Representations of All Solutions of Boolean Programming Problems

Utz-Uwe Haus and **Carla Michini***
Institute for Operations Research
Department of Mathematics
ETH Zurich
Rämistr. 101, 8092 Zürich, Switzerland

Abstract

It is well known that (reduced, ordered) binary decision diagrams (BDDs) can sometimes be compact representations of the full solution set of Boolean optimization problems. Recently they have been suggested to be useful as discrete relaxations in integer and constraint programming (Hoda, van Hoeve, and Hooker 2010). We show that for every independence system there exists a top-down (i.e., single-pass) construction rule for the BDD. Furthermore, for packing and covering problems on n variables whose bandwidth is bounded by $\mathcal{O}(\log n)$, the maximum width of the BDD is bounded by n. Furthermore, we characterize minimal width of BDDs representing the set of all solutions to a stable set problem for various basic classes of graphs.

1 Introduction

While in optimization one is typically concerned with the question of how to determine an optimal solution, it is often just as important to determine whether multiple optimal or close-to-optimal solutions exist, since they may structurally differ significantly. For some combinatorial problems there are specialized procedures to compute the k best solutions, and recent general-purpose IP solvers usually provide some option to return more than one optimal solution. Furthermore, in many applications one may even not be interested so much in optimality, but rather in understanding the structure of the feasible set near the optimum. In ultimate consequence this leads to the question of how to construct good descriptions of the set of all feasible solutions (Henk, Köppe, and Weismantel 2003; Haus, Köppe, and Weismantel 2003).

Recently, (Hoda, van Hoeve, and Hooker 2010) suggested to use binary decision diagrams (BDDs) (or their integer variant, MDDs) to encode a discrete relaxation of scheduling problems, to aid branching decisions and allow strong pruning decisions. By suitably limiting the width of the BDD one can guarantee reasonable size of the relaxation, at the cost of weakening it. The relaxations considered are conflicts and logical statements in DNF (leading to stable set and set covering problems).

*The authors acknowledge support of the Marie-Curie-ITN 289581 'NPlast', an EU FP7 project.

A *subset system* $\mathcal{L}$ is a collection of subsets of a finite set E. The subsets of the collection are called the *members* of $\mathcal{L}$, the set E is called the *ground set* of $\mathcal{L}$ and denoted by $E(\mathcal{L})$.

A subset system is called *clutter* if its members are pairwise incomparable. Given a clutter $\mathcal{C}$ and a set $Z \subseteq E(\mathcal{C})$, the clutter of all the members of $\mathcal{C}$ which have empty intersection with Z is obtained from $\mathcal{C}$ by *deletion* of the set Z and is denoted by $\mathcal{C} \setminus Z$. The clutter of all minimal members of $\{A - Z \colon A \in \mathcal{C}\}$ is obtained from $\mathcal{C}$ by *contraction* of Z and denoted by $\mathcal{C}/Z$. To avoid confusion we use the symbol $-$ to denote set difference. If $\mathcal{C}'$ is obtained from a clutter $\mathcal{C}$ by deletion or contractions of the elements in Z, then $E(\mathcal{C}') = E(\mathcal{C}) - Z$. Any clutter $\mathcal{C}'$ obtained from $\mathcal{C}$ by a sequence of deletions and contractions is a *minor* of $\mathcal{C}$. A *transversal* is a subset $T \subseteq E(\mathcal{C})$ such that $T \cap A \neq \emptyset$ for all $A \in \mathcal{C}$.

An *independence system* $\mathcal{I}$ is a nonempty subset system satisfying $A \subseteq B \in \mathcal{I} \Rightarrow A \in \mathcal{I}$. The members of $\mathcal{I}$ are called *independent sets*; the subsets of $E(\mathcal{I})$ which are not independent are called *dependent* sets; a maximal independent set is called a *base*, a minimal dependent set is called a *circuit*. It follows that a set is independent if and only if it contains no circuit; hence, the independence system is fully characterized by the family of its circuits, $\mathcal{C}_{\mathcal{I}}$. Moreover, such a family constitutes a clutter. Conversely, given any clutter $\mathcal{C}$, there exists an independence system $\mathcal{I}_{\mathcal{C}} = \{A \,:\, E(\mathcal{C}) - A \text{ is a transversal of } \mathcal{C}\}$.

In this paper we are concerned with the question of how to compactly encode the set of all members of an independence system or the set of all transversals of a clutter. In particular this encompasses problems of the form

$$\begin{array}{ccc} Ax \le 1_m & & Ax \ge 1_m \\ x \in \{0,1\}^n & \text{or} & x \in \{0,1\}^n \\ \text{(Packing)} & & \text{(Covering)} \end{array} \qquad (1)$$

where $A \in \{0,1\}^{m \times n}$, but also other problems like matchings, solutions of a knapsack constraint, or minimal terms of a term order on monomials.

Note that, given a graph G, the stable set problem and the node covering problem on G correspond to the packing and to the covering problem in (1), where A is the edge-node incidence matrix of G. In particular, the stable sets of G are an independence system, whose circuits correspond to the edges of G. Thus, $E(G)$ is a clutter, and the transversals of this clutter are the node covers of G. Note that $S \subseteq V(G)$ is a stable set if and only if $V(G) - S$ is a node cover.

We will introduce some more notation and terminology.

Let G be a finite, simple undirected graph. We denote the node set and the edge set of G as $V(G)$ and $E(G)$, respectively. Let $G = (V, E)$. We will always assume that the nodes of G are ordered, i.e. wlog $V = \{1, \dots, |V|\}$. For $j \leq |V|$, we define $V_j = \{1, \dots, j\}$. For a node $v \in V$ the set $N(v) = \{u \in V \ : \ uv \in E\}$ denotes the *neighbors* of v. For a set $U \subseteq V$, we denote the neighbors of U by $N(U) = \{v \in V : uv \in E, u \in U, v \in V - U\}$. A set of nodes $S \subseteq V$ is called *stable* if $\{uv \in E \ : \ u, v \in S\} = \emptyset$, i.e. no pair of nodes in S is linked by an edge in G.

Let D be a finite, simple directed graph. We denote the node set and the arc set of D by $V(D)$ and $A(D)$, respectively. We denote by (u, v) an arc of A that leaves u and enters v. For a node $v \in V$, $\delta^{\text{in}}(v)$ and $\delta^{\text{out}}(v)$ denote the sets of *ingoing* and *outgoing* edges of v, respectively. Similarly, for a subset $U \subseteq V$, $\delta^{\text{in}}(U) = \{(u, v) \in A : u \in V - U, v \in U\}$ and $\delta^{\text{out}}(U) = \{(u, v) \in A \ : \ u \in U, v \in V - U\}$. In a digraph a node with no incoming arcs is called a *root*, and a node with no outgoing arcs is called a *leaf* or *terminal*. For further terminology and notation in graph theory we refer to (Schrijver 2002).

Let X be an ordered index set for the variables $\{x_i \ : \ i \in X\}$ of a binary problem. A *binary decision diagram* (BDD) $B = (X, U \cup \{\top, \bot\}, A, l, d)$ is a directed acyclic graph (dag) on node set $U \cup \{\top, \bot\}$ with exactly one root, two leaves $\top$ and $\bot$, arc set A, arc label function $d : A \to \{0, 1\}$ and a function $l : U \to X$ such that:

- B is a layered digraph, i.e. its node set partitions into layers $L_i = \{u \in U \ : \ l(u) = i\}$, $i = 1, \dots, |X|$ and $L_{|X|+1} = \{\top, \bot\}$, such that $|L_1| = 1$ (the root layer) and $L_{|X|+1}$ is the terminal layer.
- Arcs only extend to layers with higher index, i.e. $\forall (u, v) \in A$, $l(u) < l(v)$.
- Every node $u \in U$ is the tail of exactly two differently labeled arcs, i.e. $\delta^{\text{out}}(u) = \{(u, v_0), (u, v_1)\}$ and $d((u, v_0)) = 0, d((u, v_1)) = 1$.
- For any two distinct nodes u, v of B, the sub-BDDs rooted at u and v are not isomorphic, i.e. $B_u \ncong B_v$. Here B_u denotes the sub-BDD of B defined by the sub-dag of $(U \cup \{\top, \bot\}, A)$ rooted at u, with node and arc label functions obtained by restriction of l and d to $V(B_u) - \{\top, \bot\}$ and $A(B_u)$, respectively. BDD-isomorphism $\cong$ is defined as isomorphism of directed graphs with identical variable set and identically evaluating functions l and d.

For each node $u \in U$, the outgoing arc labeled by 1 is called the *true-arc*, and the outgoing arc labeled by 0 is called the *false-arc*, denoted by $t(u)$ and $f(u)$, respectively. Note that BDDs according to this definition are called *reduced, ordered BDD* in the classical literature (Lee 1959; Bryant 1986; Meinel and Theobald 1998).

For $i = 1, \dots, |X|$, the *layer width* w_i of layer L_i is defined as $w_i = |L_i|$. The *BDD-width* is then $w = \max\{w_i \ : \ i \in \{1, \dots, |X|\}\}$. The *total size* of a BDD is $|U| = \sum_{i=1,\dots,|X|} w_i$.

BDDs can be used to represent solutions to Boolean programming problems, or Boolean formulas. The interpretation is as follows: every directed path P from the root of the BDD to the leaf $\top$ defines a family of feasible solutions $x(P)$. Precisely, for each $i \in X$ such that $P \cap \delta^{out}(L_i) = (u, v)$ we have $x(P)_i = d((u, v))$, while for each $i \in X$ such that $P \cap \delta^{out}(L_i) = \emptyset$, $x(P)_i \in \{0, 1\}$. Analogously, paths from the root to $\bot$ define families of infeasible solutions.

Moreover, for any directed path P from the root of the BDD to $u \in U$, we denote by $S(P, u)$ the set of variables that have been set to 1 in P, and by $\bar{S}(P, u) = \{1, \dots, i - 1\} - S(P, u)$.

The feasible points of a problem like (1) can be encoded as Boolean formulas, and hence as BDDs (of possibly exponential size). It is well-known that the universal procedure (Bryant 1986) to construct such a BDD from a Boolean formula may have a run time not even (output-)polynomial in the size of this BDD, and this construction hurdle often is a second severe limitation on the applicability of BDDs.

In (Bergman, van Hoeve, and Hooker 2011) and (Bergman et al. 2012) it is shown that for stable set problems and set covering problems it is possible to construct BDDs in time output-polynomial in their size, by a single top-down traversal construction pass. The crucial step in this construction consists in a necessary and sufficient condition to determine whether two nodes of the BDD have identical sub-dags, i.e. if they have to be merged. Such conditions are presented in the next two Lemmata.

Lemma 1 (see (Bergman et al. 2012), Theorem 1). Let $G = (V, E)$ be an undirected graph and let B be the BDD encoding the stable sets of G, according to the ordering defined on V. Let $u \in L_i$. Then for any two directed paths P and P' from the root of B to u it holds that $N(S(P, u)) - V_{i-1} = N(S(P', u)) - V_{i-1}$.

In other words, each node u of the BDD is labeled uniquely by the set of neighbors of a stable set S encoded in an arbitrary directed path P from the root of B to u. Since S can not be augmented by any of its neighbors, we will denote this set of 'forbidden' nodes by $F(u) = N(S(P, u)) - V_{i-1}$. Hence Lemma 1 immediately implies the existence of a top-down BDD compilation algorithm for set packing problems. Let $A \in \{0, 1\}^{m \times n}$ be the constraint matrix of a set packing problem and define the *intersection graph* G_A as the graph whose nodes correspond to the columns of A, and having an edge linking two nodes if and only if the corresponding columns of A both have a nonzero entry in a common row. The solution set of the set packing problem defined on A is the same as the set of stable sets on the *intersection graph* G_A.

Now let $A \in \{0, 1\}^{m \times n}$ be the constraint matrix of a set covering problem, and let $\mathcal{C}_A$ be the associated clutter with ground set $E(\mathcal{C}_A) = \{1, \dots, n\}$, and such that each (minimal) row of A is the incidence vector of a member of $\mathcal{C}_A$.

Lemma 2 (see (Bergman, van Hoeve, and Hooker 2011), Section 3.1). Let $A \in \{0, 1\}^{m \times n}$ and let B be the BDD encoding the solutions of the set covering problem on A, according to the variable ordering defined by $X = \{1, \dots, n\}$. Then for any two directed paths P and P' from the root of B to u it holds that $\mathcal{C}_A \setminus S(P,u)/\bar{S}(P,u) = \mathcal{C}_A \setminus S(P',u)/\bar{S}(P',u)$.

In other words, each node u of the BDD is labeled uniquely by the minor obtained by deletion and contraction of the variables fixed to 1 and to 0, respectively, in any directed path from the root to u. Thus, for a node $u \in U$ we will denote by $M(u)$ the minor $\mathcal{C}_A \setminus S(P,u)/\bar{S}(P,u)$, where P is an arbitrary directed path from the root to u.

The duality between independence systems and clutters has the consequence that a BDD $B = (X, U, A, l, d)$ encoding the members of the independence system $\mathcal{I}$ immediately yields the BDD B' encoding the transversals of the circuit system $\mathcal{C}_{\mathcal{I}}$ (and vice versa): $B' = (X, U, A, l, d')$ where $d'((u,v)) = 1 - d((u,v))$, i.e. only the arc labels need to be complemented.

With these observations and Lemma 2 the complexity of constructing a BDD for the members of any independence system $\mathcal{I}$ (equivalently, for the transversals of $\mathcal{C}_{\mathcal{I}}$) is delegated to that of deciding whether two minors of its circuit system $\mathcal{C}_{\mathcal{I}}$ are the same:

Lemma 3. Let $\mathcal{I}$ be an independence system on ground set E. Let two pairs of deletion and contraction sets $(D_1, C_1) \subseteq E^2$ and $(D_2, C_2) \subseteq E^2$ be given. Given an oracle that decides equality of the minors $\mathcal{C}_{\mathcal{I}} \setminus D_1/C_1$ and $\mathcal{C}_{\mathcal{I}} \setminus D_2/C_2$, one can construct a BDD for the members of $\mathcal{I}$ in output-linear time.

It should be noted that one only needs to be able to check equality of minors using the deletion and contraction sets; we need not require knowledge of the full circuit system.

For some problems it is easy to define such an oracle: For the stable set problem we can explicitly access the edge-node incidence matrix and compute minors in it. The BDD obtained by using the circuit labeling of Lemma 2 is then isomorphic to that obtained by the labeling defined in Lemma 1, but the label sets obtained from the two methods will be different (sets of forbidden nodes versus minors of the edge-node incidence matrix). Similarly, such an oracle can be defined to encode all matchings in a graph G, since matchings in G correspond to stable sets in the line graph of G. For other problems, like the knapsack problem, the oracle will have to answer an $\mathcal{NP}$-hard question: If we could check equality of two minors of the circuit system of the knapsack problem, we could e.g. solve the 2-partition problem (which can be solved if, for a knapsack constraint with sum of the coefficients equal to b, we can decide whether the set of solutions is the same for right-hand-side $\frac{b}{2}$ and $\frac{b}{2} - 1$).

2 BDD-width for encodings of stable sets of simple classes of graphs

It has been shown previously (Bergman et al. 2012) that if a graph G is a path or a clique then there exist orderings of the nodes of G for which the width of the exact reduced BDD encoding the stable sets of G is 2. Furthermore, there is a universal bound:

Theorem 1. (Bergman et al. 2012). Let G be a graph. Then there exists an ordering of the nodes of G such that each layer L_i of the exact reduced BDD encoding the stable sets of G has width $w_i \leq F_{i+1}$, where F_k is the k-th Fibonacci number.

However, it is possible to define a 'bad ordering' such that, even for a path, the BDD encoding the stable sets may have a width of similar magnitude: For a path on n nodes there exists an ordering such that the BDD has $w \geq F(n/2)$.

Complete bipartite graphs $K_{m,n}$

Theorem 2. Let $G = K_{m,n}$ be the complete bipartite graph, with $m \leq n$. Then there exists an ordering of the nodes of G for which the width of the exact reduced BDD encoding the stable sets of G is at most m.

Proof. Let V_1 and V_2 denote the two shores of $K_{m,n}$, with $|V_1| = m$ and $|V_2| = n$, and suppose that the nodes of G are ordered in such a way that $V_1 = \{1, \dots, m\}$ and $V_2 = \{m+1, \dots, m+n\}$. We prove the claim by induction on m.

The base case is $m = 1$, i.e. the case where G is a star. Clearly, $w_1 = 1$. Let $u \in L_1$ be the root of the BDD. Then $f(u) = (u, \top)$, as any subset of $\{2, \dots, n+1\}$ is a stable set of G. Let $t(u) = (u, v)$, with $v \in L_2$. Note that the sub-BDD B_v rooted at v has width 1, because $N(1) = \{2, \dots, n+1\}$, i.e. for any node $v_i \in L_i$, $i = 2, \dots, n+1$, $t(v_i) = (v_i, \bot)$. This proves that if $m = 1$ then $w = 1$.

For $m \geq 2$, let $u \in L_1$ be the root of the BDD. Let $v \in L_2$ be such that $t(u) = (u, v)$. Note that the sub-BDD B_v rooted at v has width 1, because $N(1) = \{m+1, \dots, m+n\}$, i.e. any subset of $\{1, \dots, m\}$ is stable and, for each node $v_i \in L_i$, $i \in \{m+1, \dots, m+n\}$, $t(v_i) = (v_i, \bot)$. Let $v' \in L_2$ be such that $f(u) = (u, v')$. The sub-BDD $B_{v'}$ that is rooted at v' encodes the stable sets of $K_{m-1,n}$ that, by the induction hypothesis, has width at most $m - 1$. This proves that the width of the BDD encoding the stable sets of $K_{m,n}$ cannot exceed m.

□

Corollary 1. Let G be a star. Then there exists an ordering of the nodes of G for which the width of the exact reduced BDD encoding the stable sets of G is at most 1.

Cycles

Theorem 3. Let G be a cycle. Then there exists an ordering of the nodes of G for which the width of the exact reduced BDD encoding the stable sets of G is at most 4.

Proof. Let $V(G) = \{1, \dots, n\}$ and suppose that the nodes of G are ordered in such a way that $(1, n) \in E(G)$ and $(i, i+1) \in E(G)$ for all $i = 1, \dots, n-1$. Consider any layer L_i of the BDD B encoding the stable sets of G. Clearly, $w_1 = 1$ and $w_2 = 2$. Moreover, $w_n = 1$, because for $u \in L_n$ either $n \in F(u)$, implying $t(u) = (u, \bot)$ and $f(u) = (u, \top)$, or $n \notin F(u)$, implying $u \notin V(B)$.

For $3 \leq i \leq n-1$, the nodes in L_i can have at most 4 possible labels. In fact, the only nodes in $\{i, \dots, n\}$ that can be linked to nodes in $\{1, \dots, i-1\}$ are i and n, i.e. for each $u \in L_i$, $F(u) \subseteq \{i, n\}$. Therefore a node $u \in L_i$ can be labeled in at most four different ways, depending on i and n belonging to $F(u)$, that implies $w_i \leq 4$. □

graph $G = (V, E)$	no. stable sets $F(G)$	BDD width
star	$2^{\|V\|-1} + 1$	1
complete bipartite $K_{m,n}$	$2^m + 2^n - 1$	$\min\{m, n\}$
cycle	$\mathcal{L}_n$	4
cycle with chord	$\leq \mathcal{L}_n$	4
wheel	$\mathcal{L}_{n-1} + 1$	5
tree	$F_{n+2} \leq F(G) \leq 2^{\|V\|-1} + 1$	$\frac{\|V\|}{2}$

Table 1: Width bounds for stable sets of specific graphs. F_n denoted the n-th Fibonacci number and $\mathcal{L}_n = F_{n+1} + F_{n-1}$ the n-th Lucas number.

Theorem 3 can be easily extended to the case where G is a cycle with one or more chords that are all incident to a common node of G. Assume this is node 1. Then the arguments of Theorem 3 still hold for the same ordering of the nodes. Now for some $u \in L_i$ we can have that $F(u) \subseteq \{i, j_h, \ldots, j_k, n\}$, because $(1, j_h), \ldots, (1, j_k)$ are chords of the cycle and $i \leq j_h$. However, the decisions on $\{j_h, \ldots, j_k, n\}$ all depend simultaneously on the decision taken on 1, implying that the only four possible labels for node u are $\{i, j_h, \ldots, j_k, n\}$, $\{i\}$, $\{j_h, \ldots, j_k, n\}$ and $\emptyset$.

Corollary 2. Let G be a cycle with multiple chords, that are all incident to a common node of the cycle. Then, there exists an ordering of the nodes of G for which the width of the exact reduced BDD is at most 4.

Wheels

Theorem 4. Let G be a wheel. Then there exists an ordering of the nodes of G for which the width of the exact reduced BDD encoding the stable sets of G is at most 5.

Proof. Let $V(G) = \{1, \ldots, n\}$ and suppose that the nodes of G are ordered in such a way that $(1, i) \in E(G)$ for all $i = 2, \ldots, n$, $(2, n) \in E(G)$ and $(i, i+1) \in E(G)$ for all $i = 2, \ldots, n-1$.

Let $u \in L_1$ be the root of the BDD and let $v \in L_2$ be such that $t(u) = (u, v)$. The sub-BDD B_v rooted at v has width 1, because $N(1) = \{2, \ldots, n\}$, i.e. for each node $v_i \in L_i$, $i \in \{2, \ldots, n\}$, $t(v_i) = (v_i, \perp)$. Let $v' \in L_2$ be such that $f(u) = (u, v')$. The sub-BDD $B_{v'}$ that is rooted at v' encodes the stable sets of a cycle on $n-1$ nodes, whose node ordering yields a BDD having width at most 4. This proves that the width of the BDD encoding the stable sets of G cannot exceed 5. □

Trees

We can improve the bound of (Bergman, van Hoeve, and Hooker 2011, Theorem 4) by using essentially the same proof, and by noting that the base cases of the induction actually have a better bound on the width.

Theorem 5. Let G be a tree on n nodes. Then there exists an ordering of the nodes of G for which the width of the exact reduced BDD encoding the stable sets of G is at most $\lfloor \frac{n}{2} \rfloor$.

3 Bandwidth-constrained packings and coverings

For a BDD encoding the stable sets of a graph G Lemma 1 implies that the width w_i of layer L_i is bounded by the number of possible labels of a node in L_i . If every node of G has its neighbors at most k indices distant from it in the node ordering, then there can be at most 2^{k-1} ways of constructing labels $F(u)$ for a node $u \in L_i$. Using Lemma 2 one can also argue similarly for the set covering case.

Let $A \in \{0, 1\}^{m \times n}$ be the incidence matrix of a clutter $\mathcal{C}_A$, i.e., each row of A is the incidence vector of a member of $\mathcal{C}_A$, implying that no two rows of A are comparable by componentwise $\leq$.

Definition 1 (bandwidth of a clutter, of a graph; minimum bandwidth problem)**.** The *bandwidth* $b(A)$ of A is the maximal distance between nonzeros in any row of A, i.e.

$$b(A) = \max_{i \in \{1, \ldots, m\}} \{j_2 - j_1 + 1 \ : \ A_{ij_1}, A_{ij_2} \neq 0, j_2 \geq j_1\}.$$

For a graph G, the bandwidth is the bandwidth of its edge-node incidence matrix.

The *minimum bandwidth problem* is the problem of determining a permutation of the columns of A such that $b(A)$ attains the minimum value over all permutations of columns of A.

Theorem 6. Let $\mathcal{C}_A$ be a nonempty clutter and suppose that $E(\mathcal{C}_A)$ is ordered in such a way that $b(A) = k$. Then, in the BDD that encodes the transversals of $\mathcal{C}_A$, the width of any layer is at most 2^{k-1}.

Proof. Let $E(\mathcal{C}_A) = \{1, \ldots, n\}$ and consider a node $u \in L_j$ for $j \in E(\mathcal{C}_A)$ in the BDD encoding the transversals of $\mathcal{C}_A$. First note that, since $\mathcal{C}_A$ is nonempty, each row of A has at least one nonzero entry. If the i-th row of A has all its nonzero entries before column j, then the row is deleted in $M(u)$. In fact, if this was not the case, no directed path from the root of the BDD to u could be extended to $\top$: any transversal of $\mathcal{C}_A$ must have non-empty intersection with the i-th member of $\mathcal{C}_A$, implying that it must contain an element $h \in E(\mathcal{C}_A)$ such that $h \leq j$ and $A_{ih} = 1$. By the bandwidth limit in every row i of A, if there is some $A_{ih} \neq 0$ for $h \geq j$, among the entries preceding j only those in $\{A_{i,j-(k-1)}, \ldots, A_{i,j-1}\}$ can be nonzero. Since there are at most 2^{k-1} ways of differently assigning values to the $k-1$ variables directly preceding x_j, we can construct at most 2^{k-1} different deletion/contraction minors, and therefore labels, at layer L_j. By Lemma 2 the BDD width is thus limited by 2^{k-1}. □

Corollary 3. Let $G = (V, E)$ be a graph with bandwidth k. Then the maximum width of any layer of the BDD encoding the stable sets of G is 2^{k-1}.

Since the construction of the intersection graph of a set packing problem preserves the bandwidth we find that:

Corollary 4. For packing and covering problems on n variables whose bandwidth is bounded by $\mathcal{O}(\log n)$ the maximum width of the BDD is bounded by $\mathcal{O}(n)$.

References

[Bergman et al. 2012] Bergman, D.; Cire, A. A.; van Hoeve, W.-J.; and Hooker, J. N. 2012. Variable ordering for the application of BDDs to the maximum independent set problem. In *Proceedings of the 9th international conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, CPAIOR'12, 34–49. Berlin, Heidelberg: Springer-Verlag.

[Bergman, van Hoeve, and Hooker 2011] Bergman, D.; van Hoeve, W.-J.; and Hooker, J. N. 2011. Manipulating MDD relaxations for combinatorial optimization. In *Proceedings of the 8th international conference on Integration of AI and OR techniques in constraint programming for combinatorial optimization problems*, CPAIOR'11, 20–35. Berlin, Heidelberg: Springer-Verlag.

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

[Haus, Köppe, and Weismantel 2003] Haus, U.-U.; Köppe, M.; and Weismantel, R. 2003. A primal all-integer algorithm based on irreducible solutions. *Math. Programming, Series B* 96(2):205–246.

[Henk, Köppe, and Weismantel 2003] Henk, M.; Köppe, M.; and Weismantel, R. 2003. Integral decomposition of polyhedra and some applications in mixed integer programming. *Mathematical Programming, Series B* 94(2–3):193–206.

[Hoda, van Hoeve, and Hooker 2010] Hoda, S.; van Hoeve, W. J.; and Hooker, J. N. 2010. A systematic approach to mdd-based constraint programming. In Cohen, D., ed., *CP*, volume 6308 of *Lecture Notes in Computer Science*, 266–280. Springer.

[Lee 1959] Lee, C. Y. 1959. Representation of switching circuits by binary-decision programs. *Bell Systems Technical Journal* 38(4):985–999.

[Meinel and Theobald 1998] Meinel, C., and Theobald, T. 1998. *Algorithms and Data Structures in VLSI Design: OBDD – Foundations and Applications*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

[Schrijver 2002] Schrijver, A. 2002. *Combinatorial Optimization*. Berlin, Germany: Springer.