A lower bound on DNNF encodings of pseudo-Boolean constraints

Alexis de Colnet

CRIL, CNRS & Univ Artois
decolnet@cril.fr

Abstract. Two major considerations when encoding pseudo-Boolean (PB) constraints into SAT are the size of the encoding and its propagation strength, that is, the guarantee that it has a good behaviour under unit propagation. Several encodings with propagation strength guarantees rely upon prior compilation of the constraints into DNNF (decomposable negation normal form), BDD (binary decision diagram), or some other sub-variants. However it has been shown that there exist PB-constraints whose ordered BDD (OBDD) representations, and thus the inferred CNF encodings, all have exponential size. Since DNNFs are more expressive than OBDDs, preferring encodings via DNNF to avoid size explosion seems a legitimate choice. Yet in this paper, we prove the existence of PB-constraints whose DNNFs all require exponential size.

Keywords: pseudo-Boolean constraint · knowledge compilation · DNNF

1 Introduction

Pseudo-Boolean (PB) constraints are Boolean functions over 0/1 Boolean variables $x_1, \dots, x_n$ of the form $\sum_{i=1}^{n} w_i x_i$ `'op'` θ where the w_i are integer weights, θ is an integer threshold and `'op'` is a comparison operator $<$, $\leq$, $>$ or $\geq$. PB-constraints have been studied extensively under different names (e.g. threshold functions [13], Knapsack constraints [12]) due their omnipresence in a many domains of AI and their wide range of practical applications [3, 5, 7, 14, 20].

One way to handle PB-constraints in a constraint satisfaction problem is to translate them into a CNF formula and feed it to a SAT solver. The general idea is to generate a CNF, possibly introducing auxiliary Boolean variables, whose restriction to variables of the constraint is equivalent to the constraint. Two major considerations here are the size of the CNF encoding and its propagation strength. One wants, on the one hand, to avoid the size of the encoding to explode, and on the other hand, to guarantee a good behaviour of the SAT instance under unit propagation – a technique at the very core of SAT solving. Desired propagation strength properties are, for instance, unit refutation completeness [9] or propagation completeness [4]. Many encodings to CNF follow the same two-steps method: first, each constraint is represented in a compact form such as BDD (Binary Decision Diagram) or DNNF (Decomposable Negation Normal Form). Second, the compact forms are turned into CNFs using Tseitin or other transformations. The SAT instance is then the conjunction of all obtained CNFs. Thus the first step of this approach is a knowledge compilation task.

Knowledge compilation studies different representations for knowledge [8, 18] under the general idea that some representations are more suitable than others when solving specific reasoning problems. One observation that has been made is that the more reasoning tasks can be solved efficiently with particular representations, the larger these representations get in size. In the context of constraint encodings to SAT, the conversion of compiled forms to CNFs does not reduce the size of the SAT instance, therefore it is essential to control the size of the representations obtained by knowledge compilation.

Several representations have been studied with respect to different encoding techniques with the purpose of determining which properties of a representation are sufficient to ensure propagation strength [1,2,10,11,15,16]. Popular representations in this context are DNNF and BDD and their many variants: deterministic DNNF, smooth DNNF, ordered BDD (OBDD)... As mentioned above, a problem occurring when compiling a constraint into such representations is that exponential space may be required. Most notably, it has been shown in [2, 13], that there are PB-constraints that can only be represented by OBDDs whose size is exponential in $\sqrt{n}$, where n is the number of variables. Our contribution in this paper is the proof of the following theorem where we lift the statement from OBDD to DNNF.

Theorem 1. *There is a class of PB-constraints $\mathcal{F}$ such that for any constraint $f \in \mathcal{F}$ on n^2 variables, any DNNF representation of f has size $2^{\Omega(n)}$.*

Since DNNFs are exponentially more succinct than OBDDs [8], our result is a generalisation of the result in [2,13]. The class $\mathcal{F}$ is similar to that used in [2,13], actually the only difference is the choice of the threshold for the PB-constraints. Yet, adapting proofs given in [2, 13] for OBDD to DNNF is not straightforward, thus, our proof of Theorem 1 bears very little resemblance.

It has been shown in [17] that there exist sets of PB-constraints such that the whole *set* (so a conjunction of PB-constraints) require exponential size DNNF to represent. Our result is a generalisation to *single* PB-constraints.

2 Preliminaries

Conventions of notation. Boolean variables are seen as variables over $\{0, 1\}$, the values 1 and 0 representing *true* and *false* respectively. Via this 0/1 representation, Boolean variables can be used in arithmetic expressions over $\mathbb{Z}$. For notational convenience, we keep the usual operators $\neg$, $\vee$ and $\wedge$ to denote, respectively, the negation, disjunction and conjunction of Boolean variables or functions. Given X a set of n Boolean variables, assignments to X are seen as vectors in $\{0, 1\}^n$. Single Boolean variables are written in plain text (x) while assignments to several variables are written in bold ($\mathbf{x}$). We write $\mathbf{x} \leq \mathbf{y}$ when the vector $\mathbf{y}$ dominates $\mathbf{x}$ element-wise. We write $\mathbf{x} < \mathbf{y}$ when $\mathbf{x} \leq \mathbf{y}$ and $\mathbf{x} \neq \mathbf{y}$. In this framework, a Boolean function f over X is a mapping from $\{0, 1\}^n$ to

$\{0,1\}$. f is said to *accept* an assignment $\mathbf{x}$ when $f(\mathbf{x}) = 1$, then $\mathbf{x}$ is called a *model* of f. The function is *monotone* if for any $\mathbf{x}$ model of f, all $\mathbf{y} \geq \mathbf{x}$ are models of f as well. The set of models of f is denoted $f^{-1}(1)$. Given f and g two Boolean functions over X, we write $f \leq g$ when $f^{-1}(1) \subseteq g^{-1}(1)$. We write $f < g$ when the inclusion is strict.

Pseudo-Boolean constraints. *Pseudo-Boolean constraints* (PB-constraints for short), are inequalities the form $\sum_{i=1}^{n} w_i x_i$ `'op'` θ where the x_is are 0/1 Boolean variables, the w_i and θ are integers, and `'op'` is one of the comparison operator $<$, $\leq$, $>$ or $\geq$. A PB-constraint will be associated with a Boolean function f, where the models of f are exactly the assignments to $\{x_1, \ldots, x_n\}$ that satisfy the inequality. For simplicity, we will directly consider PB-constraints as Boolean functions – although the same Boolean function may represent different constraints – while keeping the term "constraints" when referring to them. In this paper, we restrict our attention to PB-constraints where `'op'` is $\geq$ and all weights are positive integers. Note that such PB-constraints are monotone Boolean functions. Given a sequence of positive integer weights $W = (w_1, \ldots, w_n)$ and an integer threshold θ, we define the function $w : \{0,1\}^n \to \mathbb{N}$ that maps any assignment to its weight by $w(\mathbf{x}) = \sum_{i=1}^{n} w_i x_i$. With these notations, a PB-constraints over X for a given pair (W, θ) is a Boolean function whose models are exactly the $\mathbf{x}$ such that $w(\mathbf{x}) \geq \theta$.

Example 1. Let $n = 5$, $W = (1, 2, 3, 4, 5)$ and $\theta = 9$. The PB-constraint for (W, θ) is the Boolean function whose models are the assignments such that $\sum_{i=1}^{5} i x_i \geq 9$. E.g. $\mathbf{x} = (0, 1, 1, 0, 1)$ is a model of weight $w(\mathbf{x}) = 10$.

For notational clarity, given any subset $Y \subseteq X$ and denoting $\mathbf{x}|_Y$ the restriction of $\mathbf{x}$ to variables of Y, we overload w so that $w(\mathbf{x}|_Y)$ is the sum of weights activated by variables in Y set to 1 in $\mathbf{x}$.

Decomposable NNF (DNNF). A circuit in *negation normal form* (NNF), is a single output Boolean circuit whose inputs are Boolean variables and their complements, and whose gates are fanin-2 AND and OR gates. The *size* of the circuit is the number of its gates. We say that an NNF is *decomposable* (DNNF) if for any AND gate, the two sub-circuits rooted at that gate share no input variable, i.e., if x or $\neg x$ is an input of the circuit rooted at the left input of the AND gate, then neither x nor $\neg x$ is an input gate of the circuit rooted at the right input, and vice versa. A circuit is *monotone* if its inputs are non-negated Boolean variables only. We say that a Boolean function f is *encoded* by a DNNF D (or D *represents* f) if the assignments of variables for which the output of D is 1 (*true*) are exactly the models of f.

Rectangle covers. Let X be a finite set of Boolean variables and let $\Pi = (X_1, X_2)$ be a partition of X (i.e., $X_1 \cup X_2 = X$ and $X_1 \cap X_2 = \emptyset$). A *rectangle* r with respect to Π is any Boolean function over X defined as the conjunction of two Boolean functions ρ_1 and ρ_2 over X_1 and X_2 respectively. Π is called the *partition* of the rectangle. The partition and the rectangle are called *balanced* if $\frac{|X|}{3} \leq |X_1| \leq \frac{2|X|}{3}$ (thus the same holds for X_2). Whenever considering a partition (X_1, X_2), we use for any assignment $\mathbf{x}$ to X the notation $\mathbf{x}_1 \coloneqq \mathbf{x}|_{X_1}$ and $\mathbf{x}_2 \coloneqq \mathbf{x}|_{X_2}$. And for any two assignments $\mathbf{x}_1$ and $\mathbf{x}_2$ to X_1 and X_2, we note $(\mathbf{x}_1, \mathbf{x}_2)$ the assignment to X whose restrictions to X_1 and X_2 are $\mathbf{x}_1$ and $\mathbf{x}_2$. Given f a Boolean function over X, a *rectangle cover* of f is a disjunction of rectangles over X, possibly with different partitions, equivalent to f. The *size* of a rectangle cover is the number of its rectangles. A cover is called *balanced* if all its rectangles are balanced. Any function f has at least one balanced rectangle cover as one can create a balanced rectangle accepting exactly one chosen model of f. We denote by $C(f)$ the size of the smallest balanced rectangle cover of f. The following result from [6] links $C(f)$ to the size of any DNNF encoding f.

Theorem 2. *Let D be a DNNF encoding a Boolean function f. Then f has a balanced rectangle cover of size at most the size of D.*

Theorem 2 reduces the problem of finding lower bounds on the size of DNNFs encoding f to that of finding lower bounds on $C(f)$.

3 Restriction to threshold models of PB-constraints

The strategy to prove Theorem 1 is to find a PB-constraint f over n variables such that $C(f)$ is exponential in $\sqrt{n}$ and then use Theorem 2. We first show that we can restrict our attention to covering particular models of f with rectangles rather than the whole function. In this section X is a set of n Boolean variables and f is a PB-constraint over X. Recall that we only consider constraints of the form $\sum_{i=1}^{n} w_i x_i \geq \theta$ where the w_i and θ are positive integers.

Definition 1. *The* threshold models *of f are the models $\mathbf{x}$ such that $w(\mathbf{x}) = \theta$.*

Threshold models should not be confused with minimal models (or minimals).

Definition 2. *A* minimal *of f is a model $\mathbf{x}$ such that no $\mathbf{y} < \mathbf{x}$ is a model of f.*

For a monotone PB-constraint, a minimal model is such that removing any element from the sum of weights makes it pass below the threshold. Any threshold model is minimal, but not all minimals are threshold models. There even exist constraints with no threshold models (e.g. take even weights and an odd threshold) while there always are minimals for satisfiable constraints.

Example 2. The minimals of the PB constraint described in Example 1 are $(0,0,0,1,1)$, $(0,1,1,1,0)$, $(1,0,1,0,1)$ and $(0,1,1,0,1)$. But only the first three are threshold models.

Let f^* be the Boolean function whose models are exactly the threshold models of f. Note that in general, f^* is no longer monotone. In the next lemma, we prove that smallest rectangle cover of f^* has size at most $C(f)$, and thus lower bounds on $C(f^*)$ are also lower bounds on $C(f)$.

Lemma 1. *Let f^* be the Boolean function whose models are exactly the threshold models of f. Then $C(f) \geq C(f^*)$.*

Proof. Let $r \coloneqq \rho_1 \wedge \rho_2$ be a balanced rectangle with $r \leq f$ and assume r accepts some threshold models. Let $\Pi \coloneqq (X_1, X_2)$ be the partition of r. We claim that there exist two integers θ_1 and θ_2 such that $\theta_1 + \theta_2 = \theta$ and for any threshold model $\mathbf{x}$ accepted by r, there is $w(\mathbf{x_1}) = \theta_1$ and $w(\mathbf{x_2}) = \theta_2$. To see this, assume by contradiction that there exists another partition $\theta = \theta'_1 + \theta'_2$ of θ such that some other threshold model $\mathbf{y}$ with $w(\mathbf{y}_1) = \theta'_1$ and $w(\mathbf{y}_2) = \theta'_2$ is accepted by r. Then either $w(\mathbf{x_1}) + w(\mathbf{y_2}) < \theta$ or $w(\mathbf{y_1}) + w(\mathbf{x_2}) < \theta$, but since $(\mathbf{x}_1, \mathbf{y}_2)$ and $(\mathbf{y}_1, \mathbf{x}_2)$ are also models of r, r would accept a non-model of f, which is forbidden. Now let ρ_1^* (resp. ρ_2^*) be the function whose models are exactly the models of ρ_1 (resp. ρ_2) of weight θ_1 (resp. θ_2). Then $r^* \coloneqq \rho_1^* \wedge \rho_2^*$ is a balanced rectangle whose models are exactly the threshold models accepted by r.

Now consider a balanced rectangle cover of f of size $C(f)$. For each rectangle r of the cover, if r accepts no threshold model then discard it, otherwise construct r^*. The disjunction of these new rectangles is a balanced rectangle cover of f^* of size at most $C(f)$. Therefore $C(f) \geq C(f^*)$. □

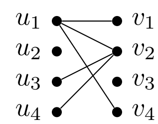
4 Reduction to covering $K_{n,n}$'s maximal matchings

We define the class of hard PB-constraints for Theorem 1 in this section. Recall that for a hard constraint f, our aim is to find an exponential lower bound on $C(f)$. We will show, using Lemma 1, that the problem can be reduced to that of covering all maximal matchings of the complete $n \times n$ bipartite graph $K_{n,n}$ with rectangles.

In this section, X is a set of n^2 Boolean variables. For presentability reasons, assignments to X are written as $n \times n$ matrices. Each variable $x_{i,j}$ has the weight $w_{i,j} \coloneqq (2^i + 2^{j+n})/2$. Define the collection of weights $W \coloneqq \{w_{i,j} : 1 \leq i, j \leq n\}$ and the threshold $\theta \coloneqq 2^{2n} - 1$. The PB-constraint f for the pair (W, θ) is such that $f(\mathbf{x}) = 1$ if and only if $\mathbf{x}$ satisfies

$$\sum_{1 \leq i,j \leq n} \left(\frac{2^i + 2^{j+n}}{2} \right) x_{i,j} \geq 2^{2n} - 1 \ . \tag{1}$$

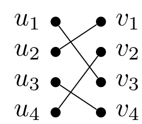
Constraints of this form constitute the class of hard constraints of Theorem 1. One may find it easier to picture f writing the weights and threshold as binary numbers of $2n$ bits. Bits of indices 1 to n form the *lower part* of the number and those of indices $n+1$ to $2n$ form the *upper part*. The weight $w_{i,j}$ is the binary number where the only bits set to 1 are the ith bit of the lower part and the jth bit of the upper part. Thus when a variable $x_{i,j}$ is set to 1, exactly one bit of value 1 is added to each part of the binary number of the sum. Assignments to X uniquely encode subgraphs of $K_{n,n}$. We denote $U = \{u_1, \ldots, u_n\}$ the nodes of the left side and $V = \{v_1, \ldots, v_n\}$ those of the right side of $K_{n,n}$. The bipartite graph encoded by $\mathbf{x}$ is such that there is an edge between the u_i and v_j if and only if $x_{i,j} = 1$.

Example 3. Take $n = 4$. The assignment $\mathbf{x} = \begin{pmatrix} 1\,1\,0\,1 \\ 0\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,1\,0\,0 \end{pmatrix}$ encodes 

Definition 3. *A* maximal matching assignment *(or maximal matching model) is an assignment* $\mathbf{x}$ *to* X *such that*

- *for any* $i \in [n]$*, there is exactly one* k *such that* $x_{i,k}$ *is set to* 1 *in* $\mathbf{x}$*,*
- *for any* $j \in [n]$*, there is exactly one* k *such that* $x_{k,j}$ *is set to* 1 *in* $\mathbf{x}$*.*

As the name suggests, maximal matching assignments are those encoding bipartite graphs whose edges form a maximal matching of $K_{n,n}$ (i.e., a maximum cardinality matching).

Example 4. The maximal matching model $\mathbf{x} = \begin{pmatrix} 0\,0\,1\,0 \\ 1\,0\,0\,0 \\ 0\,0\,0\,1 \\ 0\,1\,0\,0 \end{pmatrix}$ encodes 

For an assignment $\mathbf{x}$, denote by $\text{var}_k(\mathbf{x}) := \{j \mid x_{k,j} \text{ is set to 1 in } \mathbf{x}\}$ when $1 \leq k \leq n$ and $\text{var}_k(\mathbf{x}) := \{i \mid x_{i,k-n} \text{ is set to 1 in } \mathbf{x}\}$ when $n + 1 \leq k \leq 2n$. Note that a maximal matching model is an assignment $\mathbf{x}$ such that $|\text{var}_k(\mathbf{x})| = 1$ for all k. It is easy to see that maximal matching models are threshold models of the function f: seeing weights as binary numbers of size $2n$, for every bit in the sum of weights the value 1 is added exactly once, so all $2n$ first bits of the sum are set to 1, which gives us θ. Note that not all threshold models of f are maximal matching assignments, for instance the assignment from Example 3 does not encode a maximal matching but one can verify that it is a threshold model. Recall that f^* is the function whose models are the threshold models of f. In the next lemmas, we prove that a lower bound on the size of rectangle covers of the maximal matching models is also a lower bound on $C(f^*)$, and a fortiori on $C(f)$.

Lemma 2. *Let* $\Pi = (X_1, X_2)$ *be a partition of* X*. Let* $\mathbf{x} := (\mathbf{x}_1, \mathbf{x}_2)$ *and* $\mathbf{y} := (\mathbf{y}_1, \mathbf{y}_2)$ *be maximal matching assignments. If* $(\mathbf{x}_1, \mathbf{y}_2)$ *and* $(\mathbf{y}_1, \mathbf{x}_2)$ *both have weight* $\theta := 2^{2n} - 1$ *then both are maximal matching assignments.*

Proof. It is sufficient to show that $|\text{var}_k(\mathbf{x}_1, \mathbf{y}_2)| = 1$ and $|\text{var}_k(\mathbf{y}_1, \mathbf{x}_2)| = 1$ for all $1 \leq k \leq 2n$. We prove it for $(\mathbf{x}_1, \mathbf{y}_2)$ by induction on k. First observe that since $|\text{var}_k(\mathbf{x})| = 1$ and $|\text{var}_k(\mathbf{y})| = 1$ for all $1 \leq k \leq 2n$, the only possibilities for $|\text{var}_k(\mathbf{x}_1, \mathbf{y}_2)|$ are 0, 1 or 2.

- For the base case $k = 1$, if $|\text{var}_1(\mathbf{x}_1, \mathbf{y}_2)|$ is even then the first bit of $w(\mathbf{x}_1) + w(\mathbf{y}_2)$ is 0 and the weight of $(\mathbf{x}_1, \mathbf{y}_2)$ is not θ. So $|\text{var}_1(\mathbf{x}_1, \mathbf{y}_2)| = 1$.
- For the general case $1 < k \leq 2n$, assume that $|\text{var}_1(\mathbf{x}_1, \mathbf{y}_2)| = \cdots = |\text{var}_{k-1}(\mathbf{x}_1, \mathbf{y}_2)| = 1$. So the kth bit of $w(\mathbf{x}_1) + w(\mathbf{y}_2)$ depends only on the parity of $|\text{var}_k(\mathbf{x}_1, \mathbf{y}_2)|$: the kth bit is 0 if $|\text{var}_k(\mathbf{x}_1, \mathbf{y}_2)|$ is even and 1 otherwise. $(\mathbf{x}_1, \mathbf{y}_2)$ has weight θ so $|\text{var}_k(\mathbf{x}_1, \mathbf{y}_2)| = 1$.

The argument applies to $(\mathbf{y}_1, \mathbf{x}_2)$ analogously. ☐

Lemma 3. *Let f be the PB-constraint (1) and let $\hat{f}$ be the function whose models are exactly the maximal matching assignments. Then $C(f) \geq C(\hat{f})$.*

Proof. Recall that f^* is the function whose models are exactly the threshold models of f. We know that $\hat{f} \leq f^*$. By Lemma 1, it is sufficient to prove that $C(f^*) \geq C(\hat{f})$. Let $r \coloneqq \rho_1 \wedge \rho_2$ be a balanced rectangle of partition $\Pi \coloneqq (X_1, X_2)$ with $r \leq f^*$, and assume r accepts some maximal matching assignment. Let $\hat{\rho}_1$ (resp. $\hat{\rho}_2$) be the Boolean function over X_1 (resp. X_2) whose models are the $\mathbf{x}_1$ (resp. $\mathbf{x}_2$) such that there is a maximal matching assignment $(\mathbf{x}_1, \mathbf{x}_2)$ accepted by r. We claim that the balanced rectangle $\hat{r} \coloneqq \hat{\rho}_1 \wedge \hat{\rho}_2$ accepts exactly the maximal matching models of r. On the one hand, it is clear that all maximal matching models of r are models of $\hat{r}$. On the other hand, all models of $\hat{r}$ are threshold models of the form $(\mathbf{x}_1, \mathbf{y}_2)$, where $(\mathbf{x}_1, \mathbf{x}_2)$ and $(\mathbf{y}_1, \mathbf{y}_2)$ encode maximal matchings, so by Lemma 2, $\hat{r}$ accepts only maximal matching models of r.

Now consider a balanced rectangle cover of f^* of size $C(f^*)$. For each rectangle r of the cover, if r accepts no maximal matching assignment then discard it, otherwise construct $\hat{r}$. The disjunction of these new rectangles is a balanced rectangle cover of $\hat{f}$ of size at most $C(f^*)$. Therefore $C(f^*) \geq C(\hat{f})$. □

5 Proof of Theorem 1

Theorem 1. *There is a class of PB-constraints $\mathcal{F}$ such that for any constraint $f \in \mathcal{F}$ on n^2 variables, any DNNF encoding f has size $2^{\Omega(n)}$.*

$\mathcal{F}$ is the class of constraints defined in (1). Thanks to Theorem 2 and Lemma 3, the proof boils down to finding exponential lower bounds on $C(\hat{f})$, where $\hat{f}$ is the Boolean function on n^2 variables whose models encode exactly the maximal matchings of $K_{n,n}$. $\hat{f}$ has $n!$ models, the idea is now to prove that rectangles covering $\hat{f}$ must be relatively small, so that covering the whole function requires many of them.

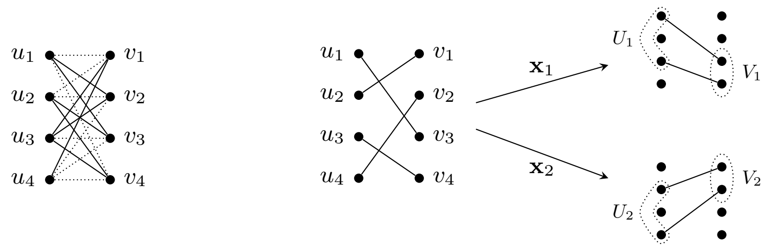
Lemma 4. *Let $\Pi = (X_1, X_2)$ be a balanced partition of X. Let r be a rectangle with respect to Π with $r \leq \hat{f}$. Then there is $|r^{-1}(1)| \leq n!/\binom{n}{n\sqrt{2/3}}$.*

The function $\hat{f}$ has already been studied extensively in the literature, see for instance section 6.2 of [19] where statement similar to Lemma 4 has been established. With Lemma 4 we can give the proof of Theorem 1.

Proof (Theorem 1). Let $\bigvee_{k=1}^{C(\hat{f})} r_k$ be a balanced rectangle cover of $\hat{f}$. We have that $\sum_{k=1}^{C(\hat{f})} |r_k^{-1}(1)| \geq |\hat{f}^{-1}(1)| = n!$. Lemma 4 gives us $C(\hat{f}) \times n!/\binom{n}{n\sqrt{2/3}} \geq n!$, and finally

$$C(\hat{f}) \geq \binom{n}{n\sqrt{2/3}} \geq \left(\frac{n}{n\sqrt{2/3}}\right)^{n\sqrt{2/3}} = \left(\frac{3}{2}\right)^{n\frac{\sqrt{2/3}}{2}} \geq 2^{n\frac{\sqrt{2/3}}{4}} = 2^{\Omega(n)}$$

where we have used $\binom{a}{b} \geq (a/b)^b$ and $3/2 \geq \sqrt{2}$. Using Lemma 3 we get that $C(f) \geq C(\hat{f}) \geq 2^{\Omega(n)}$. Theorem 2 allows us to conclude. □



(a) Balanced partition Π of $K_{4,4}$ (b) Partition of a maximal matching w.r.t. Π

Fig. 1: Partition of maximal matching

All that is left is to prove Lemma 4.

Proof (Lemma 4). Let $r \coloneqq \rho_1 \wedge \rho_2$. Let $\mathbf{x} \coloneqq (\mathbf{x}_1, \mathbf{x}_2)$ be a model of r. Then $\mathbf{x}_1$ and $\mathbf{x}_2$ both encode matchings of $K_{n,n}$. Recall that $U \coloneqq \{u_1, \dots, u_n\}$ and $V \coloneqq \{v_1, \dots, v_n\}$ are the nodes from the left and right part of $K_{n,n}$ respectively. Define $U_1 \coloneqq \{u_i \mid \text{there exists } x_{i,k} \text{ set to 1 in } \mathbf{x_1}\}$ and $V_1 \coloneqq \{v_j \mid \text{there exists } x_{k,j}$ set to 1 in $\mathbf{x_1}\}$. Note that $\mathbf{x}_1$ encodes a maximal matching between nodes of U_1 and nodes of V_1, so $|U_1| = |V_1| \coloneqq k$. Analogously we find that $\mathbf{x}_2$ encodes a maximal matching between $U_2 \coloneqq U \setminus U_1$ and $V_2 \coloneqq V \setminus V_1$. Figure 1 illustrates the construction of U_1, V_1, U_2 and V_2: a balanced partition Π of the edges of $K_{4,4}$ is shown Figure 1a (full edges form X_1, dotted edges form X_2) and the partitioning of a maximal matching according to Π is shown Figure 1b.

We claim that all models of ρ_1 encode maximal matchings between U_1 and V_1. Indeed suppose there is $\mathbf{y}_1$ a model of ρ_1 that does not encode such a maximal matching. $\mathbf{y}_1$ must encode a matching of $K_{n,n}$ otherwise r makes a mistake: either $\mathbf{y}_1$ encodes a matching that forgets some node from U_1 or V_1, but then $(\mathbf{y}_1, \mathbf{x}_2)$ does not encode a maximal matching because some node is missing, or $\mathbf{y}_1$ encodes a matching that matches some node from U_2 or V_2, but then $(\mathbf{y}_1, \mathbf{x}_2)$ does not encode a matching because some node in has two adjacent edges. Thus models of ρ_1 encode maximal matchings between U_1 and V_1. Since maximal matchings are uniquely encoded by the models, there is $|\rho_1^{-1}(1)| \leq k!$. The argument works symmetrically to prove that $|\rho_2^{-1}(1)| \leq (n-k)!$, leading to $|r^{-1}(1)| \leq k!(n-k)! = n!/\binom{n}{k}$.

Up to k^2 edges may be used to build matchings between U_1 and V_1. Since r is balanced we obtain $k^2 \leq 2n^2/3$. The same argument applied to U_2 and V_2 gives us $(n-k)^2 \leq 2n^2/3$, so $n(1 - \sqrt{2/3}) \leq k \leq n\sqrt{2/3}$. Finally, the function $k \mapsto n!/\binom{n}{k}$, when restricted to some interval $[\![k_1, k_2]\!]$, reaches its maximum at $k = \max(n - k_1, k_2)$ so we obtain the upper bound $|r^{-1}(1)| \leq n!/\binom{n}{n\sqrt{2/3}}$. $\square$

Acknowledgments

This work has been partly supported by the PING/ACK project of the French National Agency for Research (ANR-18-CE40-0011).

References

1. Abío, I., Gange, G., Mayer-Eichberger, V., Stuckey, P.J.: On CNF encodings of decision diagrams. In: Quimper, C. (ed.) Integration of AI and OR Techniques in Constraint Programming - 13th International Conference, CPAIOR 2016, Banff, AB, Canada, May 29 - June 1, 2016, Proceedings. Lecture Notes in Computer Science, vol. 9676, pp. 1–17. Springer (2016), https://doi.org/10.1007/978-3-319-33954-2_1
2. Abío, I., Nieuwenhuis, R., Oliveras, A., Rodríguez-Carbonell, E., Mayer-Eichberger, V.: A new look at BDDs for pseudo-Boolean constraints. J. Artif. Intell. Res. **45**, 443–480 (2012), https://doi.org/10.1613/jair.3653
3. Aloul, F.A., Ramani, A., Markov, I.L., Sakallah, K.A.: Generic ILP versus specialized 0-1 ILP: an update. In: Pileggi, L.T., Kuehlmann, A. (eds.) Proceedings of the 2002 IEEE/ACM International Conference on Computer-aided Design, ICCAD 2002, San Jose, California, USA, November 10-14, 2002. pp. 450–457. ACM / IEEE Computer Society (2002), https://doi.org/10.1145/774572.774638
4. Bordeaux, L., Marques-Silva, J.: Knowledge compilation with empowerment. In: SOFSEM 2012: Theory and Practice of Computer Science - 38th Conference on Current Trends in Theory and Practice of Computer Science, Špindlerův Mlýn, Czech Republic, January 21-27, 2012. Proceedings. pp. 612–624 (2012), https://doi.org/10.1007/978-3-642-27660-6_50
5. Boros, E., Hammer, P.L., Minoux, M., Rader, D.J.: Optimal cell flipping to minimize channel density in VLSI design and pseudo-Boolean optimization. Discret. Appl. Math. **90**(1-3), 69–88 (1999), https://doi.org/10.1016/S0166-218X(98)00114-0
6. Bova, S., Capelli, F., Mengel, S., Slivovsky, F.: Knowledge compilation meets communication complexity. In: Kambhampati, S. (ed.) Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016. pp. 1008–1014. IJCAI/AAAI Press (2016), http://www.ijcai.org/Abstract/16/147
7. Bryant, R.E., Lahiri, S.K., Seshia, S.A.: Deciding CLU logic formulas via Boolean and pseudo-Boolean encodings. In: In Proc. Intl. Workshop on Constraints in Formal Verification (CFV02 (2002)
8. Darwiche, A., Marquis, P.: A knowledge compilation map. J. Artif. Intell. Res. **17**, 229–264 (2002), https://doi.org/10.1613/jair.989
9. del Val, A.: Tractable databases: How to make propositional unit resolution complete through compilation. In: Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR'94). Bonn, Germany, May 24-27, 1994. pp. 551–561 (1994)
10. Eén, N., Sörensson, N.: Translating pseudo-boolean constraints into SAT. JSAT **2**(1-4), 1–26 (2006), https://satassociation.org/jsat/index.php/jsat/article/view/18
11. Gange, G., Stuckey, P.J.: Explaining propagators for s-DNNF circuits. In: Beldiceanu, N., Jussien, N., Pinson, E. (eds.) Integration of AI and OR Techniques in Contraint Programming for Combinatorial Optimzation Problems - 9th International Conference, CPAIOR 2012, Nantes, France, May 28 - June1, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7298, pp. 195–210. Springer (2012), https://doi.org/10.1007/978-3-642-29828-8_13
12. Gopalan, P., Klivans, A.R., Meka, R., Štefankovič, D., Vempala, S.S., Vigoda, E.: An FPTAS for #Knapsack and related counting problems. In: Ostrovsky, R.

(ed.) IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011. pp. 817–826. IEEE Computer Society (2011), https://doi.org/10.1109/FOCS.2011.32
13. Hosaka, K., Takenaga, Y., Kaneda, T., Yajima, S.: Size of ordered binary decision diagrams representing threshold functions. Theor. Comput. Sci. **180**(1-2), 47–60 (1997), https://doi.org/10.1016/S0304-3975(97)83807-8
14. Ivănescu, P.L.: Some network flow problems solved with pseudo-boolean programming. Operations Research **13**(3), 388–399 (1965)
15. Jung, J.C., Barahona, P., Katsirelos, G., Walsh, T.: Two encodings of DNNF theories. In: ECAI'08 Workshop on Inference methods based on Graphical Structures of Knowledge (2008)
16. Kučera, P., Savický, P.: Propagation complete encodings of smooth DNNF theories. CoRR **abs/1909.06673** (2019), http://arxiv.org/abs/1909.06673
17. Le Berre, D., Marquis, P., Mengel, S., Wallon, R.: Pseudo-Boolean constraints from a knowledge representation perspective. In: Lang, J. (ed.) Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden. pp. 1891–1897. ijcai.org (2018), https://doi.org/10.24963/ijcai.2018/261
18. Marquis, P.: Compile! In: Bonet, B., Koenig, S. (eds.) Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA. pp. 4112–4118. AAAI Press (2015), http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9596
19. Mengel, S., Wallon, R.: Revisiting graph width measures for CNF-encodings. In: Janota, M., Lynce, I. (eds.) Theory and Applications of Satisfiability Testing - SAT 2019 - 22nd International Conference, SAT 2019, Lisbon, Portugal, July 9-12, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11628, pp. 222–238. Springer (2019), https://doi.org/10.1007/978-3-030-24258-9_16
20. Papaioannou, S.G.: Optimal test generation in combinational networks by pseudo-Boolean programming. IEEE Trans. Computers **26**(6), 553–560 (1977), https://doi.org/10.1109/TC.1977.1674880