Boolean Constraint Satisfaction: Complexity Results for Optimization Problems with Arbitrary Weights

Peter Jonsson
Department of Computer and Information Science
Linköpings Universitet
S-581 83 Linköping, Sweden
Fax: +46 13 28 26 06
Email: `petej@ida.liu.se`

Abstract

A boolean constraint satisfaction problem consists of some finite set of constraints (*i.e.*, functions from 0/1-vectors to $\{0, 1\}$) and an instance of such a problem is a set of constraints applied to specified subsets of n boolean variables. The goal is to find an assignment to the variables which satisfy all constraint applications. The computational complexity of optimization problems in connection with such problems has been studied extensively but the results have relied on the assumption that the weights are non-negative. The goal of this article is to study variants of these optimization problems where arbitrary weights are allowed. For the four problems that we consider, we give necessary and sufficient conditions for when the problems can be solved in polynomial time. In addition, we show that the problems are NP-equivalent in all other cases.

Keywords: Computational complexity, boolean constraints, optimization problems

1 Introduction

A boolean constraint satisfaction problem (CSP) consists of a finite set $\mathcal{F} = \{f_1, \ldots, f_m\}$ of boolean functions $f_i : \{0,1\}^{k_i} \rightarrow \{0,1\}$ called *constraints*. An instance of such a problem is a set of *constraint applications* where each application is a constraint in $\mathcal{F}$ applied to a specified subset of n boolean variables. The decision version of this problem (denoted SAT($\mathcal{F}$)) decides whether there is an assignment to the variables such that all constraint applications are satisfied, *i.e.*, each application evaluates to 1. Schaefer [15] has studied SAT($\mathcal{F}$) and proved that for all choices of $\mathcal{F}$, SAT($\mathcal{F}$) is either polynomial or NP-complete. Furthermore, he gives six classes of boolean functions such that if all functions in $\mathcal{F}$ fall entirely within any of these classes, the problem is polynomial. Similar complete classifications of the hardness of computational problems have been given by, for instance, Fortune *et al.* [3] (the subgraph homeomorphism problem), Hell and Nešetřil [5] (H-colouring of graphs) and Creignou [1] (the maximum generalized satisfiability problem).

Previously presented complexity results for optimization versions of constraint satisfaction problems have been proved under the assumption that only non-negative weights are allowed (*cf.* Khanna *et al.* [12, 13]). In the sequel, we will study four optimization problems where we allow arbitrary weights. The first problem is MIN AW CSP($\mathcal{F}$). Here, we are given an instance of SAT($\mathcal{F}$) together with a weight for each constraint application and the objective is to minimize the sum of the weights of the unsatisfied constraints. In the maximization variant MAX AW CSP($\mathcal{F}$), the objective is to maximize the sum of the satisfied constraints instead. Many problems that have received considerable attention in the literature are subsumed by these two problem. Examples of such problems are MAX CUT, CLIQUE and NEAREST CODEWORD.

The second class of problems that we consider consists of MIN AW ONES($\mathcal{F}$) and MAX AW ONES($\mathcal{F}$). In these problems, we are given an instance of SAT($\mathcal{F}$) and a weight w_i for each variable. The objective is to find a boolean assignment $(x_1, \ldots, x_n)$ that either minimizes or maximizes the sum $\sum_{i=1}^{n} w_i x_i$ and which satisfies all constraints. Several previously studied problems can be cast as MAX/MIN AW ONES($\mathcal{F}$) problems. For example, certain variations of the ZERO/ONE LINEAR PROGRAMMING (ZOLP) problem, defined as

INSTANCE: An $m \times n$ integer matrix A, an m-vector b and an n-vector c of integers.
OBJECTIVE: Find a vector $x \in \{0,1\}^n$ that maximizes $\sum_{i=1}^{n} c_i x_i$ and satisfies $Ax \geq b$.

are MAX AW ONES($\mathcal{F}$) problems for suitable choices of $\mathcal{F}$. Consider, for instance, ZOLP restricted to constraint matrices with at most two non-zero entries per row. This problem (or variants of it) has been studied in a number of papers [4, 7] and it is easily seen to be a MAX AW ONES($\mathcal{F}$) problem. However, observe that the general ZOLP problem is not a MAX AW ONES($\mathcal{F}$) problem for any $\mathcal{F}$ since the set of constraints must be finite. Other examples are the MAX ONES($\mathcal{F}$) and MIN ONES($\mathcal{F}$)[1] problems which have been subject to studies by Khanna *et al.* [12, 13]. In MAX ONES($\mathcal{F}$), we are given an instance of SAT($\mathcal{F}$) together with a non-negative weight for each boolean variable and the objective is to find a boolean assignment of maximum weight that satisfies all constraint applications. This problem is the MAX AW ONES($\mathcal{F}$) problem restricted to non-negative weights.

In this article, we prove that the problems under consideration are either polynomial or NP-equivalent depending on the choice of the constraint set $\mathcal{F}$. Hence, we have obtained a dichotomy result similar in spirit to Schaefer's [15] dichotomy theorem for SAT($\mathcal{F}$). Furthermore, our classifications are compact; for each problem, there is only a finite number of maximal classes of constraints that make the problem solvable in polynomial time. We also show that it is a polynomial-time problem to decide whether a given constraint (presented as its truth table) is a member or not of these subclasses.

The rest of this article is structured in the following way: Section 2 contains the basic definitions and some auxiliary results. In Section 3, we prove the dichotomy theorem for MIN/MAX AW CSP($\mathcal{F}$). The dichotomy theorem for MIN/MAX AW ONES($\mathcal{F}$) are collected in Sections 4 and 5; Section 4 contains the polynomiality results and Section 5 the hardness results. In section 6, we present polynomial-time algorithms for checking membership in the polynomial-time solvable classes.

2 Preliminaries

Recall that a boolean constraint is a function $f : \{0,1\}^k \rightarrow \{0,1\}$ and a constraint application is a pair $\langle f, (i_1, \ldots, i_k) \rangle$ where the $i_j \in \{1, \ldots, n\}$ indicate to which k of the n boolean variables the constraint is applied. We require that $i_j \neq i_{j'}$ for $j \neq j'$ and we assume that all constraints are non-trivial, *i.e.*, there exist $x, y \in \{0,1\}^k$ such that $f(x) = 1$ and $f(y) = 0$. We say that a constraint is *satisfied* by an assignment of the variables iff the constraint evaluates to 1. A constraint family $\mathcal{F}$ is a finite set of constraints. Note that our definition of boolean constraints agree with that of Khanna *et al.* [12, 13] but not with Schaefer [15] nor Creignou and Hermann [2]. This

[1] The problems MAX ONES and MIN ONES are not defined consistently in the literature. The definitions by Khanna *et al.* must not be mixed up with the definitions by, for instance, Kann [9, 10].

follows from the fact that we do not allow the use of variable replication in constraint applications; we insist that the indices must be distinct. Some of the implications of using this weaker formulation of constraints are discussed in Khanna *et al.* [13].

We now describe the optimization problems considered in this article and it should be noted that all these problems are easy for NP. The first problem is MIN AW CSP($\mathcal{F}$) which is defined as follows:

INSTANCE: The instance consists of two parts: (1) a set of m constraint applications of the form $\{\langle f_j, (i_1(j), \ldots, i_{k_j}(j))\rangle\}_{j=1}^m$, on boolean variables $x_1, \ldots, x_n$ where $f_j \in \mathcal{F}$ and k_j is the arity of f_j; and (2) m weights $w_1, \ldots, w_m \in \mathbb{Z}$.
OBJECTIVE: Find a boolean assignment to x_i's which minimizes the sum of the weights of the unsatisfied constraints.

The problem MAX AW CSP($\mathcal{F}$) is defined analogously but with the objective of maximizing the sum of the weights of the satisfied constraints.

The problem MAX AW ONES($\mathcal{F}$) has the following definition.

INSTANCE: The instance consists of two parts: (1) a set of m constraint applications of the form $\{\langle f_j, (i_1(j), \ldots, i_{k_j}(j))\rangle\}_{j=1}^m$, on boolean variables $x_1, \ldots, x_n$ where $f_j \in \mathcal{F}$ and k_j is the arity of f_j; and (2) m weights $w_1, \ldots, w_m \in \mathbb{Z}$.
OBJECTIVE: Find a boolean assignment to x_i's which satisfies all the constraints and maximizes $\sum_{i=1}^n w_i x_i$.

Analogously, we have a minimization problem MIN AW ONES($\mathcal{F}$) where the objective is to minimize $\sum_{i=1}^n w_i x_i$. Given an instance Θ of an optimization problem such as MAX AW ONES, let $opt(\Theta)$ denote the optimal value of Θ. We write $opt(\Theta) = \perp$ to indicate that Θ has no feasible solution.

Let Γ be an instance of MAX AW ONES($\mathcal{F}$) with the objective of maximizing $\sum_{i=1}^n w_i x_i$ over a set $\mathcal{C}$ of constraint applications. Define the instance Γ' of MIN AW ONES($\mathcal{F}$) such that the objective is to minimize $\sum_{i=1}^n -w_i x_i$ over $\mathcal{C}$. It is easy to see that $opt(\Gamma) = -opt(\Gamma')$ and we have proved the following lemma.

Lemma 2.1 MAX AW ONES($\mathcal{F}$) is polynomial (NP-hard) iff MIN AW ONES($\mathcal{F}$) is polynomial (NP-hard).

Thus, it is sufficient to only consider MAX AW ONES($\mathcal{F}$) in the sequel.

We continue by describing some classes of constraints. Most of these definitions are taken from [2, 12, 13, 15].

Let $f(x_1, \ldots, x_k)$ be a constraint and σ a propositional formula over the variables $x_1, \ldots, x_k$. We say that f and σ are *logically equivalent* iff every assignment of the variables that satisfies f satisfies σ and vice versa. Similarly, if E is a set of equations over $GF(2)$ (the field with two elements) with variables $x_1, \ldots, x_k$, we say that f is logically equivalent to E iff every assignment of the variables that satisfies f satisfies the equations in E and vice versa.

A *clause* is a disjunction of one or more *literals*, *i.e.*, negated or unnegated propositional variables. A propositional formula is on *conjunctive normal form* (CNF) iff it is a conjunction of clauses. A clause is *unary* iff it contains exactly one literal. A clause is *implicative* iff it contains exactly one negated and one unnegated variable.

Definition 2.2 Let f be an arbitrary constraint. We say that

- f is *0-valid* (resp. *1-valid*) iff $f(0, \ldots, 0) = 1$ (resp. $f(1, \ldots, 1) = 1$);
- f is *weakly positive* (resp. *weakly negative*) iff it is logically equivalent to a CNF formula having at most one negated variable (resp. at most one unnegated variable) in each clause;
- f is *implicative* iff it is logically equivalent to a CNF formula whose every clause is either unary or implicative;
- f is *1-unary* iff it is logically equivalent to a CNF formula consisting of a single unary clause.
- f is *2cnf* iff it is logically equivalent to a CNF formula where each clause contains at most two literals.
- f is *2-monotone* iff it is logically equivalent to a formula of the form

  $$(x_{i_1} \wedge \ldots \wedge x_{i_p}) \vee (\neg x_{j_1} \wedge \ldots \wedge \neg x_{j_q})$$

  where $p, q \geq 0$.
- f is *affine* iff it is logically equivalent to a conjunction of linear equalitites over $GF(2)$; that is, if f is logically equivalent to a system of linear equations of the forms $x_1 \oplus \ldots \oplus x_n = 0$ and $x_1 \oplus \ldots \oplus x_n = 1$, where $\oplus$ denotes the addition operator in $GF(2)$.
- f is *width-2 affine* iff it is logically equivalent to a conjunction of linear equalitites over $GF(2)$ with at most two variables per equality constraint.

The above definition extend to constraint families naturally. For instance, a constraint family $\mathcal{F}$ is 0-valid iff *every* constraint $f \in \mathcal{F}$ is 0-valid.

We can now state the main results of this article.

Theorem Let $\mathcal{F}$ be a constraint family. Min AW CSP($\mathcal{F}$) and Max AW CSP($\mathcal{F}$) are polynomial-time problems iff $\mathcal{F}$ is 1-unary. Min AW Ones($\mathcal{F}$) and Max AW Ones($\mathcal{F}$) are polynomial iff $\mathcal{F}$ is implicative or width-2 affine. Otherwise, the problems are NP-equivalent.

The proof for Min/Max AW CSP can be found in Section 3 and the proof for Min/Max AW Ones in Sections 4 and 5.

3 Dichotomy Theorem for Min/Max AW CSP

This section is devoted to the proof of the dichotomy result for Min/Max AW CSP($\mathcal{F}$). We begin by showing that Min/Max AW CSP($\mathcal{F}$) can be solved in polynomial time if $\mathcal{F}$ is 1-unary.

Lemma 3.1 Let $\mathcal{F}$ be a finite set of 1-unary constraints. Then Min AW CSP($\mathcal{F}$) and Max AW CSP($\mathcal{F}$) can be solved in polynomial time.

Proof: Let $\Gamma = \{\gamma_1, \ldots, \gamma_m\}$ be an instance of Max AW CSP($\mathcal{F}$) (where γ_i has weight w_i) over the variable set $\{x_1, \ldots, x_n\}$. Since $\mathcal{F}$ is 1-unary, each constraint application $f(x_{i_1}, \ldots, x_{i_p})$ in Γ is logically equivalent to either x_j or $\neg x_j$ for some $j \in \{i_1, \ldots, i_p\}$. Given a constraint application γ_i, let $lit(\gamma_i)$ denote this literal. Now, define K as follows:

$$K = \sum_{i=1}^{n} \max(\sum\{w_j \mid lit(\gamma_j) = x_i\},\ \sum\{w_j \mid lit(\gamma_j) = \neg x_i\}).$$

It is obvious that $\sum\{w_j \mid lit(\gamma_j) = x_i\}$ is the total weight resulting from assigning 1 to x_i and $\sum\{w_j \mid lit(\gamma_j) = \neg x_i\}$ is the weight of assigning 0 to x_i. Since the constraints are 1-unary, we can greedily choose the optimal assignment for each variable so $opt(\Gamma) = K$. Since K can easily be computed in polynomial time, Max AW CSP($\mathcal{F}$) is a polynomial-time solvable problem.

Assume instead that Γ is an instance of the Min AW CSP($\mathcal{F}$) problem. Then we compute L as follows:

$$L = \sum_{i=1}^{n} \min(\sum\{w_j \mid lit(\gamma_j) = x_i\},\ \sum\{w_j \mid lit(\gamma_j) = \neg x_i\})$$

By noting that $\sum\{w_j \mid lit(\gamma_j) = x_i\}$ is the total weight resulting from assigning 0 to x_i and $\sum\{w_j \mid lit(\gamma_j) = \neg x_i\}$ is the weight of assigning 1 to x_i,

$L = opt(\Gamma)$. The polynomiality of MIN AW CSP($\mathcal{F}$) follows immediately.
□

We continue by showing that if $\mathcal{F}$ is not 1-unary, then MAX/MIN AW CSP($\mathcal{F}$) is NP-hard. The problem MAX CSP (MIN CSP) is the MAX AW CSP (MIN AW CSP) problem restricted to non-negative weights and the following theorem was proved by Khanna *et al.* [12, 13].

Theorem 3.2 Let $\mathcal{F}$ be a constraint family. MIN/MAX CSP($\mathcal{F}$) is a polynomial-time problem iff $\mathcal{F}$ is 0-valid, 1-valid or 2-monotone. Otherwise, the problem is NP-hard.

The previous theorem holds even if all weights are equal to 1 so we will use this simplified formulation of the problems in the following.

Given a constraint f, let $\neg f$ denote the converse of f, *i.e.*, $f(x_1, \ldots, x_k) = 1$ iff $\neg f(x_1, \ldots, x_k) = 0$ for every choice of $x_1, \ldots, x_k \in \{0, 1\}$.

Lemma 3.3 If f is a 2-monotone constraint which is not 1-unary, then $\neg f$ is not a 2-monotone constraint.

Proof: First observe that a 2-monotone constraint is always either 0-valid or 1-valid or both. We begin by assuming that f is simultaneously 0-valid and 1-valid. In this case, $\neg f$ is neither 0-valid nor 1-valid which implies that $\neg f$ is not 2-monotone.

Assume instead that f is 1-valid but not 0-valid. By the definition of 2-monotone constraints and the fact that f is not 1-unary, f is logically equivalent to a formula of the form

$$x_{i_1} \wedge \ldots \wedge x_{i_p}$$

where $p \geq 2$. This implies that $\neg f$ is logically equivalent to a formula of the form

$$\neg x_{i_1} \vee \ldots \vee \neg x_{i_p}$$

and $\neg f$ is not 2-monotone. The case when f is 0-valid but not 1-valid is analogous. □

Lemma 3.4 Let f be a constraint which is not 1-unary. Then MIN/MAX CSP($\{f, \neg f\}$) are NP-hard problems.

Proof: We show the lemma for MIN CSP; the other case is symmetric. Assume to the contrary that MIN CSP($\{f, \neg f\}$) is a problem which can be solved in polynomial time. It follows from Theorem 3.2 that $\{f, \neg f\}$ must be 0-valid or 1-valid or 2-monotone. By Lemma 3.3, $\{f, \neg f\}$ must be either 0-valid or 1-valid. If f is both 0- and 1-valid then $\neg f$ is neither 0-valid nor 1-valid which contradicts the polynomiality of MIN CSP($\{f, \neg f\}$). Similarly, if f is 0-valid but not 1-valid then $\neg f$ is 1-valid but not 0-valid which once again leads to a contradiction. The case when f is 1-valid but not 0-valid is analogous.

Finally, since MIN CSP($\{f, \neg f\}$) is not a polynomial-time problem, it is NP-hard by Theorem 3.2. □

Lemma 3.5 If f is a constraint which is not 1-unary, then MAX AW CSP($\{f\}$) and MIN AW CSP($\{f\}$) are NP-hard.

Proof: We show the theorem for MAX AW CSP; the other case is similar. Polynomial-time reduction from MAX CSP($\{f, \neg f\}$) which is NP-hard by Lemma 3.4.

Let Γ be an arbitrary instance of MAX CSP($\{f, \neg f\}$) containing n_f applications of the constraint f and $n_{\neg f}$ applications of $\neg f$. Construct an instance Θ of MAX AW CSP$\{f\}$ as follows: for each application $f(x_{i_1}, \ldots, x_{i_p})$ introduce an application $f(x_{i_1}, \ldots, x_{i_p})$ with weight 1 and for each application $\neg f(x_{j_1}, \ldots, x_{j_q})$ introduce an application $f(x_{j_1}, \ldots, x_{j_q})$ with weight -1. We show that $opt(\Gamma) = opt(\Theta) + n_{\neg f}$ and, thus, the NP-hardness of MAX AW CSP($\{f\}$).

We begin by showing that $opt(\Gamma) \leq opt(\Theta) + n_{\neg f}$. We assume that Γ and Θ are over the variable set $X = \{x_1, \ldots, x_n\}$ and that $a = (a_1, \ldots, a_n)$ is an optimal solution to Γ. Assume that a satisfies A constraints of type f and B constraints of type $\neg f$ in Γ, *i.e.*, $opt(\Gamma) = A + B$. This implies that a does not satisfy $n_f - A$ constraints of type f and $n_{\neg f} - B$ of type $\neg f$. If we apply a to Θ, the following holds:

- a satisfies A constraints of type f with weight 1;
- a does not satisfy $n_f - A$ constraints of type f with weight 1;
- a satisfies $n_{\neg f} - B$ constraints of type f with weight -1;
- a does not satisfy B constraints of type f with weight -1.

Hence, $opt(\Theta) \geq A + (-1) \cdot (n_{\neg f} - B) = A + B - n_{\neg f}$ and $opt(\Theta) + n_{\neg f} \geq A + B = opt(\Gamma)$. An analogous proof shows that $opt(\Theta) + n_{\neg f} \leq opt(\Gamma)$ and the lemma follows. □

Theorem 3.6 Let $\mathcal{F}$ be a constraint family. MIN/MAX AW CSP($\mathcal{F}$) is a polynomial-time solvable problem iff $\mathcal{F}$ is 1-unary and, otherwise, NP-equivalent.

Proof: If $\mathcal{F}$ is 1-unary, then MIN/MAX AW CSP($\mathcal{F}$) is a polynomial-time problem by Lemma 3.1. Otherwise, the problem is NP-hard (and thus NP-equivalent) by Lemma 3.5. $\square$

4 Polynomial Subclasses of MAX AW ONES

In this section, we will show that MAX AW ONES($\mathcal{F}$) is a polynomial-time problem when $\mathcal{F}$ is implicative or width-2 affine. To begin with, we need some standard definitions and theorems from the theory of linear optimization. The proofs can be found in standard textbooks such as [14] or [16].

Theorem 4.1 (Khachiyan [11]) Let A be an $m \times n$ integer matrix and let b, c be m- and n-vectors of integers, respectively. Finding a vector $x \in \mathbb{R}^n$ that maximizes $\sum_{i=1}^{n} c_i x_i$ under the constraint $Ax \geq b$ is a polynomial-time problem.

Definition 4.2 A square, integer matrix B is called *unimodular* if its determinant $\det(B) = \pm 1$. An integer matrix A is called *totally unimodular* (TUM) iff every square, nonsingular submatrix of A is unimodular.

Let $\mathbf{0}$ stand for all-zero vectors and $\mathbf{1}$ stand for all-one vectors of appropriate dimensions.

Theorem 4.3 (Hoffman and Kruskal [8]) Let A be an $m \times n$ TUM matrix. Let b be an m-vector and c an n-vector of integers. The following two problems have the same optimal value.

1. $\max \sum_{i=1}^{n} c_i x_i$ subject to $Ax \geq b$, $x \in \{0,1\}^n$

2. $\max \sum_{i=1}^{n} c_i x_i$ subject to $Ax \geq b$, $\mathbf{0} \leq x \leq \mathbf{1}$.

Similarly, the next two problems have the same optimal value.

1. $\max \sum_{i=1}^{n} c_i x_i$ subject to $Ax = b$, $x \in \{0,1\}^n$

2. $\max \sum_{i=1}^{n} c_i x_i$ subject to $Ax = b$, $\mathbf{0} \leq x \leq \mathbf{1}$.

We say that a matrix is *dyadic* iff each row contains at most two non-zero entries.

Theorem 4.4 (Heller and Tompkins [6]) A dyadic $(0, \pm 1)$-matrix is TUM if the columns of A can be partitioned into two sets I_1 and I_2 such that:

1. If a row has two entries of the same sign, their columns are in different sets; and

2. If a row has two entries of different signs, their columns are in the same set.

Theorem 4.4 is often stated in a form where "rows" are replaced by "columns" and vice versa. However, a matrix is TUM iff its transpose is TUM so these two formulations are equivalent.

Corollary 4.5 A $(0, \pm 1)$-matrix A is TUM if each row contains at most one positive non-zero entry and at most one negative non-zero entry.

We are now ready to show that Max AW Ones over implicative constraints can be solved in polynomial time.

Theorem 4.6 Let $\mathcal{F}$ be a finite set of implicative constraints. Then Max AW Ones($\mathcal{F}$) is a polynomial-time problem.

Proof: Implicative constraints are logically equivalent to a conjunction of clauses of the following possible forms: (x), $(\neg x)$ or $(x \vee \neg y)$. For a given implicative constraint application $f(x_1, \ldots, x_k)$ logically equivalent to a set C of clauses, let the relational form R be defined as

$$\begin{aligned} R \;=\; & \{x_i \geq 1, -x_i \geq -1 \mid (x_i) \in C\} \cup \\ & \{x_i \geq 0, -x_i \geq 0 \mid (\neg x_i) \in C\} \cup \\ & \{x_i - x_j \geq 0 \mid (x_i \vee \neg x_j) \in C\}. \end{aligned}$$

Obviously, a boolean vector $a = (a_1, \ldots, a_k)$ satisfies $f(a) = 1$ iff the relations in R are satisfied by the assignment $x_i = a_i$, $1 \leq i \leq k$. Now, assume we are given a Max AW Ones($\mathcal{F}$) problem

$$\Theta = \max \sum_{i=1}^{n} w_i x_i \text{ over } \mathcal{C}.$$

By transforming each constraint application to its corresponding relational form, we can reformulate Θ as a Zolp problem Γ:

$$\max \sum_{i=1}^{n} w_i x_i \text{ over } Ax \geq b,\ x \in \{0,1\}^n$$

where A is a $(0, \pm 1)$-matrix containing at most one 1 and at most one -1 per row. Clearly, this is a polynomial-time transformation. By Corollary 4.5, A is TUM and, by Theorem 4.3, Γ has the same optimal value as

$$\max \sum_{i=1}^{n} w_i x_i \text{ over } Ax \geq b,\ \mathbf{0} \leq x \leq \mathbf{1}.$$

By Theorem 4.1, the optimal value of this problem can be computed in polynomial time and the theorem follows. □

Next, we turn our attention to width-2 affine constraints. Before we can prove that Max AW Ones is polynomial for this class of constraints, we need two auxiliary lemmata.

Lemma 4.7 Let A be a dyadic $m \times n$ $(0,1)$-matrix. Deciding whether there exists a vector $x \in \{0,1\}^n$ such that $Ax = \mathbf{1}$ can be decided in polynomial time.

Proof: Follows from a straightforward reduction to 2SAT. For details, see [7]. □

Lemma 4.8 Let A be a dyadic $m \times n$ $(0,1)$-matrix. If there exists a vector $x \in \{0,1\}^n$ such that $Ax = \mathbf{1}$ then A is TUM.

Proof: Let $x \in \{0,1\}^n$ be such that $Ax = \mathbf{1}$. Let $I_0 = \{i \mid x_i = 0\}$ and $I_1 = \{i \mid x_i = 1\}$. If there exists a row in A with two nonzero entries then this row corresponds to an equation of the form $x_i + x_j = 1$ since A is a $(0,1)$-matrix. Consequently, exactly one of x_i, x_j equals 0 and one equals 1 so i and j are in different sets. We can thus apply Theorem 4.4 and conclude that A is TUM. □

Theorem 4.9 Let $\mathcal{F}$ be a set of width-2 affine constraints. Then Max AW Ones($\mathcal{F}$) is a polynomial-time problem.

Proof: A width-2 affine constraint is logically equivalent to some conjunction of equations over $GF(2)$ of the following possible forms: $(x = 0)$, $(x = 1)$, $(x \oplus y = 0)$ or $(x \oplus y = 1)$. For a given width-2 affine constraint $f(x_1, \ldots, x_k)$ defined by a set E of equations, let the relational form R be defined as follows: Let $V = \{y\} \cup \{z_{ij} \mid (x_i \oplus x_j = 0) \in E\}$ be a set of fresh variables and

$$\begin{aligned}
R \;=\; & \{y = 1\} \cup \\
& \{x_i + y = 1 \mid (x_i = 0) \in E\} \cup \\
& \{x_i = 1 \mid (x_i = 1) \in E\} \cup \\
& \{x_i + z_{ij} = 1, x_j + z_{ij} = 1 \mid (x_i \oplus x_j = 0) \in E\} \cup \\
& \{x_i + x_j = 1 \mid (x_i \oplus x_j = 1) \in E\}.
\end{aligned}$$

It is easy to see that a vector $a = (a_1, \ldots, a_k)$ satisfies $f(a) = 1$ iff there exists an assignment of the variables in V such that a together with this assignment satisfy the relations in R. Now, assume we are given a MAX AW ONES($\mathcal{F}$) problem

$$\Theta = \max \sum_{i=1}^{n} w_i x_i \text{ over } \mathcal{C}.$$

Transform the constraint applications in Θ to their corresponding relational form and assume that l fresh variables $v_1, \ldots, v_l$ are introduced. As is easily seen, this transformation can be carried out in polynomial time. Now, we can reformulate Θ as a ZOLP problem Γ of the following appearance:

$$\max \sum_{i=1}^{n} w_i x_i \text{ over } A \begin{pmatrix} x \\ v \end{pmatrix} = \mathbf{1}, \ \begin{pmatrix} x \\ v \end{pmatrix} \in \{0,1\}^{n+l}$$

where A is a dyadic $(0,1)$-matrix. By Lemma 4.7, we can decide whether there exists a vector $\begin{pmatrix} x \\ v \end{pmatrix} \in \{0,1\}^{n+l}$ such that $A \begin{pmatrix} x \\ v \end{pmatrix} = \mathbf{1}$ in polynomial time. Furthermore, either the set of equations has no solution or A is TUM, by Lemma 4.8. In the former case, Θ has no solution and we are done. Otherwise, A is TUM and the optimal value can be determined in polynomial time by Theorems 4.3 and 4.1. $\square$

5 Dichotomy Theorem for MAX AW ONES

The goal of this section is to show that MAX AW ONES($\mathcal{F}$) is polynomial only if $\mathcal{F}$ is implicative or width-2 affine. By combining this result with the results of the previous section, we have proved the dichotomy theorem for MAX AW ONES.

To begin with, we need some results by Khanna *et al.* [12, 13]. Let MAX ONES denote the MAX AW ONES problem restricted to non-negative weights and define MIN ONES analogously. We have the following result.

Theorem 5.1 (Khanna *et al.* [13]) Let $\mathcal{F}$ be a constraint family. MAX ONES($\mathcal{F}$) is polynomial iff $\mathcal{F}$ is 1-valid or weakly positive or width-2 affine and, otherwise, NP-equivalent. MIN ONES($\mathcal{F}$) is polynomial iff $\mathcal{F}$ is 0-valid or weakly negative or width-2 affine and, otherwise, NP-equivalent.

The previous theorem holds even if all weights are equal to 1. To simplify the forthcoming proofs, we use the unweighted versions of MAX/MIN ONES in the sequel.

Define the constraints T and F such that $T(x) = x$ and $F(x) = 1 - x$.

Lemma 5.2 Let $\mathcal{F}$ be a constraint family such that $T \in \mathcal{F}$. If Min Ones($\mathcal{F}$) is NP-hard then Max AW Ones($\mathcal{F} - \{T\}$) is NP-hard.

Proof: By Lemma 2.1, Max AW Ones and Min AW Ones are polynomially interreducible. For the sake of simplicity, we will consider the Min AW Ones problem in this proof. We will prove the lemma by constructing a polynomial-time Turing reduction from Min Ones($\mathcal{F}$) to Min AW Ones($\mathcal{F} - \{T\}$).

Let Γ be an arbitrary instance of Min Ones($\mathcal{F}$). Assume Γ is over the variable set $X = \{x_1, \ldots, x_n\}$. Then, Γ has the following form:

$$\min \sum_{i=1}^{n} x_i \text{ over } \mathcal{C} \cup \mathcal{U}$$

where $\mathcal{C}$ contains the applications of the constraints $\mathcal{F} - \{T\}$ and $\mathcal{U}$ contains the applications of the constraint T. Let $X_T = \{i \mid T(x_i) \in \mathcal{U}\}$, $M = 2n$ and $L = -M \cdot |X_T|$.

Now, consider the following instance Θ of Min AW Ones($\mathcal{F} - \{T\}$) over the variable set X:

$$\min \sum_{i=1}^{n} x_i - \sum_{i \in X_T} M \cdot x_i \text{ over } \mathcal{C}.$$

We begin by showing that $L \leq opt(\Theta) \leq L + n$ iff $opt(\Gamma) \neq \bot$.

Assume $opt(\Gamma) \neq \bot$ and let $a = (a_1, \ldots, a_n)$ be an optimal solution to Γ. Naturally, a is a feasible solution to Θ and by noting that $a_i = 1$ if $i \in X_T$, it follows that

$$L \leq \sum_{i=1}^{n} a_i - \sum_{i \in X_T} M \cdot a_i \leq L + n$$

and $L \leq opt(\Theta) \leq L + n$.

Conversely, assume that $L \leq opt(\Theta) \leq L + n$ and let $b = (b_1, \ldots, b_n)$ be an optimal solution to Θ. Since $L \leq opt(\Theta) \leq L + n$, $i \in X_T$ implies that $b_i = 1$ by the choice of M. Thus, b satisfies all constraints in $\mathcal{C} \cup \mathcal{U}$ and $opt(\Gamma) \neq \bot$.

Finally, we show that if $L \leq opt(\Theta) \leq L + n$ then $opt(\Theta) + M \cdot |X_T| = opt(\Gamma)$. As we have already proved, any optimal solution of Θ is a feasible solution of Γ in this case. Furthermore, all variables with indices in X_T necessarily have the value 1. By substituting 1 for the variables with indices in X_T, Θ and Γ collapse to the following two problems:

$$\Theta' : \max \sum_{1 \leq i \leq n} x_i - M \cdot |X_T| \text{ over } \mathcal{C}$$

$$\Gamma' : \max \sum_{1 \le i \le n} x_i \text{ over } \mathcal{C}$$

and $opt(\Theta') + M \cdot |X_T| = opt(\Gamma')$. Consider the following reduction.

1. Given an instance Γ of MIN ONES($\mathcal{F}$), transform it to an instance Θ of MIN AW ONES($\mathcal{F} - \{T\}$) as above.

2. Solve Θ by using an oracle for MIN AW ONES($\mathcal{F} - \{T\}$).

3. If $opt(\Theta) = \bot$ or $opt(\Theta)$ is not in the interval $[L, \ldots, L+n]$ then report that there is no feasible solution to Γ.

4. Otherwise, report that the optimal value of Γ is $opt(\Theta) + M \cdot |X_T|$.

Clearly, this is a polynomial-time Turing reduction from MIN ONES($\mathcal{F}$) to MIN AW ONES($\mathcal{F} - \{T\}$) which concludes the proof. □

The proof of the next lemma is omitted since it is analogous to the proof of Lemma 5.2.

Lemma 5.3 Let $\mathcal{F}$ be a constraint family such that $F \in \mathcal{F}$. If MAX ONES($\mathcal{F}$) is NP-hard then MAX AW ONES($\mathcal{F} - \{F\}$) is NP-hard.

Lemma 5.4 Let g be a constraint which is not width-2 affine. Then the following hold:

1. If g is not weakly positive then MAX AW ONES($\{g\}$) is NP-hard.

2. If g is not weakly negative then MAX AW ONES($\{g\}$) is NP-hard.

Proof: Assume that g is not weakly positive and consider the following set of constraints: $\mathcal{C} = \{g, F\}$. Clearly, F is not 1-valid, g is not weakly positive and g is not width-2 affine. Thus, by Theorem 5.1, MAX ONES($\mathcal{C}$) is NP-hard and, by Lemma 5.3, MAX AW ONES($\{g\}$) is NP-hard. The case when g is not weakly negative is similar but uses the reduction of Lemma 5.2 instead. □

Before the proof of the dichotomy theorem, we need a technical lemma.

Lemma 5.5 (Creignou and Hermann [2]) A constraint f is both weakly positive and weakly negative iff f is implicative.

Theorem 5.6 Let $\mathcal{F}$ be a constraint family. MAX AW ONES($\mathcal{F}$) is a polynomial-time problem iff $\mathcal{F}$ is implicative or width-2 affine. Otherwise, MAX AW ONES($\mathcal{F}$) is NP-equivalent.

Proof: *if:* Follows from Theorems 4.6 and 4.9.

only-if: Assume that $\mathcal{F}$ is not width-2 affine nor implicative. Then $\mathcal{F}$ is either not weakly positive or not weakly negative by Lemma 5.5 so, by Lemma 5.4, Max AW Ones($\mathcal{F}$) is NP-hard. Since Max AW Ones is NP-easy, Max AW Ones($\mathcal{F}$) is NP-equivalent. □

6 Recognition of Tractable Classes

We have seen that 1-unary, implicative and width-2 affine constraints have favourable computational properties. We shall provide polynomial-time algorithms for identifying such constraints in this section.

Let f be a constraint of arity k. We denote by $\mathrm{Sat}(f)$ the set of assignments which satisfy f. Thus, $\mathrm{Sat}(f)$ is a set of vectors in $\{0,1\}^k$ and given such a vector s, let $s(i)$ denote its i:th component. We define three operations on such vectors.

1. $s = s_1 \oplus s_2$ is defined by $s(i) = 1$ iff $s_1(i) \neq s_2(i)$ and $s(i) = 0$ otherwise;
2. $s = s_1 \cap s_2$ is defined by $s(i) = 1$ iff $s_1(i) = s_2(i) = 1$ and $s(i) = 0$ otherwise;
3. $s = s_1 \cup s_2$ is defined by $s(i) = 0$ iff $s_1(i) = s_2(i) = 0$ and $s(i) = 1$ otherwise.

Proposition 6.1 Recognizing 1-unary and implicative constraints can be done in polynomial time. □

Proof: If f is a 1-unary constraint of arity k, then there exists a position $1 \leq i \leq k$ and a constant $c \in \{0,1\}$ such that $(a_1, \ldots, a_k) \in \mathrm{Sat}(f)$ iff $a_i = c$. The existence of i and c can easily be checked in polynomial time given the truth table of f.

To verify that f is implicative, we only have to note that a constraint f is implicative iff for all $s, t \in \mathrm{Sat}(f)$, $s \cap t \in \mathrm{Sat}(f)$ and $s \cup t \in \mathrm{Sat}(f)$ (for a proof, see [2]). This can be verified in polynomial time and the proposition follows. □

Proposition 6.2 It is a polynomial-time task to decide whether a constraint f is width-2 affine or not. □

Proof: Schaefer [15] has shown that f is affine iff

for all $s_1, s_2, s_3 \in \mathrm{Sat}(f)$, $s_1 \oplus s_2 \oplus s_3 \in \mathrm{Sat}(f)$;

and f is 2cnf iff

$$\text{for all } s_1, s_2, s_3 \in \text{Sat}(f),\ (s_1 \cup s_2) \cap (s_2 \cup s_3) \cap (s_3 \cup s_1) \in \text{Sat}(f).$$

Obviously, these two conditions can be checked in polynomial time. To prove the proposition, we will show that f is width-2 affine iff f is affine and 2cnf.

The only-if direction is trivial since we can express every width-2 affine constraint as a 2cnf formula (for instance, $x \oplus y = 1$ can be expressed as $(x \vee y) \wedge (\neg x \vee \neg y)$).

To show the other direction, we assume that f is 2cnf but not width-2 affine and show that f cannot be affine. We begin the proof by exploiting an idea by Khanna *et al.* [12]. Consider the application $f(x_1, \ldots, x_k)$. Since f is 2cnf, it can be viewed as a conjunction of implications of the following types: $x_i \to x_j$, $x_i \to \neg x_j$ and $\neg x_i \to \neg x_j$. Construct a directed graph G_f on $2k$ vertices (one for each literal x_i or $\neg x_i$) which has a directed edge from literal l_i to l_j iff this is a constraint imposed by f. Now, the graph G_f must have vertices l_i and l_j such that there is a directed path from l_i to l_j but not the other way round. Note that if this is not the case, then the graph is undirected and corresponds to equality and inequality constraints which can be expressed by equations of the form $x_i \oplus x_j = 0$ and $x_i \oplus x_j = 1$. This implies that f is width-2 affine which contradicts our initial assumptions.

It should be clear that whenever we assign values to x_i and x_j which satisfy the constraint imposed by $l_i \to l_j$, we can find values to the other variables such that $f(x_1, \ldots, x_k)$ is satisfied. Likewise, if we assign values that do not satisfy $l_i \to l_j$ then we cannot assign values to the other variables such that the constraint becomes satisfied.

Without loss of generality, we can assume that literal l_i corresponds to variable x_1 and l_j to x_2. We carry out the rest of the proof for the case when both l_i and l_j are unnegated literals; the other three cases are similar.

Now, we know that there does not exist any $s \in \text{Sat}(f)$ such that $s(1) = 1$ and $s(2) = 0$ but we know that there exists $s_1, s_2, s_3 \in \text{Sat}(f)$ such that

1. $s_1(1) = 0$, $s_1(2) = 0$; and
2. $s_2(1) = 1$, $s_2(2) = 0$; and
3. $s_3(1) = 1$, $s_3(2) = 1$.

As was observed earlier, f is affine iff for arbitrary $t, u, v \in \text{Sat}(f)$, $t \oplus u \oplus v \in \text{Sat}(f)$. However,

$$\begin{aligned}
& s_1 \oplus s_2 \oplus s_3 \\
&= (0, 1, s_1(3) \oplus s_2(3), \ldots s_1(k) \oplus s_2(k)) \oplus (1, 1, s_3(3), \ldots, s_3(k)) \\
&= (1, 0, s_1(3) \oplus s_2(3) \oplus s_3(3), \ldots, s_1(k) \oplus s_2(k) \oplus s_3(k)) \notin \text{Sat}(f)
\end{aligned}$$

so f is not affine which concludes the proof. $\square$

Acknowledgements

The author thanks Thomas Drakengren and the anonymous reviewers for providing helpful comments and suggestions.

References

[1] N. Creignou, A dichotomy theorem for maximum generalized satisfiability problems, *Journal of Computer and System Sciences* **51**(3) (1995) 511–522.

[2] N. Creignou and M. Hermann, Complexity of generalized satisfiability counting problems, *Information and Computation* **125** (1996) 1–12.

[3] S. Fortune, J. Hopcroft, and J. Wyllie, The directed subgraph homeomorphism problem, *Theoretical Computer Science* **10**(2) (1980) 111–121.

[4] D. Gusfield and L. Pitt, A bounded approximation for the minimum cost 2-Sat problem, *Algorithmica* **8** (1992) 103–117.

[5] P. Hell and N. Nešetřil, On the complexity of H-coloring, *Journal of Combinatorial Theory, ser. B* **48** (1990) 92–110.

[6] I. Heller and C. B. Tompkins, An extension of a theorem of Dantzig, in: H. W. Kuhn and A. W. Tucker, eds., *Linear Inequalities and Related Systems*, (Princeton University Press, New York, 1956) 247–252.

[7] D. Hochbaum, N. Megiddo, J. Naor, and A. Tamir, Tight bounds and 2-approximation algorithms for integer programs with two variables per inequality, *Mathematical Programming* **62** (1993) 69–84.

[8] A. J. Hoffman and J. B. Kruskal, Integral boundary points of convex polyhedra, in: H. W. Kuhn and A. W. Tucker, eds., *Linear Inequalities and Related Systems*, (Princeton University Press, New York, 1956) 223–246.

[9] V. Kann, On the approximability of NP-complete optimization problems, Ph.D. Thesis, Royal Institute of Technology, Stockholm, 1992.

[10] V. Kann, Polynomially bounded minimization problems that are hard to approximate, *Nordic Journal of Computing* **1** (1994) 317–331.

[11] L. G. Khachiyan, A polynomial algorithm in linear programming, *Soviet Mathematics Doklady* **20** (1979) 191–194.

[12] S. Khanna, M. Sudan, and L. Trevisan, Constraint satisfaction: The approximability of minimization problems, in: *Proceedings of the 12th Annual IEEE Conference on Computational Complexity* (1997) 282–296.

[13] S. Khanna, M. Sudan, and D. P. Williamson, A complete classification of the approximability of maximization problems derived from boolean constraint satisfaction, in: *Proceedings of the 29th ACM Symposium on Theory of Computing* (1997) 11–20.

[14] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity* (Prentice-Hall, New Jersey, 1982).

[15] T. J. Schaefer., The complexity of satisfiability problems, in: *Proceedings of the 10th ACM Symposium on Theory of Computing* (1978) 216–226.

[16] A. Schrijver, *Theory of Linear and Integer Programming*, (John Wiley & Sons, 1986).