



## INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:  
<http://pubsonline.informs.org>

### On Solving MAX-SAT Using Sum of Squares

Lennart Sinjorgo, Renata Sotirov

To cite this article:

Lennart Sinjorgo, Renata Sotirov (2024) On Solving MAX-SAT Using Sum of Squares. *INFORMS Journal on Computing* 36(2):417-433. <https://doi.org/10.1287/ijoc.2023.0036>

This work is licensed under a Creative Commons Attribution 4.0 International License. You are free to copy, distribute, transmit and adapt this work, but you must attribute this work as “*INFORMS Journal on Computing*.” Copyright © 2023 The Author(s). <https://doi.org/10.1287/ijoc.2023.0036>, used under a Creative Commons Attribution License: <https://creativecommons.org/licenses/by/4.0/>.”

Copyright © 2023 The Author(s)

Please scroll down for article—it is on subsequent pages



With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes. For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

# On Solving MAX-SAT Using Sum of Squares

Lennart Sinjorgo,<sup>a,\*</sup> Renata Sotirov<sup>a</sup><sup>a</sup>CentER, Department of Econometrics and Operations Research, Tilburg University, 5037 AB Tilburg, Netherlands

\*Corresponding author

Contact: l.m.sinjorgo@tilburguniversity.edu,  <https://orcid.org/0000-0003-0516-6360> (LS); r.sotirov@tilburguniversity.edu, <https://orcid.org/0000-0002-3298-7255> (RS)

Received: February 13, 2023

Revised: July 18, 2023


Accepted: August 7, 2023

Published Online in Articles in Advance:  
November 7, 2023<https://doi.org/10.1287/ijoc.2023.0036>

Copyright: © 2023 The Author(s)

**Abstract.** We consider semidefinite programming (SDP) approaches for solving the maximum satisfiability (MAX-SAT) problem and weighted partial MAX-SAT. It is widely known that SDP is well-suited to approximate (MAX-)2-SAT. Our work shows the potential of SDP also for other satisfiability problems by being competitive with some of the best solvers in the yearly MAX-SAT competition. Our solver combines sum of squares (SOS)-based SDP bounds and an efficient parser within a branch-and-bound scheme. On the theoretical side, we propose a family of semidefinite feasibility problems and show that a member of this family provides the rank-two guarantee. We also provide a parametric family of semidefinite relaxations for MAX-SAT and derive several properties of monomial bases used in the SOS approach. We connect two well-known SDP approaches for (MAX)-SAT in an elegant way. Moreover, we relate our SOS-SDP relaxations for partial MAX-SAT to the known SAT relaxations.

**History:** Accepted by Andrea Lodi, Area Editor for Design & Analysis of Algorithms – Discrete.

 **Open Access Statement:** This work is licensed under a Creative Commons Attribution 4.0 International License. You are free to copy, distribute, transmit and adapt this work, but you must attribute this work as "INFORMS Journal on Computing. Copyright © 2023 The Author(s). <https://doi.org/10.1287/ijoc.2023.0036>, used under a Creative Commons Attribution License: <https://creativecommons.org/licenses/by/4.0/>."

**Supplemental Material:** The online appendix is available at <https://doi.org/10.1287/ijoc.2023.0036>.

**Keywords:** SAT • MAX-SAT • weighted partial MAX-SAT • semidefinite programming • sum of squares • Peaceman–Rachford splitting method

## 1. Introduction

In this paper, we investigate semidefinite programming (SDP) approaches for the satisfiability (SAT) problem, maximum satisfiability (MAX-SAT) problem, and their variants. Given a logical proposition built from a conjunction of clauses, SAT asks whether there exists a truth assignment to the variables such that all clauses are satisfied. The optimization variant of SAT, known as MAX-SAT, is to determine a truth assignment that satisfies the largest number of clauses.

SAT is a central problem in mathematical logic and computer science and finds various applications, including software or hardware verification (Marques-Silva and Sakallah 2000) and planning in artificial intelligence (Kautz and Selman 1999). Cook (1971) proves that SAT is  $\mathcal{NP}$ -complete. SAT is the first problem proven to be  $\mathcal{NP}$ -complete, which implies that any problem contained in the complexity class  $\mathcal{NP}$  can be efficiently recast as a SAT instance. Thus, algorithms for SAT can also solve a wide variety of other problems, such as timetabling (Asín Achá and Nieuwenhuis 2014, Gattermann et al. 2016) and product line engineering (Mendonca et al. 2009).

SDP approaches to SAT are proposed first by de Klerk et al. (2000) and later extended by Anjos (2004a, b; 2005; 2006; 2007). Goemans and Williamson (1995) are first to apply SDP to MAX-SAT. They show that, for a specific class of MAX-SAT instances known as MAX-2-SAT (in MAX- $k$ -SAT, each clause is a disjunction of at most  $k$  variables), MAX-SAT is equivalent to optimizing a multivariate quadratic polynomial, which is naturally well-suited for semidefinite relaxations. In the same paper, Goemans and Williamson (1995) propose a 0.878-approximation algorithm for MAX-2-SAT based on SDP. This result is later improved to 0.940 in Lewin et al. (2002). Further, Karloff and Zwick (1997) obtain an optimal  $7/8$  approximation algorithm for MAX-3-SAT, and Halperin and Zwick (2001) obtain a nearly optimal approximation algorithm for MAX-4-SAT. In van Maaren et al. (2008), the authors exploit sum of squares (SOS) optimization to compute bounds for MAX-SAT.

Despite the great success in designing approximation algorithms using SDP, most modern MAX-SAT solvers do not exploit SDP. A possible reason for this is the fact that medium- to large-size SDP problems are computationally challenging to solve. Interior point methods, the conventional approach for solving SDPs, struggle from large

memory requirements and prohibitive computation time per iteration already for medium-size SDPs. Recently, first order methods, such as the alternating direction method of multipliers (Gabay and Mercier 1976, Boyd et al. 2011) and the Peaceman–Rachford splitting method (PRSM) (Peaceman and Rachford 1955) show a great success in solving SDPs (see, e.g., Oliveira et al. 2018, de Meijer and Sotirov 2021, Graham et al. 2022). Motivated by these results, we design a MAX-SAT solver that incorporates SDP bounds and the PRSM within a branch-and-bound (B&B) scheme.

In particular, we further exploit the SOS approach from van Maaren et al. (2008) to derive SOS-based SDP relaxations that provide strong upper bounds to the optimal MAX-SAT solution. The derived SDP relaxations are strengthened SDP duals of the Goemans and Williamson (1995) MAX-SAT relaxation. The strength of the upper bounds and the required time to compute the relaxations depend on the chosen monomial basis. We experiment with different monomial bases and propose a class of bases that provide good trade-offs between these effects. Moreover, we derive several properties of monomial bases that are exploited in the design of our solver. We extend the SOS approach to weighted partial MAX-SAT, a variant of MAX-SAT in which clauses are divided into soft and hard clauses. Here, the goal is to maximize the weighted sum of soft clauses, satisfying all the hard clauses. We strengthen SDP bounds for weighted partial MAX-SAT using the SAT resolution rule. To the best of our knowledge, we are the first to exploit SDP for solving weighted partial MAX-SAT.

We show that the Peaceman–Rachford splitting method is well-suited for exploiting the structure of the SOS-based SDP relaxations. Therefore, we implement the PRSM to (approximately) solve large-scale SDP relaxations and obtain upper bounds for the (weighted partial) MAX-SAT. The resulting algorithm is very efficient; for example, it can compute upper bounds with matrix variables of order 1,800 in less than two minutes and for matrices of order 2,400 in less than four minutes. Our numerical results show that the upper bounds are strong, in particular, when larger monomial bases are used. We also exploit the output of the PRSM to efficiently compute lower bounds for the MAX-SAT problem.

We design an SOS-SDP-based MAX-SAT solver (named SOS-MS) that exploits SOS-based SDP relaxations and the PRSM. SOS-MS is one of the first SDP-based MAX-SAT solvers. The only alternative SDP-based solver is the MIXSAT algorithm (Wang and Zico Kolter 2019) that is designed to solve MAX-2-SAT instances. SOS-MS is able to solve (weighted partial) MAX- $k$ -SAT instances for  $k \leq 3$ . To solve a MAX-SAT instance, SOS-MS has to approximately solve multiple SDP subproblems. A crucial component of SOS-MS is, therefore, its ability to quickly construct the program parameters of the required SDPs, that is, the process of parsing. We design an efficient parsing method, which is also applicable to other problems and publicly available. Another efficient feature of our solver is warm starts. Namely, our solver uses the approximate PRSM solution at a node as warm starts for the corresponding children's node. We are able to solve a variety of MAX-SAT instances in a reasonable time, solving some instances faster than the best solvers in the 11th Evaluation of Max-SAT Solvers (MSE-2016). Moreover, we solve three previously unsolved MAX-3-SAT instances from the MSE-2016. To our knowledge, we are first to use SDP for solving weighted partial MAX-SAT. Our results provide new perspectives on solving MAX-SAT and all its variants by using SDPs.

Our paper provides various theoretical results. We propose a family of semidefinite feasibility problems and show that one member of this family provides the rank-two guarantee. That is, whenever the semidefinite relaxation admits a feasible matrix of rank two or less, the underlying SAT instance is satisfiable. This result relates to a similar rank-two guarantee result by Anjos (2004a). The rank value can be seen as a measure of the strength of the relaxation. We also provide a parametric family of semidefinite relaxations for the (weighted partial) MAX-SAT. This parameter can be finely tuned to determine the strength of the relaxation, and any such relaxation can easily be incorporated within SOS-MS. This allows the solver to be adapted per (class of) problem instances.

Further, we show how the SOS approach to MAX-SAT of van Maaren et al. (2008) and the, here generalized, moment relaxations of SAT from Anjos (2004a, b; 2005; 2006; 2007) are related. This is done by exploiting the duality theory of the moment and SOS approaches. Our result generalizes a result by van Maaren et al. (2008), who show the connection between the two approaches only for restricted cases. By exploiting duality theory, we also relate the SOS relaxations for the partial MAX-SAT problem to SAT relaxations from Anjos (2004a).

Finally, we investigate MAX-SAT resolution, a powerful technique used by many MAX-SAT solvers (Abramé and Habet 2014) in relation to the SDP approach to MAX-SAT. Standard MAX-SAT solvers use resolution to determine upper bounds on the MAX-SAT solution, whereas SOS-MS determines upper bounds through SDP. We show how resolution is related to the monomial basis. We also show how SAT resolution can be exploited for weighted partial MAX-SAT.

This paper is organized as follows. We provide notation and preliminaries in Section 1.1 and assumptions in Section 1.2. Section 2 provides an overview of the Goemans and Williamson (1995) approach to MAX-SAT. Section 3 first outlines previous SDP approaches to SAT and then generalizes them. Section 4 provides the details of the SOS theory applied to MAX-SAT. We also derive various properties of monomial bases in that section. Section 5

concerns the combination of MAX-SAT resolution and SOS. In Section 6, we show how two SDP approaches to (MAX-)SAT (i.e., Anjos 2004a, van Maaren et al. 2008) are connected. Section 7 introduces the PRSM for SOS.<sup>1</sup> We extend the SOS approach to weighted partial MAX-SAT and connect the resulting program to the relaxations in Anjos (2004a) in Section 8. Online Appendix E provides an overview and pseudocode of our solver SOS-MS. Online Appendix F presents numerical results that include SOS-SDP bounds and performance of SOS-MS. Concluding remarks are given in Section 9.

### 1.1. Preliminaries and Notation

For any  $n \in \mathbb{N}$ , we write  $[n] = \{1, \dots, n\}$ . We denote by  $\phi$  a propositional formula, in variables  $x_1$  up to  $x_n$  and assume that  $\phi$  is in conjunctive normal form (CNF). That is,  $\phi$  is given by a conjunction of  $m$  clauses,

$$\phi = \bigwedge_{j=1}^m C_j. \tag{1}$$

We mostly use  $n$  to refer to the number of variables and  $m$  to refer to the number of clauses. Each clause  $C_j$  is a disjunction of (possibly negated) variables. We define each clause  $C_j$  as a subset of  $[n]$ , indicating the variables appearing in  $C_j$ . Moreover, we define  $I_j^+ \subseteq C_j$  as the set of unnegated variables appearing in  $C_j$ . Similarly,  $I_j^- \subseteq C_j$  is defined as the set of negated variables appearing in  $C_j$ . Thus, the clause associated with  $C_j$  reads

$$\bigvee_{i \in I_j^+} x_i \vee \bigvee_{i \in I_j^-} \neg x_i. \tag{2}$$

We refer to both  $x_i$  and  $\neg x_i$  as literals. For example, the literal  $\neg x_i$  is true if  $x_i$  is false. We denote the length of a clause by  $\ell_j$ ; thus,  $\ell_j := |C_j|$ . We say that  $\phi$  constitutes a (MAX-)k-SAT instance if  $\max_{j \in [m]} \ell_j = k$ .

The SAT problem is to decide, given  $\phi$ , whether a satisfying truth assignment to the variables  $x_i, i \in [n]$  exists. The MAX-SAT problem is to find an assignment that satisfies the largest number of clauses. We associate to each clause a vector  $a_j \in \{0, \pm 1\}^n$ , having entries  $a_{j,i}$  according to

$$a_{j,i} = \begin{cases} -1, & \text{if } i \in I_j^-, \\ 0, & \text{if } i \notin I_j^+ \cup I_j^-, \\ 1, & \text{if } i \in I_j^+. \end{cases} \tag{3}$$

We write  $S^n$  for the set of symmetric  $n \times n$  matrices and  $S_+^n$  for the cone of symmetric positive semidefinite matrices of size  $n \times n$ . If the context is clear, the superscript  $n$  is omitted. We denote by  $\mathbf{1}_n \in \mathbb{R}^n$  and  $\mathbf{0}_n \in \mathbb{R}^n$  vectors of all ones and zeroes, respectively (subscripts are omitted when the context is clear). The identity matrix of order  $n$  is written as  $I_n$ . Matrix  $\mathbf{0}_{m \times n}$  denotes the zero matrix of  $m$  rows and  $n$  columns. For  $X, Y \in \mathcal{S}$ , we define the trace inner product as  $\langle X, Y \rangle := \text{Tr}(XY)$ . We write  $X \succeq Y$  if and only if  $X - Y \in S_+$ . The Frobenius norm of a symmetric matrix  $X$  is given by  $\|X\|_F = \sqrt{\langle X, X \rangle}$ . For  $X \in \mathcal{S}$  and  $\mathcal{M} \subseteq \mathcal{S}$ , we set

$$\mathcal{P}_{\mathcal{M}}(X) := \arg \min_{Z \in \mathcal{M}} \|Z - X\|_F, \tag{4}$$

as the projection of  $X$  onto  $\mathcal{M}$ . For  $X \in \mathbb{R}^{n \times n}$ ,  $\text{diag}(X) \in \mathbb{R}^n$  denotes the vector equal to the diagonal of  $X$ . For  $x \in \mathbb{R}^n$ , we write  $\|x\| := \sqrt{x^\top x}$  and

$$x^\alpha := \prod_{i \in \alpha} x_i, \tag{5}$$

for some  $\alpha \subseteq [n]$ . We evaluate the empty product as one. Note that (5) differs from the more common notation in SOS literature, in which the  $\alpha$  are considered as vectors in  $\mathbb{N}^n$ ; see, for example, Lasserre (2007).

### 1.2. Assumptions

In the rest of this work, we assume that all logical propositions  $\phi$  on  $n$  variables and  $m$  clauses satisfy the following three properties:

1. For all  $j \in [m]$ ,  $|C_j| \geq 2$ .
2. For all  $j \in [m]$ ,  $I_j^+ \cap I_j^- = \emptyset$ .
3. Each variable is contained in at least two clauses along with  $\phi$  being in CNF.

Property 1 can be assumed for SAT instances. If a SAT instance contains a clause  $C_j$  with  $|C_j| = 1$  (such a clause is known as a unit clause), the literal in  $C_j$  must be satisfied. The variable corresponding to this literal can, thus, be

given an appropriate truth value, and  $\phi$  can be reduced (such a reduction of  $\phi$  is referred to as unit resolution). For MAX-SAT instances, it is possible that an optimal truth assignment might leave unit clauses unsatisfied. We note, however, that the MAX-SAT benchmark instances we consider satisfy all these assumptions.

In Online Appendix B, we show that properties 2 and 3 can be assumed without loss of generality (w.l.o.g.).

## 2. MAX-SAT Formulation and Relaxation

We outline the approach of Goemans and Williamson (1995) for formulating MAX-SAT as a polynomial optimization problem. We also present their SDP relaxation for MAX-2-SAT.

Let  $x_1, x_2, \dots, x_n$  be the variables of the MAX-SAT instance given by a logical proposition  $\phi$ . We, thus, assume that  $\phi$  is given in conjunctive normal form, see (1), and contains  $m$  clauses. As customary in the SDP SAT literature, we associate +1 with `true` and -1 with `false`. An assignment of the  $x_i$  values in  $\{\pm 1\}$  is referred to as a truth assignment. As proposed by Goemans and Williamson (1995), we define a truth function  $v: \{\pm 1\}^n \rightarrow \{0, 1\}$  such that, given a logical proposition  $\phi'$  evaluated for some truth assignment,  $v(\phi') = 1$  if and only if  $\phi'$  is satisfied and zero otherwise. This property uniquely determines  $v$ ; that is,  $v(x_i) = \frac{1+x_i}{2}$  and  $v(-x_i) = \frac{1-x_i}{2}$ . And, in general, for a clause  $C_j \subseteq [n]$  of length  $\ell_j$ , we have

$$v(C_j) := 1 - \prod_{i \in I_j^+} v(-x_i) \prod_{i \in I_j^-} v(x_i) = 1 - 2^{-\ell_j} \left( \sum_{\gamma \subseteq C_j} (-1)^{|\gamma|} a_j^\gamma x^\gamma \right), \quad (6)$$

for  $a_j$  as in (3) and both  $a_j^\gamma$  and  $x^\gamma$  as in (5). The last equality in (6) follows from the product expansion of  $v(C_j)$  as shown in proposition 1 in Anjos (2006). In Goemans and Williamson (1995), an extra variable  $x_0 \in \{\pm 1\}$  is defined with the purpose of deciding the truth value; that is,  $\phi'$  is true if and only if  $v(\phi') = x_0$ . We set  $x_0 = 1$  without loss of generality for sake of clarity. The MAX-SAT problem given by  $\phi$  is to maximize the following polynomial:

$$v_\phi := \sum_{j \in [m]} v(C_j) = \sum_{\alpha \subseteq [n]} v_\phi^\alpha x^\alpha, \quad (7)$$

subject to  $x_i \in \{\pm 1\}$  for all  $i$  and for appropriate  $v_\phi^\alpha \in \mathbb{R}$  and  $x^\alpha$  as in (5). Observe that  $v_\phi$  is a  $k$ th degree polynomial if  $\phi$  represents a MAX- $k$ -SAT instance. A MAX-2-SAT instance, thus, corresponds to a quadratic polynomial and is, therefore, well-suited for approximation by SDP. We return to  $v_\phi$  in Section 6.

Assuming now that  $\phi$  represents a MAX-2-SAT instance on  $n$  variables, the corresponding MAX-2-SAT can be formulated as

$$\max \langle W, X \rangle \quad \text{s.t.} \quad \text{diag}(X) = \mathbf{1}, \quad X \succeq 0, \quad X \in \{\pm 1\}^{(n+1) \times (n+1)}, \quad (8)$$

where  $\langle W, X \rangle$  describes the quadratic polynomial  $v_\phi$ . Observe that the constraints of (8) enforce  $X$  to satisfy  $X = xx^\top$  for some  $x \in \{\pm 1\}^{n+1}$ . The size of this vector  $x$  is one more than the number of variables  $n$  to account for the additional truth value variable  $x_0$ .

A semidefinite relaxation of (8) is obtained by omitting the integrality constraint or, equivalently, nonconvex rank-one constraint. This constitutes the well-known Goemans and Williamson (1995) SDP relaxation of MAX-2-SAT. That is,

$$\max \langle W, X \rangle \quad \text{s.t.} \quad \text{diag}(X) = \mathbf{1}, \quad X \in \mathcal{S}_+^{n+1}. \quad (9)$$

Goemans and Williamson (1995) show that the optimal matrix for (9) can be used to obtain a 0.878-approximation algorithm to MAX-2-SAT. Assuming  $\mathcal{P} \neq \mathcal{NP}$ , for any  $\varepsilon > 0$  there exists no  $(\frac{21}{22} + \varepsilon)$  approximation algorithm to MAX-2-SAT (Håstad 2001). Karloff and Zwick (1997) introduce a canonical way of obtaining SDP relaxations for any MAX-SAT that is exploited to obtain approximation algorithms to MAX-3-SAT and MAX-4-SAT in Karloff and Zwick (1997) and Halperin and Zwick (2001), respectively. To solve MAX-2-SAT instances, rather than approximate, Wang and Zico Kolter (2019) propose the MIXSAT algorithm, which combines (9) with a B&B scheme.

## 3. SAT as a Semidefinite Feasibility Problem

In this section, we first present a brief overview of the work done by de Klerk et al. (2000) and Anjos (2004a, b; 2005; 2006; 2007). Their relaxations of SAT involve semidefinite feasibility problems: infeasibility of the semidefinite program implies unsatisfiability of the corresponding SAT instance. The difference between relaxations is the size of the SDP variable and the method of encoding the structure of the SAT instance in the SDP relaxation. Then, we propose a family of semidefinite feasibility problems that contains relaxations from Anjos (2004a, b; 2005; 2006; 2007)

and de Klerk et al. (2000) as special cases and show that a particular member of the family provides a rank-two guarantee; see Theorem 1.

We reconsider first Program (9), which attempts to satisfy the maximum number of clauses through its objective function. For SAT specifically, one might move the clause satisfaction part from the objective to the feasible set of a semidefinite program. This idea is first proposed by de Klerk et al. (2000) and later extended by Anjos (2004a). To be precise: de Klerk et al. (2000) propose the so-called GAP relaxation, or GAP for short, which is a semidefinite feasibility problem, given by

$$\begin{aligned} \text{find } & Y \in \mathcal{S}_+^n, y \in \mathbb{R}^n \\ \text{s.t. } & a_j^\top Y a_j - 2a_j^\top y \leq \ell_j(\ell_j - 2), \quad \forall j \in [m], \text{diag}(Y) = \mathbf{1}, Y \succeq yy^\top, \end{aligned} \tag{GAP}$$

for  $a_j$  as in (3). It is noted in de Klerk et al. (2000) that, for  $\ell_j \leq 2$ , the corresponding inequalities in GAP relaxation may be changed to equalities. The GAP relaxation is suited for instances that contain a clause of length two. If  $\ell_j \geq 3, \forall j \in [m]$ , then  $(Y, y) = (I, \mathbf{0})$  is always feasible for GAP whether the underlying SAT instance is satisfiable or not.

We now state the SDP relaxations of SAT by Anjos (2004a, b; 2005; 2006; 2007) that are not restricted to the lengths of the clauses in instances. Let  $\phi$  be a proposition on  $n$  variables and  $m$  clauses and  $x \in \{\pm 1\}^n$  the truth assignment to the variables. Consider a family of subsets  $\mathcal{F} = \{\alpha_1, \dots, \alpha_s\}$ , let  $\mathbf{x} = (x^{\alpha_1}, \dots, x^{\alpha_s})^\top$ , and define  $Y := \mathbf{x}\mathbf{x}^\top$ . It is clear that  $\text{rank}(Y) = 1, \text{diag}(Y) = \mathbf{1}$ , and  $Y \succeq 0$ . Later, to obtain a semidefinite relaxation of SAT, we omit the rank-one constraint.

We index the matrix  $Y$  with the elements of  $\mathcal{F}$  and define for all subsets  $\gamma$  contained in some clause of  $\phi$  the expression

$$Y(\gamma) := Y_{\alpha, \beta}, \text{ for some } \alpha, \beta \in \mathcal{F} \text{ jointly contained in a single clause s.t. } \alpha \Delta \beta = \gamma, \tag{10}$$

where  $\Delta$  is the symmetric difference operator, which is induced by the fact that, for  $x \in \{\pm 1\}^n$ , we have  $Y_{\alpha, \beta} = x^\alpha x^\beta = x^{\alpha \Delta \beta} = x^\gamma$ ; see (5). In general,  $Y(\emptyset)$  refers to a diagonal entry of  $Y$ ; hence,  $Y(\emptyset) = 1$ . We may have  $Y(\gamma) = Y_{\emptyset, \gamma}$ , and we assume that, for all  $\gamma$  contained in a clause of  $\phi$ , we can always find  $\alpha$  and  $\beta$  as in (10).

The expression  $Y(\gamma)$  can refer to multiple entries of  $Y$ . By construction of  $Y$ , these entries are equal. Stated formally, we have  $Y \in \bigcap_{j \in [m]} \Delta_j$ , where

$$\Delta_j := \{Y \in \mathcal{S} \mid Y_{\alpha_1, \beta_1} = Y_{\alpha_2, \beta_2} \quad \forall (\alpha_1, \alpha_2, \beta_1, \beta_2) \in \mathcal{F} \text{ s.t. } \alpha_1 \Delta \beta_1 = \alpha_2 \Delta \beta_2 \subseteq C_j\}. \tag{11}$$

Observe that the sets  $\Delta_j$  do not capture all equalities present in  $Y$  because of the restriction  $\alpha_1 \Delta \beta_1 = \alpha_2 \Delta \beta_2 \subseteq C_j$ . In this section, we choose to include only the equalities captured by  $\Delta_j$ . This keeps our work in line with previous relaxations by Anjos (2004a), and these equalities suffice to prove the main theorem in this section; see Theorem 1. In Section 6, we consider an SDP relaxation of SAT that considers all equalities present in  $Y$ .

If  $x$  is a satisfying assignment to  $\phi$ , then  $v(C_j) = 1$  (see (6)) for all  $j \in [m]$ . We can rewrite this constraint in terms of  $Y(\gamma)$ ; see (10). We now omit the rank-one constraint on  $Y$  to obtain the following semidefinite feasibility program, denoted by  $R_{\mathcal{F}}(\phi)$ :

$$\begin{aligned} \text{find } & Y \in \mathcal{S}_+^{|\mathcal{F}|} \\ \text{s.t. } & \sum_{\gamma \subseteq C_j} (-1)^{|\gamma|} a_j^\gamma Y(\gamma) = 0, \quad \forall j \in [m], \text{diag}(Y) = \mathbf{1}, Y \in \bigcap_{j \in [m]} \Delta_j. \end{aligned} \tag{R_{\mathcal{F}}(\phi)}$$

The program  $R_{\mathcal{F}}(\phi)$  contains both the GAP relaxation and the relaxations proposed by Anjos (2004b) as special cases. Specifically, one obtains the GAP relaxation from  $R_{\mathcal{F}}(\phi)$  by taking  $\mathcal{F} = \{\alpha \subseteq [n] \mid |\alpha| \leq 1\}$ . de Klerk et al. (2000) prove that, whenever GAP is feasible for a 2-SAT instance, the 2-SAT instance is satisfiable. Note that 2-SAT is decidable in linear time (Aspvall et al. 1979) unlike the NP-complete  $k$ -SAT with  $k \geq 3$ .

The GAP relaxation can be considered as a semidefinite program in the first level of the well-known Lasserre (2001) hierarchy. Anjos (2004b) proposes semidefinite relaxations of SAT in approximately levels two and three of the Lasserre (2001) hierarchy by only adding a subset of products of variables to the moment relaxation. For example, Anjos (2004a) proposes the  $R_2$  relaxation, which can be obtained from  $R_{\mathcal{F}}(\phi)$  by taking

$$\mathcal{F} = \{\alpha \mid \alpha \subseteq C_j \text{ for some } j, |\alpha| \text{ odd, or } \alpha = \emptyset\}. \tag{12}$$

Anjos (2004a) proves that the  $R_2$  relaxation attains a rank-two guarantee on 3-SAT instances: whenever the SDP admits a feasible matrix of rank two or lower, the corresponding SAT instance is satisfiable. We now prove that, for a different  $\mathcal{F}$  than (12), the resulting relaxation  $R_{\mathcal{F}}(\phi)$  provides the same rank-two guarantee.

**Theorem 1.** Let  $\phi$  be a 3-SAT instance and  $\mathcal{F} = \{\alpha \subseteq [n] \mid \alpha \subseteq C_j \text{ for some } j, |\alpha| \neq 1\}$ . Let  $Y$  be the matrix variable of  $R_{\mathcal{F}}(\phi)$  indexed by elements of  $\mathcal{F}$ . If  $R_{\mathcal{F}}(\phi)$  admits a feasible rank-two matrix, then  $\phi$  is satisfiable.

**Proof.** See Online Appendix A.  $\square$

## 4. Sum of Squares and MAX-SAT

In Section 4.1, we first provide an overview of the approach of van Maaren et al. (2008) for deriving relaxations for MAX-SAT. Their approach exploits SOS optimization, which has received much attention in the literature (see, e.g., Lasserre 2007, Laurent 2009, Scheiderer 2009, Parrilo and Thomas 2020). Relaxations depend on a basis that is used to compute them. We introduce a parametric family of monomial bases with increasing complexity. In Section 4.2, we derive several properties of monomial bases that are later used in our computations.

### 4.1. General Overview

For a given logical proposition  $\phi$  on  $n$  variables and  $m$  clauses, the value

$$F_{\phi}(x) := \sum_{j=1}^m \frac{1}{2^{\ell_j}} \prod_{i \in C_j} (1 - a_{j,i} x_i), \quad (13)$$

for  $a_{j,i}$  as in (3) equals the number of unsatisfied clauses by truth assignment  $x \in \{\pm 1\}^n$ . Hence, we are interested in minimizing  $F_{\phi}$  over  $\{\pm 1\}^n$ , on which  $F_{\phi}$  is nonnegative. Let  $\mathbb{R}[x]$  be the set of real polynomials in  $x$ . We define

$$\mathcal{V} := \left\{ f \mid f \equiv \sum_{j=1}^k f_j^2 \pmod{\mathcal{I}}, f_j \in \mathbb{R}[x] \quad \forall j \in [k], k \in \mathbb{N} \right\} \quad (14)$$

as the set of SOS polynomials modulo  $\mathcal{I}$ , where  $\mathcal{I}$  is the vanishing ideal of  $\{\pm 1\}^n$ . That is,

$$\mathcal{I} = \langle 1 - x_1^2, 1 - x_2^2, \dots, 1 - x_n^2 \rangle. \quad (15)$$

By Putinar's Positivstellensatz (Putinar 1993),  $\mathcal{V}$  is the set of nonnegative polynomials on  $\{\pm 1\}^n$ . Generally, optimization over  $\mathcal{V}$  is intractable because of its size, which is why we consider

$$\mathcal{V}_{\mathbf{x}} := \{f \mid f \equiv \mathbf{x}^T M \mathbf{x} \pmod{\mathcal{I}}, M \succeq 0\}, \quad (16)$$

where  $\mathbf{x}$  is some monomial basis. Because  $M \succeq 0$ , it follows that all polynomials of  $\mathcal{V}_{\mathbf{x}}$  are nonnegative on  $\{\pm 1\}^n$ . Therefore,  $\mathcal{V}_{\mathbf{x}} \subseteq \mathcal{V}$ , and we may approximate the minimum of  $F_{\phi}$  by

$$\min_{x \in \{\pm 1\}^n} F_{\phi} = \sup\{\lambda \mid F_{\phi} - \lambda \in \mathcal{V}\} \geq \sup\{\lambda \mid F_{\phi} - \lambda \in \mathcal{V}_{\mathbf{x}}\}. \quad (17)$$

The description of  $\mathcal{V}_{\mathbf{x}}$  shows that computing this lower bound can be done via SDP.

It is important to note that, in the quotient ring of  $\mathbb{R}[x]$  modulo  $\mathcal{I}$ , all terms  $x_i^2 \equiv 1$ , and thus, it suffices to consider only monomials in  $\mathbf{x}$  for which the highest power is at most one. Thus, we can write

$$F_{\phi}(x) = \sum_{\alpha \subseteq [n]} p_{\phi}^{\alpha} x^{\alpha}, \quad (18)$$

where  $p_{\phi}^{\alpha} \in \mathbb{R}$  for all  $\alpha \subseteq [n]$  and  $x^{\alpha}$  as in (5). For the constant term of  $F_{\phi}(x)$ , we have

$$p_{\phi}^{\emptyset} = \sum_{j=1}^m \frac{1}{2^{\ell_j}}. \quad (19)$$

We say that monomial basis  $\mathbf{x}$  represents a logical proposition  $\phi$  if matrix  $\mathbf{X} \equiv \mathbf{x} \mathbf{x}^T \pmod{\mathcal{I}}$  contains all monomials  $x^{\alpha}$  for which  $p_{\phi}^{\alpha} \neq 0$ . We index this matrix  $\mathbf{X}$  and the matrix  $M$  from (16) with subsets  $\alpha \subseteq [n]$  for which  $x^{\alpha} \in \mathbf{x}$ . Note that for such  $\alpha, \beta \subseteq [n]$ , we have

$$\mathbf{X}_{\alpha, \beta} \equiv x^{\alpha \Delta \beta} \pmod{\mathcal{I}}. \quad (20)$$

For  $\alpha \subseteq [n]$ , we write  $x^{\alpha} \in \mathbf{X}$  if  $\mathbf{X}$  has an entry equal to  $x^{\alpha}$  (modulo  $\mathcal{I}$ ). van Maaren et al. (2008) propose multiple monomial bases  $\mathbf{x}$ , among them basis  $SOS_p$ , given by

$$\mathbf{x} = 1 \cup \{x_i \mid i \in [n]\} \cup \{x_i x_j \mid i \text{ and } j \text{ appear together in a clause}\}. \quad (21)$$

It is stated in van Maaren et al. (2008) that  $SOS_p$  represents 2-SAT and 3-SAT instances. Whereas this is true, this basis also represents 4-SAT instances (see Lemma 2). We additionally define for  $Q \in \mathbb{N}$ , as an extension to  $SOS_p$ , the basis

$$SOS_p^Q := SOS_p \cup \{x_i x_j \mid i \text{ and } j \text{ are both in the top } Q \text{ appearing variables}\}. \quad (22)$$

Basis  $SOS_p^Q$  takes basis  $SOS_p$  and adds all the  $\binom{Q}{2}$  quadratic terms of the  $Q$  variables appearing in the highest number of clauses of  $\phi$ . Any basis  $\mathbf{x}$  is considered to have duplicate monomials removed, and so, for small values of  $Q$ , bases  $SOS_p^Q$  and  $SOS_p$  might coincide.

We also define the basis  $SOS_s^\theta$  for  $\theta \in [0, 1]$ , which is suited for (MAX-)2-SAT instances. This basis consists of all the monomials of degrees one and zero plus a percentage  $\theta$  of all quadratic monomials appearing in  $SOS_p$ . The included quadratic monomials are those that appear in  $SOS_p$  and attain the highest monomial weight  $w$ , which is defined as  $w(x^\alpha) := \sum_{i \in \alpha} w(i)$ , where  $w(i) := |\{C \in \phi \mid i \in C\}|$  for  $i \in [n]$ . This results in the following chain of inclusions:

$$\{x^\alpha \mid |\alpha| \leq 1\} = SOS_s^0 \subseteq SOS_s^\theta \subseteq SOS_s^1 = SOS_p = SOS_p^0 \subseteq SOS_p^Q \subseteq SOS_p^n = \{x^\alpha \mid |\alpha| \leq 2\}. \quad (23)$$

We now define, for all  $\gamma \subseteq [n]$  such that  $x^\gamma \in \mathbf{X}$ , a set of ordered pairs as follows:

$$\mathbf{x}^\gamma := \{(\alpha, \beta) \subseteq [n]^2 \mid \alpha \Delta \beta = \gamma, x^\alpha \in \mathbf{x}, x^\beta \in \mathbf{x}\}. \quad (24)$$

Set  $\mathbf{x}^\gamma$  contains the index pairs  $(\alpha, \beta)$  such that  $\mathbf{X}_{\alpha, \beta} \equiv x^\gamma$ . Therefore,  $F_\phi \equiv \mathbf{x}^\top M \mathbf{x}$  if and only if

$$\sum_{(\alpha, \beta) \in \mathbf{x}^\gamma} M_{\alpha, \beta} = p_\phi^\gamma, \quad \forall \gamma \text{ such that } x^\gamma \in \mathbf{X}. \quad (25)$$

Constraints of this form are sometimes referred to as coefficient matching conditions in SOS literature (Zheng et al. 2017). We define

$$\mathcal{M}_\phi := \left\{ M \in S^{|\mathbf{X}|} \mid \sum_{(\alpha, \beta) \in \mathbf{x}^\gamma} M_{\alpha, \beta} = p_\phi^\gamma, \quad \forall \gamma \neq \emptyset \text{ such that } x^\gamma \in \mathbf{X} \right\}, \quad (26)$$

as the set of matrices satisfying the coefficient matching conditions for all monomials except  $x^\emptyset$ .

Note that  $M$  is constrained to be symmetric, which is reflected in the definition of  $\mathbf{x}^\gamma$ , because  $(\alpha, \beta) \in \mathbf{x}^\gamma$  if and only if  $(\beta, \alpha) \in \mathbf{x}^\gamma$ . Moreover,  $\mathbf{x}^\emptyset$  contains the index pairs of the diagonal entries of  $M$ , which correspond to zero-degree monomials in  $\mathbf{X}$ . Hence, if  $F_\phi - \lambda \equiv \mathbf{x}^\top M \mathbf{x}$ , then  $p_\phi^\emptyset - \lambda = \langle I, M \rangle$ ; see (17) and (19). To maximize the lower bound on  $F_\phi$  (see (17)), we maximize  $\lambda$ , which is, thus, equivalent to minimizing  $\langle I, M \rangle$ . We can, therefore, compute this lower bound by solving the following SDP:

$$\min \langle I, M \rangle \quad \text{s.t. } M \in \mathcal{M}_\phi \cap \mathcal{S}_+. \quad (\mathbb{P}_\phi)$$

We note that, for the purpose of solving  $\mathbb{P}_\phi$  through interior point methods, program  $\mathbb{P}_\phi$  is strictly feasible: for any feasible matrix  $M$ , matrix  $M + I$  is strictly feasible. The existence of any such feasible matrix  $M$  follows from the non-negativity of  $F_\phi$  on  $\{\pm 1\}^n$ . We postpone the derivation of the dual of  $\mathbb{P}_\phi$  to Section 6, in which we also show its strict feasibility in Theorem 2.

## 4.2. Properties of $SOS_p^Q$

We provide several properties of monomial bases that are exploited within the PRSM; see Section 7.

Denote by  $|\mathbf{x}^\gamma|$  the cardinality of the set  $\mathbf{x}^\gamma$ ; see (24). Because of the symmetry of  $\mathbf{X}$  (see (20)),  $|\mathbf{x}^\gamma|$  is an even number and greater than or equal to two. In particular, when  $|\mathbf{x}^\gamma| = 2$ , say  $\mathbf{x}^\gamma = \{(\alpha, \beta), (\beta, \alpha)\}$ , we have

$$M_{\alpha, \beta} + M_{\beta, \alpha} = p_\phi^\gamma \text{ and } M_{\alpha, \beta} = M_{\beta, \alpha} \Rightarrow M_{\alpha, \beta} = M_{\beta, \alpha} = p_\phi^\gamma / 2. \quad (27)$$

Thus, whenever  $|\mathbf{x}^\gamma| = 2$ , the constraint involving  $\mathbf{x}^\gamma$  in  $\mathcal{M}_\phi$  (see (26)), simply fixes two entries of  $M$ . van Maaren et al. (2008) refer to these constraints arising from  $|\mathbf{x}^\gamma| = 2$  as unit constraints. In section 7 of van Maaren et al. (2008), the authors empirically show that a high percentage of the constraints of  $\mathcal{M}_\phi$  are unit constraints. van Maaren et al. (2008) propose as future work the development of an SDP solver that is able to exploit the large number of unit constraints. We propose an algorithm for approximately solving  $\mathbb{P}_\phi$  in Section 7 that is able to do so.

The following lemma describes the subsets  $\gamma$  that induce unit constraints.

**Lemma 1.** Let  $\phi$  be a (MAX-)SAT instance on  $n$  variables and  $m$  clauses and  $\mathbf{x}$  a monomial basis that contains at least all of the monomials induced by the  $SOS_p$  basis; see (21). Then, for all  $\gamma \subseteq [n]$ , we have  $|\mathbf{x}^\gamma| = 2 \Rightarrow p_\phi^\gamma = 0$ , where  $p_\phi^\gamma$  is a coefficient of  $F_\phi(x)$ ; see (18).

**Proof.** See Online Appendix A.  $\square$

Lemma 1 implies that, in an implementation that uses the  $SOS_p$  basis, it is not required to store the coefficients corresponding to unit constraints (because these all equal zero), but only the indices restricted by the unit constraints. The converse of Lemma 1 is generally not true. That is, there can exist many subsets  $\gamma \subseteq [n]$  for which  $p_\phi^\gamma = 0$ , but  $|\mathbf{x}^\gamma| > 2$ .

**Lemma 2.** Let  $\phi$  be an SAT instance on  $n$  variables and  $\mathbf{x}$  its monomial basis according to  $SOS_p^Q$  for some  $Q \in \mathbb{N}$ . Let  $\gamma \subseteq [n]$ . Then,

1.  $|\gamma| \in \{1, 2\} \Rightarrow |\mathbf{x}^\gamma| \leq 2n$ .
2.  $|\gamma| \in \{3, 4\} \Rightarrow |\mathbf{x}^\gamma| \leq 6$ .
3.  $|\gamma| > 4 \Rightarrow |\mathbf{x}^\gamma| = 0$ .

**Proof.** See Online Appendix A.  $\square$

Part 3 of Lemma 2 shows that the  $SOS_p^Q$  bases are only suited for (MAX-) $k$ -SAT when  $k \leq 4$ .

## 5. Resolution and Monomial Bases

In this section, we consider resolution in combination with the SOS approach to MAX-SAT. Resolution is a technique from mathematical logic and widely employed by MAX-SAT solvers (Prasad et al. 2005). Resolution takes as inputs two clauses of a proposition  $\phi$  and returns a set of new clauses named the resolvent clauses. The resolvent clauses transform  $\phi$  into  $\phi'$  by either replacing the original clauses or adding the resolvent clauses to  $\phi$  (depending on which resolution rule is used). We show in this section that the MAX-SAT resolution rule might not be beneficial for the SOS approach applied to MAX-SAT and can even decrease its effectiveness. However, in Section 8.2, we show how to benefit from the SAT resolution rule in partial MAX-SAT.

We show this using an example. For  $k \geq 3$ , we define the following proposition on  $k$  variables:

$$\phi_k := \begin{cases} \neg x_1 \wedge (x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge \neg x_3 & \text{if } k = 3, \\ \neg x_1 \wedge (x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge \left[ \bigwedge_{j=3}^{k-1} (\neg x_j \vee x_{j+1}) \right] \wedge \neg x_k & \text{else.} \end{cases} \quad (28)$$

It is clear that  $\phi_k$  is unsatisfiable. If one satisfies the initial two unit clauses and performs unit resolution, more unit clauses appear. Repeating this process leads to an all-false truth assignment, leaving clause  $x_2 \vee x_3$  unsatisfied. Therefore, any truth assignment leaves at least one clause unsatisfied, and hence,  $\min_{\mathbf{x} \in \{\pm 1\}^k} F_{\phi_k} = F_{\phi_k}(-\mathbf{1}_k) = 1$  for  $F_{\phi_k}$  as in (13). In the following lemma, we show that the  $SOS_p$  basis (see (21)), suffices for proving optimality of this assignment.

**Lemma 3.** For all  $k \geq 3$ , we have  $\max\{\lambda | F_{\phi_k} - \lambda \in \mathcal{V}_x\} = 1$ , where  $\mathbf{x} = SOS_p(\phi_k)$ , and  $\mathcal{V}_x$  as in (16).

**Proof.** See Online Appendix A.  $\square$

Let us now present the MAX-SAT resolution rule (see, e.g., Abramé and Habet 2014). For clauses  $C_1$  and  $C_2$  of some proposition  $\phi$ , on literals  $x, z_i, i \in [s]$  and  $y_i, i \in [t]$  construct the clauses below the horizontal line:

$$\begin{array}{c} C_1 = [x \vee z_1 \vee \dots \vee z_s], C_2 = [\neg x \vee y_1 \vee \dots \vee y_t] \\ \hline z_1 \vee \dots \vee z_s \vee y_1 \vee \dots \vee y_t, \\ [C_1 \vee \neg y_1 \vee y_2 \vee \dots \vee y_t], [C_1 \vee \neg y_2 \vee y_3 \vee \dots \vee y_t], \dots, [C_1 \vee \neg y_t], \\ [C_2 \vee \neg z_1 \vee z_2 \vee \dots \vee z_s], [C_2 \vee \neg z_2 \vee z_3 \vee \dots \vee z_s], \dots, [C_2 \vee \neg z_s]. \end{array} \quad (29)$$

The MAX-SAT resolution rule states that one may replace clauses  $C_1$  and  $C_2$  in  $\phi$  with a subset of the  $1 + s + t$  resolvent clauses below the horizontal line. Namely, clauses that are trivially satisfied, such as  $x \vee \neg x$ , are not part of this subset. We refer to the resulting new proposition, obtained after resolution, as  $\phi'$ . In Bonet et al. (2007), theorem 4, it is proven that  $F_\phi = F_{\phi'}$ ; see (13). Note that, in Bonet et al. (2007), definition 6, function  $F_\phi$  is stated in terms of  $\{0, 1\}$  variables.

For standard MAX-SAT solvers, one of the goals of resolution is to create new unit clauses, which are used to compute upper bounds on the MAX-SAT solution (Abramé and Habet 2014). For our SDP approach, assuming a fixed monomial basis, the sets  $\mathcal{M}_\phi$  and  $\mathcal{M}_{\phi'}$  (see (26)), depend only on the coefficients of  $F_\phi$  and  $F_{\phi'}$ . Because  $F_\phi = F_{\phi'}$ , these coefficients are equal, and thus,  $\mathcal{M}_\phi = \mathcal{M}_{\phi'}$ . Note that the feasible set of  $\mathbb{P}_\phi$  is defined in terms of  $\mathcal{M}_\phi$ . Hence, it follows that, if given the same basis, program  $\mathbb{P}_\phi$  equals program  $\mathbb{P}_{\phi'}$ . This equivalence of programs suggests that MAX-SAT resolution does not change our approach; however, we find that, in general,  $SOS_p(\phi) \neq SOS_p(\phi')$ ; see (21). We investigate the effect of this difference.

Returning to the example of  $\phi_k$  in (28), let us define  $C_q = \neg x_q \vee x_{q+1}$ . Observe that, for  $3 \leq q \leq k-1$ ,  $C_q \in \phi_k$  (assuming  $k > 3$ ). Let us fix some  $q$ ,  $3 \leq q \leq k-3$ , and consider the clauses  $C_q, C_{q+1}, C_{q+2} \in \phi_k$ . We perform resolution as

$$\frac{C_q = [\neg x_q \vee x_{q+1}], C_{q+1} = [\neg x_{q+1} \vee x_{q+2}]}{[\neg x_q \vee x_{q+2}], [\neg x_q \vee x_{q+1} \vee \neg x_{q+2}], [x_q \vee \neg x_{q+1} \vee x_{q+2}]} \quad (30)$$

We perform resolution again on the third new clause obtained in (30) and  $C_{q+2}$  to obtain

$$\frac{[x_q \vee \neg x_{q+1} \vee x_{q+2}], C_{q+2} = [\neg x_{q+2} \vee x_{q+3}]}{[x_q \vee \neg x_{q+1} \vee x_{q+3}], [x_q \vee \neg x_{q+1} \vee x_{q+2} \vee \neg x_{q+3}], [\neg x_q \vee \neg x_{q+1} \vee \neg x_{q+2} \vee x_{q+3}], [x_{q+1} \vee \neg x_{q+2} \vee x_{q+3}]} \quad (31)$$

and, here, the resolution rule states that one may replace the original clauses  $C_1, C_2$ , and  $C_3$  with the six new resolvent clauses obtained from (30) and (31) (the third resolvent from (30) is not counted because it is replaced in the resolution in (31)).

Observe that the  $SOS_p$  basis generates six quadratic monomials for the new resolvent clauses, whereas originally, only three quadratic monomials are generated for  $C_q, C_{q+1}$ , and  $C_{q+2}$ . We now define  $\phi'_k$  for  $k \geq 6$  as the logical proposition, obtained by taking  $\phi_k$  and performing resolution as in (30) and (31) for each triple of clauses  $\{C_q, C_{q+1}, C_{q+2}\}$  for each  $q \in \{3, 6, 9, \dots, k-3\}$  (let us assume here that  $k$  is a multiple of three). Note that proposition  $\phi'_k$  constitutes a MAX-4-SAT instance, and therefore, basis  $SOS_p$  is applicable. Let us compare the sizes of the resulting  $SOS_p$  bases, denoted as  $|SOS_p|$ . We have  $|SOS_p(\phi_k)| = 2k < 3k - 3 = |SOS_p(\phi'_k)|$ .

Thus, compared with  $SOS_p(\phi_k)$ , basis  $SOS_p(\phi'_k)$  adds approximately  $k$  monomials. None of these monomials strengthens the bound because  $SOS_p(\phi_k)$  is already sufficient for proving optimality by Lemma 3. It is clear that having a larger basis without offering a stronger bound is inefficient because solving  $\mathbb{P}_\phi$  requires more time for larger matrices.

The example of  $\phi'_k$  and  $\phi_k$  shows that not all monomials are (equally) useful in determining bounds. It also shows that resolution can decrease the effectiveness of the SOS approach to MAX-SAT by providing “bad” monomial bases or it can occur that the  $SOS_p$  basis misses “good” monomials. Our proposed basis  $SOS_p^Q$  (see (22)) attempts to solve this issue.

## 6. Relating Sum of Squares and Method of Moments

In this section, we show how the SOS-SDP relaxation of van Maaren and van Norden (2005) and van Maaren et al. (2008) and moment relaxations of Anjos (2004a, b; 2005; 2006; 2007) are related. The relaxations of Anjos as described in Section 3 are first introduced in Anjos (2004a) and can be considered as extensions of the GAP relaxation via the well-known Lasserre (2001) hierarchy. van Maaren and van Norden (2005) propose the SOS approach to (MAX-)SAT. Subsequently, van Maaren and van Norden (2005) show that the SOS relaxation, using monomial basis  $SOS_{pt}$  that is larger than  $SOS_p$  (see (21)) outperforms the  $R_3$  relaxation of Anjos (2005) in deciding on the satisfiability of 3-SAT instances. The  $R_3$  relaxation is known to dominate the  $R_2$  relaxation; see (12). Anjos (2007) strengthened his  $R_3$  relaxation further and left it as future work to determine which SDP relaxation was the strongest.

We complete that work here by showing a simple relation between the two approaches. In particular, Anjos’ relaxations can be considered as method of moments in the Lasserre (2001) hierarchy. It is well-known that method of moments is dual to SOS optimization (see Lasserre 2001), and we work out the details here. Let us first derive the dual of the SOS program  $\mathbb{P}_\phi$  (see Theorem 2) and then relate it to the here proposed strengthened version of Anjos’ relaxations.

To this end, we require the following intermediate result on  $v_\phi$ ; see (7).

**Lemma 4.** Let  $\phi = \bigwedge_{j=1}^m C_j$  be a logical proposition,  $v_\phi = \sum_{\alpha \subseteq [n]} v_\phi^\alpha x^\alpha$  (see (7)), and  $F_\phi = \sum_{\alpha \subseteq [n]} p_\phi^\alpha x^\alpha$ ; see (18). Then,  $v_\phi = m - F_\phi$ , and  $v_\phi^\alpha = -p_\phi^\alpha$  for all nonempty  $\alpha \subseteq [n]$ .

**Proof.** See Online Appendix A.  $\square$

Let  $\mathbf{x}$  be a given monomial basis,  $S \in \mathcal{S}^{|\mathbf{x}|}$ . Matrix  $S$  is indexed by all  $\alpha \subseteq [n]$  for which  $x^\alpha \in \mathbf{x}$ . To simplify the comparison between the SOS approach and the relaxations of Anjos (2004a), we define the set

$$\mathcal{X}_\phi := \{S \in \mathcal{S} \mid \text{diag}(S) = \mathbf{1}, S_{\alpha,\beta} = S_{\alpha',\beta'} \quad \forall (\alpha,\beta,\alpha',\beta') \subseteq [n] \text{ s.t. } \alpha \Delta \beta = \alpha' \Delta \beta'\}, \quad (32)$$

for a proposition  $\phi$  on  $n$  variables and  $m$  clauses. Note that  $\mathcal{X}_\phi \subseteq \bigcap_{j \in [m]} \Delta_j$  (see (11)) because  $\Delta_j$  only restricts entries  $S_{\alpha,\beta}$  whenever  $\alpha$  and  $\beta$  are jointly contained in a single clause. We use  $\mathcal{X}_\phi$  in the following theorem.

**Theorem 2.** Let  $\phi$  be a logical proposition and  $\mathbf{x}$  a monomial basis. The SOS program  $\mathcal{P}_\phi$  defined by  $\phi$  and  $\mathbf{x}$  is equivalent to

$$\max \langle C, S \rangle \quad \text{s.t.} \quad S \in \mathcal{X}_\phi \cap \mathcal{S}_+, \quad (33)$$

where  $\mathcal{X}_\phi$  is given by (32) and  $C \in \mathcal{S}^{|\mathbf{x}|}$ , indexed by the subsets  $\alpha \subseteq [n]$  for which  $x^\alpha \in \mathbf{x}$ , is any matrix that satisfies  $\sum_{(\alpha,\beta) \in \mathbf{x}^\gamma} C_{\alpha,\beta} = v_\phi^\gamma \quad \forall \gamma \neq \emptyset, \mathbf{x}^\gamma \neq \emptyset$  for  $v_\phi^\gamma$  as in (7). Moreover, (33) is strictly feasible.

**Proof.** We rewrite program  $\mathcal{P}_\phi$  by splitting the matrix variable  $M$  as follows:

$$v := \min \langle I, M \rangle \quad \text{s.t.} \quad M = Z, M \in \mathcal{M}_\phi, Z \in \mathcal{S}_+, \quad (34)$$

where  $\mathcal{M}_\phi$  as in (26). We dualize the constraint  $M = Z$ , and set

$$g(S) := \min_{M \in \mathcal{M}_\phi, Z \geq 0} \langle I, M \rangle + \langle S, M - Z \rangle,$$

for some  $S \in \mathcal{S}$ . Clearly,  $g(S) \leq v$  for all  $S$ , and we, thus, look to maximize  $g(S)$ ; that is,

$$\begin{aligned} \max_S g(S) &= \max_S \left[ \min_{M \in \mathcal{M}_\phi} \langle I + S, M \rangle + \min_{Z \geq 0} \langle S, -Z \rangle \right] = \max_{S \geq 0} \min_{M \in \mathcal{M}_\phi} \langle I + S, M \rangle \\ &= \max_{S \geq 0} \min_{M \in \mathcal{M}_\phi} \langle I - S, M \rangle. \end{aligned} \quad (35)$$

We now determine the set  $\mathcal{X}_\phi$  such that, whenever  $S \in \mathcal{X}_\phi$ , the minimization over  $M \in \mathcal{M}_\phi$  in (35) is bounded. Observe that  $\mathcal{M}_\phi$  places no restrictions on the diagonal. To guarantee a bounded minimum, set  $\mathcal{X}_\phi$  should restrict  $\text{diag}(I - S) = \mathbf{0}$ . Each off-diagonal element of a matrix in  $\mathcal{M}_\phi$  is restricted by a single constraint of the form (25). Therefore, solving (35) for  $M$  can be done by considering separately the elements of  $M$  restricted by a single constraint. That is,

$$\min_{M \in \mathcal{S}} \sum_{\gamma \in \mathbf{X}} \sum_{(\alpha,\beta) \in \mathbf{x}^\gamma} -S_{\alpha,\beta} M_{\alpha,\beta} \quad \text{s.t.} \quad \sum_{(\alpha,\beta) \in \mathbf{x}^\gamma} M_{\alpha,\beta} = p_\phi^\gamma, \quad (36)$$

where  $\mathbf{x}^\gamma$  and  $\mathbf{X}$  are defined in (24) and (20), respectively. This minimization problem is bounded if and only if

$$S_{\alpha,\beta} = S_{\alpha',\beta'}, \quad \forall (\alpha,\beta), (\alpha',\beta') \in \mathbf{x}^\gamma \quad \forall \gamma \in \mathbf{X}, \quad (37)$$

or equivalently,  $S_{\alpha,\beta} = S_{\alpha',\beta'}$  for all possible index pairs  $(\alpha,\beta)$  and  $(\alpha',\beta')$  that satisfy  $\alpha \Delta \beta = \alpha' \Delta \beta'$ . It follows that  $\mathcal{X}_\phi$  is given by (32). Now, for fixed  $S \in \mathcal{X}_\phi \cap \mathcal{S}_+$ , any matrix  $M \in \mathcal{M}_\phi$  obtains the same value in (35). Note also that, w.l.o.g., we may fix  $M = \mathcal{P}_{\mathcal{M}_\phi}(\mathbf{0})$ , that is, the projection of the zero matrix onto  $\mathcal{M}_\phi$  (see Lemma 5), which has zero diagonal. This yields the equivalent program of the form (33) for  $C = -M = -\mathcal{P}_{\mathcal{M}_\phi}(\mathbf{0})$ . Written explicitly,  $C_{\alpha,\beta} = -p_\phi^\gamma / |\mathbf{x}^\gamma|$  for all  $\alpha,\beta \subseteq [n]$  such that  $\alpha \Delta \beta = \gamma$  (i.e.,  $(\alpha,\beta) \in \mathbf{x}^\gamma$ ). This, combined with Lemma 4, proves the claim on matrix  $C$ . Finally, observe that the identity matrix of appropriate size is strictly feasible for (33).  $\square$

We define, for  $S \in \mathcal{X}_\phi$  and each clause  $C_j$ , the function  $v^{\text{SDP}}(S, C_j)$ , which is obtained by taking (6) and replacing each  $x^\gamma$  by  $S_{\alpha,\beta}$  for some  $(\alpha,\beta) \in \mathbf{x}^\gamma$ . By (37), we are allowed to pick any such  $(\alpha,\beta)$ . By Lemma 4, for any nonempty  $\gamma \subseteq [n]$ ,  $S \in \mathcal{X}_\phi$  and  $C$  as in Theorem 2, we have

$$\sum_{(\alpha,\beta) \in \mathbf{x}^\gamma} C_{\alpha,\beta} S_{\alpha,\beta} = \sum_{(\alpha,\beta) \in \mathbf{x}^\gamma} \frac{-p_\phi^\gamma}{|\mathbf{x}^\gamma|} S_{\alpha,\beta} = -p_\phi^\gamma S_{\alpha,\beta} = v_\phi^\gamma S_{\alpha,\beta}. \quad (38)$$

Hence, maximizing  $\langle C, S \rangle$  is equivalent to maximizing the semidefinite relaxation of  $v_\phi$  (see (7)), which equals  $\sum_{j \in [m]} v^{\text{SDP}}(S, C_j)$ .

Moreover, in the relaxations of Anjos (2004a) outlined in Section 3, the matrix variable is restricted to satisfy  $v^{\text{SDP}}(S, C_j) = 1$ . Now, we can easily observe the difference between the SOS-SDP relaxations and those proposed by

Anjos. We present the equivalent dual formulation of the SOS approach on the left-hand side and the latter (in slightly adapted form) on the right.

$$v^* = \max \sum_{j \in [m]} v^{\text{SDP}}(S, C_j)$$

$$\text{s.t. } S \in \mathcal{X}_\phi \cap \mathcal{S}_+. \tag{39}$$

$$\text{find } S \in \mathcal{X}_\phi \cap \mathcal{S}_+$$

$$\text{s.t. } v^{\text{SDP}}(S, C_j) = 1, \forall C_j. \tag{40}$$

Note again the difference between (40) and the relaxations described in Section 3, resulting from using set  $\mathcal{X}_\phi$  instead of the intersection of the  $\Delta_j$ ; see (11). Thus, we compare the SOS approach with a strengthened variant of the relaxation proposed by Anjos. In Section 8.3, we determine the dual of (40).

Program (39) proves the unsatisfiability of  $\phi$  if  $v^* < m$  (with some margin of error because of numerical precision), whereas (40) does so whenever the program is infeasible. These programs are not equivalent in this sense: we empirically find instances  $\phi$  for which  $v^* \geq m$ , whereas (40) is infeasible. Neither program can directly prove satisfiability. However, solutions to both programs can be used to guide the search toward satisfying assignments (should they exist); see Online Appendix C.

If (40) admits a feasible matrix  $S^*$ , then matrix  $S^*$  is clearly also feasible for (39) and attains an objective value of  $m$ . Consequently, in this case, we have  $v^* \geq m$ . Thus, if (40) does not prove unsatisfiability of  $\phi$ , then neither does (39). In Online Appendix F, we show that (39) can be computed efficiently by applying the PRSM to its dual.<sup>2</sup> It is currently unclear whether a good algorithm for solving (40) exists, and if so, how efficient it would be. Previous numerical experiments on (40) use general purpose SDP solvers. An immediate improvement might be to use an SDP feasibility problem solver (see Henrion and Malick 2011, Drusvyatskiy et al. 2017).

Finally, the objective value of (39) is more useful for MAX-SAT: if the underlying instance is infeasible,  $v^*$  provides an upper bound to the number of satisfiable clauses, which is useful in a B&B scheme. Program (40) might also show unsatisfiability of the same instance, but its infeasibility offers no additional value as to how unsatisfiable the instance is.

## 7. The Peaceman–Rachford Splitting Method for MAX-SAT

In this section, we introduce the Peaceman–Rachford splitting method (Peaceman and Rachford 1955) for solving SOS-SDP problems and apply it to the MAX-SAT SOS program  $\mathcal{P}_\phi$ . Conventionally, interior point methods are used to solve SDP problems. However, for medium- and large-size instances, interior point methods suffer from a large computation time and memory demand, which has recently motivated researchers to consider first order methods, such as the PRSM. For recent applications of PRSM to SDP, see, for example, de Meijer and Sotirov (2021) and Graham et al. (2022).

Section 7.2 and Online Appendix C provide details on obtaining valid upper and lower bounds from the output of the PRSM algorithm.

### 7.1. The PRSM for SOS Relaxations of MAX-SAT

We start from the reformulation of  $\mathcal{P}_\phi$  given in (34). The augmented Lagrangian function of (34) with respect to (w.r.t.) the constraint  $M = Z$  for a penalty parameter  $\beta > 0$  is

$$L_\beta(Z, M, S) = \langle I, M \rangle + \langle S, M - Z \rangle + \frac{\beta}{2} \|M - Z\|_F^2. \tag{41}$$

Here,  $S \in \mathcal{S}^n$  is the Lagrange multiplier and  $\|\cdot\|_F$  denotes the Frobenius matrix norm; see Section 1.1.

The PRSM now entails iteratively optimizing over the variables  $Z$  and  $M$  separately and updating  $S$  twice per cycle. We write superscript  $k$  to denote the the value of the variable at iteration  $k$ :

$$\begin{cases} Z^{k+1} = \arg \min_{Z \geq 0} L_\beta(Z, M^k, S^k) = \mathcal{P}_{\mathcal{S}_+} \left( M^k + \frac{1}{\beta} S^k \right), \\ S^{k+1/2} = S^k + \gamma_1 \beta (M^k - Z^{k+1}), \\ M^{k+1} = \arg \min_{M \in \mathcal{M}_\phi} L_\beta(Z^{k+1}, M, S^{k+1/2}) = \mathcal{P}_{\mathcal{M}_\phi} \left( Z^{k+1} - \frac{1}{\beta} [I + S^{k+1/2}] \right), \\ S^{k+1} = S^{k+1/2} + \gamma_2 \beta (M^{k+1} - Z^{k+1}). \end{cases} \tag{42}$$

Here,  $\mathcal{M}_\phi$  is as in (26), and  $\mathcal{P}$  is the projection operator as in (4). We use that

$$\begin{aligned} \arg \min_{Z \geq 0} L_\beta(Z, M, S) &= \arg \min_{Z \geq 0} \langle I, M \rangle - \frac{1}{2\beta} \|S\|_F^2 + \frac{\beta}{2} \left\| Z - \left( M + \frac{1}{\beta} S \right) \right\|_F^2 \\ &= \arg \min_{Z \geq 0} \frac{\beta}{2} \left\| Z - \left( M + \frac{1}{\beta} S \right) \right\|_F^2, \end{aligned} \quad (43)$$

see, for example, Oliveira et al. (2018). In an implementation of (42), one should not store matrix  $S^k$  directly but, rather, the matrix  $\frac{1}{\beta} S^k$ ; see Online Appendix G. When  $X \in \mathcal{S}$  has eigenvalues  $\lambda_i$  and corresponding eigenvectors  $v_i$ , it is well-known that the projection onto the positive semidefinite cone is given by

$$\mathcal{P}_{\mathcal{S}_+}(X) = \sum_{\{i \mid \lambda_i > 0\}} \lambda_i v_i v_i^\top = X - \sum_{\{i \mid \lambda_i < 0\}} \lambda_i v_i v_i^\top. \quad (44)$$

Depending on the number of positive eigenvalues of  $X$ , one of these expressions is cheaper to compute. The next lemma shows how to compute a projection onto  $\mathcal{M}_\phi$ .

**Lemma 5.** Let matrices  $M, \widehat{M} \in \mathcal{S}$ , indexed by subsets of  $[n]$  such that  $\widehat{M} = \mathcal{P}_{\mathcal{M}_\phi}(M)$ , where the projection operator  $\mathcal{P}_{\mathcal{M}_\phi}(\cdot)$  is given by (4). Then,  $\text{diag}(\widehat{M}) = \text{diag}(M)$  and

$$\widehat{M}_{\delta, \mu} = M_{\delta, \mu} - \frac{1}{|\mathbf{x}^\gamma|} \left( \sum_{(\alpha, \beta) \in \mathbf{x}^\gamma} M_{\alpha, \beta} - p_\phi^\gamma \right), \quad (45)$$

for  $(\delta, \mu) \in \mathbf{x}^\gamma$ ,  $\gamma \neq \emptyset$ . In particular, when  $|\mathbf{x}^\gamma| = 2$ , (45) reduces to  $\widehat{M}_{\delta, \mu} = \widehat{M}_{\mu, \delta} = p_\phi^\gamma/2$ .

**Proof.** See Online Appendix A.  $\square$

Because of the presence of many unit constraints (see (27)), these projections are computationally cheap to compute, and hence, the PRSM is well-suited to exploit this. Finally, it is proven (He et al. 2016) that (42) converges for  $(\gamma_1, \gamma_2) \in \mathcal{D}$ , where

$$\mathcal{D} = \left\{ (\gamma_1, \gamma_2) \mid \gamma_1 + \gamma_2 > 0, |\gamma_1| < 1, 0 < \gamma_2 < \frac{1 + \sqrt{5}}{2}, |\gamma_1| < 1 + \gamma_2 - \gamma_2^2 \right\}. \quad (46)$$

The values that we choose for  $(\gamma_1, \gamma_2)$  and other parameters are given in Online Appendix F.

## 7.2. Upper Bounds, Lower Bounds, and Early Stopping

After each PRSM iterate  $k$ , we obtain a triple  $(Z^k, M^k, S^k)$  and the resulting  $\langle I, M^k \rangle$ . Although this value converges to the optimal solution of the SDP, the convergence is (typically) not monotonic, and therefore, this value does not necessarily provide a valid upper bound for the problem. In this section, we describe how to obtain a valid upper bound from the output of the PRSM.

Observe that the feasible set of  $\mathcal{P}_\phi$  depends on the chosen monomial  $\mathbf{x}$  through  $\mathcal{V}_\mathbf{x}$ ; see (16). Hence, by (17), we have

$$p_\phi^0 - \min_{M \in \mathcal{M}_\phi \cap \mathcal{S}_+} \langle I, M \rangle = \sup_{\lambda \in \mathbb{R}} \{ \lambda \mid F_\phi - \lambda \in \mathcal{V}_\mathbf{x} \} \leq \min_{x \in \{\pm 1\}^n} F_\phi, \quad (47)$$

for  $p_\phi^0$  as in (19). From this, it follows that the maximum number of satisfiable clauses of  $\phi$  is bounded from above by

$$m - p_\phi^0 + \min_{M \in \mathcal{M}_\phi \cap \mathcal{S}_+} \langle I, M \rangle, \quad (48)$$

for  $m$  equal to the number of clauses in  $\phi$ . Because the number of satisfied clauses is an integer, the bound (48) can be improved by rounding down the result.

Ideally, the PRSM algorithm (42) computes the upper bound (48) by finding the optimal  $M$  in the set  $\mathcal{M}_\phi \cap \mathcal{S}_+$  (up to some given numerical precision). However, in practice, one terminates the PRSM algorithm before this optimal  $M$  is found. Let matrix  $M^k$  then be defined as in (42) and let  $\lambda_{\min}(M^k)$  be its smallest eigenvalue. Note that

$$\widetilde{M}^k = M^k - \lambda_{\min}(M^k)I \in \mathcal{M}_\phi \cap \mathcal{S}_+, \quad (49)$$

and so  $\tilde{M}^k$  is feasible for  $\mathbb{P}_\phi$ . Thus, a valid upper bound at iteration  $k$  is obtained as follows:

$$\lfloor m - p_\phi^0 + \langle I, \tilde{M}^k \rangle \rfloor. \tag{50}$$

In Online Appendix C, we outline a method for determining lower bounds when using the PRSM.

### 8. Weighted Partial MAX-SAT

In this section, we extend the SOS approach from MAX-SAT to weighted partial MAX-SAT. We also show that the dual formulation of the SOS program for certain partial MAX-SATs equals the relaxations by Anjos (2005).

In weighted MAX-SAT, each clause is given a weight, and the objective is to maximize the sum of the weights of the satisfied clauses. In partial MAX-SAT, clauses are divided into soft and hard clauses. The aim is to maximize the number of satisfied soft clauses, satisfying all the hard clauses. The combination of weighted and partial MAX-SAT is clear and referred to as weighted partial MAX-SAT (Li and Manyá 2021).

Consider again a logical proposition  $\phi$ . Let  $w_j \in \mathbb{R}$  be the weight associated to clause  $C_j$ . The generalization of (13) for (unweighted) MAX-SAT to weighted MAX-SAT follows by setting

$$F_\phi^W(x) = \sum_{j=1}^m \frac{w_j}{2^{\ell_j}} \prod_{i \in C_j} (1 - a_{j,i} x_i), \tag{51}$$

and then minimizing  $F_\phi^W$  for  $x \in \{\pm 1\}^n$ . This minimization can be approximated by SOS optimization, using directly the semidefinite program  $\mathbb{P}_\phi$ .

For weighted partial MAX-SAT, consider a logical proposition  $\phi$  on  $n$  variables,  $m$  soft clauses  $C_j$ , and  $q$  hard clauses  $C_p^H$ . To each hard clause  $C_p^H$ ,  $p \in [q]$ , we associate the polynomial  $f_p = \prod_{i \in [n]} (1 - a_{p,i} x_i)$ , similar to (51). Note that  $f_p$  vanishes for all truth assignments that satisfy clause  $C_p^H$ . Similar to (14) and (16), we define the sets

$$\mathcal{H}_x := \left\{ \sum_{p \in [q]} c_p f_p \text{ mod } \mathcal{I} \mid c_p \in \mathbb{R} \ \forall p \in [q] \right\} \subseteq \mathcal{H} := \left\{ \sum_{p \in [q]} g_p f_p \text{ mod } \mathcal{I} \mid g_p \in \mathbb{R}[x] \ \forall p \in [q] \right\}. \tag{52}$$

Let  $\text{SAT} \subseteq \{\pm 1\}^n$  be the set of all truth assignments satisfying the hard clauses (which we assume to be nonempty). From Putinar (1993), it follows that

$$\min_{x \in \text{SAT}} F_\phi^W(x) = \sup\{\lambda \mid F_\phi^W - \lambda \in \mathcal{V} + \mathcal{H}\} \geq \sup\{\lambda \mid F_\phi^W - \lambda \in \mathcal{V}_x + \mathcal{H}_x\}, \tag{53}$$

where “+” denotes the Minkowski sum of sets. We proceed by writing the lower bound in (53) as an explicit SDP for which we introduce the following sets:  $\mathbf{H}^\gamma := \{p \in [q] \mid \gamma \in C_p^H\}$  for  $\gamma \subseteq [n]$ . Set  $\mathbf{H}^\gamma$  contains all  $p$  for which  $f_p$ , when expanded, contains the term  $\pm x^\gamma$ . The sign here is determined by the parity of  $|\gamma \cap I_p^+|$ ; see (2). Additionally, we define as analogue to  $\mathcal{M}_\phi$  (see (26)), the set  $\mathcal{M}_\phi^H$ . This set contains all matrices  $M$  and vectors  $c$  such that  $F_\phi^W - \lambda \equiv \mathbf{x}^\top M \mathbf{x} + \sum_{p \in [q]} c_p f_p \text{ mod } \mathcal{I}$ . It is, therefore, defined as

$$\mathcal{M}_\phi^H := \left\{ (M, c) \in \mathcal{S} \times \mathbb{R}^q \mid \sum_{(\alpha, \beta) \in \mathcal{X}^\gamma} M_{\alpha, \beta} + \sum_{p \in \mathbf{H}^\gamma} (-1)^{|\gamma \cap I_p^+|} c_p = p_\phi^\gamma, \ \forall \gamma \neq \emptyset \text{ s.t. } x^\gamma \in \mathbf{X} \right\}. \tag{54}$$

This allows us to adapt  $\mathbb{P}_\phi$  to weighted partial MAX-SAT as follows:

$$\min \langle I, M \rangle + \sum_{p \in [q]} c_p \text{ s.t. } (M, c) \in \mathcal{M}_\phi^H, \ M \in \mathcal{S}_+. \tag{55}$$

We approximately solve (55) by the PRSM; see Section 8.1. Let us elaborate on how to adapt the monomial bases to (weighted) partial MAX-SAT. We make no distinction between soft and hard clauses for the  $\text{SOS}_p$  basis; see (21). For basis  $\text{SOS}_s^\theta$  (see (23)), we determine the variable weights as  $w(i) := \sum_{\{j \mid i \in C_j\}} w_j + \sum_{\{p \mid i \in C_p^H\}} \bar{w}$  for  $\bar{w}$  the mean of

all soft clause weights  $w_j$ . For basis  $SOS_p^Q$ , we add all  $\binom{Q}{2}$  quadratic terms of the  $Q$  variables that attain the highest value of  $w(i)$ . For unweighted partial MAX-SAT instances, we consider all  $w_j$  to equal one.

### 8.1. The PRSM for SOS Relaxations of Weighted Partial MAX-SAT

We show here how to solve (55) by the PRSM. We first rewrite (55) by introducing the matrix variable  $Z$  (see also (34)):

$$\min \langle I, M \rangle + \sum_{p \in [q]} c_p \quad \text{s.t.} \quad M = Z, (M, c) \in \mathcal{M}_\phi^H, Z \in \mathcal{S}_+. \quad (56)$$

Then, the augmented Lagrangian function of (56) w.r.t.  $Z=M$  and for a penalty parameter  $\beta > 0$  is  $L_\beta(Z, M, S, c) := L_\beta(Z, M, S) + \mathbf{1}^\top c$ ; see (41). The PRSM is iteratively and separately optimizing over  $(M, c) \in \mathcal{M}_\phi^H$  and  $Z \in \mathcal{S}_+$  and updating  $S$  twice per cycle, similarly to (42). However, in this case, the  $M$ -subproblem from (42) is replaced by the  $(M, c)$ -subproblem.

We now show that minimization over  $(M, c) \in \mathcal{M}_\phi^H$  can be performed efficiently. By derivations similar to (43), we have

$$\arg \min_{(M, c) \in \mathcal{M}_\phi^H} L_\beta(Z, M, S, c) = \arg \min_{(M, c) \in \mathcal{M}_\phi^H} \|M - X\|_F^2 + \frac{2}{\beta} \mathbf{1}^\top c, \quad (57)$$

where  $X := Z - (S + I)/\beta$ . This is a convex quadratic program (QP) that we solve in two steps. First, consider the matrix entries  $M_{\alpha, \beta}$  with  $(\alpha, \beta) \in \mathcal{X}'$  (see (24)) and  $\mathbf{H}' = \emptyset$ . Because  $\mathbf{H}' = \emptyset$ , these entries are unaffected by the  $c_p$  variables. This implies that these  $M_{\alpha, \beta}$  variables are not coupled with the other entries of  $M$ , and one can minimize separately over such  $M_{\alpha, \beta}$ . This separate minimization problem can be solved by applying Lemma 5. Second, the remaining QP

$$\min \sum_{\{\gamma \mid \mathbf{H}' \neq \emptyset\}} \sum_{(\alpha, \beta) \in \mathcal{X}'} (M_{\alpha, \beta} - X_{\alpha, \beta})^2 + \frac{2}{\beta} \mathbf{1}^\top c, \quad (58)$$

can be simplified by the following observation. If  $M^*$  is an optimal solution to (57), then  $(\alpha, \beta), (\alpha', \beta') \in \mathcal{X}' \Rightarrow M_{\alpha, \beta}^* - X_{\alpha, \beta} = M_{\alpha', \beta'}^* - X_{\alpha', \beta'}$ . Hence, (58) can be simplified by substituting each term  $\sum_{(\alpha, \beta) \in \mathcal{X}'} (M_{\alpha, \beta} - X_{\alpha, \beta})^2$  with a single squared variable. We solve the resulting QP either by solving the Karush–Kuhn–Tucker conditions using the lower-upper decomposition or via MOSEK ApS (2023). The solving method depends on the underlying QP.

### 8.2. Strengthening the Bounds

We demonstrate a simple technique for improving the upper bounds given by Program (55). This technique is based on the SAT resolution rule, which is given as follows. For two hard clauses of some proposition  $\phi$ , on literals  $x, z_i, i \in [s]$  and  $y_i, i \in [t]$ , construct the clause below the horizontal line:

$$\frac{[x \vee z_1 \vee \dots \vee z_s], [\neg x \vee y_1 \vee \dots \vee y_t]}{z_1 \vee \dots \vee z_s \vee y_1 \vee \dots \vee y_t}. \quad (59)$$

In contrast to the MAX-SAT resolution rule (29), the SAT resolution rule states that one may add the clause below the horizontal line to  $\phi$  without changing its (un)satisfiability (we say that the new clause is implied by the original two clauses). We may apply this SAT resolution rule to the hard clauses of a partial MAX-SAT instance to generate more hard clauses. As each new clause induces a new variable  $c_p$ , the bound of Program (55) can only improve. One may also regard SAT resolution as extending the set  $\mathcal{H}_x$  (see (52)), by including terms of the form  $c_p x^\alpha f_p$  for some  $\alpha \subseteq [n]$ , where  $c_p \in \mathbb{R}$ .

Additionally, SAT resolution can generate hard unit clauses. This is advantageous because hard unit clauses reduce the number of variables in MAX-SAT; see Section 1.2.

### 8.3. Duality in Partial MAX-SAT

Now, we consider a partial MAX-SAT with only hard clauses. Solving such instances is, thus, equivalent to determining the satisfiability of the given hard clauses. We show that, by taking the dual of the resulting SOS program, one obtains (a stronger version of) the relaxations of Anjos (2004a) given by (40).

We define, for  $A \in \mathcal{S}^n$ ,  $\text{vec}(A) \in \mathbb{R}^{n^2}$  the vector whose entries are the columns of  $A$  stacked together. We start from Program (55) and perform variable splitting on  $M$ , similar to (34). We take the dual  $g(S)$  of this formulation, similar

to (35), and consider the problem

$$\max_S g(S) = \max_{S \in \mathcal{X}_\phi \cap \mathcal{S}_+} \min_{(M, c) \in \mathcal{M}_\phi^H} \langle S, -M \rangle + \mathbf{1}^\top c, \quad (60)$$

for  $\mathcal{X}_\phi$  as in (32). The steps that show  $S \in \mathcal{X}_\phi \cap \mathcal{S}_+$  is necessary for the expression to be finite are provided in the proof of Theorem 2. We rewrite the inner minimization problem in (60) as

$$\min_{(M, c) \in \mathcal{M}_\phi^H} \begin{bmatrix} \text{vec}(S) \\ \mathbf{1} \end{bmatrix}^\top \begin{bmatrix} \text{vec}(-M) \\ c \end{bmatrix}, \quad (61)$$

and proceed to show under which conditions this value is bounded. Observe that the coefficients  $p_\phi^\gamma = 0$  (see (54)) because there are no soft clauses. Moreover, the set  $\mathcal{M}_\phi^H$  places only linear constraints on the entries of  $M$  and  $c$ . Therefore, there exists a matrix  $D$  that satisfies

$$(M, c) \in \mathcal{M}_\phi^H \Leftrightarrow D \begin{bmatrix} \text{vec}(-M) \\ c \end{bmatrix} = \mathbf{0}. \quad (62)$$

Hence, (61) is bounded if and only if  $[\text{vec}(S)^\top \mathbf{1}^\top]$  is contained in the row space of  $D$ . This is precisely the requirement that  $v^{\text{SDP}}(S, C_p^H) = 1, \forall p \in [q]$  as in (40). We provide one example of this claim in Online Appendix D.

Thus, (61) is bounded if  $[\text{vec}(S)^\top \mathbf{1}^\top] \in \text{row}(D)$ , in which case, the value equals zero. Hence, Program (60) is equivalent to (40).

## 9. Conclusions and Future Work

In this paper, we consider SOS optimization for solving MAX-SAT and weighted partial MAX-SAT. We design an SOS-SDP-based exact MAX-SAT solver called SOS-MS. Our solver is competitive with the best known solvers on solving various (weighted partial) MAX-SAT instances. We are also first to compute SDP bounds for weighted partial MAX-SAT.

In Section 3, we propose a family of semidefinite feasibility problems  $R_{\mathcal{F}}(\phi)$  and show that one member of this family provides the rank-two guarantee; see Theorem 1. That is, the existence of a feasible rank-two matrix implies satisfiability of the corresponding SAT instance. In Section 4, we outline the SOS approach to MAX-SAT from van Maaren et al. (2008) and propose new bases. We introduce the  $\text{SOS}_s^\theta$  and  $\text{SOS}_p^Q$  bases (see (23)) and provide several theoretical results related to these bases; see Lemmas 1 and 2. Clearly, the strength of the SOS-SDP-based relaxations and the required time to compute them depend on the chosen monomial basis. The SOS-SDP relaxation for MAX-SAT is denoted by  $P_\phi$ . We consider MAX-SAT resolution in Section 5 and show that resolution might not be beneficial for the SOS approach applied to MAX-SAT.

In Section 6, we elegantly show a connection between the SOS approach to MAX-SAT and the family of semidefinite feasibility problems  $R_{\mathcal{F}}(\phi)$ . This is done by deriving the dual problem to  $P_\phi$ ; see Theorem 2. In Section 7, we propose the PRSM for solving  $P_\phi$ . We show that PRSM is well-suited for exploiting the structure of  $P_\phi$ , in particular, the unit constraints; see (27). We, thus, provide an affirmative answer to the key question posed by van Maaren et al. (2008) regarding whether SDP software can be developed dealing with unit constraints efficiently.

We extend the SOS approach for MAX-SAT to weighted partial MAX-SAT in Section 8. Here, the variables are restricted to satisfy a set of hard clauses. We show that such hard clauses can be incorporated in the SOS program  $P_\phi$  by adding scalar variables. We show in Section 8.1 that the resulting Program (55) is also well-suited for the PRSM.

In Online Appendix E, we provide implementation details of our SOS-SDP-based MAX-SAT solver, whose pseudocode is given in Online Algorithm 1. SOS-MS is a B&B algorithm and has two crucial components. The first one is the use of warm starts to program  $P_\phi$ , in order to quickly obtain strong bounds. The second one is its ability to quickly parse  $P_\phi$  as outlined in Online Appendix E.1. Our algorithm parses a basis that contains 4,439,449 monomials in (approximately) one second (!).

In Online Appendix F, we provide extensive numerical results that verify efficiency of our exact solver SOS-MS and quality of SOS upper bounds. We show that SOS-MS can solve a variety of MAX-SAT instances in reasonable time, solving some instances faster than the best solvers in the MSE-2016. We show that the  $\text{SOS}_p^Q$  bases (22) are able to prove optimality of some MAX-SAT instances and the parameter  $Q$  provides the option to adjust the trade-off between quality of the bounds and computation time. We also test our B&B algorithm for (weighted) partial MAX-SAT instances in Online Appendices F.2 and F.4. Our solver is able to solve many (weighted) partial MAX-SAT instances in a reasonable time.

This paper demonstrates the strong performance of SOS-MS on (weighted partial) MAX-SAT instances from the MSE random track. In the future, we hope to also solve instances with SOS-MS from the so-called industrial and crafted tracks. These tracks currently impose two challenges on SOS-MS. First, these instances induce prohibitively large  $SOS_p$  bases, which hinders the computation of strong bounds. To solve this, we require a more sophisticated method for choosing a smaller, manageable, basis, such as  $SOS_s^\theta$ . Second, these instances can possess clauses of length  $k$ , where  $k \geq 4$ . This is problematic in the current settings because  $F_\phi$  (see (13)) is a  $k$ th degree polynomial, which requires a large basis to be represented. One possible way to overcome these challenges is through exploiting the structure present in these instances. For example, function  $F_\phi$  might have few nonzero coefficients, which allows for finding SOS decompositions with small monomial bases, using methods proposed in Wang et al. (2021); see also Ahmadi et al. (2017).

## Endnotes

<sup>1</sup> Our PRSM implementation is available at [https://github.com/LMSinjorgo/SOS-SDP\\_MAXSAT](https://github.com/LMSinjorgo/SOS-SDP_MAXSAT).

<sup>2</sup> Program (39) can also be directly solved with the PRSM as projecting onto  $\mathcal{X}_\phi$  is computationally cheap.

## References

- Abramé A, Habet D (2014) Ahmaxsat: Description and evaluation of a branch and bound Max-SAT solver. *J. Satisfiability Boolean Model. Comput.* 9(1):89–128.
- Ahmadi AA, Hall G, Papachristodoulou A, Saunderson J, Zheng Y (2017) Improving efficiency and scalability of sum of squares optimization: Recent advances and limitations. Astolfi A, chairman. *2017 IEEE 56th Annual Conf. Decision Control* (IEEE, Piscataway, NJ), 453–462.
- Anjos MF (2004a) On semidefinite programming relaxations for the satisfiability problem. *Math. Methods Oper. Res.* 60(3):349–367.
- Anjos MF (2004b) Proofs of unsatisfiability via semidefinite programming. Ahr D, Fahrion R, Oswald M, Reinelt G, eds. *Operations Research Proceedings 2003* (Springer-Verlag, Berlin).
- Anjos MF (2005) An improved semidefinite programming relaxation for the satisfiability problem. *Math. Programming* 102(3):589–608.
- Anjos MF (2006) Semidefinite optimization approaches for satisfiability and maximum-satisfiability problems. *J. Satisfiability Boolean Model. Comput.* 1(1):1–47.
- Anjos MF (2007) An extended semidefinite relaxation for satisfiability. *J. Satisfiability Boolean Model. Comput.* 4(1):15–31.
- Asín Achá R, Nieuwenhuis R (2014) Curriculum-based course timetabling with SAT and MaxSAT. *Ann. Oper. Res.* 218(1):71–91.
- Aspvall B, Plass MF, Tarjan RE (1979) A linear-time algorithm for testing the truth of certain quantified Boolean formulas. *Inform. Processing Lett.* 8(3):121–123.
- Bonet ML, Levy J, Manyá F (2007) Resolution for Max-SAT. *Artificial Intelligence* 171(8–9):606–618.
- Boyd S, Parikh N, Chu E, Peleato B, Eckstein J (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations Trends Machine Learning* 3(1):1–122.
- Cook SA (1971) The complexity of theorem-proving procedures. Lewis PM, chairman. *Proc. Third Annual ACM Sympos. Theory Comput.* (Association for Computing Machinery, New York), 151–158.
- de Klerk E, van Maaren H, Warners J (2000) Relaxations of the satisfiability problem using semidefinite programming. *J. Automatic Reasoning* 24(1):37–65.
- de Meijer F, Sotirov R (2021) SDP-based bounds for the quadratic cycle cover problem via cutting-plane augmented Lagrangian methods and reinforcement learning: Inform's journal on computing meritorious paper awardee. *INFORMS J. Comput.* 33(4):1262–1276.
- Drusvyatskiy D, Li G, Wolkowicz H (2017) A note on alternating projections for ill-posed semidefinite feasibility problems. *Math. Programming* 162(1):537–548.
- Gabay D, Mercier B (1976) A dual algorithm for the solution of nonlinear variational problems via finite element approximations. *Comput. Math. Appl.* 2(7):17–40.
- Gattermann P, Großmann P, Nachtigall K, Schöbel A (2016) Integrating passengers' routes in periodic timetabling: A SAT approach. Goerigk M, Werneck RF, eds. *16th Workshop Algorithmic Approaches Transportation Model. Optim. Systems* (Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Wadern, Germany), 3:1–3:15.
- Goemans MX, Williamson DP (1995) Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM* 42(6):1115–1145.
- Graham N, Hu H, Im J, Li X, Wolkowicz H (2022) A restricted dual Peaceman-Rachford splitting method for a strengthened DNN relaxation for QAP. *INFORMS J. Comput.* 34(4):2125–2143.
- Halperin E, Zwick U (2001) Approximation algorithms for MAX 4-SAT and rounding procedures for semidefinite programs. *J. Algorithms* 40(2):184–211.
- Håstad J (2001) Some optimal inapproximability results. *J. ACM* 48(4):798–859.
- He B, Ma F, Yuan X (2016) Convergence study on the symmetric version of ADMM with larger step sizes. *SIAM J. Imaging Sci.* 9(3):1467–1501.
- Henrion D, Malick J (2011) Projection methods for conic feasibility problems: Applications to polynomial sum-of-squares decompositions. *Optim. Methods Software* 26(1):23–46.
- Karloff H, Zwick U (1997) A 7/8-approximation algorithm for MAX 3SAT? Torwick I, ed. *Proc. 38th Annual Sympos. Foundations Comput. Sci.* (IEEE, Piscataway, NJ), 406–415.
- Kautz H, Selman B (1999) Unifying SAT-based and graph-based planning. Dean T, ed. *Proc. Sixteenth Internat. Joint Conf. Artificial Intelligence—Volume 1* (Morgan Kaufmann Publishers Inc., San Francisco), 318–325.
- Lasserre JB (2001) Global optimization with polynomials and the problem of moments. *SIAM J. Optim.* 11(3):796–817.
- Lasserre JB (2007) A sum of squares approximation of nonnegative polynomials. *SIAM Rev.* 49(4):651–669.

- Laurent M (2009) Sums of squares, moment matrices and optimization over polynomials. Putinar M, Sullivant S, eds. *Emerging Applications of Algebraic Geometry* (Springer, New York), 157–270.
- Lewin M, Livnat D, Zwick U (2002) Improved rounding techniques for the MAX 2-SAT and MAX DI-CUT problems. Cook WJ, Schulz AS, eds. *Internat. Conf. Integer Programming Combin. Optim.* (Springer), 67–82.
- Li CM, Manyá F (2021) MaxSAT, hard and soft constraints. Biere A, Heule M, van Maaren H, Walsh T, eds. *Handbook of Satisfiability*, Frontiers in Artificial Intelligence and Applications, vol. 336 (IOS Press, Amsterdam), 903–927.
- Marques-Silva JP, Sakallah KA (2000) Boolean satisfiability in electronic design automation. De Micheli G, chairman. *Proc. 37th Annual Design Automation Conf.* (Association for Computing Machinery, New York), 675–680.
- Mendonca M, Wąsowski A, Czarnecki K (2009) SAT-based analysis of feature models is easy. McGregor JD, Muthig D, eds. *Proc. 13th Internat. Software Product Line Conf.* (Carnegie Mellon University, Pittsburgh), 231–240.
- MOSEK ApS (2023) The MOSEK optimization toolbox for MATLAB manual, Version 10.0.20. Accessed January 2023, <https://docs.mosek.com/10.0/toolbox/index.html>.
- Oliveira DE, Wolkowicz H, Xu Y (2018) ADMM for the SDP relaxation of the QAP. *Math. Programming Comput.* 10(4):631–658.
- Parrilo PA, Thomas RR, eds. (2020) Sum of Squares: Theory and Applications. *Proc. Sympos. Appl. Math.*, vol. 77 (American Mathematical Society, Providence, RI).
- Peaceman DW, Rachford HH Jr (1955) The numerical solution of parabolic and elliptic differential equations. *J. Soc. Indust. Appl. Math.* 3(1):28–41.
- Prasad MR, Biere A, Gupta A (2005) A survey of recent advances in SAT-based formal verification. *Internat. J. Software Tools Tech. Transfer* 7(2):156–173.
- Putinar M (1993) Positive polynomials on compact semi-algebraic sets. *Indiana Univ. Math. J.* 42(3):969–984.
- Scheiderer C (2009) Positivity and sums of squares: A guide to recent results. Putinar M, Sullivant S, eds. *Emerging Applications of Algebraic Geometry* (Springer, New York), 271–324.
- van Maaren H, van Norden L (2005) Sums of squares, satisfiability and maximum satisfiability. Bacchus F, Walsh T, eds. *Internat. Conf. Theory Appl. Satisfiability Testing* (Springer, New York), 294–308.
- van Maaren H, van Norden L, Heule MJ (2008) Sums of squares based approximation algorithms for MAX-SAT. *Discrete Appl. Math.* 156(10):1754–1779.
- Wang J, Magron V, Lasserre JB (2021) Chordal-TSSOS: A moment-SOS hierarchy that exploits term sparsity with chordal extension. *SIAM J. Optim.* 31(1):114–141.
- Wang PW, Zico Kolter J (2019) Low-rank semidefinite programming for the MAX2SAT problem. Van Hentenryck P, Zhou Z-H, program chairs. *Proc. Conf. AAAI Artificial Intelligence*, vol. 33 (AAAI Press, Palo Alto, CA), 1641–1649.
- Zheng Y, Fantuzzi G, Papachristodoulou A (2017) Exploiting sparsity in the coefficient matching conditions in sum-of-squares programming using ADMM. *IEEE Control Systems Lett.* 1(1):80–85.