented as OBDDs. In *STACS 1998*, volume 1373 of *LNCS*, pages 216–226. Springer, 1998.
7. R. Gavalda and D. Guijarro. Learning Ordered Binary Decision Diagrams. In *Algorithmic Learning Theory 1995*, volume 997 of *LNCS*, pages 228–238. Springer, 1995.
8. R. Gentilini, C. Piazza, and A. Policriti. Computing strongly connected components in a linear number of symbolic steps. In *SODA 2003*, pages 573–582. ACM Press, 2003.
9. R. Gentilini and A. Policriti. Biconnectivity on symbolically represented graphs: A linear solution. In *ISAAC 2003*, volume 2906 of *LNCS*, pages 554–564. Springer, 2003.
10. G.D. Hachtel and F. Somenzi. *Logic Synthesis and Verification Algorithms.* Kluwer Academic Publishers, Boston, 1996.
11. G.D. Hachtel and F. Somenzi. A symbolic algorithm for maximum flow in 0–1 networks. *Formal Methods in System Design*, 10:207–219, 1997.
12. H. Jin, A. Kuehlmann, and F. Somenzi. Fine-grain conjunction scheduling for symbolic reachability analysis. In *TACAS 2002*, volume 2280 of *LNCS*, pages 312–326. Springer, 2002.
13. K. Ravi, R. Bloem, and F. Somenzi. A comparative study of symbolic algorithms for the computation of fair cycles. In *FMCAD 2000*, volume 1954 of *LNCS*, pages 143–160. Springer, 2000.
14. D. Sawitzki. Implicit flow maximization by iterative squaring. In *SOFSEM 2004*, volume 2932 of *LNCS*, pages 301–313. Springer, 2004.
15. D. Sawitzki. A symbolic approach to the all-pairs shortest-paths problem. In *WG 2004*, volume 3353 of *LNCS*, pages 154–167. Springer, 2004.
16. D. Sawitzki. Lower bounds on the OBDD size of graphs of some popular functions. In *SOFSEM 2005*, volume 3381 of *LNCS*, pages 298–309. Springer, 2005.
17. D. Sawitzki. The complexity of problems on implicitly represented inputs. To appear in *SOFSEM 2006*, 2006.
18. I. Wegener. *Branching Programs and Binary Decision Diagrams.* SIAM, Philadelphia, 2000.
19. P. Woelfel. New bounds on the OBDD-size of integer multiplication via universal hashing. In *STACS 2001*, volume 2010 of *LNCS*, pages 563–574. Springer, 2001.
20. P. Woelfel. Symbolic topological sorting with OBDDs. In *MFCS 2003*, volume 2747 of *LNCS*, pages 671–680. Springer, 2003.
21. A. Xie and P.A. Beerel. Implicit enumeration of strongly connected components. In *ICCAD 1999*, pages 37–40. ACM Press, 1999.