Approximability of Bounded Occurrence Max Ones

Fredrik Kuivinen[*]

Department of Computer and Information Science, Linköpings Universitet,
S-581 83 Linköping, Sweden, `freku@ida.liu.se`

Abstract. We study the approximability of MAX ONES when the number of variable occurrences is bounded by a constant. For conservative constraint languages (i.e., when the unary relations are included) we give a complete classification when the number of occurrences is three or more and a partial classification when the bound is two.

For the non-conservative case we prove that it is either trivial or equivalent to the corresponding conservative problem under polynomial-time many-one reductions.

Keywords: Approximability, Bounded occurrence, Constraint satisfaction problems, Matching, Max Ones

1 Introduction

Many combinatorial optimisation problems can be formulated as various variants of constraint satisfaction problems (CSPs). MAX ONES is a boolean CSP where we are not only interested in finding a solution but also the measure of the solution. In this paper we study a variant of MAX ONES when the number occurrences of each variable is bounded by a constant.

We denote the set of all n-tuples with elements from $\{0, 1\}$ by $\{0, 1\}^n$. A subset $R \subseteq \{0, 1\}^n$ is a *relation* and n is the *arity* of R. A *constraint language* is a finite set of relations. A constraint language is said to be *conservative* if every unary relation is included in the language. In the boolean case this means that the relations $\{(0)\}$ and $\{(1)\}$ are in the language. The constraint satisfaction problem over the constraint language Γ, denoted $\text{CSP}(\Gamma)$, is defined to be the decision problem with instance (V, C), where V is a set of variables and C is a set of constraints $\{C_1, \ldots, C_q\}$, in which each constraint C_i is a pair (R_i, s_i) with s_i a list of variables of length n_i, called the constraint scope, and R_i an n_i-ary relation over the set $\{0, 1\}$, belonging to Γ, called the constraint relation. The question is whether there exists a solution to (V, C) or not. A solution to (V, C) is a function $s : V \to \{0, 1\}$ such that, for each constraint $(R_i, (v_1, v_2, \ldots, v_{n_i})) \in C$, the image of the constraint scope is a member of the constraint relation, i.e., $(s(v_1), s(v_2), \ldots, s(v_{n_i})) \in R_i$.

The optimisation problem W-MAX ONES can be defined as follows:

Definition 1 (W-MAX ONES)**.** W-MAX ONES *over the constraint language Γ is defined to be the optimisation problem with*

Instance: *Tuple (V, C, w), where (V, C) is an instance of $\textsc{Csp}(\Gamma)$ and $w : V \to \mathbb{N}$ is a function.*

Solution: *An assignment $f : V \to \{0, 1\}$ to the variables which satisfies the $\textsc{Csp}(\Gamma)$ instance (V, C).*

Measure: $\sum\limits_{v \in V} w(v) \cdot f(v)$

The function $w : V \to \mathbb{N}$ is called a *weight function*. In the corresponding unweighted problem, denoted $\textsc{Max Ones}(\Gamma)$, the weight function is restricted to map every variable to 1. The approximability of (W-)$\textsc{Max Ones}$ has been completely classified by Khanna et al. [20]. Several well-known optimisation problems can be rephrased as (W-)$\textsc{Max Ones}$ problems, in particular $\textsc{Independent Set}$. We will study W-$\textsc{Max Ones}(\Gamma)$ with a bounded number of variable occurrences, denoted by W-$\textsc{Max Ones}(\Gamma)$-$k$ for an integer k. In this problem the instances are restricted to contain at most k occurrences of each variable. The corresponding bounded occurrence variant of $\textsc{Csp}(\Gamma)$ will be denoted by $\textsc{Csp}(\Gamma)$-k.

Schaefer [26] classified the complexity of $\textsc{Csp}(\Gamma)$ for every constraint language Γ. Depending on Γ, Schaefer proved that $\textsc{Csp}(\Gamma)$ is either solvable in polynomial time or is **NP**-complete. The conservative bounded occurrence variant of $\textsc{Csp}(\Gamma)$ has been studied by a number of authors [12,14,15,16]. One result of that research is that the difficult case to classify is when the number of variable occurrences are restricted to two, in all other cases the bounded occurrence problem is no easier than the unrestricted problem. Kratochvíl et al. [21] have studied k-$\textsc{Sat}$-l, i.e., satisfiability where every clause have length k and there are at most l occurrences of each variable. k-$\textsc{Sat}$-l is a *non-conservative* constraint satisfaction problem. The complexity classification seems to be significantly harder for such problems compared to the conservative ones. In particular, Kratochvíl et al [21] proves that there is a function f such that k-$\textsc{Sat}$-l is trivial if $l \leq f(k)$ (every instance has a solution) and **NP**-complete if $l \geq f(k) + 1$. Some bounds of f is given in [21], but the exact behaviour of f is unknown.

$\textsc{Max Ones}(\Gamma)$-$k$ can represent many well-known problems. For $k \geq 3$, we have for example, that $\textsc{Independent Set}$ in graphs of maximum degree k is precisely $\textsc{Max Ones}(\{\{(0, 0), (1, 0), (0, 1)\}\})$-$k$. However, the more interesting case is perhaps $k = 2$ due to its connection to matching problems. (See [24] for definitions and more information about the matching problems mentioned below.) Ordinary weighted maximum matching in graphs is, for example, straightforward to formulate and we get certain generalisations "for free" (because they can be rephrased as ordinary matching problems), such as f-factors and capacitated b-matchings. The general factor problem can also be rephrased as a $\textsc{Max Ones}(\cdot)$-2 problem. A dichotomy theorem for the existence problem of general factors has been proved by Cornuéjols [9]. Some research has also been done on the optimisation problem [8].

In this paper, we start the classification of bounded occurrence $\textsc{Max Ones}$. Our first result is a complete classification of W-$\textsc{Max Ones}(\Gamma)$-$k$ when $k \geq 3$ and $\{(0)\}$ and $\{(1)\}$ are included in Γ. We show that, depending on Γ, this problem is either in **PO**, **APX**-complete or **poly-APX**-complete. Our second result is a partial classification of W-$\textsc{Max Ones}(\Gamma)$-2. We also give hardness results for the non-conservative case.

The outline of the paper is as follows: in Section 2 we define our notation and present the tools we use. Section 3 and 4 contains our results for three or more occur-

rences and two occurrences, respectively. Section 5 contains our results for the general case, i.e., when the constraint language is not necessarily conservative. Section 6 contains some concluding remarks. Due to lack of space most of the proofs can be found in the appendix.

2 Preliminaries

For an integer n we will use $[n]$ to denote the set $\{1, 2, \ldots, n\}$. The Hamming distance between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ will be denoted by $d_H(\boldsymbol{x}, \boldsymbol{y})$. For a tuple or vector $\boldsymbol{x}$ the n:th component will be denoted by $\boldsymbol{x}[n]$.

Unless explicitly stated otherwise we assume that the constraint languages we are working with are *conservative*, i.e., every unary relation is a member of the constraint language (in the boolean domain, which we are working with, this means that $\{(0)\}$ and $\{(1)\}$ are in the constraint language).

We define the following relations

- $NAND^m = \{(x_1, \ldots, x_m) \mid x_1 + \ldots + x_m < m\}$,
- $EQ^m = \{(x_1, \ldots, x_m) \mid x_1 = x_2 = \ldots = x_m\}$,
- $IMPL = \{(0, 0), (0, 1), (1, 1)\}, c_0 = \{(0)\}, c_1 = \{(1)\}$

and the function $h_n(x_1, x_2, \ldots, x_{n+1}) = \bigvee_{i=1}^{n+1}(x_1 \wedge \ldots \wedge x_{i-1} \wedge x_{i+1} \wedge \ldots \wedge x_{n+1})$. For a relation R of arity r, we will sometimes use the notation $R(x_1, \ldots, x_r)$ with the meaning $(x_1, \ldots, x_r) \in R$, i.e., $R(x_1, \ldots, x_r) \iff (x_1, \ldots, x_r) \in R$. If r is the arity of R and $I = \{i_1, \ldots, i_n\} \subseteq [r], i_1 < i_2 < \ldots < i_n$, then we denote the projection of R to I by $R\big|_I$, i.e., $R\big|_I = \{(x_{i_1}, x_{i_2}, \ldots, x_{i_n}) \mid (x_1, x_2, \ldots, x_r) \in R\}$

Representations (sometimes called implementations) have been central in the study of constraint satisfaction problems. We need a notion of representability which is a bit stronger that the usual one, because we have to be careful with how many occurrences we use of each variable.

Definition 2 (k-representable). *An n-ary relation R is k-representable by a set of relations F if there is a collection of constraints $C_1, \ldots, C_l$ with constraint relations from F over variables $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ (called* primary *variables) and $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$ (called* auxiliary *variables) such that,*

- *the primary variables occur at most once in the constraints,*
- *the auxiliary variables occur at most k times in the constraints, and*
- *for every tuple $\boldsymbol{z}$, $\boldsymbol{z} \in R$ if and only if there is an assignment to $\boldsymbol{y}$ such that $\boldsymbol{x} = \boldsymbol{z}$ satisfies all of the constraints $C_1, C_2, \ldots, C_l$.*

The intuition behind the definition is that if every relation in Γ_1 is k-representable by relations in Γ_2 then W-MAX ONES(Γ_2)-k is no easier than W-MAX ONES(Γ_1)-k. This is formalised in Lemma 6.

2.1 Approximability, Reductions, and Completeness

A *combinatorial optimisation problem* is defined over a set of *instances* (admissible input data) $\mathcal{I}$; each instance $I \in \mathcal{I}$ has a finite set $\text{SOL}(I)$ of *feasible solutions* associated with it. The objective is, given an instance I, to find a feasible solution of *optimum* value with respect to some measure function m defined for pairs (x, y) such that $x \in \mathcal{I}$ and $y \in \text{SOL}(x)$. Every such pair is mapped to a non-negative integer by m. The optimal value is the largest one for *maximisation* problems and the smallest one for *minimisation* problems. A combinatorial optimisation problem is said to be an **NPO** problem if its instances and solutions can be recognised in polynomial time, the solutions are polynomially-bounded in the input size, and the objective function can be computed in polynomial time (see, e.g., [1]).

Definition 3 (*r-approximate*). *A solution* $s \in \text{SOL}(I)$ *to an instance* I *of an* **NPO** *problem* Π *is* r-approximate *if* $\max \left\{ \frac{m(I,s)}{\text{OPT}(I)}, \frac{\text{OPT}(I)}{m(I,s)} \right\} \leq r$, *where* $\text{OPT}(I)$ *is the optimal value for a solution to* I.

An approximation algorithm for an **NPO** problem Π has *performance ratio* $\mathcal{R}(n)$ if, given any instance I of Π with $|I| = n$, it outputs an $\mathcal{R}(n)$-approximate solution.

Definition 4 (**PO, APX, poly-APX**). **PO** *is the class of* **NPO** *problems that can be solved (to optimality) in polynomial time. An* **NPO** *problem* Π *is in the class* **APX** *if there is a polynomial-time approximation algorithm for* Π *whose performance ratio is bounded by a constant. Similarly,* Π *is in the class* **poly-APX** *if there is a polynomial-time approximation algorithm for* Π *whose performance ratio is bounded by a polynomial in the size of the input.*

Completeness in **APX** and **poly-APX** is defined using AP-reductions [1]. However, we do not need AP-reductions in this paper, the simpler L- and S-reductions are sufficient for us.

Definition 5 (*L-reduction*). *An* **NPO** *problem* Π_1 *is said to be* L-reducible *to an* **NPO** *problem* Π_2, *written* $\Pi_1 \leq_L \Pi_2$, *if two polynomial-time computable functions* F *and* G *and positive constants* β *and* γ *exist such that*

- *given any instance* I *of* Π_1, *algorithm* F *produces an instance* $I' = F(I)$ *of* Π_2, *such that* $\text{OPT}(I') \leq \beta \cdot \text{OPT}(I)$.
- *given* $I' = F(I)$, *and any solution* s' *to* I', *algorithm* G *produces a solution* s *to* I *such that* $|m_1(I, s) - \text{OPT}(I)| \leq \gamma \cdot |m_2(I', s') - \text{OPT}(I')|$, *where* m_1 *is the measure for* Π_1 *and* m_2 *is the measure for* Π_2.

It is well-known (see, e.g., Lemma 8.2 in [1]) that, if Π_1 is L-reducible to Π_2 and $\Pi_1 \in$ **APX** then there is an AP-reduction from Π_1 to Π_2.

S-reductions are similar to L-reductions but instead of the condition $\text{OPT}(I') \leq \beta \cdot \text{OPT}(I)$ we require that $\text{OPT}(I') = \text{OPT}(I)$ and instead of $|m_1(I, s) - \text{OPT}(I)| \leq \gamma \cdot |m_2(I', s') - \text{OPT}(I')|$ we require that $m_1(I, s) = m_2(I', s')$. If there is an S-reduction from Π_1 to Π_2 (written as $\Pi_1 \leq_S \Pi_2$) then there is an AP-reduction from Π_1 to Π_2. An **NPO** problem Π is **APX**-*hard* (**poly-APX**-*hard*) if every problem in

APX (**poly-APX**) is AP-reducible to it. If, in addition, Π is in **APX** (**poly-APX**), then Π is called **APX**-*complete* (**poly-APX**-*complete*).

We will do several reductions from INDEPENDENT SET (hereafter denoted by MIS) which is **poly-APX**-complete [19]. We will also use the fact that for any $k \geq 3$, MIS restricted to graphs of degree at most k is **APX**-complete [22]. We will denote the latter problem by MIS-k.

The following lemma shows the importance of k-representations in our work.

Lemma 6. *For constraint languages Γ_1 and Γ_2 if every relation in Γ_1 can be k-represented by Γ_2 then* W-MAX ONES(Γ_1)-$k \leq_S$ W-MAX ONES(Γ_2)-k.

Proof. Given an arbitrary instance $I = (V, C, w)$ of W-MAX ONES(Γ_1)-k, we will construct an instance $I' = (V', C', w')$ of W-MAX ONES(Γ_2)-k, in polynomial time. For each $c \in C$, add the k-representation of c to C' and also add all variables which participate in the representation to V' in such a way that the auxiliary variables used in the representation are distinct from all other variables in V'. Let $w'(x) = w(x)$ for all $x \in V$ and $w(x) = 0$ if $x \notin V$ (i.e., all auxiliary variables will have weight zero).

It is not hard to see that: (a) every variable in I' occurs at most k times (b) OPT$(I') =$ OPT(I), and (c) given a solution s' to I' we can easily construct a solution s to I (let $s(x) = s'(x)$ for every $x \in V$) such that $m(I, s) = m(I', s')$. Hence, there is an S-reduction from W-MAX ONES(Γ_1)-k to W-MAX ONES(Γ_2)-k. $\square$

2.2 Co-clones and Polymorphisms

Given an integer k, a function $f : \{0,1\}^k \rightarrow \{0,1\}$ can be extended to a function over tuples as follows: let $\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots, \boldsymbol{t_k}$ be k tuples with n elements each then $f(\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots, \boldsymbol{t_k})$ is defined to be the tuple $(f(\boldsymbol{t_1}[1], \boldsymbol{t_2}[1], \ldots, \boldsymbol{t_k}[1]), \ldots, f(\boldsymbol{t_1}[n], \boldsymbol{t_2}[n], \ldots, \boldsymbol{t_k}[n]))$. Given a n-ary relation R we say that R is *invariant* (or, closed) under f if $\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots, \boldsymbol{t_k} \in R \Rightarrow f(\boldsymbol{t_1}, \boldsymbol{t_2}, \ldots, \boldsymbol{t_n}) \in R$. Conversely, for a function f and a relation R, f is a *polymorphism* of R if R is invariant under f. For a constraint language Γ we say that Γ is invariant under f if every relation in Γ is invariant under f. We analogously extend the notion of polymorphisms to constraint languages, i.e., a function f is a polymorphism of Γ if Γ is invariant under f. Those concepts has been very useful in the study of the complexity of various constraint satisfaction problems (see, e.g., [17]) and play an important role in this work, too.

The set of polymorphisms for a constraint language Γ will be denoted by Pol(Γ), and for a set of functions C the set of all relations which are invariant under C will be denoted by Inv(B). The sets Pol(Γ) are *clones* in the sense of universal algebra. For a clone C, Inv(C) is called a relational clone or a co-clone. Over the boolean domain Emil Post has classified all such co-clones and their inclusion structure in [23].

For a set of relations Γ we define a closure operator $\langle \Gamma \rangle$ as the set of relations that can be expressed with relations from Γ using existential quantification and conjunction (note that we are only allowed to use the relations in Γ, hence equality is not necessarily allowed). Intuitively $\langle \Gamma \cup \{EQ^2\} \rangle$ is the set of relations which can be simulated by Γ in CSP(Γ). An alternative classification of this set is $\langle \Gamma \cup \{EQ^2\} \rangle = $ Inv(Pol(Γ)) [25]. These few paragraphs barely scratch the surface of the rich theory of clones and their

relation to the computational complexity of various constraint satisfaction problems, for a more thorough introduction see [5,6,10].

We say that a set of relations B is a *plain basis* for a constraint language Γ if every relation in Γ can be expressed with relations from B using relations from $B \cup \{=\}$ and conjunction. Note that this differs from the definition of the closure operator $\langle \cdot \rangle$ as we do not allow existential quantification. See [11] for more information on plain bases.

We can not only study the co-clones when we try to classify MAX ONES(Γ)-k because the complexity of the problem do not only depend on the co-clone $\langle \Gamma \rangle$. However, the co-clone lattice with the corresponding plain bases and invariant functions will help us in our classification effort. Furthermore, as we mostly study the conservative constraint languages we can concentrate on the co-clones which contain c_0 and c_1. Figure 1 contains the conservative part of Post's lattice and Table 1 contains the plain bases for the relational clones which will be interesting to us (co-clones at and below IV_2 have been omitted as MAX ONES is in **PO** there).
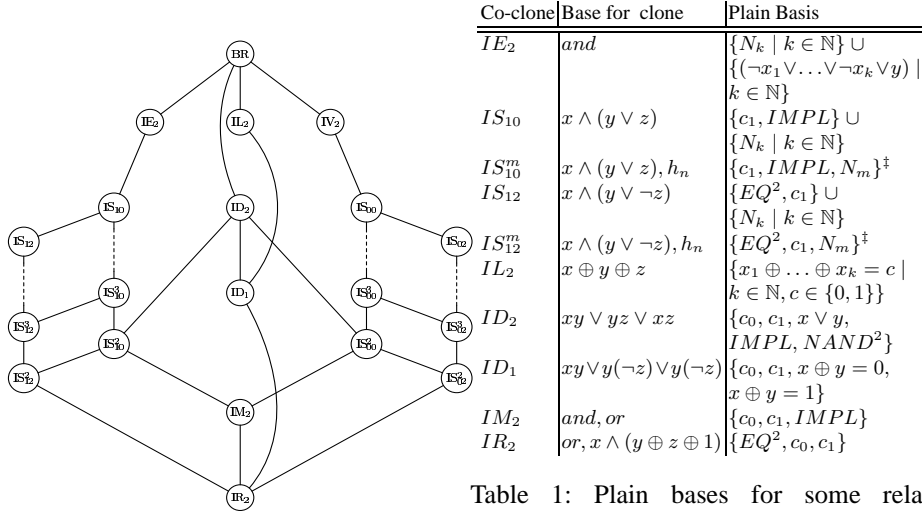

Fig. 1: Lattice of idempotent co-clones

Co-clone	Base for clone	Plain Basis
IE_2	and	$\{N_k \mid k \in \mathbb{N}\} \cup \{(\neg x_1 \vee \ldots \vee \neg x_k \vee y) \mid k \in \mathbb{N}\}$
IS_{10}	$x \wedge (y \vee z)$	$\{c_1, IMPL\} \cup \{N_k \mid k \in \mathbb{N}\}$
IS_{10}^m	$x \wedge (y \vee z), h_n$	$\{c_1, IMPL, N_m\}^{\ddagger}$
IS_{12}	$x \wedge (y \vee \neg z)$	$\{EQ^2, c_1\} \cup \{N_k \mid k \in \mathbb{N}\}$
IS_{12}^m	$x \wedge (y \vee \neg z), h_n$	$\{EQ^2, c_1, N_m\}^{\ddagger}$
IL_2	$x \oplus y \oplus z$	$\{x_1 \oplus \ldots \oplus x_k = c \mid k \in \mathbb{N}, c \in \{0,1\}\}$
ID_2	$xy \vee yz \vee xz$	$\{c_0, c_1, x \vee y, IMPL, NAND^2\}$
ID_1	$xy \vee y(\neg z) \vee y(\neg z)$	$\{c_0, c_1, x \oplus y = 0, x \oplus y = 1\}$
IM_2	and, or	$\{c_0, c_1, IMPL\}$
IR_2	$or, x \wedge (y \oplus z \oplus 1)$	$\{EQ^2, c_0, c_1\}$

Table 1: Plain bases for some relational clones. The list of plain bases are from [11].‡

3   Three or More Occurrences

In this section we will prove a classification theorem for W-MAX ONES(Γ)-k where $k \geq 3$. The main result of this section is the following theorem.

Theorem 7. *Let Γ be a conservative constraint language and $k \geq 3$,*

---

‡ In [11] the listed plain basis for IS_{12}^m is $\{EQ^2, c_1\} \cup \{N_k | k \leq m\}$ however, if we have N_m then N_{m-1} can be represented without auxiliary variables by $N_{m-1}(x_1, x_2, \ldots, x_{m-1}) \iff N_m(x_1, x_1, x_2, x_3, \ldots, x_{m-1})$, hence the set of relations listed in Table 1 is a plain basis for IS_{12}^m. The same modification has been done to IS_{10}^m.

1. *If $\Gamma \subseteq IV_2$ then* W-MAX ONES(Γ)-k *is in* **PO**.
2. *Else if $IS_{12}^2 \subseteq \langle \Gamma \rangle \subseteq IS_{12}$ then* (W-)MAX ONES(Γ)-k *is* **APX**-*complete if EQ^2 is not k-representable by Γ and* W-MAX ONES(Γ)-k *is* **poly-APX**-*complete otherwise.*
3. *Otherwise,* W-MAX ONES(Γ) *and* W-MAX ONES(Γ)-k *are equivalent under S-reductions.*

The first part of Theorem 7 follows from Khanna et al.'s results for MAX ONES [20]. Intuitively the second part follows from the fact that W-MAX ONES$(\{NAND^2\})$ is equivalent to MIS, hence if we have access to the equality relation then the problem gets **poly-APX**-complete. On the other hand, if we do not have the equality relation then we essentially get MIS-k, for some k, which is **APX**-complete. The third part follows from Lemmas 8, 9, 10, and 11.

Dalmau and Ford proved the following lemma in [12].

Lemma 8. *If there is a relation R in the constraint language Γ such that $R \notin IE_2$, then either $x \vee y$ or $x \neq y$ can be 3-represented by Γ. By duality, if there is a relation $R \in \Gamma$ such that $R \notin IV_2$, then either $NAND^2$ or $x \neq y$ can be 3-represented.*

We can use the lemma above to get a 3-representation of either EQ^2 or $IMPL$. We will later, in Lemma 11, show that those relations makes the problem as hard as the unbounded occurrence variant.

Lemma 9. *If there is a relation R in the constraint language Γ such that $R \notin IE_2$ and $R \notin IV_2$, then either EQ^2 or $IMPL$ can be 3-represented by Γ.*

Proof. From Lemma 8 we know that either $x \neq y$ or both $x \vee y$ and $NAND^2$ are 3-representable. In the first case $\exists z : x \neq z \wedge z \neq y$ is a 3-representation of EQ^2. In the second case $\exists z : NAND^2(x, z) \wedge (z \vee y)$ is a 3-representation of $IMPL(x, y)$. $\square$

To get the desired hardness results for the IS_{10} chain we need to prove that we can represent EQ_2 or $IMPL$ in that case too. To this end we have the following lemma.

Lemma 10. *If there is a relation R in the constraint language Γ such that $R \in IE_2$ and $R \notin IS_{12}$, then either EQ^2 or $IMPL$ can be 3-represented by Γ.*

Proof. Let r be the arity of R then, as $R \notin IS_{12}$, there exists a set of minimal cardinality $I \subseteq [r]$, such that $R\big|_I \notin IS_{12}$.

As $g(x, y) = x \wedge y$ is a base of the clone which corresponds to IE_2, $R\big|_I \in IE_2$ implies that g is a polymorphism of $R\big|_I$. Furthermore, as $f(x, y, z) = x \wedge (y \vee \neg z)$ is a base of the clone which corresponds to IS_{12}, $R\big|_I \notin IS_{12}$ implies that f is *not* a polymorphism of $R\big|_I$. Hence, there exists tuples $\boldsymbol{t_1}, \boldsymbol{t_2}, \boldsymbol{t_3} \in R\big|_I$ such that $f(\boldsymbol{t_1}, \boldsymbol{t_2}, \boldsymbol{t_3}) = \boldsymbol{t} \notin R\big|_I$.

There exists a coordinate $l_1, 1 \leq l_1 \leq r$ such that $(\boldsymbol{t_1}[l_1], \boldsymbol{t_2}[l_1], \boldsymbol{t_3}[l_1]) = (1, 0, 1)$, because otherwise $f(\boldsymbol{t_1}, \boldsymbol{t_2}, \boldsymbol{t_3}) = \boldsymbol{t_1}$. Similarly there exists a coordinate $l_2, 1 \leq l_2 \leq r$ such that $(\boldsymbol{t_1}[l_2], \boldsymbol{t_2}[l_2], \boldsymbol{t_3}[l_2])$ is equal to one of $(0, 1, 0)$, $(0, 1, 1)$ or $(1, 0, 0)$. Because otherwise $f(\boldsymbol{t_1}, \boldsymbol{t_2}, \boldsymbol{t_3}) = \boldsymbol{t_2}$. From now on, the case $(\boldsymbol{t_1}[l_2], \boldsymbol{t_2}[l_2], \boldsymbol{t_3}[l_2]) = (1, 0, 0)$ will be denoted by (*). Finally, there also exists a coordinate $l_3, 1 \leq l_3 \leq r$ such that

$(\boldsymbol{t_1}[l_3], \boldsymbol{t_2}[l_3], \boldsymbol{t_3}[l_3])$ is equal to one of $(0, 0, 1)$, $(0, 1, 1)$, $(1, 0, 0)$, $(1, 0, 1)$ or $(1, 1, 0)$, because otherwise $f(\boldsymbol{t_1}, \boldsymbol{t_2}, \boldsymbol{t_3}) = \boldsymbol{t_3}$. The case $(\boldsymbol{t_1}[l_3], \boldsymbol{t_2}[l_3], \boldsymbol{t_3}[l_3]) = (1, 0, 0)$ will be denoted by (**).

As $R\big|_I$ is invariant under g we can place additional restrictions on l_1, l_2 and l_3. In particular, there has to be coordinates l_1, l_2 and l_3 such that we have at least one of the cases (*) or (**), because otherwise $f(\boldsymbol{t_1}, \boldsymbol{t_2}, \boldsymbol{t_3}) = g(\boldsymbol{t_1}, \boldsymbol{t_2})$, which is in $R\big|_I$ and we have assumed that $f(\boldsymbol{t_1}, \boldsymbol{t_2}, \boldsymbol{t_3}) \notin R\big|_I$. There is no problem in letting $l_2 = l_3$ since we will then get both (*) and (**). This will be assumed from now on. We can also assume, without loss of generality, that $l_1 = 1$ and $l_2 = l_3 = 2$. We can then construct a 3-representation as $R_\phi(x, y) \iff \exists z_3 \ldots z_r : R\big|_I(x, y, z_3, \ldots, z_r) \wedge c_{k_3}(z_3) \wedge c_{k_4}(z_4) \wedge \ldots \wedge c_{k_r}(z_r)$ where $k_i = f(\boldsymbol{t_1}[i], \boldsymbol{t_2}[i], \boldsymbol{t_3}[i])$ for $3 \le i \le r$. We will now prove that R_ϕ is equal to one of the relations we are looking for.

If $(0, 1) \in R_\phi$, then we would have $\boldsymbol{t} \in R\big|_I$, which is a contradiction, so $(0, 1) \notin R_\phi$. We will now show that $(0, 0) \in R_\phi$. Assume that $(0, 0) \notin R_\phi$. Then, $R^* = R\big|_{I \setminus \{l_2\}}$ is not in IS_{12} which contradicts the minimality of I. To see this consider the following table of possible tuples in $R\big|_I$,

	$1 = l_1$	$2 = l_2 = l_3$	3	4	$\ldots$
$\boldsymbol{t_1}$	1	1	$\boldsymbol{t_1}[3]$	$\boldsymbol{t_1}[4]$	$\ldots$
$\boldsymbol{t_2}$	0	0	$\boldsymbol{t_2}[3]$	$\boldsymbol{t_2}[4]$	$\ldots$
$\boldsymbol{t_3}$	1	0	$\boldsymbol{t_3}[3]$	$\boldsymbol{t_3}[4]$	$\ldots$
a	0	1	$f(\boldsymbol{t_1}[3], \boldsymbol{t_2}[3], \boldsymbol{t_3}[3])$	$f(\boldsymbol{t_1}[4], \boldsymbol{t_2}[4], \boldsymbol{t_3}[4])$	$\ldots$
b	0	0	$f(\boldsymbol{t_1}[3], \boldsymbol{t_2}[3], \boldsymbol{t_3}[3])$	$f(\boldsymbol{t_1}[4], \boldsymbol{t_2}[4], \boldsymbol{t_3}[4])$	$\ldots$

We know that $\boldsymbol{t_1}, \boldsymbol{t_2}, \boldsymbol{t_3} \in R\big|_I$ and we also know that $a \notin R\big|_I$. Furthermore, if $b \notin R\big|_I$, then $f(\boldsymbol{t_1}, \boldsymbol{t_2}, \boldsymbol{t_3})\big|_{I \setminus \{l_2\}} \notin R^*$ which means that I is not minimal. The conclusion is that we must have $(0, 0) \in R_\phi$. In the same way it is possible to prove that unless $(1, 1) \in R_\phi$, I is not minimal.

To conclude, we have proved that $(0, 0), (1, 1) \in R_\phi$ and $(0, 1) \notin R_\phi$, hence we either have $R_\phi = EQ^2$ or $R_\phi = \{(0, 0), (1, 0), (1, 1)\}$. $\qquad\square$

It is now time to use our implementations of EQ^2 or $IMPL$ to prove hardness results. To this end we have the following lemma.

Lemma 11. *If EQ^2 or $IMPL$ is 3-representable by the constraint language Γ then* W-MAX ONES$(\Gamma) \le_S$ W-MAX ONES(Γ)-3.

The proof can be found in the appendix. As either EQ^2 or $IMPL$ is available we can construct a cycle of constraints among variables and such a cycle force every variable in the cycle to obtain the same value. Furthermore, each variable occurs only twice in such a cycle so we have one occurrence left for each variable.

4 Two Occurrences

In this section, we study W-MAX ONES(Γ)-2. We are not able to present a complete classification but a partial classification is achieved. We completely classify the co-

clones IL_2 and ID_2. For Γ such that $\Gamma \not\subseteq IL_2, ID_2$ we show that if there is a relation which is not a Δ-matroid relation (those are defined below) in Γ then W-MAX ONES(Γ)-2 is **APX**-hard if W-MAX ONES(Γ) is not tractable.

4.1 Definitions and Results

Most of the research done on CSP(Γ)-k (e.g., in [14,12,15]) has used the theory of Δ-matroids. Those objects are a generalisation of matroids and has been widely studied, cf. [4,3]. It turns out that the complexity of W-MAX ONES(Γ)-2 depend to a large degree on if there is a relation which is not a Δ-matroid relation in the constraint language. Δ-matroid relations are defined as follows.

Definition 12 (Δ-matroid relation [12]). *Let $R \subseteq \{0,1\}^r$ be a relation. If $\boldsymbol{x}, \boldsymbol{x}' \in \{0,1\}^r$, then $\boldsymbol{x}'$ is a step from $\boldsymbol{x}$ to $\boldsymbol{y}$ if $d_H(\boldsymbol{x}, \boldsymbol{x}') = 1$ and $d_H(\boldsymbol{x}, \boldsymbol{x}') + d_H(\boldsymbol{x}', \boldsymbol{y}) = d_H(\boldsymbol{x}, \boldsymbol{y})$. R is a Δ-matroid relation if it satisfies the following two-step axiom: $\forall \boldsymbol{x}, \boldsymbol{y} \in R$ and $\forall \boldsymbol{x}'$ a step from $\boldsymbol{x}$ to $\boldsymbol{y}$, either $\boldsymbol{x}' \in R$ or $\exists \boldsymbol{x}'' \in R$ which is a step from $\boldsymbol{x}'$ to $\boldsymbol{y}$.*

As an example of a Δ-matroid relation consider $NAND^3$. It is not hard to see that $NAND^3$ satisfies the two-step axiom for every pair of tuples as there is only one tuple which is absent from the relation. EQ^3 is the simplest example of a relation which is not a Δ-matroid relation. The main theorem of this section is the following partial classification result for W-MAX ONES(Γ)-2. We say that a constraint language Γ *is Δ-matroid* if every relation in Γ is a Δ-matroid relation.

Theorem 13. *Let Γ be a conservative constraint language,*

1. *If $\Gamma \subseteq IV_2$ or $\Gamma \subseteq ID_1$ then W-MAX ONES(Γ)-2 is in **PO**.*
2. *Else if $\Gamma \subseteq IL_2$ and,*
   – *Γ is not Δ-matroid then, W-MAX ONES(Γ)-2 is **APX**-complete.*
   – *otherwise, W-MAX ONES(Γ)-2 is in **PO**.*
3. *Else if $\Gamma \subseteq ID_2$ and,*
   – *Γ is not Δ-matroid then, W-MAX ONES(Γ)-2 is **poly-APX**-complete.*
   – *otherwise, W-MAX ONES(Γ)-2 is in **PO**.*
4. *Else if $\Gamma \subseteq IE_2$ and Γ is not Δ-matroid then W-MAX ONES(Γ)-2 is **APX**-hard.*
5. *Else if Γ is not Δ-matroid then it is **NP**-hard to find feasible solutions to W-MAX ONES(Γ)-2.*

Part 1 of the theorem follows from the known results for W-MAX ONES [1]. Part 4 follows from results for CSP(Γ)-2 [14, Theorem 4]. The other parts follows from the results in Sections 4.3 and 4.4 below.

4.2 Tractability Results for W-MAX ONES($\boldsymbol{\Gamma}$)-2

Edmonds and Johnson [13] has shown that the following integer linear programming problem is solvable in polynomial time: maximise $\boldsymbol{wx}$ subject to the constraints $\boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{1}$, $\boldsymbol{b_1} \leq A\boldsymbol{x} \leq \boldsymbol{b_2}$ and $\boldsymbol{x}$ is an integer vector. Here A is a matrix with integer entries such that the sum of the absolute values of each column is at most 2. $\boldsymbol{b_1}$, $\boldsymbol{b_2}$ and $\boldsymbol{w}$ are arbitrary real vectors of appropriate dimensions. We will denote this problem by ILP-2. With the polynomial solvability of ILP-2 it is possible to prove the tractability of a number of W-MAX ONES(Γ)-2 problems.

4.3 Classification of ID_2 and IL_2

When $\text{Pol}(\Gamma) = \text{Pol}(ID_2)$ or $\text{Pol}(\Gamma) = \text{Pol}(IL_2)$ we prove a complete classification result. We start with the hardness results for ID_2, which consists of the following lemma.

Lemma 14. *Let Γ be a constraint language such that $\text{Pol}(\Gamma) = \text{Pol}(ID_2)$. If there is a relation $R \in \Gamma$ which is not a Δ-matroid relation, then* W-MAX ONES(Γ)-2 *is* **poly-APX***-complete.*

The main observations used to prove the lemma is that since $\text{Pol}(\Gamma) = \text{Pol}(ID_2)$ we can 2-represent every two-literal clause. This has been proved by Feder in [14]. Furthermore, if we have access to every two-literal clause and also have a non-Δ-matroid relation then it is possible to make variables participate in three clauses, which was also proved in [14]. The hardness result then follows with a reduction from MIS.

We will use some additional notation in the following proofs. For a tuple $x = (x_1, x_2, \ldots, x_k)$ and a set of coordinates $A \subseteq [k]$, $x \oplus A$ is defined to be the tuple $(y_1, y_2, \ldots, y_k)$ where $y_i = x_i$ if $i \notin A$ and $y_i = 1 - x_i$ otherwise. We extend this notation to relations: if $R \subseteq \{0,1\}^n$ and $A \subseteq [n]$ then $R \oplus A = \{t \oplus A \mid t \in R\}$.

We will now define a constraint language denoted by $\mathcal{Q}$. We will later prove that W-MAX ONES$(\mathcal{Q})$-2 is in **PO**. $\mathcal{Q}$ is the smallest constraint language such that:

– $\emptyset$, c_0, c_1, EQ^2 and $\{(0,1),(1,0)\}$ are in $\mathcal{Q}$.
– Every relation definable as $\{t \mid d_H(\mathbf{0}, t) \leq 1\}$ is in $\mathcal{Q}$.
– If $R, R' \in \mathcal{Q}$ then their cartesian product $\{(t, t') \mid t \in R, t' \in R'\}$ is also in $\mathcal{Q}$.
– If $R \in \mathcal{Q}$ and n is the arity of R then $R \oplus A \in \mathcal{Q}$ for every $A \subseteq [n]$.
– If $R \in \mathcal{Q}$, n is the arity of R and $f : [n] \to [n]$ is a permutation on $[n]$ then $\{(t_{f(1)}, t_{f(2)}, \ldots, t_{f(n)}) \mid t \in R\}$ is in $\mathcal{Q}$.

The relation between $\mathcal{Q}$ and the Δ-matroid relations in ID_2 is given by the following lemma.

Lemma 15. *If $R \in ID_2$ is a Δ-matroid relation, then $R \in \mathcal{Q}$.*

As for the tractability part we have the following lemma.

Lemma 16. *Let Γ be a constraint language such that $\Gamma \subseteq ID_2$, if all relations in Γ are Δ-matroid relations then* W-MAX ONES(Γ)-2 *is in* **PO**.

The idea behind the proof is that W-MAX ONES$(\mathcal{Q})$-2 can be seen as an ILP-2 problem and is therefore solvable in polynomial time.

As for IL_2 the result is the same, non Δ-matroids give rise to **APX**-complete problems and absence of such relations makes the problem tractable. Also in this case the tractability follows from a reduction to ILP-2.

4.4 IE_2, IS_{12} and IS_{10}

The structure of the Δ-matroids do not seem to be as simple in IS_{12} and IS_{10} as they are in ID_2 and IL_2. There exists relations in IS_{12} which are Δ-matroid relations

but for which we do not know of any polynomial time algorithm. One such relation is $R(x, y, z, w) \iff NAND^3(y, z, w) \wedge NAND^3(x, z, w) \wedge NAND^2(x, y)$. However, we get tractability results for some relations with the algorithm for ILP-2. In particular if the constraint language is a subset of $\{NAND^m \mid m \in \mathbb{N}\} \cup \{IMPL\}$ then W-MAX ONES$(\cdot)$-2 is in **PO**.

We manage to prove hardness results for every non-Δ-matroid relation contained in those co-clones. The main part of our hardness results for the non-Δ-matroid relations is the following lemma.

Lemma 17. *Let* $R(x_1, x_2, x_3) \iff NAND^2(x_1, x_2) \wedge NAND^2(x_2, x_3)$, *then* W-MAX ONES$(\{c_0, c_1, R\})$-2 *is* **APX**-*complete.*

Note that R is not a Δ-matroid relation. With Lemma 17 and a careful enumeration of the types of non-Δ-matroid relations that exists in IE_2, we can deduce the desired result: if there is a non-Δ-matroid relation in the constraint language, then W-MAX ONES$(\cdot)$-2 is **APX**-hard. The proof builds upon the work in [14,18,2].

5   Non-conservative Constraint Languages

In this section we will take a look at the non-conservative case, i.e., we will look at constraint languages which do not necessarily contain c_0 and c_1. A relation R is said to be *1-valid* if it contains the all ones tuple, i.e., R is 1-valid if $(1, 1, \ldots, 1) \in R$. A constraint language is said to be 1-valid if every relation in the language is 1-valid.

Theorem 18. *For any constraint language* Γ *which is not 1-valid, if* W-MAX ONES$(\Gamma \cup \{c_0, c_1\})$-$k$ *is* **NP**-*hard for some integer* k *then so is* W-MAX ONES(Γ)-k.

Note that for constraint languages Γ which are 1-valid W-MAX ONES(Γ) is trivial: the all-ones solution is optimal. The idea in the proof is that we can simulate c_1 constraints by giving the variable a large weight. Furthermore, if there are relations which are not 1-valid then we can represent c_0 constraints when we have access to c_1 constraints. It fairly easy to see why this fails to give us any inapproximability results: due to the large weight used to simulate c_1 any feasible solution is a good approximate solution.

6   Conclusions

We have started the study of the approximability properties of bounded occurrence MAX ONES. We have presented a complete classification for the weighted conservative case when three or more variable occurrences are allowed. Furthermore, a partial classification of the two occurrence case has been presented. In the latter case we have proved that non-Δ-matroid relations give rise to problems which are **APX**-hard if the unbounded occurrence variant is not tractable. We have also given complete classifications for the IL_2 and ID_2 co-clones.

There are still lots of open questions in this area. For example, what happens with the complexity if the weights are removed? Many constraint satisfaction problems such as MAX ONES and MAX CSP do not get any harder when weights are added. Such

results are usually proved by scaling and replicating variables and constraints a suitable number of times. However, such techniques do not work in the bounded occurrence setting and we do not know of any substitute which is equally general.

Except for the IS_{12} and IS_{10} chains the open questions in the two occurrence case are certain constraint languages Γ such that Γ only contains Δ-matroid relations and $\mathrm{Pol}(\Gamma) = \mathrm{Pol}(BR)$. It would be very interesting to find out the complexity of W-MAX ONES$(\cdot)$-2 for some of the classes of Δ-matroid relations which have been proved to be tractable for CSP$(\cdot)$-2 in [14,12,16,15]. Instead of trying to classify the entire IS_{12} or IS_{10} chain one could start with IS_{12}^3 or IS_{10}^3. The approximability of the non-conservative case is also mostly open. In light of [21] the computational structure of those problems seems to be quite complex.

Appendix

Proofs for Results in Section 3

Proof (Of Lemma 11). Let $I = (V, C, w)$ be an instance of W-MAX ONES(Γ). We will start with the case when $IMPL$ is 3-representable.

If $IMPL$ is 3-representable we can reduce I to an instance $I' = (V', C', w')$ of W-MAX ONES(Γ)-2 as follows: for each variable $v_i \in V$, let o_i be the number of occurrences of v_i in I, we introduce the variables $v_i^1, \ldots, v_i^{o_i}$ in V'. We let $w'(v_i^1) = w(v_i)$ and $w'(v_i^j) = 0$ for $j \neq 1$. We also introduce the constraints $IMPL(v_i^k, v_i^{k+1})$ for $k, 1 \leq k \leq o_i - 1$ and $IMPL(v_i^{o_i}, v_i^1)$ into C'. For every $i, 1 \leq i \leq |V|$ those constraints makes the variables $v_i^1, \ldots, v_i^{o_i}$ have the same value in every feasible solution of I'.

For every constraint $c = (R, s) \in C$ the constraint scope $s = (v_{l_1}, \ldots, v_{l_m})$ is replaced by $s' = (v_{l_1}^{k_1}, \ldots, v_{l_m}^{k_m})$ and (R, s') is added to C'. The numbers $k_1, \ldots, k_m$ are chosen in such a way that every variable in V' occur exactly three times in I'. This is possible since there are o_i variables in V' for every $v_i \in V$.

It is clear that the procedure described above is an S-reduction from W-MAX ONES(Γ) to W-MAX ONES(Γ)-3.

I can easily be S-reduced to an instance I' of W-MAX ONES$(\Gamma \cup \{EQ^2\})$-3. And as EQ^2 is 3-representable by Γ we are done, as every constraint involving EQ^2 can be replaced by the 3-representation of EQ^2 and any auxiliary variables used in the representation can be assigned the weight zero. $\square$

We need a couple of lemmas before we can state the proof of the classification theorem (Theorem 7). The following lemma will be used in several places to prove hardness results.

Lemma 19. *Let Γ be a constraint language such that $\mathrm{Pol}(\Gamma) = \mathrm{Pol}(IS_{1\alpha}^m)$ for some integer m and $\alpha \in \{0, 2\}$, then $NAND^m$ can be 2-represented by Γ.*

Proof. As $\mathrm{Pol}(\Gamma) = \mathrm{Pol}(IS_{1\alpha}^m)$, Γ is invariant under h_m and not invariant under h_{m-1}. Let r be the arity of R and let $X \subseteq [r]$ be a set of minimal cardinality such that there exist tuples $x_1, x_2, \ldots, x_m \in R|_X$ which satisfies $h_{m-1}(x_1, x_2, \ldots, x_m) = z \notin R|_X$. If there is a coordinate $i \in X$ such that $x_1[i] = x_2[i] = \ldots = x_m[i]$ then $z[i] = x_1[i]$ and as X is minimal we must have $z \oplus i \in R|_X$. However, this means that $h_m(x_1, x_2, \ldots, x_m, z \oplus i) = z \notin R|_X$ which is a contradiction with the assumption that R is invariant under h_m. We conclude that no coordinate is constant in every $x_1, x_2, \ldots, x_m$.

Now assume that there is a coordinate $j \in X$ such that $z[j] = 0$, then for X to be minimal we must have $z \oplus j \in R|_X$. However, $h_m(x_1, x_2, \ldots, x_m, z \oplus j) = z \notin R|_X$, a contradiction, hence there is no $j \in X$ such that $z[j] = 0$.

We can assume that $|X| \geq m$ because every relation of arity less than m which is invariant under h_m is also invariant under h_{m-1} [7, Proposition 3.6].

We do now know three things, no coordinate in X is constant in every $x_1, x_2, \ldots, x_m$, $z = (1, 1, \ldots, 1)$ and $|X| \geq m$. As $z = (1, 1, \ldots, 1)$ there is at most one zero for every given coordinate $i \in X$ among $x_1[i], x_2[i], \ldots, x_m[i]$, however as there is no constant

coordinate and $|X| \geq m$ we must have at least one zero in every $\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_m}$. We can in fact assume that there is exactly one zero entry, because if it is two distinct coordinates $i, j \in X$ such that $\boldsymbol{x_1}[i] = \boldsymbol{x_1}[j] = 0$ then as $\boldsymbol{z} = (1, 1, \ldots, 1)$ no other tuple can have $\boldsymbol{x_k}[i] = 0$ or $\boldsymbol{x_k}[j] = 0$. The conclusion is that $R\big|_{X \setminus \{j\}}$ is not invariant under h_{m-1} either.

This implies that $\boldsymbol{x_i} = (1, 1, \ldots, 1) \oplus i$. It is not hard to see that by using the fact that R is invariant under *and* we can get any tuple $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$ such that $y_1 + y_2 + \ldots + y_m < m$ by applying *and* to the $\boldsymbol{x_i}$s an appropriate number of times. Hence, we must have $R\big|_X = NAND^m$. $\qquad\square$

Lemma 20. *If* $\mathrm{Pol}(\{R\}) = \mathrm{Pol}(IS_{12}^m)$ *for some* $m \geq 2$ *and* R *cannot represent* EQ^2, *then* $\langle \{R, c_0, c_1\} \rangle = \langle \{NAND^m, c_1\} \rangle$.

Proof. We will denote $NAND^m$ by N. Let r be the arity of R then $B = \{N, EQ^2, c_1\}$ is a plain basis for IS_{12}^m (see Table 1). As B is a plain basis for R there is an implementation of R on the following form,

$$
\begin{aligned}
R(x_1, \ldots, x_r) \iff & N(x_{k_1^1}, x_{k_1^2}, \ldots, x_{k_1^m}) \wedge \ldots \wedge N(x_{k_n^1}, \ldots, x_{k_n^m}) \wedge \\
& EQ^2(x_{l_1^1}, x_{l_1^2}) \wedge \ldots \wedge EQ^2(x_{l_c^1}, x_{l_c^2}) \\
& c_1(x_{c_1}) \wedge \ldots \wedge c_1(x_{c_w})
\end{aligned}
$$

for some n, c and w such that $k_i^j \in [r]$, $l_i^j \in [r]$ and $c_i \in [r]$.

Assume that the representation above is minimal in the sense that it contains a minimal number of constraints. Hence, the only equalities that are possible are of the form $EQ^2(x_i, x_j)$ for $i \neq j$. If there is such an equality there are a number of cases to consider,

1. $R\big|_{\{i,j\}} = \{(0,0), (1,1)\}$,
2. $R\big|_{\{i,j\}} = \{(1,1)\}$, and
3. $R\big|_{\{i,j\}} = \{(0,0)\}$.

We cannot have equalities of type 1 because then EQ^2 would be representable by R. Furthermore, equalities of type 2 and 3 can be replaced by constraints of the form $c_1(x_i) \wedge c_1(x_j)$ and $N(x_i, \ldots, x_i) \wedge N(x_j, \ldots, x_j)$, respectively.

The conclusion is that R can be represented without EQ^2 and hence it is representable by $\{N, c_1\}$ alone. We have thus proved that $\langle \{R, c_0, c_1\} \rangle \subseteq \langle \{N, c_1\} \rangle$. The other inclusion, $\langle \{N, c_1\} \rangle \subseteq \langle \{R, c_0, c_1\} \rangle$, is given by Lemma 19. $\qquad\square$

As for the containment we have the following lemma.

Lemma 21. *Let* Γ *be a constraint language if* $\Gamma \subseteq IS_{12}^m$ *for some* m *and* Γ *cannot represent* EQ^2 *then* W-MAX ONES(Γ)-k *is in* **APX**.

Proof. Lemma 20 tells us that $\langle \Gamma \rangle = \langle \{NAND^m, c_1\} \rangle$, hence an instance J of W-MAX ONES(Γ)-k can be reduced to an instance J' of W-MAX ONES($\{NAND^m, c_1\}$)-k' for some constant k'. To prove the lemma it is therefore sufficient show that W-MAX ONES($\{NAND^m, c_1\}$)-l is in **APX** for every fixed l.

Let $I = (V, C, w)$ be an arbitrary instance of W-MAX ONES($\{NAND^m, c_1\}$)-l, for some l, and assume that $V = \{x_1, \ldots, x_n\}$. By Schaefer's result [26] we can decide in polynomial time whether I have a solution or not. Hence, we can safely assume that I has a solution. If a variable occurs in a constant constraint, say $c_1(x)$, then x must have the same value in every model of I. Thus, we can eliminate all such variables and assume that I only contains constraints of the type $NAND^m(x_1, \ldots, x_m)$.

We will give a polynomial-time algorithm that creates a satisfying assignment s to I with measure at least $\frac{1}{l+1}\text{OPT}(I)$. Hence we have a $\frac{1}{l+1}$-approximate algorithm proving that W-MAX ONES(IS_{12})-l is in **APX**.

The algorithm is as follows: Repeatedly delete from I any variable x_i having maximum weight and all variables that appear together with x_i in a clause of size two. In s we assign 1 to x_i and 0 to all variables appearing together with x_i in a clause of size two.

For simplicity, assume that the algorithm chooses variables $x_1, x_2, \ldots, x_t$ before it stops. If the algorithm at some stage choose a variable x with weight $w(x)$, then, in the worst case, it is forced to set l (remember that no variable occurs more than l times in I) variables to 0 and each of these variables have weight $w(x)$. This implies that $(l + 1) \cdot \sum_{i=1}^{t} w(x_i) \geq \sum_{i=1}^{n} w(x_i)$ and

$$m(I, s) = \sum_{i=1}^{t} w(x_i) \geq \frac{1}{l + 1} \sum_{i=1}^{n} w(x_i) \geq \frac{\text{OPT}(I)}{l + 1}.$$

$\square$

Lemma 22. *Let Γ be a constraint language such that $IS_{12}^2 \subseteq \langle \Gamma \rangle \subseteq IS_{12}$ then W-MAX ONES(Γ)-k is* **APX***-hard for $k \geq 3$.*

Proof. Note that MIS-3 is exactly the same as MAX ONES($\{NAND^2\}$)-3. The lemma then follows from the fact that MIS-3 is **APX**-hard, Lemma 19, and Lemma 6. $\square$

We are now ready to give the proof of the classification theorem for three or more occurrences.

Proof (Of Theorem 7, part 1). Follows directly from Khanna et al's results for MAX ONES [20]. $\square$

Proof (Of Theorem 7, part 2). The **APX**-hardness follows from Lemma 22. Containment in **APX** follows from Lemma 21. If EQ^2 is k-representable by Γ then the result follows from Lemma 11 and Khanna et al's results for MAX ONES [20]. $\square$

Proof (Of Theorem 7, part 3). There are two possibilities, the first one is that $\Gamma \not\subseteq IE_2$ and $\Gamma \not\subseteq IV_2$, the second case is that $\Gamma \subseteq IE_2$ and $\Gamma \not\subseteq IS_{12}$.

In the first case we can use the 3-representation of EQ^2 or $IMPL$ from Lemma 9. The result then follows from Lemma 11. In the second case the result follows from Lemma 10 and Lemma 11. $\square$

Proofs for Results in Section 4

We will start with the case when $\mathrm{Pol}(\Gamma) = \mathrm{Pol}(ID_2)$. We need the following lemma before we can give the proof of Lemma 14.

Lemma 23. *Let Γ be a constraint language such that $\mathrm{Pol}(\Gamma) = \mathrm{Pol}(ID_2)$ then $x \vee y$, $IMPL$ and $NAND^2$ are 2-representable by Γ.*

Proof. A part of the proof of Theorem 3 in [14] is the following: let F be a constraint language such that there are relations $R_1, R_2, R_3 \in F$ with the following properties:

- R_1 is not closed under $f(x, y) = x \vee y$.
- R_2 is not closed under $g(x, y) = x \wedge y$.
- R_3 is not closed under $h(x, y, z) = x + y + z \pmod 2$.

then F can 2-represent every two-literal clause. As we have assumed that $\mathrm{Pol}(\Gamma) = \mathrm{Pol}(ID_2)$ there are relations in Γ which full fills the conditions above. The lemma follows. $\qquad\square$

Proof (Of Lemma 14). We will do an S-reduction from the **poly-APX**-complete problem MIS, which is precisely MAX ONES($\{NAND^2\}$). Let $I = (V, C)$ be an arbitrary instance of MAX ONES($\{NAND^2\}$). We will construct an instance $I' = (V', C', w)$ of W-MAX ONES(Γ)-2. From Lemma 23 we know that we can 2-represent every two-literal clause. It is easy to modify I so that each variable occur at most three times. For a variable $x \in V$ which occur k times, introduce k fresh variables $y_1, y_2, \ldots, y_k$ and add the constraints $IMPL(y_1, y_2), IMPL(y_2, y_3), \ldots, IMPL(y_k, y_1)$. Each occurrence of x is then replaced with one of the y_i variables. In every solution each of the y_i variables will obtain the same value, furthermore they occur three times each. Hence, if we can create a construction which allows us to let a variable participate in three clauses we are done with our reduction.

In Theorem 4 in [14] it is shown that given a relation which is not a Δ-matroid we can make variables participate in three clauses if we have access to all clauses.

If we assign appropriate weights to the variables in V' it is clear that $\mathrm{OPT}(I) = \mathrm{OPT}(I')$ and each solution to I' corresponds to a solution of I with the same measure. Hence, we get an S-reduction. $\qquad\square$

We will now give the proof of Lemma 15 which describes the structure of the Δ-matroid relations in ID_2.

For a relation $R \in \mathcal{Q}$ if R can be decomposed (possibly after a permutation of the coordinates of R) into a cartesian product of other relations, $P_1, P_2, \ldots, P_n \in \mathcal{Q}$ then $P_1, P_2, \ldots, P_n$ will be called the *factors* of R.

Proof (Of Lemma 15). In this proof we will denote the majority function by m, i.e., $m(x, y, z) = (x \wedge y) \vee (y \wedge z) \vee (x \wedge z)$. Note that every relation in ID_2 is invariant under m. Let R be a relation which contradicts the lemma, i.e., $R \in ID_2$, R is a Δ-matroid and $R \notin \mathcal{Q}$. Let n be the arity of R. We can assume without loss of generality that R consists of one factor, i.e., it is not possible to decompose R into a cartesian product of other relations. In particular, R do not contain any coordinate which has the same value in all tuples.

As every relation of arity less than or equal to two is in $\mathcal{Q}$ we can assume that $n \geq 3$. If for every pair of tuples $\boldsymbol{t}, \boldsymbol{t}' \in R$ we have $d_H(\boldsymbol{t}, \boldsymbol{t}') \leq 2$ then $R \in \mathcal{Q}$ which is a contradiction. To see this let $\boldsymbol{t_1}, \boldsymbol{t_2}, \boldsymbol{t_3}$ be three distinct tuples in R (if there are less than three tuples in R then either R is not a Δ-matroid relation or there is some coordinate which is constant in all tuples). Then $\boldsymbol{t_2} = \boldsymbol{t_1} \oplus A$, $\boldsymbol{t_3} = \boldsymbol{t_1} \oplus B$ for some $A, B \subseteq [n]$ such that $|A|, |B| \leq 2$ and $|A \cap B| \leq 1$. If $|A \cup B| \leq 2$ for all such sets then R is either of arity 2 or there is a coordinate in R which is constant. Hence, assume that $|A \cup B| = 3$, which implies $|A| = |B| = 2$. Let $\boldsymbol{t} = m(\boldsymbol{t_1}, \boldsymbol{t_2}, \boldsymbol{t_3})$. We will prove that for every tuple $\boldsymbol{t}' \in R$ we have $d_H(\boldsymbol{t}', \boldsymbol{t}) \leq 1$. To this end, let $\boldsymbol{t}' = \boldsymbol{t} \oplus C$, with $|C| = 2$ ($|C| \leq 1$ implies $d_H(\boldsymbol{t}', \boldsymbol{t}) \leq 1$), be an arbitrary tuple in R. If $|A \cap C| = 0$ (or, $|B \cap C| = 0$) then $d_H(\boldsymbol{t_2}, \boldsymbol{t}') \geq 3$ ($d_H(\boldsymbol{t_3}, \boldsymbol{t}') \geq 3$). Hence, we must have $|A \cap C|, |B \cap C| \geq 1$ but this implies $d_H(\boldsymbol{t}, \boldsymbol{t}') \leq 1$ or $d_H(\boldsymbol{t}, \boldsymbol{t}') \geq 3$, but the latter is not possible. We conclude that for every tuple $\boldsymbol{t}' \in R$ we have $d_H(\boldsymbol{t}, \boldsymbol{t}') \leq 1$, hence $R \in \mathcal{Q}$ which contradicts our assumption that $R \notin \mathcal{Q}$.

Hence, there exists tuples $\boldsymbol{t}, \boldsymbol{t}' \in R$ such that $d_H(\boldsymbol{t}, \boldsymbol{t}') \geq 3$. If for every pair of such tuples it is the case that every step, $\boldsymbol{s}$, from $\boldsymbol{t}$ to $\boldsymbol{t}'$ we have $\boldsymbol{s} \in R$, then as no coordinate is constant, we must have $(0, 0, \ldots, 0) \in R$ and $(1, 1, \ldots, 1) \in R$. However, if $(0, 0, \ldots, 0), (1, 1, \ldots, 1) \in R$ and every step from the former to the latter is in R then every tuple with one coordinate set to 1 is in R, too. We can continue this way and get every tuple with two coordinates set to one and then every tuple with k coordinates set to 1 for $k \in [n]$. Hence, we must have $R = \{0, 1\}^n \in \mathcal{Q}$.

We can therefore assume that there exists an coordinate l such that the step $\boldsymbol{s} = \boldsymbol{t} \oplus l$ from $\boldsymbol{t}$ to $\boldsymbol{t}'$ is not in R. Then, as R is a Δ-matroid relation, there exist another coordinate K such that $\boldsymbol{s} \oplus \{K\} \in R$ is a step from $\boldsymbol{s}$ to $\boldsymbol{t}'$. Let X denote the set of coordinates i such that $\boldsymbol{t} \oplus i \notin R$ but $\boldsymbol{t} \oplus \{K, i\} \in R$, furthermore choose $\boldsymbol{t}$ and K such that $|X|$ is maximised and let $X' = X \cup \{K\}$.

Our goal in the rest of the proof is to show that if $X' = [n]$ then $R \in \mathcal{Q}$ and otherwise it is possible to decompose R into a cartesian product with $R\big|_{X'}$ in one factor and $R\big|_{[n] \setminus X'}$ in the other factor. As we have assumed that R cannot be decomposed into a cartesian product we get a contradiction and hence the relation R cannot exist.

Case 1: $|X'| = 2$

We will start with the case when $|X'| = 2$. Assume, without loss of generality, that $X' = \{x, K\}$ then $\boldsymbol{t}, \boldsymbol{t} \oplus \{x, K\} \in R$ and $\boldsymbol{t} \oplus x \notin R$. We will now prove that we cannot have any tuples $\boldsymbol{v}$ in R such that $\boldsymbol{v}\big|_{X'} = (\boldsymbol{t} \oplus x)\big|_{X'}$. If we had such a tuple then $m(\boldsymbol{v}, \boldsymbol{t}, \boldsymbol{t} \oplus \{x, K\}) = \boldsymbol{w} \in R$ due to the fact that $R \in ID_2$ and m is a polymorphism of ID_2. Furthermore, $\boldsymbol{w}$ must have the same value as $\boldsymbol{t}$ on every coordinate except for possibly x and K, this follows from the fact that $\boldsymbol{t}$ has the same value as $\boldsymbol{t} \oplus \{x, K\}$ on every coordinate except for x and K. Hence, the only coordinates for which we do not know the value of $\boldsymbol{w}$ are x and K. However, $\boldsymbol{v}[K] = \boldsymbol{t}[K]$ (due do the construction of $\boldsymbol{v}$ and the fact that $K \in X'$). Hence we must get $\boldsymbol{w}[K] = \boldsymbol{t}[K]$. For $\boldsymbol{w}[x]$ note that $\boldsymbol{v}[x] = (\boldsymbol{t} \oplus \{x, K\})[x]$, hence $\boldsymbol{w}[x] = (\boldsymbol{t} \oplus x)[x]$. We can finally conclude $\boldsymbol{w} = \boldsymbol{t} \oplus x$ which is a contradiction with the construction of X'.

Similar arguments as the above will be used repeatedly in this proof. However, the presentation will not be as detailed as the one above.

17

We split the remaining part of case 1 into two subcases, when $t \oplus K \notin R$ (subcase 1a) and $t \oplus K \in R$ (subcase 1b).

Subcase 1a: $t \oplus K \notin R$ Assume that $t \oplus K \notin R$, then $(t \oplus K)\big|_{X'} \notin R\big|_{X'}$, because given a tuple v such that $v\big|_{X'} = (t \oplus K)\big|_{X'}$ then $m(t, t \oplus \{x, K\}, v) = t \oplus K$, which is not in R by the assumption we made.

Furthermore, for any tuple $v \in R$, $v \oplus x$ is a step from v to either t or $t \oplus \{x, K\}$, but $v \oplus x \notin R$ (because either $v\big|_{X'} = t\big|_{X'}$ which would imply $v \oplus x \notin R$, or $v\big|_{X'} = (t \oplus \{x, K\})\big|_{X'}$ which implies $(v \oplus x)\big|_{X'} = (t \oplus K)\big|_{X'} \notin R\big|_{X'}$).

The only way to get from $v \oplus x$ to something which is in R is by flipping coordinate K, hence $v \oplus \{x, K\} \in R$. This is the end of the case when $t \oplus K \notin R$, because what we have proved above is that R can be decomposed into a cartesian product with the coordinates X' in one factor and $[n] \setminus X'$ in the other factor.

Subcase 1b: $t \oplus K \in R$ We know that $(t \oplus x)\big|_{X'} \notin R\big|_{X'}$. We will now show that for any $v \in R$ such that, $v\big|_{X'}$ is either $t\big|_{X'}$ or $(t \oplus \{x, K\})\big|_{X'}$, we have $v \oplus \{x, K\} \in R$.

To this end, let v be an arbitrary tuple in R satisfying one of the conditions above. We will consider the two possible cases separately.

- If $v\big|_{X'} = t\big|_{X'}$ then $v \oplus x$ is a step from v to $t \oplus \{x, K\}$ and $v \oplus x \notin R$. Furthermore, the only way to get into R is by flipping K hence $v \oplus \{x, K\} \in R$.
- If $v\big|_{X'} = (t \oplus \{x, K\})\big|_{X'}$ then $v \oplus K$ is a step from v to t and $v \oplus K \notin R$. Furthermore, the only way to get into R is by flipping x hence $v \oplus \{x, K\} \in R$.

Now, let v be an arbitrary tuple in R such that $v\big|_{X'} = (t \oplus K)\big|_{X'}$ then $v \oplus K$ is a step from v to t. If $v \oplus K \in R$ or $v \oplus x \in R$ then we are done with this step, so assume that $v \oplus K, v \oplus x \notin R$. However, as R is a Δ-matroid relation there has to exist a coordinate l such that $v \oplus \{K, l\} \in R$. Then we get, $(v \oplus \{K, l\})\big|_{X'} = t\big|_{X'}$ which implies $v \oplus \{x, l\} \in R$ by the argument above. However, this means that $|X|$ is not maximal we could have chosen v, l and X' instead of t, K and X. We conclude that $v \oplus K \in R$.

Finally, let v be an arbitrary tuple in R such that $v\big|_{X'} = t\big|_{X'}$ then $v \oplus K \in R$. To see this note that $m(t \oplus K, v, v \oplus \{x, K\}) = v \oplus K$.

We have now proved that R can be decomposed into a cartesian product with the coordinates X' in one factor and $[n] \setminus X'$ in the other factor for this case too.

As we have assumed that the arity of R is strictly greater than two we have $X' \neq [n]$. Hence, $[n] \setminus X' \neq \emptyset$.

Case 2: $|X'| > 2$

The rest of the proof will deal with the case when $|X'| > 2$. We will begin with establishing a number of claims of R. Assuming that $X' \neq [n]$, our main goal is still to show that R can be decomposed into a cartesian product with X' in one factor and $[n] \setminus X'$ in one factor. If $X' = [n]$ we will show that $R \in \mathcal{Q}$.

Claim 1: if $d_H(x\big|_X, t\big|_X) = 1$ and $x[K] = t[K]$ then $x \notin R$

Let x be a tuple which satisfies the precondition in the claim, assume that $x \in R$, and let $i \in X$ be a coordinate where x differs from t. By the construction of X we have that $t \oplus \{K, i\} \in R$, hence we get $m(t, t \oplus \{K, i\}, x) = t \oplus \{i\} \in R$, which is a contradiction.

Claim 2: if $d_H(x\big|_X, t\big|_X) = m$, for any m such that $2 \leq m \leq |X|$, then $x \notin R$

We will prove this claim by induction on m. For the base case, let $m = 2$. Let $x \in X$ be some coordinate such that $x[x] \neq t[x]$, if $x \in R$ and $x[K] = t[K]$ then $m(t, x, t \oplus \{x, K\}) = t \oplus x \notin R$. Hence $x[K] = t[K]$ is not possible.

On the other hand if $x[K] \neq t[K]$ then $x \oplus K$ is a step from x to t. By the argument in the preceding paragraph we get $x \oplus K \notin R$ (note that $K \notin X$ hence we have $d_H((x \oplus K)\big|_X, t\big|_X) = m$). Furthermore as R is a Δ-matroid we can flip some coordinate $l \in X$ such that $t[l] \neq x[l]$ to get a tuple which is in R ($l \notin X$ will not work as the argument in the preceding paragraph still applies in that case). However, $d_H((x \oplus \{K, l\})\big|_X, t\big|_X) = 1$ hence by claim 1 we get a contradiction.

Now, assume that claim 2 holds for $m = m'$. We will prove that it also holds for $m = m' + 1$ such that $2 < m \leq |X|$. Note that we can use exactly the same argument as above except for the very last sentence in which we appeal to claim 2 with $m = m'$ instead of using claim 1. As we have assumed that claim 2 holds for $m = m'$ we are done.

Claim 3: there is a tuple $z \in R\big|_{X'}$ such that for any tuple $x \in R\big|_{X'}$ we have $d_H(z, x) \leq 1$

If $|X'| > 2$, then there are tuples $t \oplus \{i, K\}$ and $t \oplus \{j, K\}$ for distinct $i, j, K \in X'$ in R. Hence, the tuple $z' = m(t, t \oplus \{i, K\}, t \oplus \{j, K\}) = t \oplus K \in R$. Let $z = z'\big|_{X'}$. We will now show that $d_H(z, x\big|_{X'}) \leq 1$ for every tuple x in R. To this end, let x be an arbitrary tuple in R. By claim 2 we must have $d_H(x\big|_X, t\big|_X) \leq 1$, furthermore if $x[K] = z'[K] \neq t[K]$ then we are done as $d_H(x\big|_{X'}, z'\big|_{X'}) = 1$ in this case. On the other hand, if $x[K] = t[K]$ then claim 1 and claim 2 tells us that we must have $d_H(x\big|_X, t\big|_X) = 0$ in which case claim 3 follows.

Claim 4: if $x \in R$ and $x\big|_{X'} = z \oplus \{i\}$ for some $i \in X'$, then $x \oplus \{i, j\} \in R$ for every $j \in X'$.

Given $j \in X', j \neq i$, there is at least one tuple $v \in R$ such that $v[j] \neq x[j]$ since otherwise the coordinate j would be constant and R could be decomposed into a cartesian product. Hence, $x' = x \oplus j$ is a step from x to v, but claim 3 tells us that $x' \notin R$ and the only way to full fill the two-step axiom is if $x \oplus \{i, j\} \in R$ (due to claim 3 we cannot have $d_H(x, v) = 1$).

We will now prove that R can be decomposed into cartesian product where the coordinates X' make up one factor and $[n] \setminus X'$ make up the other factor. Let $P = R\big|_{X'}$. Our goal is to show that for any $p \in P$ and $v \in R\big|_{[n] \setminus X'}$ we have $(p, v) \in R$ (we have assumed that $X' = \{1, 2, 3, \ldots, |X'|\}$ here).

To this end, let v and v' be arbitrary tuples in R. By claim 3 there either is a coordinate $i \in X'$ such that $(v \oplus i)\big|_{X'} = z$ or $v\big|_{X'} = z$. The same is true for v'; either there is a coordinate $i' \in X'$ such that $(v' \oplus i')\big|_{X'} = z$ or $v'\big|_{X'} = z$,

If $v'\big|_{X'} = v\big|_{X'}$ or $v'\big|_{[n] \setminus X'} = v'\big|_{[n] \setminus X'}$ then we are done, so assume that neither holds.

If $\boldsymbol{v'}\big|_{X'} \neq \boldsymbol{z}$ and $\boldsymbol{v}\big|_{X'} \neq \boldsymbol{z}$ then $\boldsymbol{s} = \boldsymbol{v} \oplus i'$ is a step from $\boldsymbol{v}$ to $\boldsymbol{v'}$ but by claim 3 $\boldsymbol{s} \notin R$ and the only way to go a step from $\boldsymbol{s}$ to $\boldsymbol{v'}$ and get into R is $\boldsymbol{s'} = \boldsymbol{s} \oplus i$, hence $\boldsymbol{s'} \in R$.

For the other case, when $\boldsymbol{v'}\big|_{X'} = \boldsymbol{z}$ and $\boldsymbol{v}\big|_{X'} \neq \boldsymbol{z}$, if there is a coordinate $j \in X'$ such that $\boldsymbol{v'} \oplus j \in R$ then we are back to the previous case, so assume that such a j do not exist. As R is a Δ-matroid there must be a coordinate $x \notin X'$ such that $\boldsymbol{s} = \boldsymbol{v'} \oplus \{i, x\} \in R$, because for some appropriate x, $\boldsymbol{s}$ is a step from $\boldsymbol{v'} \oplus i$ to $\boldsymbol{v}$. Due to claim 4 we will then have $\boldsymbol{v'} \oplus \{x, y\} \in R$ for every $y \in X'$. However, this contradicts the maximality of X since we could have chosen $\boldsymbol{v'}$, x, and X' instead of $\boldsymbol{t}$, K, and X. The conclusion is that if $X' \neq [n]$, then R can be decomposed into a cartesian product. On the other hand, if $X' = [n]$, then we can easily deduce from claim 3 that $R \in \mathcal{Q}$. $\qquad\square$

Proof (Of Lemma 16). Let I be an arbitrary instance of W-MAX ONES(Γ)-2. We will show that the problem is in **PO** by reducing it to an instance I' of ILP-2. For any relation $R \in \Gamma$ of arity n we know, from Lemma 15, that $R \in \mathcal{Q}$. We can assume that R is not the cartesian product of any other two relations, because if it is then every use of R can be substituted by the factors in the cartesian product. If R is unary we can replace $R(x)$ by $x = 0$ or $x = 1$. If $R = EQ^2$ then we can replace $R(x, y)$ by $x = y$ and if $R(x, y) \iff x \neq y$ then we replace $R(x, y)$ by $x = y - 1$.

Now, assume that none of the cases above occur. We will show that

$$R(t_1, t_2, \ldots, t_n) \iff \sum_{i=1}^{n} a_i t_i \leq b \tag{1}$$

for some $a_i \in \{-1, 1\}$ and integer b. Let N be set of negated coordinates of R, i.e., let $N \subseteq [n]$ such that

$$R = \{(f(t_1, 1), f(t_2, 2), \ldots, f(t_n, n)) \mid d_H(\boldsymbol{0}, \boldsymbol{t}) \leq 1\}$$

where $f : \{0, 1\} \times [n] \to \{0, 1\}$ and $f(x, i) = \neg x$ if $i \in N$ and $f(x, i) = x$ otherwise. According to the definition of $\mathcal{Q}$, R can be written on this form. Let $a_i = -1$ if $i \in N$ and $a_i = 1$ otherwise. Furthermore, let $b = 1 - |N|$. It is now easy to verify that (1) holds.

As every variable occur at most twice in I every variable occur at most twice in I' too. Furthermore, the coefficient in front of any variable in I' is either $-1, 0$ or 1, hence the sum of the absolute values in any column in I' is bounded by 2. I' is therefore an instance of ILP-2. If we let the weight function of I' be the same as the weight function in I it easily seen that any solution s to I' is also a solution to I with the same measure. $\qquad\square$

As we are done with ID_2 we will continue with IL_2. A linear system of equations over GF(2) with n equations and m variables can be represented by a matrix A, a constant column vector $\boldsymbol{b}$ and a column vector $\boldsymbol{x} = (x_1, \ldots, x_m)$ of variables. The system of equations is then given by $A\boldsymbol{x} = \boldsymbol{b}$. Assuming that the rows of A are linearly independent the set of solutions to $A\boldsymbol{x} = \boldsymbol{b}$ are

$$\left\{(\boldsymbol{x'}, \boldsymbol{x''}) \mid \boldsymbol{x''} \in \mathbb{Z}_2^{n-m} \text{ and } \boldsymbol{x'} = A'\boldsymbol{x''} + \boldsymbol{b'}\right\}.$$

where $\boldsymbol{x'} = (x_1, \ldots, x_m)$, $\boldsymbol{x''} = (x_{m+1}, \ldots, x_n)$ and A' and $\boldsymbol{b'}$ are suitably chosen.

If there is a column in A' with more than one entry which is equal to 1 (or, equivalently more than one non-zero entry), then we say that the system of equations is *coupled*.

Lemma 24. *Let Γ be a conservative constraint language such that $\Gamma \subseteq IL_2$. If there is a relation $R \in \Gamma$ such that R is the set of solutions to a coupled linear system of equations over $GF(2)$ then* W-MAX ONES(Γ) $\leq_L$ W-MAX ONES(Γ)-2*, otherwise* W-MAX ONES(Γ)-2 *is in* **PO**.

Proof. First note that every relation $R \in IL_2$ is the set of solutions to a linear system of equations over GF(2) [11].

We will start with the hardness proof. To this end we will construct a 2-representation of EQ^3. Let $R \in \Gamma$ be defined by

$$R = \left\{ (\boldsymbol{x'}, \boldsymbol{x''}) \mid \boldsymbol{x''} \in \mathbb{Z}_2^{n-m} \text{ and } \boldsymbol{x'} = A'\boldsymbol{x''} + \boldsymbol{b'} \right\}.$$

for some n, m, A' and $\boldsymbol{b'}$. Furthermore, we can assume that there is one column (say j) in A' with more than one entry equal to 1 (say i and i'). Hence, A'_{ij} and $A'_{i'j}$ are equal to 1. Our first implementation consists of R and a number of c_0 constraints,

$$Q(x_{j+m}, x_i, x_{i'}) \iff R(x_1, \ldots, x_n) \bigwedge_{\substack{k:\ m+1 \leq k \leq n \\ k \notin \{j+m, i, i'\}}} c_0(x_k).$$

This implementation leaves us with three cases, the first one is $Q = EQ^3$, in which case we are done. The other two cases are $Q = \{(0, 0, 1), (1, 1, 0)\}$ and $Q = \{(0, 1, 0), (1, 0, 1)\}$. We will give an implementation of EQ^3 with the first case, the other one is similar. Note that

$$EQ^3(y_1, y_2, y_3) \iff \exists z, z' : Q(y_1, y_2, z) \land Q(y_3, z', z).$$

For the containment proof note that every relation in Γ is the set of solutions to some non-coupled linear system of equations over GF(2). The set of feasible solutions to an instance of W-MAX ONES(Γ)-2 is therefore the set of solutions to a linear system of equations over GF(2) with the property that every variable occurs at most twice. This problem is solvable by Edmonds and Johnson's method [13]. $\square$

Corollary 25. *Let Γ be a conservative constraint language such that* $\mathrm{Pol}(\Gamma) = \mathrm{Pol}(IL_2)$ *if there is a relation $R \in \Gamma$ such that R is not a Δ-matroid relation then* W-MAX ONES(Γ)-2 *is* **APX**-*complete, otherwise* W-MAX ONES(Γ)-2 *is in* **PO**.

Proof. Given a constraint language Γ such that $\mathrm{Pol}(\Gamma) = \mathrm{Pol}(IL_2)$ then W-MAX ONES(Γ) is **APX**-complete [20].

It is not hard to see that for a relation $R \in \Gamma$, R is not a Δ-matroid relation if and only if R is the set of solutions to a coupled system of equations. (The "if"-part follows directly from the representation of relation Q in Lemma 24.)

Hence, if there is a relation $R \in \Gamma$ such that R is not a Δ-matroid relation then we get **APX**-completeness for W-MAX ONES(Γ)-2 from Lemma 24. On the other hand, if there is no non-Δ-matroid relation in Γ then no relation is the set of solutions to a coupled system of equations and hence we get tractability from Lemma 24. $\square$

The final sub-case is $\Gamma \subseteq IE_2$.

Proof (Of Lemma 17). For the containment note that the algorithm in Lemma 21 can be used, as an instance of W-MAX ONES$(\{c_0, c_1, R\})$-2 can easily be reduced to an instance of W-MAX ONES$(\{c_0, c_1, NAND^2\})$-4.

We will do a reduction from MAX 2SAT-3 (i.e., MAX 2SAT where every variable occurs at most three times), which is **APX**-complete [1, Chap. 8]. The reduction is based on Theorem 1 in [2], which in turn is based on some of Viggo Kann's work on 3-dimensional matching [18].

We will do the reduction in two steps, we will first reduce MAX 2SAT-3, to a restricted variant of MIS-3. More precisely the graphs produced by the reduction will have maximum degree three and it will be possible to "cover" the graphs with R (we will come back to this soon).

Let $I = (V, C)$ be an arbitrary instance of MAX 2SAT-3. We will construct an instance $I' = (G, w)$, where $G = (V', E')$ and $w : V' \to \mathbb{Q}$, of weighted maximum independent set. We can assume, without loss of generality, that each variable in I occurs at least once unnegated and at least once negated. For a node $v \in V$ construct four paths with three nodes each. Sequentially label the nodes in path number x by p_{x1}, p_{x2}, p_{x3}. Construct three complete binary trees with four leaves each and label the roots of the trees with $v_1, \neg v_1, v_2$ (or $v_1, \neg v_1, \neg v_2$ if v occurs once unnegated and twice negated). Finally, identify the leaves of each of the trees with the nodes in the paths with similar labels, where two labels p_{xy} and p_{uv} are similar if $y = v$. Figure 2 contains this gadget for our example variable, v.
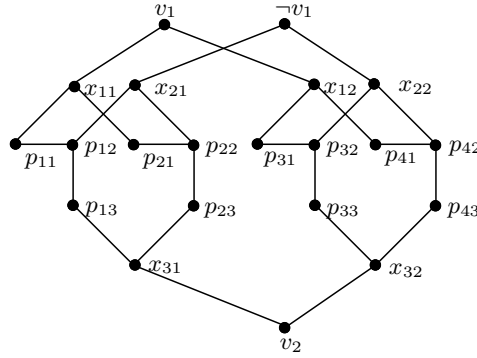

Fig. 2: The graph gadget for the variable v which occurs three times, two times unnegated and one time negated.

Let the w be defined as follows, $w(p_{12}) = w(p_{22}) = w(p_{32}) = w(p_{42}) = 2.25$, $w(x_{21}) = w(x_{22}) = 2$ and $w(\cdot) = 1$ otherwise.

Denote the disjoint union of those paths and trees for all variables by X. A solution S for the independent set problem for X will be called *consistent* if for each variable, v, (which occurs twice unnegated and once negated) we have $v_1, v_2 \in S$ and $\neg v_1 \notin S$ or

vice versa (i.e., $v_1, v_2 \notin S$ and $\neg v_1 \in S$). It is not hard to verify (e.g., with a computer assisted search) that the optimal solutions to X are consistent. Furthermore, for each consistent solution there is a solution which is optimal and includes or excludes the v_i's and $\neg v_i$'s in the same way.

For each clause $c \in C$, containing the literals l_1 and l_2, add two fresh nodes l_1 and l_2 to G'. Connect l_1 and l_2 with an edge and connect l_1 with the node which is labelled with this literal (one of the roots of the trees). Do the same thing for l_2.

We deduce that given a solution to I' it is possible to construct a consistent solution with a measure which is greater than or equal to the measure of the original solution. The only case we have to be careful about is when we are given a solution S where $v_1, v_2, \neg v_1 \notin S$. In this case the measure of the gadget is strictly less than the locally optimal solution. Hence, we can add $\neg v_1$ which, in the worst case, will force us to remove one node which was attached to $\neg v_1$ due to the clause which $\neg v_1$ is in. However, this loss will be made up for as we can assign an optimal solution to the gadget.

We have $\mathrm{OPT}(I') \leq |V|K + \mathrm{OPT}(I)$ where $K = 14$ is the optimum value for our gadget. As $\mathrm{OPT}(I) \geq |C|/2$ and $|V| \leq 3|C|$ we get $\mathrm{OPT}(I') \leq 3K|C| + \mathrm{OPT}(I) \leq (6K+1)\mathrm{OPT}(I)$, hence $\beta = 6K + 1$ is an appropriate parameter for an L-reduction.

For any consistent solution S' to I' we can construct a solution s to I as follows, for each variable $v \in V$ let $s(v) = \mathrm{TRUE}$ if $v_i \notin S'$ and $s(v) = \mathrm{FALSE}$ otherwise. We will then have $|\mathrm{OPT}(I) - m(I, s)| = |\mathrm{OPT}(I') - m(I', S')|$. Hence, $\gamma = 1$ is an appropriate parameter for the L-reduction.

Using c_0 and R it is possible to 2-represent $NAND^2(x, y)$. To reduce I' to an instance of W-MAX ONES$(\{R\})$-2 note that we can "cover" each variable gadget with R and $NAND^2$, see Figure 3 how this is done. Furthermore, in the covering we have only used v_1, $\neg v_1$ and v_2 once so it wont be any problems with connecting the gadgets to each other with $NAND^2$ constraints. $\qquad\square$

We need the following result which has been proved by Feder [14, Theorem 3, fact 1].

Lemma 26. *Given a relation R which is not closed under $f(x, y) = x \vee y$, then R can 2-represent either $NAND^2$ or $x \neq y$.*

Corollary 27. *Given a relation $R \in IE_2$ which is not closed under $f(x, y) = x \vee y$, then R can 2-represent $NAND^2$.*

Proof. From Lemma 26 we deduce that R can 2-represent either $NAND^2$ or $x \neq y$, but the latter is not contained in IE_2, hence we must have the former. $\qquad\square$

Lemma 28. *Let Γ be a conservative constraint language, if $IS_{12}^2 \subseteq \langle \Gamma \rangle \subseteq IE_2$, and there is a relation $R \in \Gamma$ such that R is not a Δ-matroid relation, then* W-MAX ONES(Γ)-2 *is* **APX***-hard.*

Some parts of the following proof is similar to Feder's proof in [14] that non-Δ-matroids causes $\mathrm{CSP}(\cdot)$-2 to be no easier than $\mathrm{CSP}(\cdot)$.

Proof. As R is not a Δ-matroid relation there exists tuples $t, t' \in R$ such that $d_H(t, t') \geq 3$ and a step $s \notin R$ from t to t' such that no step from s to t' is contained in R.

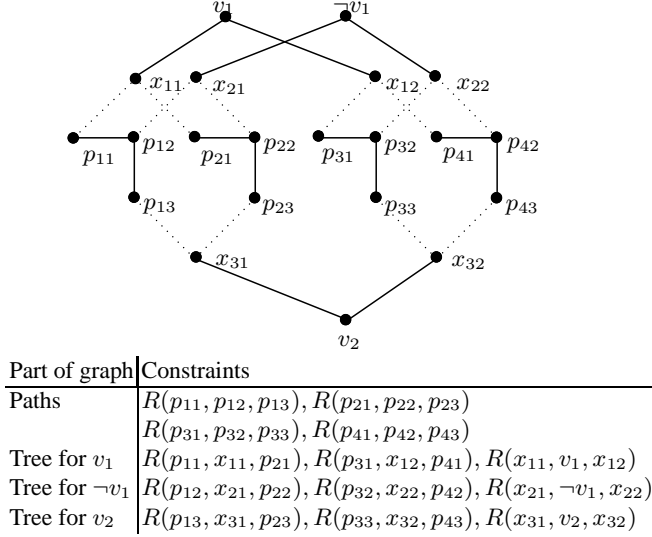| Part of graph	Constraints
Paths	$R(p_{11}, p_{12}, p_{13}), R(p_{21}, p_{22}, p_{23})$
	$R(p_{31}, p_{32}, p_{33}), R(p_{41}, p_{42}, p_{43})$
Tree for v_1	$R(p_{11}, x_{11}, p_{21}), R(p_{31}, x_{12}, p_{41}), R(x_{11}, v_1, x_{12})$
Tree for $\neg v_1$	$R(p_{12}, x_{21}, p_{22}), R(p_{32}, x_{22}, p_{42}), R(x_{21}, \neg v_1, x_{22})$
Tree for v_2	$R(p_{13}, x_{31}, p_{23}), R(p_{33}, x_{32}, p_{43}), R(x_{31}, v_2, x_{32})$

Fig. 3: The gadget for the variable v covered by the relation R. Note that each variable occurs at most twice and that v_1, v_2 and $\neg v_1$ occurs once. Constraints with overlapping nodes are represented by two different line styles in the graph: solid and dotted.

Let n be the arity of R and let $X \subseteq [n]$ be the set of coordinates where t differs from t', i.e., $t = t' \oplus X$. Furthermore, let $k \in [n]$ be the coordinate where s differs from t.

By using projections and the c_0 and c_1 constraints together with R we can 2-represent a new relation, P, which is not a Δ-matroid relation and has arity 3. To do this, choose a subset $X' \subset X$ of minimal cardinality such that $k \in X'$ and $t \oplus X' \in R$. Note that $|X'| \geq 3$. Let a and b be two distinct coordinates in X' which differs from k. Construct P as follows:

$$P(x_k, x_a, x_b) \iff R(x_1, x_2, \ldots, x_n) \bigwedge_{\substack{l \in [n] \setminus X' \\ t[l]=1}} c_1(x_l) \bigwedge_{\substack{l \in [n] \setminus X' \\ t[l]=0}} c_0(x_l).$$

Furthermore, let $v = t\big|_{\{k,a,b\}}$ and $v' = t'\big|_{\{k,a,b\}}$ we then have $v, v' \in P$ and $v \oplus 1, v \oplus \{1,2\}, v \oplus \{1,3\} \notin P$. Hence, depending on v and which other tuples that are in P we get a number of possibilities. We will use the following notation: $a = v \oplus 2, b = v \oplus 3$ and $c = v \oplus \{2,3\}$. Zero or more of a, b and c may be contained in P. Tables 4–7 list the possible relations we can get, up to permutations of the coordinates. Note that $a \in P, b, c \notin P$ and $b \in P, a, c \notin P$ are equivalent if we disregard permutations of the coordinates. Similarly $a, c \in P, b \notin P$ and $b, c \in P, a \notin P$ are equivalent.

Some of the relations are not in IE_2 and can therefore be omitted from further consideration (it is clear that if P is not in IE_2 then R is not in IE_2 either, which is a contradiction with the assumptions in the lemma). Others can 2-represent EQ^3, or can

do so together with $NAND^2$. As an example consider $A5$, then

$$\exists y_1, y_2, y_3, z_1, z_2, z_3 : A5(y_1, x_1, z_1) \wedge NAND^2(z_1, y_2) \wedge$$
$$A5(y_2, x_2, z_2) \wedge NAND^2(z_2, y_3) \wedge$$
$$A5(y_3, x_3, z_3) \wedge NAND^2(z_3, y_1)$$

is a 2-representation of $EQ^3(x_1, x_2, x_3)$. Similar constructions works for some of the other relations. If we can 2-represent EQ^3 then we get **poly-APX**-hardness due to the construction in Lemma 19, Lemma 6 and a simple reduction from MIS. Information about which relations this applies to is contained in Table 2.

Furthermore, some of the relations can 2-represent other relations in the table, see Table 3 for those. This implies that the only relation that is left to prove **APX**-hardness for is $ABC1$. We will do this with a reduction from MIS-3. Let $G = (V, E)$ be an instance of MIS-3, we will construct an instance $I' = (V', C', w')$ of W-MAX ONES(Γ)-2 with the assumption that $ABC1 \in \Gamma$. Furthermore, due to Lemma 19 and Corollary 27 we are free to assume that $NAND^2 \in \Gamma$. For every variable $v \in V$, if there are three occurrences of v in I add one fresh variable for each occurrence of v in I to V', name those fresh variables v_1, v_2 and v_3. If there are less than three occurrences add v to V'. Furthermore, for each edge (v, x) for some $x \in V$ add a $NAND^2(v_i, x_j)$ constraint to C'. So far I' is an instance where each variable occurs at most twice and the variables which corresponds to nodes in G with degree three occurs once in I'.

For each node $v \in V$ with degree three add the constraint $ABC1(v_1, v_2, v_3)$ to C'. Finally, let $w(x) = 1$ for every $x \in V$ with degree less than three and $w(v_1) = 1$ and $w(v_2) = w(v_3) = 0$ for every $v \in V$ with degree three. For every solution s to I' we can construct a solution S to I such that $m(I', s) = m(I, S)$ to see this note that if $s(x) = 1$ for some variable x then due to the $ABC1$ constraints the other occurrences of x also have the value 1. On the other hand, if $s(x) = 0$ then we can set the other occurrences of x to 0 without changing the measure of the solution and without conflicts with any constraints. This implies that there is an S-reduction from MIS-3 to W-MAX ONES(Γ)-2. $\square$

The results obtained in Lemma 28 is not optimal for all non-Δ-matroids. It is noted in the proof that we get **poly-APX**-hardness results for some of the relations, but we do not get this for all of them. In particular we do not get this for $A3$, $AB1$, $BC4$, $ABC1$, $ABC3$, $ABC5$ and $ABC6$. However, $ABC5$ is contained in **APX** by Lemma 17.

We are now finally ready to state the proof of the classification theorem for two variable occurrences.

Proof (Of Theorem 13, part 1). Follows from Khanna et al's results on W-MAX ONES [20]. $\square$

Proof (Of Theorem 13, part 2). Follows from Corollary 25, Lemma 14 and Lemma 16. $\square$

Proof (Of Theorem 13, part 3). Follows from Lemma 28. $\square$

Proof (Of Theorem 13, part 3). Follows from [14, Theorem 4]. $\square$

Relation	Implementation or comment	Relation	Implementation or comment
1	EQ^2	BC2	Not in IE_2
2	Not in IE_2	BC3	Not in IE_2
A1	EQ^2	BC4	See Table 3
A2	Not in IE_2	AB1	See Table 3
A3	See Table 3	AB2	Not in IE_2
A4	Not in IE_2	AB3	Not in IE_2
A5	$NAND^2$	AB4	Not in IE_2
A6	EQ^2	AB5	Not in IE_2
C1	EQ^2	AB6	Not in IE_2
C2	$NAND^2$	ABC1	See Lemma 28
C3	Not in IE_2	ABC2	Not in IE_2
C4	Not in IE_2	ABC3	See Table 3
C5	Not in IE_2	ABC4	Not in IE_2
C6	EQ^2	ABC5	See Lemma 17
BC1	EQ^2	ABC6	See Table 3

Table 2: Non-Δ-matroid relations in Lemma 28. If there is a relation in the "Implementation or comment" column then this relation can 2-represent EQ^3 together with the noted relation. If this second relation is EQ^2 then the relation can in fact 2-represent EQ^3 on its own, EQ^2 is not needed.

Relation	Implements	Implementation
A3	$ABC5$	$\exists x' : A3(x_1, x', x_3) \wedge NAND^2(x', x_2)$
AB1	$ABC5$	$\exists x', x'' : AB1(x_1, x', x'') \wedge NAND^2(x', x_2) \wedge NAND^2(x'', x_3)$
BC4	$ABC5$	$\exists x' : BC4(x_1, x', x_2) \wedge NAND^2(x', x_3)$
ABC3	$ABC5$	$\exists x' : ABC3(x_1, x', x_3) \wedge NAND^2(x', x_2)$
ABC6	$ABC1$	$\exists x' : ABC6(x', x_2, x_3) \wedge NAND^2(x', x_1)$

Table 3: Implementations in Lemma 28

		000	100	010	110	101	111
111	110	111	011	101	001	010	000
000	001	010	110	000	100	111	101
1	2	A1	A2	A3	A4	A5	A6

Table 4: Non-Δ-matroid relations where $a, b, c \notin P$ followed by the relations where $a \in P$ and $b, c \notin P$.

000	100	010	110	011	111
111	011	101	001	100	000
011	111	001	101	011	100
C1	C2	C3	C4	C5	C6

Table 5: Non-Δ-matroid relations where only $c \in P$.

000	100	010	001	000	100	010	110	011	111
111	011	101	110	111	011	101	001	100	000
001	101	011	000	010	110	000	100	001	101
011	111	001	010	001	101	011	111	010	110
$\overline{BC1}$	$\overline{BC2}$	$\overline{BC3}$	$\overline{BC4}$	$\overline{AB1}$	$\overline{AB2}$	$\overline{AB3}$	$\overline{AB4}$	$\overline{AB5}$	$\overline{AB6}$

Table 6: Non-Δ-matroid relations where $b, c \in P$ and $a \notin P$ followed by relations where $a, b \in P$ and $c \notin P$.

000	100	010	110	011	111
111	011	101	001	100	000
010	110	000	100	001	101
001	101	011	111	010	110
011	111	001	101	000	100
$\overline{ABC1}$	$\overline{ABC2}$	$\overline{ABC3}$	$\overline{ABC4}$	$\overline{ABC5}$	$\overline{ABC6}$

Table 7: Non-Δ-matroid relations where $a, b, c \in P$.

Proofs for Results in Section 5

Proof (Of Theorem 18). Let Γ be a non-1-valid constraint language and k an integer such that W-MAX ONES$(\Gamma \cup \{c_0, c_1\})$-$k$ (this problem will hereafter be denoted by Π_{01}) is **NP**-hard. We will prove the theorem with a reduction from Π_{01} to W-MAX ONES(Γ)-k (hereafter denoted by Π).

As Γ is not 1-valid there exists a relation $R \in \Gamma$ such that $(1, \ldots, 1) \notin R$. Let r be the arity of R and let t be the tuple in R with the maximum number of ones. Assume, without loss of generality, that $t = (0, 1, \ldots, 1)$.

The assumption in the theorem implies that it is **NP**-hard to decide the following question: given an instance $I = (V, C, w)$ of Π_{01} and an integer K is OPT$(I) \geq K$?

Let $I = (V, C, w), K$ be an arbitrary instance of the decision variant of Π_{01}. We will transform I into an instance $I' = (V', C', w'), K'$ of the decision variant of Π by first removing constraint applications using c_0 and then removing constraint applications using c_1.

At the start of the reduction let $V' = V$ and $C' = C$. For each constraint $(c_0, (v)) \in C'$ replace this constraint with $(R, (v, v_1, \ldots, v_{r-1}))$ where $v_1, \ldots, v_{r-1}$ are fresh variables, furthermore add the constraint $(c_1, (v_k))$ for $k = 1, \ldots, r-1$ to C'.

Let c be the number of variables which are involved in c_1 constraints. For each constraint using c_1, $(c_1, (v)) \in C'$, remove this constraint and set $w'(v) = L + w(v)$, where L is a sufficiently large integer ($L = 1 + \sum_{v \in V} w(v)$ is enough). For every variable v which is not involved in a c_1 constraint let $w'(v) = w(v)$.

Finally let $K' = K + cK$. Given a solution s' to I' such that $m(I', s') \geq K'$ it is clear that this solution also is a solution to I such that $m(I, s') \geq K$. Furthermore, if there is a solution s to I such that $m(I, s) \geq K$ then s is a solution I' such that $m(I', s) \geq K'$. $\square$

27

References

1. G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and approximation: Combinatorial optimization problems and their approximability properties*. Springer, 1999.

2. P. Berman and T. Fujito. On the approximation properties of independent set problem in degree 3 graphs. In *WADS '95*, pages 449–460, 1995.

3. A. Bouchet. Matchings and Δ-matroids. *Discrete Appl. Math.*, 24(1-3):55–62, 1989.

4. A. Bouchet and W. H. Cunningham. Delta-matroids, jump systems, and bisubmodular polyhedra. *SIAM J. Discret. Math.*, 8(1):17–32, 1995.

5. E. Böhler, N. Creignou, S. Reith, and H. Vollmer. Playing with boolean blocks, part I: Post's lattice with applications to complexity theory. *SIGACT News*, 34(4):38–52, 2003.

6. E. Böhler, N. Creignou, S. Reith, and H. Vollmer. Playing with boolean blocks, part II: Constraint satisfaction problems. *SIGACT News*, 35(1):22–35, 2004.

7. E. Böhler, S. Reith, H. Schnoor, and H. Vollmer. Simple bases for boolean co-clones. Technical report, Theoretische Informatik, Universität Würzburg, 2005.

8. R. Carr and O. Parekh. A 1/2-integral relaxation for the a-matching problem. *Operations Research Letters*. In Press, Available online 22 September 2005.

9. G. Cornuéjols. General factors of graphs. *J. Comb. Theory Ser. B*, 45(2):185–198, 1988.

10. N. Creignou, S. Khanna, and M. Sudan. *Complexity classifications of boolean constraint satisfaction problems*. Society for Industrial and Applied Mathematics, 2001.

11. N. Creignou, P. Kolaitis, and B. Zanuttini. Preferred representations of boolean relations. Technical Report TR05-119, ECCC, 2005.

12. V. Dalmau and D. Ford. Generalized satisfability with limited occurrences per variable: A study through delta-matroid parity. In *MFCS '03*, pages 358–367, 2003.

13. J. Edmonds and E. L. Johnson. Matching: A well-solved class of integer linear programs. In *Combinatorial Optimization – Eureka, You Shrink!*, pages 27–30, 2001.

14. T. Feder. Fanout limitations on constraint systems. *Theor. Comput. Sci.*, 255(1-2):281–293, 2001.

15. T. Feder and D. Ford. Classification of bipartite boolean constraint satisfaction through delta-matroid intersection. Technical Report TR05-016, ECCC, 2005.

16. G. Istrate. Looking for a version of schaefer's dichotomy theorem when each variable occurs at most twice. Technical Report TR652, University of Rochester Computer Science Department, 1997.

17. P. Jeavons, D. Cohen, and M. Gyssens. Closure properties of constraints. *Journal of the ACM*, 44:527–548, 1997.

18. V. Kann. Maximum bounded 3-dimensional matching in max snp-complete. *Inf. Process. Lett.*, 37(1):27–35, 1991.

19. S. Khanna, R. Motwani, M. Sudan, and U. V. Vazirani. On syntactic versus computational views of approximability. *SIAM J. Comput.*, 28(1):164–191, 1998.

20. S. Khanna, M. Sudan, L. Trevisan, and D. P. Williamson. The approximability of constraint satisfaction problems. *SIAM J. Comput.*, 30(6):1863–1920, 2000.

21. J. Kratochvíl, P. Savický, and Z. Tuza. One more occurrence of variables makes satisfiability jump from trivial to np-complete. *SIAM J. Comput.*, 22(1):203–210, 1993.

22. C. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. In *STOC '88*, pages 229–234. ACM Press, 1988.

23. E. Post. The two-valued iterative systems of mathematical logic. *Annals of Mathematical Studies*, 5:1–122, 1941.

24. W. R. Pulleyblank. Matchings and extensions. In *Handbook of combinatorics (vol. 1)*, pages 179–232. MIT Press, 1995.

25. R. Pöschel and L. Kaluznin. *Funktionen- und Relationenalgebren*. DVW, Berlin, 1979.
26. T. J. Schaefer. The complexity of satisfiability problems. In *STOC '78*, pages 216–226. ACM Press, 1978.