On Symbolic Representations of Maximum Matchings and (Un)directed Graphs

Beate Bollig

LS2 Informatik, TU Dortmund,
44221 Dortmund, Germany

Abstract. The maximum matching problem is one of the most fundamental algorithmic graph problems and OBDDs are one of the most common dynamic data structures for Boolean functions. Since in some applications graphs become larger and larger, a research branch has emerged which is concerned with the theoretical design and analysis of so-called symbolic algorithms for classical graph problems on OBDD-represented graph instances. Typically problems get harder when their input is represented symbolically, nevertheless not many concrete nontrivial lower bounds are known. Here, it is shown that symbolic OBDD-based algorithms for the maximum matching problem need exponential space (with respect to the OBDD size of the input graph). Furthermore, it is shown that OBDD-representations for undirected graphs can be exponentially larger than OBDD-representations for their directed counterparts and vice versa.

Keywords: Computational complexity, lower bounds, maximum matching, ordered binary decision diagrams, symbolic algorithms.

1 Introduction

Since modern applications require huge graphs, explicit representations by adjacency matrices or adjacency lists may cause conflicts with memory limitations and even polynomial time algorithms seem to be not applicable any more. As time and space resources do not suffice to consider individual vertices, one way out seems to be to deal with sets of vertices and edges represented by their characteristic functions. Ordered binary decision diagrams, denoted OBDDs, introduced by Bryant in 1986 [6], are well suited for the representation and manipulation of Boolean functions, therefore, a research branch has emerged which is concerned with the theoretical design and analysis of so-called symbolic algorithms for classical graph problems on OBDD-represented graph instances (see, e.g., [11,12], [13], [18,19], [21], [23,24], and [28]). Symbolic algorithms have to solve problems on a given graph instance by efficient functional operations offered by the OBDD data structure.

Representing graphs with regularities by means of data structures smaller than adjacency matrices or adjacency lists seems to be a natural idea. But problems typically get harder when their input is represented implicitly. For circuit representations this has been shown in [1,10,20]. These results do not directly carry

over to problems on OBDD-represented inputs since there are Boolean functions like some output bits of integer multiplication whose OBDD complexity is exponentially larger than its circuit size [2,7]. In [8] it has been shown that even the very basic problem of deciding whether two vertices s and t are connected in a directed graph G, the so-called graph accessibility problem GAP, is PSPACE-complete on OBDD-represented graphs. Nevertheless, OBDD-based algorithms are successful in many applications and despite the hardness results there are not many concrete non-trivial lower bounds known for the complexity of problems on OBDD-represented graph instances. In [23] exponential lower bounds on the space complexity of OBDD-based algorithms for the single-source shortest paths problem, the maximum flow problem, and a restricted class of algorithms for the reachability problem have been presented. Recently, the last result has been generalized and an exponential lower bound on the space complexity of all OBDD-based algorithms for reachability analysis has been shown in [3]. The results are not very astonishing but the proofs present worst-case examples which could be helpful to realize why OBDD-based algorithms are successful in many applications by characterizing the special cases that can be handled efficiently and the cases that are difficult to process. In this paper one aim is to present concrete exponential lower bounds and not only existence proofs that there have to be objects of large size or that exponential blow-ups may happen for various problems.

Due to the problem's rich area of applications the maximum matching problem has received a considerable amount of attention for explicit graph representations. Answering an open question posed by Sawitzki (page 186, table 7.4.1 in [22]), we prove that OBDD-based representations of maximum matchings can be exponentially larger than the OBDD representation of the input graph. Using simple counting arguments it can be shown that there exists a complete bipartite graph whose OBDD complexity is small and for which there exists a maximum matching whose OBDD complexity is large. In order to present concrete proofs we present such a graph and a corresponding maximum matching. Searching for advantageous properties of real-world instances that cause an essentially better behavior than in the worst-case, the complexity of graph problems with respect to structured properties of input and/or output OBDDs is interesting. In [21] and [28] symbolic algorithms for maximum flow in 0-1 networks and topological sorting have been presented which have polylogarithmic running time with respect to the number of vertices of a given grid graph. These results rely on the very structured input graph and on restrictions on the width of occuring OBDDs during the computation. Our first result on the size of maximum (perfect) matchings shows that constant input OBDD width is not sufficient to guarantee polynomial space complexity for the maximum matching problem. Afterwards we present a graph whose edge set can be represented by OBDDs of small size but for which the implicit representation of its unique maximum matching needs exponential OBDD size.

By simple counting arguments it is easy to see that almost all graphs on N vertices cannot be represented by OBDDs of polylogarithmic size with respect

to N. On the other hand, it is quite obvious that very simple structured graphs, e.g., grid graphs, have a small OBDD representation. Therefore, in [18,19] the question has been investigated whether succinct OBDD representations can be found for significant graph classes. In this paper we consider whether undirected graphs can be exponentially larger than their so-called directed counterparts and vice versa. Our results can be summarized as follows.

Theorem 1. *Symbolic OBDD-based algorithms for the maximum matching problem need exponential space with respect to the size of the implicit representation of the input graph.*

Theorem 2. *There exists a directed graph G_d and a corresponding undirected graph G_u, obtained from G_d by changing the directed edges into undirected ones, such that the symbolic OBDD representation of G_u is exponentially larger than the OBDD representation of G_d.*

The paper is organized as follows. In Section 2 we define some notation and present some basics concerning OBDDs, symbolic graph representations, and the maximum matching problem. Section 3 contains the proof of Theorem 1. Finally, in Section 4 Theorem 2 is shown and we discuss why the result is not as obvious as it seems to be at first glance. For a slightly more general model than OBDDs the representation size for the corresponding undirected counterparts of directed graphs can only be by a factor of 2 larger than the size for the directed graph. Furthermore, we will look at an undirected graph G_U and a corresponding directed graph G_D, obtained from G_U by changing each undirected edges into one directed edge, such that the symbolic OBDD representation of G_D is exponentially larger than the OBDD representation of G_U.

2   Preliminaries

In order to make the paper self-contained we briefly recall the main notions we are dealing with in this paper.

2.1   Ordered Binary Decision Diagrams

When working with Boolean functions as in circuit verification, synthesis, and model checking, ordered binary decision diagrams are one of the most often used data structures that support efficiently all fundamental operations on Boolean functions, like binary operators, quantifications or satisfiability tests. (For a history of results on binary decision diagrams see, e.g., the monograph of Wegener [27]).

Definition 1. *Let $X_n = \{x_1, \ldots, x_n\}$ be a set of Boolean variables. A variable ordering π on X_n is a permutation on $\{1, \ldots, n\}$ leading to the ordered list $x_{\pi(1)}, \ldots, x_{\pi(n)}$ of the variables.*

In the following a variable ordering π is sometimes identified with the corresponding ordering $x_{\pi(1)}, \ldots, x_{\pi(n)}$ of the variables if the meaning is clear from the context.

Definition 2. *A π-OBDD on X_n is a directed acyclic graph $G = (V, E)$ whose sinks are labeled by Boolean constants and whose non-sink (or decision) nodes are labeled by Boolean variables from X_n. Each decision node has two outgoing edges one labeled by 0 and the other by 1. The edges between decision nodes have to respect the variable ordering π, i.e., if an edge leads from an x_i-node to an x_j-node, then $\pi^{-1}(i) \leq \pi^{-1}(j)$ (x_i precedes x_j in $x_{\pi(1)}, \ldots, x_{\pi(n)}$). Each node v represents a Boolean function $f_v \in B_n$, i.e., $f_v : \{0, 1\}^n \to \{0, 1\}$, defined in the following way. In order to evaluate $f_v(b)$, $b \in \{0, 1\}^n$, start at v. After reaching an x_i-node choose the outgoing edge with label b_i until a sink is reached. The label of this sink defines $f_v(b)$. The width of a π-OBDD is the maximum number of nodes labeled by the same variable. The size of a π-OBDD G is equal to the number of its nodes and the π-OBDD size of a function f is the size of the minimal π-OBDD representing f.*

It is well known that the size of an OBDD representing a function f that depends essentially on n Boolean variables (a function g depends essentially on a Boolean variable z if $g_{|z=0} \neq g_{|z=1}$) may be different for different variable orderings and may vary between linear and exponential size with respect to n.

Definition 3. *The OBDD size or OBDD complexity of f is the minimum of all π-OBDD(f).*

The size of the reduced π-OBDD representing f is described by the following structure theorem [25].

Theorem 3. *The number of $x_{\pi(i)}$-nodes of the minimal π-OBDD for f is the number s_i of different subfunctions $f_{|x_{\pi(1)}=a_1,\ldots,x_{\pi(i-1)}=a_{i-1}}$, $a_1, \ldots, a_{i-1} \in \{0, 1\}$, that essentially depend on $x_{\pi(i)}$.*

Theorem 3 implies the following simple observation which is helpful in order to prove lower bounds. Given an arbitrary variable ordering π the number of nodes labeled by a variable x in the reduced π-OBDD representing a given function f is not smaller than the number of x-nodes in a reduced π-OBDD representing any subfunction of f.

Partitioned binary decision diagrams, denoted PBDDs, have been introduced in [14] as a generalized OBDD model allowing a restricted use of nondeterminism and different variable orderings. They are restricted enough such that most of the essential operations can be performed efficiently and they allow polynomial-size representations for more Boolean functions than OBDDs.

Definition 4. *A k-PBDD consists of k OBDDs whose variable orderings may be different. The output value for an input b is defined as 1 iff at least one of the OBDDs computes 1 on b. A PBDD is a k-PBDD for some k. The size of a k-PBDD is the sum of the sizes of the k OBDDs.*

2.2   Symbolic OBDD-Based Graph Representations and the
Maximum Matching Problem

In the following for $z = (z_{n-1}, \ldots, z_0) \in \{0,1\}^n$ let $|z| := \sum_{i=0}^{n-1} z_i 2^i$. Let $G = (V, E)$ be a graph with N vertices $v_0, \ldots v_{N-1}$. The edge set E can be represented by an OBDD for its characteristic function, where

$$\mathcal{X}_E(x, y) = 1 \Leftrightarrow (|x|, |y| < N) \wedge (v_{|x|}, v_{|y|}) \in E, x, y \in \{0, 1\}^n \text{ and } n = \lceil \log N \rceil.$$

Undirected edges are represented by symmetric directed ones. In the rest of the paper we assume that N is a power of 2 since it has no bearing on the essence of our results. OBDD-represented graphs on N vertices are typically only defined on $\log N$ Boolean variables in comparison to other implicit graph representations where at least $c \log N$ bits for some constant $c > 1$ are allowed [16,26]. One of the reasons is that the number of variables of intermediate OBDDs during a symbolic algorithms can be seen as a performance parameter. Multiplying the number of variables on which an OBDD depends by a constant c enlarge the worst-case size asymptotically from S to S^c. (See, e.g., [9] for the importance to keep the number of variables as low as possible.)

A matching in an undirected graph $G = (V, E)$ is a subset $M \subseteq E$ such that no two edges of M are adjacent. A matching M is maximum if there exists no matching $M' \subseteq E$ such that $|M'| > |M|$, where $|S|$ denotes the cardinality of a set S. A perfect matching is a matching of cardinality $|V|/2$. In the symbolic setting the maximum (perfect) matching problem is the following one. Given an OBDD for the characteristic function of the edge set of an undirected input graph G, the output is an OBDD that represents the characteristic function of a maximum (perfect) matching in G. A graph $G = (V, E)$ is bipartite, if V can be partitioned into two disjoint nonempty sets U and W, such that for all edges $(u, w) \in E$ it holds $u \in U$ and $w \in W$ or vice versa.

3   The Maximum Matching Problem on
OBDD-Represented Graphs

In this section we prove Theorem 1 and demonstrate that an exponential blow-up from input to output size for the maximum matching problem is possible in the symbolic setting.

Our proof structure is the following one. First, we define an input graph G for the maximum matching problem. It is not difficult to see that the size of the corresponding OBDD representation for the characteristic function of its edge set is polynomial with respect to the number of Boolean variables. Afterwards we prove that there exists a maximum matching in G represented by its edge set for which the corresponding characteristic function has exponential OBDD complexity. Therefore, every OBDD-based algorithm solving the maximum matching problem need exponential space with respect to its input length. We start with a very simple input graph and show that there exists a maximum matching whose OBDD complexity is exponentially larger than the OBDD complexity of the

input graph. The investigated maximum matching is also a perfect matching. Afterwards we present an example where the maximum matching is unique but not a perfect matching because the input graph contains many isolated vertices. Now, we make our ideas more precise.

1) The definition of the input graph G:

Our input graph $G = (V, E)$ is a complete bipartite graph on 2^{n^2+1} vertices. The vertex set V is partitioned into two nonempty sets U and W of equal size such that there exists no edge incident to two vertices in U respectively W. The Boolean encoding of a vertex $v \in V$ consists of $n^2 + 1$ Boolean variables $z, x_{11}, \ldots, x_{nn}$, the variable z indicates whether v is in U ($z = 0$) or in W ($z = 1$). The x-variables can be seen as a Boolean matrix of dimension $n \times n$.

2) The polynomial upper bound on the OBDD size for the characteristic function of the edge set of G:

G can be represented by an OBDD of size 5. The characteristic function $\mathcal{X}_E$ of E is defined on the variables $((z^1, x_{11}^1, \ldots, x_{nn}^1), (z^2, x_{11}^2, \ldots, x_{nn}^2))$. The function value is 1 iff $z^1 \oplus z^2 = 1$.

3) A maximum matching in G and an exponential lower bound on the OBDD size for its characteristic function:

It remains to show that there exists a maximum matching in G defined by the characteristic function of its edge set whose OBDD complexity is exponential. Note, that our aim is to present a constructive and not only an existence proof. A vertex v in G has the property $\mathcal{P}$ iff the x-variables of its Boolean encoding correspond to a Boolean matrix that contains exactly one 1-entry in each row and in each column. Now, we are ready to define a maximum matching in G whose OBDD complexity is exponential. The crucial idea for the definition of a perfect matching with large OBDD size is the following. Vertices with property $\mathcal{P}$ are matched to vertices with property $\mathcal{P}$ and vertices without $\mathcal{P}$ to vertices without property $\mathcal{P}$. To be more precise, let $z^u, x_{11}^u, \ldots, x_{nn}^u$ be the variables of the Boolean encoding of a vertex $u \in U$ and $z^w, x_{11}^w, \ldots, x_{nn}^w$ those of a vertex $w \in W$. The vertices u and w are matched if both have the property $\mathcal{P}$ and $x_{ij}^u = x_{ji}^w$ for all $i, j \in \{1, \ldots, n\}$ or if both have not the property $\mathcal{P}$ and $x_{ij}^u = x_{ij}^w$ for all $i, j \in \{1, \ldots, n\}$. Obviously, this is a complete definition of a perfect matching in G. Let $\mathcal{X}_M$ be the characteristic function of this edge set. Next, we prove that the OBDD complexity of $\mathcal{X}_M$ is exponential. In [15,17] exponential lower bounds on the size of so-called nondeterministic read-once branching programs (a more general OBDD model) representing the Boolean function PERM_n, the test, whether a Boolean matrix contains exactly one 1-entry in each row and in each column, are presented. In the following we consider an arbitrary OBDD for a carefully chosen subfunction of $\mathcal{X}_M$ and we investigate several paths from the source of the OBDD to the 1-sink. Here, for the choice of the considered paths and for the estimation of the number of different chosen subpaths some of the ideas presented in [15,17] are used but because of the different definition of our investigated function we have to add some ideas.

Let G_M be an OBDD on the variables $((z', x_{11}, \ldots, x_{nn}), (z'', y_{11}, \ldots, y_{nn}))$ for the representation of $\mathcal{X}_M$ and G'_M be the OBDD obtained from G_M by replacing the variables x_{23}, y_{32} by 1 and x_{32}, y_{23} by 0. The reason for these replacements is that all 1-inputs for the subfunction of $\mathcal{X}_M$ represented by G'_M correspond to edges between vertices in the input graph G with property $\mathcal{P}$. As a result it is a little bit easier to argue that G'_M and therefore G_M need an exponential number of nodes. Furthermore, we set z' to 1 and z'' to 0. This is not crucial for our lower bound proof but convenient to keep our proof as simple as possible.

Our aim is to show that there is an exponential number of nodes in the OBDD G'_M. For the ease of notations we assume w.l.o.g. that n is an even number. We investigate the paths in G'_M from the source to the 1-sink called accepting paths. There are $2n - 2$ 1-edges, i.e., variables set to 1, on these paths and the number of these paths is $(n - 2)(n - 2)!$. Now, we separate each accepting path p into its initial part p_u and into the remaining part p_ℓ to the 1-sink. Here, we have to use a different cut as considered in [15,17]. A pair (x_{ij}, y_{ji}), $i, j \in \{1, \ldots, n\}$, is called (x, y)-pair. We define a cut through all accepting paths after for exactly $n/2 - 2$ (x, y)-pairs there exists at least one variable set to 1 for the first time. Let R_{p_u} (C_{p_u}) be the set of indices i for which a variable x_{i*} or y_{*i} $(x_{*i}$ or $y_{i*})$ is set to 1 on p_u. If $n/2 - 1$ rows and columns are fixed, there are $(n/2 - 1)!$ possibilities to map the indices of the rows to the indices of the columns. Each initial part of an accepting path can be continued by at most $(n/2)!$ subpath to the 1-sink. Therefore, there is a set P of different initial paths from the source to the cut, $|P| \geq \binom{n-1}{n/2-1}$, such that for two different paths p'_u and p''_u in P we know that $R_{p'_u} \neq R_{p''_u}$ or $C_{p'_u} \neq C_{p''_u}$. Due to our choice of the considered paths, there are extensions p'_ℓ of p'_u and p''_ℓ of p''_u which lead to the 1-sink. Since $R_{p'_u} \neq R_{p''_u}$ or $C_{p'_u} \neq C_{p''_u}$, p'_u concatenate with p''_ℓ cannot correspond to a Boolean encoding, where in each row and in each column is exactly one 1-entry, and therefore, cannot be an accepting path. Here, we make use of the fact that we investigate a subfunction of $\mathcal{X}_M$ whose 1-inputs correspond to edges between vertices with property $\mathcal{P}$ in the input graph. Therefore, the paths in P cannot lead to the same node in G'_M and the size of the set P is a lower bound on the size of G'_M. Using Stirling's formula we obtain a lower bound of $\Omega(n^{-1/2}2^n)$ and we are done.

Summarizing, we have shown that the maximum matching problem may cause exponential space requirements on OBDD-represented graphs by generating instances with an exponential gap between the input and the output OBDD size. On the other hand, there exists a perfect matching in G whose OBDD complexity is linear. Therefore, the representation sizes for maximum matchings in a graph can be quite different. Now, we show that an exponential gap between input and output size is also possible if the maximum matching is unique. In this way we demonstrate that every symbolic OBDD-based algorithm for the maximum matching problem need exponential space. We start with the definition of a function which is well known in the BDD literature.

Definition 5. *The hidden weighted bit function* $\mathrm{HWB}_n : \{0, 1\}^n \to \{0, 1\}$ *computes the bit* b_{sum} *on the input* $b = (b_1, \ldots, b_n)$, *where* $sum := \sum_{i=1}^{n} b_i$ *and* $b_0 := 0$.

Bryant [7] has introduced this function as a very simple version of storage access where each variable is control and data variable. He has also already shown that the OBDD complexity of HWB_n is $\Omega(2^{(1/5-\epsilon)n})$ which has been slightly improved up to $\Omega(2^{n/5})$ in [5].

1) The definition of the input graph G_n:
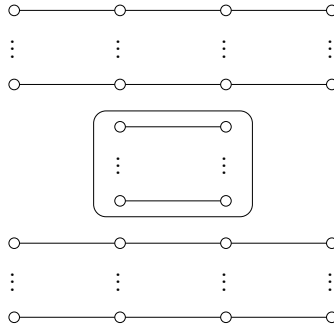
Fig. 1. The input graph G_n and the set of hidden difficult edges

The graph $G_n = (V, E)$ consists of 2^{2n+2} vertices v_{i_1,i_2,i_3}, $i_1 \in \{0,\ldots,3\}$, $i_2, i_3 \in \{0,\ldots,2^n-1\}$. Let $b^i = (b_0^i,\ldots,b_{n-1}^i)$ be the binary representation of an integer $i \in \{0,\ldots,2^n-1\}$. There exists an edge between a vertex v_{i_1,i_2,i_3} and a vertex v_{j_1,j_2,j_3} if one of the following requirements is fulfilled:

- $i_1 = 0, i_2 = 2^k, \sum_{\ell=0}^{n-1} b_\ell^{i_3} = k$ and $b_{k-1}^{i_3} = 1$,
  $j_1 = 1, j_2 = 0$, and $j_3 = i_3$, or
- $i_1 = 2, i_2 = 0, \sum_{\ell=0}^{n-1} b_\ell^{i_3} = k$ and $b_{k-1}^{i_3} = 1$,
  $j_1 = 3, j_2 = 2^k$, and $j_3 = i_3$, or
- $i_1 = 1, i_2 = 0, j_1 = 2, j_2 = 0$, and $j_3 = i_3$.

Figure 1 shows the structure of the input graph G_n, where isolated vertices are missing. Obviously, the maximum matching in G_n is unique (see Figure 2). The important property of G_n is that an edge from a vertex v_{1,i_2,i_3} to a vertex v_{2,j_2,j_3} belongs to the maximum matching iff $i_2 = j_2 = 0$, $i_3 = j_3$, and the binary representation of i_3 respectively j_3 corresponds to an input that belongs to $\mathrm{HWB}_n^{-1}(0)$. The characteristic function of this edge set is a difficult function but in our input graph this edge set is in some sense hidden (see Figure 1) such that the characteristic function of the edge set of the input graph can be represented by OBDDs of small size.

2) The polynomial upper bound on the OBDD size of $\mathcal{X}_E$:

Let $x_0^1, x_1^1, x_0^2, \ldots, x_{n-1}^2, x_0^3, \ldots, x_{n-1}^3$ be the variables of the Boolean encoding of a vertex v_{i_1,i_2,i_3}, where x_0^1, x_0^2, and x_0^3 denote the least significant bits, the x^1-variables represent i_1, the x^2-variables i_2, and the x^3-variables i_3. The
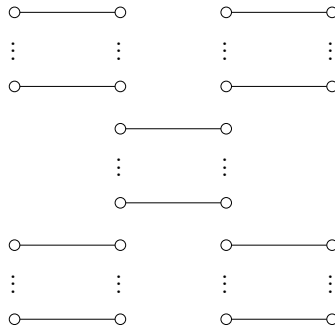
Fig. 2. The unique maximum matching in G_n

characteristic function $\mathcal{X}_E$ of the edge set depends on $2(2n+2)$ Boolean variables $((x_0^1, x_1^1, x_0^2, \ldots, x_{n-1}^2, x_0^3, \ldots, x_{n-1}^3), (y_0^1, y_1^1, y_0^2, \ldots, y_{n-1}^2, y_0^3, \ldots, y_{n-1}^3))$.

Our aim is to prove that $\mathcal{X}_E$ can be represented by OBDDs of size $O(n^2)$ according to the variable ordering

$$x_0^1, y_0^1, x_1^1, y_1^1, x_0^2, y_0^2, \ldots, x_{n-1}^2, y_{n-1}^2, x_0^3, y_0^3, \ldots, x_{n-1}^3, y_{n-1}^3.$$

There are three different disjoint edge sets, from $v_{0,\cdot,\cdot}$- to $v_{1,\cdot,\cdot}$-, from $v_{1,\cdot,\cdot}$- to $v_{2,\cdot,\cdot}$-, and from $v_{2,\cdot,\cdot}$- to $v_{3,\cdot,\cdot}$-vertices. We prove that each of them can be represented by OBDDs of small size. Since the different edge sets can be identified by the assignments to the x^1- and y^1-variables which are tested at the beginning of the OBDD, it suffices to add the OBDD sizes in order to obtain an upper bound on the OBDD complexity of $\mathcal{X}_E$.

If $x_0^1 = x_1^1 = 0$, $y_0^1 = 0$, and $y_1^1 = 1$, it is checked whether $y_0^2 = \ldots = y_{n-1}^2 = 0$, and there exists exactly one x^2-variable set to 1. If $|x^2| = 2^i$, the number of x^3-variables is counted. The function value is 1 if $\sum_{\ell=0}^{n-1} x_\ell^3 = i$, $x_{i-1}^3 = 1$, and $y_\ell^3 = x_\ell^3$, $0 \le \ell \le n - 1$. Since we only have to distinguish n different values for $|x^2|$, this can be done by an OBDD of width $O(n)$.

If $x_0^1 = 0$, $x_1^1 = 1$, $y_0^1 = y_1^1 = 0$, the roles of the x- and y-variables are exchanged. The cases $x_0^1 = 1$, $x_1^1 = 0$, $y_0^1 = y_1^1 = 1$, and $x_0^1 = x_1^1 = 1$, $y_0^1 = 0$, $y_1^1 = 1$ are similar.

If $x_0^1 = 1$, $x_1^1 = 0$, $y_0^1 = 0$, $y_1^1 = 1$, or $x_0^1 = 0$, $x_1^1 = 1$, $y_0^1 = 1$, $y_1^1 = 0$, it is checked whether $|x^2| = |y^2| = 0$ and $|x^3| = |y^3|$. This can be done by an OBDD of constant width.

Altogether, we have seen that $\mathcal{X}_E$ can be represented by an OBDD of size $O(n^2)$.

3) The exponential lower bound on the OBDD size for the characteristic function of the maximum matching $\mathcal{X}_M$ in G_n:

Due to our definition of G_n the maximum matching contains an edge from a vertex v_{1,i_2,i_3} to a vertex v_{2,j_2,j_3} if $i_2 = j_2 = 0$, $i_3 = j_3$, and the binary representation of i_3 respectively j_3 corresponds to an input that belongs to $\text{HWB}_n^{-1}(0)$.

Our aim is to adapt the ideas for the exponential lower bound on the OBDD size of HWB_n presented in [7]. Therefore, we consider the subfunction of $\mathcal{X}_M$, where $x_0^1 = 1$, $x_1^1 = 0$, $y_0^1 = 0$, $y_1^1 = 1$, and the x^2- and y^2-variables are replaced by 0. In the following we assume that n is a multiple of 10 because it has no bearing on the essence of the proof. Let π be an arbitrary but fixed variable ordering. A pair (x_ℓ^3, y_ℓ^3) is called (x, y)-pair and x_ℓ^3 a partner of y_ℓ^3 and vice versa. Now, we define a cut in the variable ordering after for the first time for exactly $(6/10)n$ (x, y)-pairs there exist at least one variable. T contains the variables before the cut according to π and B the remaining variables. Let P_H be the set of all pairs (x_i^3, y_i^3), $i \in \{(5/10)n + 1, \ldots, (9/10)n\}$, and P_L be the set of all pairs (x_j^3, y_j^3), $j \in \{(1/10)n + 1, \ldots, (5/10)n\}$. Obviously, T contains at least for $(2/10)n$ pairs in P_H or at least for $(2/10)n$ pairs in P_L at least one variable. W.l.o.g. we assume that T contains at least for $(2/10)n$ pairs in P_L at least one variable. In the following we only consider assignments where variables that belong to the same (x, y)-pair are replaced by the same constant. We consider all assignments to the variables in T where exactly $(1/10)n$ pairs in P_L are replaced by 1, all other variables in T are set to 0. There are at least $\binom{(2/10)n}{(1/10)n} = \Omega(n^{-1/2}2^{n/5})$ different assignments. Using Theorem 3 it is sufficient to prove that these assignments lead to different subfunctions. For this reason we consider two different assignments b and b' to the variables in T. Let $(x_{\ell-1}^3, y_{\ell-1}^3)$ be an (x, y)-pair for which at least one variable is replaced differently in b and b'. W.l.o.g. $x_{\ell-1}^3$ is set to 0 in b and to 1 in b'. Now, we consider the following assignment b_r to the variables in B. The variables for which there is a partner in T are replaced by the assignment to the partner according to b. The remaining variables are replaced in such a way that there are exactly $\ell - (1/10)n$ pairs that are set to 1. This can be done because there are $(4/10)n$ pairs for which both variables are in B and $\ell \leq (5/10)n$. Obviously, the function value of the subfunction induced by b on b_r is 1. The function value for the subfunction induced by b' on b_r is 0 because either $|x^3| \neq |y^3|$ or $x^3 \in HWB_n^{-1}(1)$.

Altogether, we have shown that the OBDD complexity of $\mathcal{X}_M$ is at least $\Omega(n^{-1/2}2^{n/5})$.

4   Exponential Blow-Ups for the OBDD-Complexity of Directed and Undirected Graphs

In this section we prove Theorem 2 and consider the OBDD size of directed and undirected graphs.

Definition 6. *An undirected graph $G_u = (V, E_u)$ is called the counterpart of a directed graph $G_d = (V, E_d)$ iff for all edges $(u, w) \in E_d$ the edge (u, w) is in E_u. An asymmetric directed graph $G_D = (V, E_D)$ is called a counterpart of an undirected graph $G_U = (V, E_U)$ if for all edges $(u, w) \in E_U$ the edge (w, u) is not in E_D but $(u, w) \in E_D$ or vice versa.*

In order to prove Theorem 2, we investigate the following directed bipartite graph $G_d = (V, E_d)$ defined on 2^{n^2+1} vertices. V is partitioned into the sets U
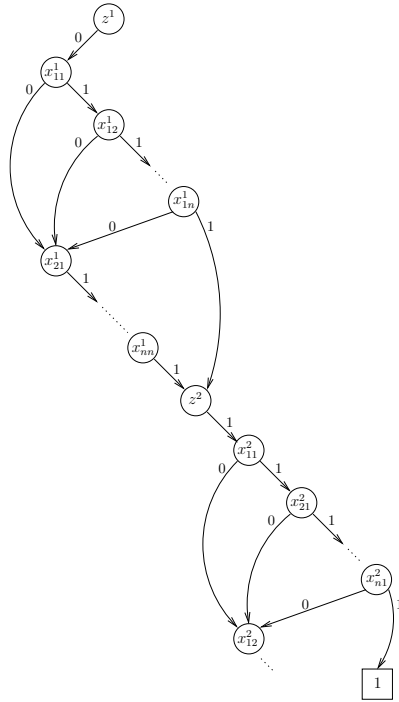
Fig. 3. An OBDD for $\mathcal{X}_{E_d}$. Missing edges are leading to the 0-sink.

and W of equal size. The Boolean encoding of a vertex $v \in V$ consists of $n^2 + 1$ Boolean variables $z, x_{11}, \ldots, x_{nn}$, where the variable z indicates whether $v \in U$ ($z = 0$) or in W ($z = 1$). The x-variables can be seen as a Boolean matrix X of dimension $n \times n$. There exists an edge from a node $u \in U$ to a node $w \in W$ if there exists a row that consists only of 1-entries in the Boolean encoding X of u and a column that contains only 1-entries in the Boolean encoding according to the x-variables of w. Next, we prove that G_d can be represented by OBDDs of linear size with respect to the number of Boolean variables but the undirected counterpart G_u needs exponential size. The characteristic function of the edge set of G_d is defined on the variables $(z^1, x_{11}^1, \ldots, x_{nn}^1), (z^2, x_{11}^2, \ldots, x_{nn}^2)$. Our aim is to prove that $\mathcal{X}_{E_d}$ can be represented by OBDDs of size $O(n^2)$ and constant width according to the variable ordering

$$z^1, x_{11}^1, x_{12}^1, \ldots, x_{nn}^1, z^2, x_{11}^2, x_{21}^2, \ldots, x_{nn}^2,$$

i.e., the first x-variables are tested in a row-wise manner, variables that belong to the same row are tested one after another, and the last x-variables are tested in a column-wise manner. Applying Theorem 3 is is sufficient to prove that there are only a constant number of different subfunctions obtained by replacements of the first i variables for all $i \in \{1, \ldots, 2n^2 + 2\}$ with respect to the considered

variable ordering. If z^1 is 1, the function value is 0 because there are no edges from a vertex $w \in W$ to a vertex $u \in U$. If z^1 is 0, the x^1-variables are tested row-wise and it is checked whether there exists a row that consists only of 1-entries. This can be done by an OBDD of width 1. If the test is negative, the function value is 0. If the test is positive, the variable z^2 is tested. The function value is 0, if z^2 is 0, because there are no edges between vertices in U. Afterwards the x^2-variables are tested column-wise and the function value is 1 iff there exists a column that contains only 1-entries. This can also be done by an OBDD of width 1. Figure 3 shows an OBDD for $\mathcal{X}_{E_d}$.

For the lower bound proof on the OBDD size for the characteristic function of the undirected counterpart G_u let π be an arbitrary but fixed variable ordering. We define a cut in π where for the first time $n/2 - 1$ rows or $n/2 - 1$ columns have a tested x^1- or x^2-variable. X_U contains the x-variables before the cut in π, X_L the remaining x-variables.

Case 1: There are $n/2 - 1$ rows for which an x^1-variable or $n/2 - 1$ columns for which an x^2-variable is in X_U.

W.l.o.g. we assume that there are $n/2 - 1$ rows for which an x^1-variable is in X_U. First, we set z^1 to 0 and z^2 to 1. The x^2-variables are replaced by constants in the following way:

  - the variables $x_{11}^2, x_{21}^2, \ldots, x_{n1}^2$ are set to 1,
  - the remaining x^2-variables are set to 0.

As a result we obtain a subfunction of $\mathcal{X}_{E_u}$ whose function value is 1 iff the Boolean matrix defined by $x_{11}^1, \ldots, x_{nn}^1$ contains a row that consists only of 1-entries. Now, we prove that a π-OBDD for this subfunction needs exponential size. We set the x^1-variables tested first in the first $n/2 - 1$ rows with respect to π to all possible assignments. Variables in X_U that belong to the same row are set to the same constant. Next, we prove that two different assignments b and b' of these $2^{n/2-1}$ partial assignments lead to different subfunctions. For this reason we consider the following assignment b_r to the remaining x^1-variables in X_L. Let ℓ be a row for which the x^1-variables in b are set to 1 and in b' to 0 (or vice versa). W.l.o.g. we assume that the $x_{\ell\cdot}$-variables in b are set to 1 and in b' to 0. In b_r the $x_{\ell\cdot}^1$-variables are set to 1, the remaining x^1-variables are set to 0. There exist $x_{\ell\cdot}^1$-variables in X_L because otherwise there are n rows that have a variable in X_U. The function value of the considered subfunction obtained by b respectively b' for b_r is 1 respectively 0, therefore the induced subfunctions are different and we are done.

Case 2: There are $n/2 - 1$ columns for which an x^1-variable or $n/2 - 1$ rows for which an x^2-variable is in X_U.

By replacing the variables z^1 by 1 and z^2 by 0 the case is similar to the first one and we are done.

One might think that our result is not very astonishing because the orientation of an edge can store some kind of information and without this information the representation size may enlarge. Nevertheless, if the characteristic function $\mathcal{X}_{E_d}$

of a directed graph $G_d = (V, E_d)$ is represented by a circuit, the representation size for the undirected counterpart can only enlarge by a factor of 2. In the following we show that the same holds for 2-PBDDs which are slightly more general than OBDDs. Let E_d^R be the set of edges obtained by replacing each directed edge $(u, v) \in E_d$ by the directed edge (v, u). Let G_{E_d} be an OBDD for the characteristic function $\mathcal{X}_{E_d}$ which is defined on x- and y-variables. The PBDD consists of two OBDDs according to different variable orderings. The first part represents the edges in E_d and is equal to G_{E_d}, the second part represents $\mathcal{X}_{E_d}^R$. For the second part of the 2-PBDD we change the variable ordering of G_{E_d} by renaming the x-and the y-variables, i.e., nodes labeled by x_i are now y_i-nodes and vice versa (for all indices). Obviously, the second part of the 2-PBDD has the same size as G_{E_d} and represents $\mathcal{X}_{E_d}^R$. Altogether, our result on the OBDD size of directed graphs and their undirected counterparts is not as obvious as it seems to be.

In the design and analysis of symbolic graph algorithms OBDDs are often ordered according to so-called interleaved variable orderings, where x- and y-variables of the same significance (or with the same indices) are tested consecutively (see, e.g., [21] and [28]). This seems to be reasonable, since the characteristic function of an undirected graph is symmetric. If a directed graph $G_d = (V, E_d)$ can be represented symbolically by an OBDD G_{E_d} of small size according to an interleaved variable ordering, its undirected counterpart $G_u = (V, E_u)$ can also be represented symbolically by an OBDD of small size. To be more precise, if S is the size of G_{E_d}, the OBDD size of $\mathcal{X}_{E_u}$ is bounded above by $O(S^2)$. The reason is the following one. If we modify a π-OBDD for a function f into a π'-OBDD for f, where π' can be obtained from π by only exchanging the position of neighbored variables, the size of the π'-OBDD can only be by a factor of 3 larger than the size of the π-OBDD (see Theorem 4 in [4]). Afterwards an OBDD for $\mathcal{X}_{E_u}$ can be obtained by applying an $\lor$-synthesis on the OBDDs for $\mathcal{X}_{E_d}$ and $\mathcal{X}_{E_d^R}$ because now the OBDDs are ordered with respect to the same variable ordering. Using the well-known results on the worst-case complexity of the synthesis-operation (see, e.g., [6]), the OBDD size for $\mathcal{X}_{E_u}$ can be bounded above by the product of the OBDD sizes for $\mathcal{X}_{E_d}$ and $\mathcal{X}_{E_d^R}$.

In the rest of this section we show that also between undirected graphs and their directed counterparts an exponential blow-up in the representation size is possible. Again, we consider the complete undirected bipartite graph $G_U = (V, E_U)$ defined in Section 3. For each undirected edge in E_U we have two possibilities to choose the orientation. Using a simple counting argument, it is easy to see that there exists a directed counterpart of G_U whose OBDD complexity is exponential. Finally, we present a concrete directed counterpart of G_U whose OBDD size is exponential. The graph is defined in the following way. There exists an edge from $u \in U$ to a vertex $w \in W$ iff the x-variables of the Boolean encoding of u correspond to a Boolean matrix that contains exactly one 1-entry in each row and in each column. If we replace z^1 by 0 and the variables $z^2, x_{11}^2, \ldots, x_{nn}^2$ by 1, we obtain the function PERM_n. As already mentioned in Section 3 the OBDD size of PERM_n is $\Omega(n^{-1/2} 2^n)$. Therefore, we are done.

Concluding Remarks

Symbolic graph algorithms on OBDD-based representations are implicitly parallel, since vertices or edges are treated simultaneously if they share their OBDD representations. Sawitzki [24] has shown that a problem is in the complexity class NC, which contains the problems that can be solved efficiently in parallel, if it can be solved with a polylogarithmic number of OBDD-operations with respect to the number of the vertices in a given graph. It is an open problem whether the maximum matching problem is in NC. Nevertheless, we have seen that symbolic algorithms for this problem need exponential space (with respect to the number of Boolean variables). Since in the complete bipartite graph presented in Section 3 there exists a maximum matching of linear OBDD size (a node in U in matched to a node in W if the x-variables in the Boolean encoding are equal) and a maximum matching whose representation size is exponential even for a more general model than OBDDs called nondeterministic read-once branching programs, we have seen that the representation sizes for maximum matchings in an input graph can be quite different.

Acknowledgment

References

1. Balcázar, J.L., Lozano, A.: The complexity of graph problems for succinctly represented graphs. In: Nagl, M. (ed.) WG 1989. LNCS, vol. 411, pp. 277–285. Springer, Heidelberg (1990)
2. Bollig, B.: On the OBDD complexity of the most significant bit of integer multiplication. In: Agrawal, M., Du, D.-Z., Duan, Z., Li, A. (eds.) TAMC 2008. LNCS, vol. 4978, pp. 306–317. Springer, Heidelberg (2008)
3. Bollig, B.: Symbolic OBDD-based reachability analysis needs exponential space. In: van Leeuwen, J., Muscholl, A., Peleg, D., Pokorný, J., Rumpe, B. (eds.) SOFSEM 2010. LNCS, vol. 5901, pp. 224–234. Springer, Heidelberg (2010)
4. Bollig, B., Löbbing, M., Wegener, I.: On the effect of local changes in the variable ordering of ordered decision diagrams. Information Processing Letters 59, 233–239 (1996)
5. Bollig, B., Löbbing, M., Sauerhoff, M., Wegener, I.: On the complexity of the hidden weighted bit function for various BDD models. Theoretical Informatics and Applications 33, 103–115 (1999)
6. Bryant, R.E.: Graph-based algorithms for Boolean function manipulation. IEEE Trans. on Computers 35, 677–691 (1986)
7. Bryant, R.E.: On the complexity of VLSI implementations and graph representations of Boolean functions with application to integer multiplication. IEEE Trans. on Computers 40, 205–213 (1991)

8. Feigenbaum, J., Kannan, S., Vardi, M.V., Viswanathan, M.: Complexity of problems on graphs represented as OBDDs. In: Meinel, C., Morvan, M. (eds.) STACS 1998. LNCS, vol. 1373, pp. 216–226. Springer, Heidelberg (1998)

9. Fisler, K., Vardi, M.Y.: Bisimulation, minimization, and symbolic model checking. Formal Methods in System Design 21(1), 39–78 (2002)

10. Galperin, H., Wigderson, A.: Succinct representations of graphs. Information and Control 56, 183–198 (1983)

11. Gentilini, R., Piazza, C., Policriti, A.: Computing strongly connected components in a linear number of symbolic steps. In: Proc. of SODA, pp. 573–582. ACM Press, New York (2003)

12. Gentilini, R., Piazza, C., Policriti, A.: Symbolic graphs: linear solutions to connectivity related problems. Algorithmica 50, 120–158 (2008)

13. Hachtel, G.D., Somenzi, F.: A symbolic algorithm for maximum flow in $0 - 1$ networks. Formal Methods in System Design 10, 207–219 (1997)

14. Jain, J., Bitner, J., Fussell, D.S., Abraham, J.A.: Functional partitioning for verification and related problems. In: Brown MIT VLSI Conference, pp. 210–226 (1992)

15. Jukna, S.: The effect of null-chains on the complexity of contact schemes. In: Csirik, J.A., Demetrovics, J., Gecseg, F. (eds.) FCT 1989. LNCS, vol. 380, pp. 246–256. Springer, Heidelberg (1989)

16. Kannan, S., Naor, M., Rudich, S.: Implicit representations of graphs. SIAM Journal on Discrete Mathematics 5, 596–603 (1992)

17. Krause, M., Meinel, C., Waack, S.: Separating the eraser Turing machine classes L_e, NL_e, co-NL_e and P_e. Theoretical Computer Science 86, 267–270 (1991)

18. Meer, K., Rautenbach, D.: On the OBDD size for graphs of bounded tree- and clique-width. Discrete Mathematics 309(4), 843–851 (2009)

19. Nunkesser, R., Woelfel, P.: Representation of graphs by OBDDs. Discrete Applied Mathematics 157(2), 247–261 (2009)

20. Papadimitriou, C.H., Yannakakis, M.: A note on succinct representations of graphs. Information and Control 71, 181–185 (1986)

21. Sawitzki, D.: Implicit flow maximization by iterative squaring. In: Van Emde Boas, P., Pokorný, J., Bieliková, M., Štuller, J. (eds.) SOFSEM 2004. LNCS, vol. 2932, pp. 301–313. Springer, Heidelberg (2004)

22. Sawitzki, D.: Algorithmik und Komplexität OBDD-repräsentierter Graphen. PhD thesis, University of Dortmund (2006) (in German)

23. Sawitzki, D.: Exponential lower bounds on the space complexity of OBDD-based graph algorithms. In: Correa, J.R., Hevia, A., Kiwi, M. (eds.) LATIN 2006. LNCS, vol. 3887, pp. 781–792. Springer, Heidelberg (2006)

24. Sawitzki, D.: The complexity of problems on implicitly represented inputs. In: Wiedermann, J., Tel, G., Pokorný, J., Bieliková, M., Štuller, J. (eds.) SOFSEM 2006. LNCS, vol. 3831, pp. 471–482. Springer, Heidelberg (2006)

25. Sieling, D., Wegener, I.: NC-algorithms for operations on binary decision diagrams. Parallel Processing Letters 48, 139–144 (1993)

26. Talamo, M., Vocca, P.: Representing graphs implicitly using almost optimal space. Discrete Applied Mathematics 108, 193–210 (2001)

27. Wegener, I.: Branching Programs and Binary Decision Diagrams - Theory and Applications. SIAM Monographs on Discrete Mathematics and Applications (2000)

28. Woelfel, P.: Symbolic topological sorting with OBDDs. Journal of Discrete Algorithms 4(1), 51–71 (2006)