Math. Program., Ser. A (2019) 173:281–312
https://doi.org/10.1007/s10107-017-1221-9

FULL LENGTH PAPER

Affine reductions for LPs and SDPs

Gábor Braun[1] · Sebastian Pokutta[1] · Daniel Zink[1]

Received: 29 December 2015 / Accepted: 18 December 2017 / Published online: 16 January 2018

Abstract We define a reduction mechanism for LP and SDP formulations that degrades approximation factors in a controlled fashion. Our reduction mechanism is a minor restriction of classical hardness reductions requiring an additional independence assumption and it allows for reusing many hardness reductions that have been used to show inapproximability in the context of PCP theorems. As a consequence we establish strong linear programming inapproximability (for LPs with a polynomial number of constraints) for many problems. In particular we obtain a $\frac{3}{2} - \varepsilon$ inapproximability for VertexCover answering an open question in Chan et al. (Proceedings of FOCS, pp. 350–359, 2013, https://doi.org/10.1109/FOCS.2013.45) and prove an inapproximability factor of $\frac{1}{2} + \varepsilon$ for bounded degree IndependentSet . In the case of SDPs, we obtain inapproximability results for these problems relative to the SDP-inapproximability of MaxCUT. Moreover, using our reduction framework we are able to reproduce various results for CSPs from Chan et al. (Proceedings of FOCS, pp. 350–359, 2013, https://doi.org/10.1109/FOCS.2013.45) via simple reductions from Max-2-XOR.

Mathematics Subject Classification 90C22 · 68Q17 · 05C70

✉ Sebastian Pokutta
sebastian.pokutta@isye.gatech.edu

Gábor Braun
gabor.braun@isye.gatech.edu

Daniel Zink
daniel.zink@gatech.edu

1 ISyE, Georgia Institute of Technology, Atlanta, GA, USA

1 Introduction

Linear Programming (LP) and Semidefinite Programming (SDP) formulations are ubiquitous in combinatorial optimization problems and are often an important ingredient in the construction of approximation algorithms. The theory of extended formulations studies the size of such LP or SDP formulations. So far most strong lower bounds have been obtained by ad-hoc analyses (three notable exceptions are [16,37,38] that reuse Sherali–Adams gap instances in the LP case and Lasserre gap instances in the SDP case). This is in stark contrast to the computational complexity world, where we often resort to reduction mechanisms to establish hardness.

In this work we establish an affine reduction mechanism for approximate LP and SDP formulations. Having a reduction mechanism in place has the following main appeals.

(i) The reduction mechanism allows for the propagation of hardness results across optimization problems very similar in spirit to the computational complexity approach. In particular, no specific knowledge of how the hardness of the base problem has been established is required. As such it turns establishing LP/SDP hardness into a routine task for many problems allowing a much broader community to benefit from results from extended formulations.

(ii) Any future improvements to the strength of the lower bounds of the base problems immediately propagate to all problems that one can reduce to. In fact, between the extended abstract of this work and the current version there have been several results strengthening the base problems, e.g., the lower bounds in [16] were significantly strengthened in [37].

Our reduction mechanism is compatible with numerous known reductions in the literature that have been used to analyze computational complexity, so that we can reuse these reductions in the context of LPs and SDPs. As the approach is very similar in all cases, i.e., to verify that the additional conditions of our mechanism are met, we will only state a select few that are of specific interest. We obtain new LP inapproximability results for problems such as, e.g., VertexCover and bounded degree IndependentSet which are not 0/1 CSPs and hence not captured by the approach in [16]. We also obtain the same inapproximability bounds for CSPs as established earlier in [16], however by means of a direct reductions from Max-2-XOR. Thus in some sense Max-2-XOR could be considered the actual driver of complexity for most of these problems.

Our reduction mechanism is based on the abstract view of extended formulations from [16] to capture linear programming formulations independent of the specific linear encoding introduced in the context of CSPs where the feasible region is the whole 0/1 cube. For completeness we present its natural generalization to arbitrary combinatorial optimization problems in Sect. 2 as this model will form the basis for the reductions. We note that while this model is essentially equivalent to previous approaches studying approximations via polyhedral pairs (see [6,8]), it removes reliance on a specific polyhedral representation. This perspective significantly simplifies the treatment of approximate LP/SDP formulations and it enables the formulation of the reduction mechanism. We set out to answer questions of the form:

Given a combinatorial optimization problem, what is the smallest size of any of its LPs or SDPs?

Contribution

Our main contribution is a reduction mechanism for LPs and SDPs that allows us to transfer hardness results between problems. The reduction mechanism is compatible with many computational complexity reductions from the literature and as such these reductions can be reused to establish hardness of approximation in terms of small LPs and SDPs for a wide range of combinatorial optimization problems. We stress that all results are *independent* of P versus NP (or P/poly vs. NP) and pertain to expressing combinatorial problems with LPs or SDPs.

Reduction mechanism We provide a purely combinatorial and conceptually simple framework for reductions (similar to L-reductions) of optimization problems in the context of LPs and SDPs, where approximations are inherent without the need of any polyhedra as compared to, e.g., [6,8]. We believe that this model is the right approach for capturing reductions for LPs and SDPs. We also stress that a reduction in this model can be applied to *any* other conic programming paradigm (such as e.g., copositive programming, or hyperbolic programming) however the hardness of the base problem needs to be adjusted.

As such a reduction between problems for LPs immediately provides a reduction between the same problems for SDPs etc. Many reductions in the context of PCP inapproximability are compatible with our mechanism and hence can be reused. Contrasting the above, so far LP and SDP inapproximability results have been only obtained for very restricted classes of problems (see [6,8,16,37,38]) requiring either a case-by-case analysis or are hierarchy-based (requiring rather involved hierarchy lower bounds). We stress though that not automatically all reductions can be reused. For example the hardness of approximation reductions for TSP in [21, §2] and [34, §6] cannot be reused as they translate feasible solutions depending on the objective functions.

LP inapproximability and conditional SDP inapproximability of specific problems Our reduction mechanism opens up the possibility to reuse previous hardness results to establish inapproximability of problems. As a case in point, we establish the first LP inapproximability result for VertexCover and IndependentSet as well as reproduce the results in [16], all by simple and direct reductions from the LP-inapproximability of Max-2-XOR within a factor better than $\frac{1}{2}$ established in [16]. In particular, we answer an open question regarding the inapproximability of VertexCover (see [16]). In the SDP case we present two sets of results. The first set uses the MaxCUT SDP-inapproximability of 15/16 from [13] as a base hard problem and the second set is formulated under the assumption that the Goemans–Williams SDP for MaxCUT is optimal (see Conjecture 4.9), which is compatible with the Unique Games Conjecture. We summarize the obtained hardness results in Tables 1 and 2.

Table 1 Summary of LP inapproximability and LP approximability factors for some of the problems considered in this paper

Problem	Inapproximability		Approximability
	(LP)	(PCP)	(LP)
(Bounded degree) VertexCover (new)	$\frac{3}{2} - \varepsilon$	$1.361 - \varepsilon$	2
Min-2-CNFDeletion (new)	$\omega(1)$	$2.889 - \varepsilon$	–
MinUnCUT (new)	$\omega(1)$	$C_{\text{uncut}} - \varepsilon$	–
Max-MULTI-k-CUT (new)	$\frac{2c(k)+1}{2c(k)+2} + \varepsilon$	$1 - \frac{1}{34k} + \varepsilon$	$\frac{1}{2(1-1/k)}$
(Bounded degree) IndependentSet (new)	$\frac{1}{2} + \varepsilon$	$O\left(\frac{\log^4 \Delta}{\Delta}\right)$	–
Max-2-SAT	$\frac{3}{4} + \varepsilon$ (optimal)	$\frac{21}{22} + \varepsilon$	$\frac{3}{4}$
Max-3-SAT	$\frac{3}{4} + \varepsilon$	$\frac{7}{8} + \varepsilon$	$\frac{19}{27}$
Max-DICUT	$\frac{1}{2} + \varepsilon$ (optimal)	$\frac{12}{13} + \varepsilon$	$\frac{1}{2}$
Max-2-CSP	$\frac{1}{2} + \varepsilon$		
Max-2-CONJSAT	$\frac{1}{2} + \varepsilon$ (optimal)	$\frac{9}{10} + \varepsilon$	$\frac{1}{2}$

The LP inapproximability factor cannot be achieved with LPs of size less than $2^{n^{c(\varepsilon)}}$ where n is the underlying size of the considered problem and $c(\varepsilon)$ is a constant only depending on ε however not on n and $c(k)$ is a constant depending on k defined in Sect. 6.1

Table 2 Summary of SDP hardness results

Problem	SDP Inapproximability with size	
	$n^{\Omega(\log n/\log\log n)}$	Superpolynomially (under Conjecture 4.9)
VertexCover (new)	$\frac{25}{24} - \varepsilon$	$1.12144 - \varepsilon$
Min-2-CNFDeletion (new)	$\frac{5}{4} - \varepsilon$	$\omega(1)$
MinUnCUT (new)	$\frac{5}{4} - \varepsilon$	$\omega(1)$
Max-MULTI-k-CUT (new)	$\frac{20c(k)+15}{20c(k)+16}$	$\frac{c(k)+c_{GW}}{c(k)+1}$
IndependentSet (new)	$\frac{15}{16} + \varepsilon$	$c_{GW} + \varepsilon$
Max-2-SAT (new)	$\frac{35}{36} + \varepsilon$	$\frac{1+c_{GW}}{2} + \varepsilon \approx 0.93928 + \varepsilon$
Max-3-SAT	$\frac{35}{36} + \varepsilon$	$\frac{1+c_{GW}}{2} + \varepsilon \approx 0.93928 + \varepsilon$
Max-DICUT	$\frac{15}{16} + \varepsilon$	$c_{GW} + \varepsilon$
Max-2-CSP	$\frac{15}{16} + \varepsilon$	$c_{GW} + \varepsilon$
Max-2-CONJSAT	$\frac{15}{16} + \varepsilon$	$c_{GW} + \varepsilon$

We report the inapproximability factor which cannot be achieved by an SDP smaller than the size given in the column. A factor of 1 means that there is no exact representation of the problem as an SDP of that size. Here $c_{GW} \approx 0.87856$ is the approximation factor of the algorithm for MaxCUT from [27], and $c(k)$ is a constant depending on k defined in Sect. 6.1

Note that our conditional SDP inapproximability of Max-3-SAT within $(1 + c_{GW})/2 + \varepsilon \approx 0.93928 + \varepsilon$ is better than the current best unconditional inapproximability of 7/8 from [38, Theorem 1.5]. In a similar vein a simple reduction from Max-2-XOR provides an unconditional SDP inapproximability of Max-2-SAT within $\frac{35}{36} + \varepsilon$; to the best of the authors' knowledge no Lasserre gap instance is known for Max-2-SAT, so that [38] cannot be applied; see Sect. 6 for details. We would also like to mention that the affine reductions are compatible with symmetric LPs and SDPs reducing potentially from [11]; the details are left to the interested reader. Finally, we provide a construction to turn perturbations of slack matrices into approximate LPs and SDPs in Theorem 7.2.

Notions of LP and SDP rank The key element in the analysis of extended formulations is Yannakakis's celebrated Factorization Theorem (see [48,49]) and its generalizations (see e.g., [6,8,16,29]) equating the minimal size of an extended formulation with a property of a slack matrix, e.g., in the linear case the nonnegative rank. Rephrasing [16], we provide an abstract factorization theorem (Theorem 3.5) for combinatorial optimization problems *and their approximations* together with new notions of LP rank and SDP rank. The factorization theorem for polyhedral pairs (see [6,8,40]) states that the nonnegative rank and extension complexity might differ by 1, which was slightly elusive. Using the notion of LP rank we clarify in Theorem 3.5 that this 1 is the difference between the LP rank and nonnegative rank, i.e., formulation complexity is a *property of the slack matrix*; similar remarks apply to the SDP case. We stress however, that the abstract view on extended formulations should be primarily considered an enabler and a prerequisite for the reduction mechanism and the analysis of approximate LP/SDP formulations.

Follow-up work

An extended abstract of this article has appeared as [10] and the current version includes full proofs, additional examples, and material providing a more complete picture. Also, since the extended abstract there has been significant follow-up work strengthening hardness results for the base hard problems, and through the reductions these directly translate to significantly improved inapproximability results for many other problems. The current version incorporates these recent developments. Moreover, as more follow-up is expected in terms of strengthening and providing new base problems, we formulate all hardness results *relative to the hardness of the base problems* and we will report the currently best known bound for the base hard problems.

We briefly discuss follow-up work. In [13] the reductions that we present here have been extended to fractional programming and general proof systems. This allows for proving hardness of approximation results for problems such as sparsest cut and it is also shown that MaxCUT cannot be approximated by small SDPs within a factor better than $\frac{15}{16}$ (compared to $\frac{16}{17}$ assuming $P \neq NP$), which we use as the unconditional base hard problem for the SDP case in addition to the conditional SDP hardness of MaxCUT assuming the Unique Games Conjecture. In [3] our reductions have been used to strengthen our VertexCover result showing that no small LP can approximate VertexCover within a factor better than 2 reducing from Unique Games instead of

MaxCUT; note that Unique Games is unconditionally hard to approximate by small LPs within any constant factor. In [28] it was shown that there exist graphs on n vertices, so that their stable set polytopes require LPs of size at least $2^{\Omega(n/\log n)}$ and in [37] the lower bound $n^{\Omega(\log n/\log\log n)}$ of [16] was strengthened to $2^{\Omega(n^{\delta})}$ (see Theorem 4.6) for LPs approximating MaxCUT within a factor better than $\frac{1}{2}$, which we will use as LP base hard problem.

2 Optimization problems

We intend to study the required size of a linear program or semidefinite program capturing a combinatorial optimization problem with specified approximation guarantees. In our context an optimization problem is defined as follows.

Definition 2.1 (*Optimization problems*) An *optimization problem* $\mathcal{P} = (\mathcal{S}, \mathcal{F}, \mathrm{val})$ consists of a set $\mathcal{S}$ of *feasible solutions* and a set $\mathcal{F}$ of *instances*, together with a real-valued objective function $\mathrm{val}\colon \mathcal{S} \times \mathcal{F} \to \mathbb{R}$.

A wide class of examples consists of constraint satisfaction problems (CSPs):

Definition 2.2 (*Maximum Constraint Satisfaction Problem* (*CSP*)) A *constraint family* $\mathcal{C} = \{C_1, \ldots, C_m\}$ on the boolean variables $x_1, \ldots, x_n$ is a family of boolean functions C_i in $x_1, \ldots, x_n$. The C_i are *constraints* or *clauses*. The problem $\mathcal{P}(\mathcal{C})$ corresponding to a constraint family $\mathcal{C}$ has

(i) **feasible solutions** all 0/1 assignments s to $x_1, \ldots, x_n$;
(ii) **instances** all nonnegative weightings $w_1, \ldots, w_m$ of the constraints $C_1, \ldots, C_m$
(iii) **objective function** the weighted sum of satisfied constraints: $\mathrm{val}_{w_1,\ldots,w_m}(s) = \sum_i w_i C_i(s)$.

The goal is to maximize the weights of satisfied constraints, in particular CSPs are maximization problems. A *maximum Constraint Satisfaction Problem* is an optimization problem $\mathcal{P}(\mathcal{C})$ for some constraint family $\mathcal{C}$. A k-CSP is a CSP where every constraint depends on at most k variables.

For brevity, we shall simply use CSP for a maximum CSP, when there is no danger of confusion with a minimum CSP. In the following, we shall restrict to instances with 0/1 weights, i.e., an instance is a subset $L \subseteq \mathcal{C}$ of constraints, and the objective function computes the number $\mathrm{val}_L(s) = \sum_{C\in L} C(s)$ of constraints in L satisfied by assignment s. Restriction to specific instances clearly does not increase hardness of the problem measured in formulation complexity as defined below.

As a special case, the Max-k-XOR problem restricts to constraints, which are XORs of k literals. Here we shall write the constraints in the equivalent equation form $x_{i_1} \oplus \cdots \oplus x_{i_k} = b$, where $\oplus$ denotes the addition modulo 2.

Definition 2.3 (Max-k-XOR) For fixed k and n, the problem Max-k-XOR is the CSP for variables $x_1, \ldots, x_n$ and the family $\mathcal{C}$ of all constraints of the form $x_{i_1} \oplus \cdots \oplus x_{i_k} = b$ with $1 \le i_1 < \cdots < i_k \le n$ and $b \in \{0, 1\}$.

An even stronger important restriction is MaxCUT, a subproblem of Max-2-XOR as we will see soon. The aim is to determine the maximum size of cuts for all graphs G with $V(G) = [n]$.

Definition 2.4 (MaxCUT) The problem MaxCUT has instances all simple graphs G with vertex set $V(G) = [n]$, and feasible solutions all cuts on $[n]$, i.e., functions $s : [n] \to \{0, 1\}$. The objective function val computes the number of edges $\{i, j\}$ of G cut by the cut, i.e., with $s(i) \neq s(j)$.

The problem MaxCUT_Δ is the subproblem of MaxCUT considering only graphs G with maximum degree at most Δ.

We have $\mathrm{val}_G^{\mathsf{MaxCUT}}(s) = \mathrm{val}_{L(G)}^{\mathsf{Max\text{-}2\text{-}XOR}}(s)$, for the constraint set $L(G) = \left\{x_i \oplus x_j = 1 \mid \{i, j\} \in E(G)\right\}$, realizing MaxCUT as a subproblem of Max-2-XOR with the same feasible solutions.

We are interested in approximately solving an optimization problem $\mathcal{P}$ by means of a linear program or a semidefinite program. Recall that a typical PCP inapproximability result states that it is hard to decide between $\max \mathrm{val}_f \leq S(f)$ and $\max \mathrm{val}_f \geq C(f)$ for a class of instances f and some easy-to compute functions S and C usually refereed to as *soundness* and *completeness*. Here and below $\max \mathrm{val}_f$ denotes the maximum value of the function val_f over the respective set of feasible solutions. We adopt the terminology to linear programs and semidefinite programs. We start with the linear case.

Definition 2.5 (*LP formulation of an optimization problem*) Let $\mathcal{P} = (\mathcal{S}, \mathcal{F}, \mathrm{val})$ be an optimization problem with real-valued functions C, S on $\mathcal{F}$, called *completeness guarantee* and *soundness guarantee*, respectively. If $\mathcal{P}$ is a maximization problem, then let $\mathcal{F}^S := \left\{f \in \mathcal{F} \mid \max \mathrm{val}_f \leq S(f)\right\}$ denote the set of instances, for which the maximum is upper bounded by soundness guarantee S. If $\mathcal{P}$ is a minimization problem, then let $\mathcal{F}^S := \left\{f \in \mathcal{F} \mid \min \mathrm{val}_f \geq S(f)\right\}$ denote the set of instances, for which the minimum is lower bounded by soundness guarantee S.

A (C, S)-*approximate LP formulation* of $\mathcal{P}$ is a linear program $Ax \leq b$ with $x \in \mathbb{R}^d$ together with the following *realizations*:

(i) **Feasible solutions** as vectors $x^s \in \mathbb{R}^d$ for every $s \in \mathcal{S}$ so that

$$Ax^s \leq b \qquad \text{for all } s \in \mathcal{S}, \tag{1}$$

i.e., the system $Ax \leq b$ is a relaxation (superset) of $\mathrm{conv}\, x^s \mid s \in \mathcal{S}$.

(ii) **Instances** as affine functions $w^f : \mathbb{R}^d \to \mathbb{R}$ for every $f \in \mathcal{F}^S$ such that

$$w^f(x^s) = \mathrm{val}_f(s) \qquad \text{for all } s \in \mathcal{S}, \tag{2}$$

i.e., we require that the linearization w^f of val_f is exact on all x^s with $s \in \mathcal{S}$.

(iii) **Achieving guarantee** C via requiring

$$\max\left\{w^f(x) \mid Ax \leq b\right\} \leq C(f) \qquad \text{for all } f \in \mathcal{F}^S, \tag{3}$$

for maximization problems (resp. $\min \left\{w^f(x) \mid Ax \leq b\right\} \geq C(f)$ for minimization problems).

The *size* of the formulation is the number of inequalities in $Ax \leq b$. Finally, the *LP formulation complexity* $\mathrm{fc}_+(\mathcal{P}, C, S)$ of the problem $\mathcal{P}$ is the minimal size of all its LP formulations.

For all instances $f \in \mathcal{F}$ soundness and completeness should satisfy $C(f) \geq S(f)$ in the case of maximization problems and $C(f) \leq S(f)$ in the case of minimization problems in order to capture the notion of relaxations and we assume this condition in the remainder of the paper.

Remark 2.6 We use affine maps instead of linear maps to allow easy shifting of functions. At the cost of an extra dimension and an extra equation, affine functions can be realized as linear functions.

Remark 2.7 (*Inequalities vs. Equations*) Traditionally in extended formulations, one would separate the description into equations and inequalities and one would only count inequalities. In our framework, equations can be eliminated by restricting to the affine space defined by them, and parametrizing it as a vector space. However, note that restricting to linear functions, one might need an equation to represent affine functions by linear functions.

For determining the exact maximum of a maximization problem, one chooses $C(f) = S(f) := \max \mathrm{val}_f$. To show inapproximability within an approximation factor $0 < \rho \leq 1$, one chooses guarantees satisfying $\rho C(f) \geq S(f)$. This choice is motivated to be comparable with factors of approximation algorithms finding a feasible solution s with $\mathrm{val}_f(s) \geq \rho \max \mathrm{val}_f$. For minimization problem, $C(f) = S(f) := \min \mathrm{val}_f$ in the exact case, and $\rho C(f) \leq S(f)$ for an approximation factor $\rho \geq 1$ provided val_f is nonnegative.

We will now adjust Definition 2.5 to the semidefinite case. For symmetric matrices, as customary, we use the Frobenius product as scalar product, i.e., $\langle A, B\rangle = \mathrm{Tr}[AB]$. Recall that the psd-cone is self-dual under this scalar product.

Definition 2.8 (*SDP formulation of an optimization problem*) Let $\mathcal{P} = (\mathcal{S}, \mathcal{F}, \mathrm{val})$ be a maximization problem with real-valued functions C, S on $\mathcal{F}$ and let $\mathcal{F}^S := \left\{f \in \mathcal{F} \mid \max \mathrm{val}_f \leq S(f)\right\}$ as in Definition 2.5.

A (C, S)-approximate *SDP formulation* of $\mathcal{P}$ consists of a linear map $\mathcal{A}\colon \mathbb{S}^d \to \mathbb{R}^k$ and a vector $b \in \mathbb{R}^k$ (defining a semidefinite program $\left\{X \in \mathbb{S}^d_+ \mid \mathcal{A}(X) = b\right\}$). Moreover, we require the following *realizations* of the components of $\mathcal{P}$:

(i) **Feasible solutions** as vectors $X^s \in \mathbb{S}^d_+$ for every $s \in \mathcal{S}$ so that

$$\mathcal{A}(X^s) = b \tag{4}$$

i.e., the system $\mathcal{A}(X) = b$, $X \in \mathbb{S}^d_+$ is a relaxation of $\mathrm{conv}\, X^s [s \in \mathcal{S}]$.

(ii) **Instances** as affine functions $w^f : \mathbb{S}^d \to \mathbb{R}$ for every $f \in \mathcal{F}^S$ with

$$w^f(X^s) = \mathrm{val}_f(s) \qquad \text{for all } s \in \mathcal{S}, \tag{5}$$

i.e., we require that the linearization w^f of val_f is exact on all X^s with $s \in \mathcal{S}$.

(iii) **Achieving guarantee** C via requiring

$$\max\left\{w^f(X) \mid \mathcal{A}(X^s) = b,\ X^s \in \mathbb{S}^d_+\right\} \le C(f) \qquad \text{for all } f \in \mathcal{F}, \tag{6}$$

for maximization problems, and the analogous inequality for minimization problems.

The *size* of the formulation is the parameter d. The *SDP formulation complexity* $\mathrm{fc}_\oplus(\mathcal{P}, C, S)$ of the problem $\mathcal{P}$ is the minimal size of all its SDP formulations.

2.1 Relation to approximate extended formulations

Traditionally in extended formulations, one would start from an initial polyhedral representation of the problem and bound the size of its smallest possible lift in higher-dimensional space. In the linear case for example, the minimal number of required inequalities would constitute the extension complexity of that polyhedral representation. Our notion of *formulation complexity* can be understood as a combinatorial version of extension complexity, formulated independently of any encoding. This independence of encoding addresses previous concerns that the obtained lower bounds are polytope-specific or encoding-specific and alternative linear encodings (i.e., different initial polyhedron) of the same problem might admit smaller formulations. Actually, formulation complexity is equivalent to the extension complexity of the minimal polyhedral encoding, where only the linear combination of the objective functions of the instances are linear. In the literature, fortunately, the minimal encoding is used in almost all cases, as this seems to be the natural encoding. However, other encodings are possibles, e.g., for the matching problem one could consider a simplex where every vertex corresponds to a matching. We will see later in Theorem 3.5 that the standard notion of extension complexity and formulation complexity are essentially equivalent (up to an additive ± 1 due to using a modification of nonnegative rank), however the more abstract perspective simplifies the handling of approximations and reductions as we will see in Sect. 4.

The notion of *LP formulation*, its size, and LP formulation complexity are closely related to *polyhedral pairs* and linear encodings (see [6,8], and also [40]). In particular, given a (C, S)-approximate LP formulation of a maximization problem $\mathcal{P}$ with linear program $Ax \le b$, representations $\{x^s \mid s \in \mathcal{S}\}$ of feasible solutions and $\left\{w^f \mid f \in \mathcal{F}^S\right\}$ of instances, one can define a polyhedral pair encoding $\mathcal{P}$

$$\begin{aligned} P &:= \operatorname{conv} x^s[s \in \mathcal{S}], \\ Q &:= \left\{x \in \mathbb{R}^d \mid \langle w^f, x\rangle \leqslant C(f), \quad \forall f \in \mathcal{F}^S\right\}. \end{aligned}$$

Then for $K := \{x \mid Ax \le b\}$, we have $P \subseteq K \subseteq Q$. Note that there is no need for the approximating polyhedron K to reside in extended space, as P and Q already live there.

Put differently, the LP formulation complexity of $\mathcal{P}$ is the minimum size of an extended formulation over all possible linear encodings of $\mathcal{P}$. The semidefinite case is similar, with the only difference being that K is now a spectrahedron, being represented by a semidefinite program instead of a linear program.

3 Factorization theorem and slack matrix

We provide an algebraic characterization of formulation complexity via the *slack matrix of an optimization problem*, similar in spirit to factorization theorems for extended formulations (see e.g., [6,8,29,48,49]), with a fundamental difference pioneered in [16] that there is no linear system to start from. The linear or semidefinite program is constructed from scratch using a matrix factorization. This also extends [12], by allowing affine functions, and using a modification of nonnegative rank, to show that formulation complexity depends only on the slack matrix.

Definition 3.1 (*Slack matrix of* $\mathcal{P}$) Let $\mathcal{P} = (\mathcal{S}, \mathcal{F}, \mathrm{val})$ be an optimization problem with guarantees C, S. The (C, S)*-approximate slack matrix* of $\mathcal{P}$ is the nonnegative $\mathcal{F}^S \times \mathcal{S}$ matrix M, with entries

$$M(f, s) := \begin{cases} C(f) - \mathrm{val}_f(s) & \text{if } \mathcal{P} \text{ is a maximization problem,} \\ \mathrm{val}_f(s) - C(f) & \text{if } \mathcal{P} \text{ is a minimization problem.} \end{cases}$$

We introduce the *LP factorization* of a nonnegative matrix, which for slack matrices captures the LP formulation complexity of the underlying problem.

Definition 3.2 (*LP factorization of a matrix*) A *size-r LP factorization* of $M \in \mathbb{R}_+^{m\times n}$ is a factorization $M = TU + \mu\mathbb{1}$ where $T \in \mathbb{R}_+^{m\times r}$, $U \in \mathbb{R}_+^{r\times n}$ and $\mu \in \mathbb{R}_+^{m\times 1}$. Here $\mathbb{1}$ is the $1 \times n$ matrix with all entries being 1. The *LP rank* $\mathrm{rank}_{\mathrm{LP}}\, M$ *of* M is the minimum r such that there exists a size-r LP factorization of M.

A size-r LP factorization is equivalent to a decomposition $M = \sum_{i\in[r]} u_i v_i^{\mathsf{T}} + \mu\mathbb{1}$ for some (column) vectors $u_i \in \mathbb{R}_+^m$, $v_i \in \mathbb{R}_+^n$ with $i \in [r]$ and a column vector $\mu \in \mathbb{R}_+^m$. It is a slight modification of a nonnegative matrix factorization, disregarding simultaneous shift of all columns by the same vector, i.e., allowing an additional term $\mu \cdot \mathbb{1}$ *not* contributing to the size, so clearly, $\mathrm{rank}_{\mathrm{LP}} \le \mathrm{rank}_+ M \le \mathrm{rank}_{\mathrm{LP}}\, M + 1$.

One similarly defines SDP factorizations of nonnegative matrices.

Definition 3.3 A *size-r SDP factorization* of $M \in \mathbb{R}_+^{m\times n}$ is a factorization is a collection of matrices $T_1, \ldots, T_m \in \mathbb{S}_+^r$ and $U_1, \ldots, U_n \in \mathbb{S}_+^r$ together with $\mu \in \mathbb{R}_+^{m\times 1}$ so that $M_{ij} = \mathrm{Tr}[T_i U_j] + \mu(i)$. The *SDP rank* $\mathrm{rank}_{\oplus}\, M$ *of* M is the minimum r such that there exists a size-r SDP factorization of M.

For the next theorem, we need the folklore formulation of linear duality using affine functions, see e.g., [44, Corollary 7.1h].

Lemma 3.4 (Affine form of Farkas's Lemma) *Let* $P := \{x \mid A_j x \leq b_j,\ j \in [r]\}$ *be a non-empty polyhedron. An affine function* Φ *is nonnegative on* P *if and only if there are nonnegative multipliers* λ_j, λ_0 *with*

$$\Phi(x) \equiv \lambda_0 + \sum_{j \in [r]} \lambda_j (b_j - A_j x).$$

We are ready for the factorization theorem for optimization problems.

Theorem 3.5 (Factorization theorem for formulation complexity) *Consider an optimization problem* $\mathcal{P} = (\mathcal{S}, \mathcal{F}, \mathrm{val})$ *with* (C, S)*-approximate slack matrix* M. *Then we have*

$$\mathrm{fc}_+(\mathcal{P}, C, S) = \mathrm{rank}_{\mathrm{LP}}\, M, \qquad \textit{and} \qquad \mathrm{fc}_\oplus(\mathcal{P}, C, S) = \mathrm{rank}_{\mathrm{SDP}}\, M.$$

for linear formulations and semidefinite formulations, respectively.

Note also that for the slack matrix of a polytope, every row contains a 0 entry, and hence the $\mu\mathbb{1}$ term in any LP factorization must be 0. Therefore the nonnegative rank and LP rank coincide for polytopes. Similar remarks apply to SDP factorizations.

Proof of Theorem 3.5—the linear case We will confine ourselves to the case of $\mathcal{P}$ being a maximization problem. For minimization problems, the proof is analogous.

To prove $\mathrm{rank}_{\mathrm{LP}}\, M \leq \mathrm{fc}_+(\mathcal{P}, C, S)$, let $Ax \leq b$ be an arbitrary (C, S)-approximate, size-r LP formulation of $\mathcal{P}$, with realizations $\{w^f \mid f \in \mathcal{F}^S\}$ of instances and $\{x^s \mid s \in \mathcal{S}\}$ of feasible solutions. We shall construct a size-r LP factorization of M. As $\max_{x : Ax \leq b} w^f(x) \leq C(f)$ by Condition (3), via the affine form of Farkas's lemma, Lemma 3.4 we have

$$C(f) - w^f(x) = \sum_{j=1}^{r} T(f, j) \left(b_j - \langle A_j, x\rangle\right) + \mu(f)$$

for some nonnegative multipliers $T(f, j), \mu(f) \in \mathbb{R}_+$ with $1 \leq j \leq r$. By taking $x = x^s$, we obtain

$$M(f, s) = \sum_{j=1}^{r} T(f, j) U(j, s) + \mu(f), \quad \text{with} \quad U(j, s) := b_j - \langle A_j, x^s\rangle \quad \text{for } j > 0.$$

i.e., $M = TU + \mu\mathbb{1}$. By construction, T and μ are nonnegative. By Condition (1) we also obtain that U is nonnegative. Therefore $M = TU + \mu\mathbb{1}$ is a size-r LP factorization of M.

For the converse, i.e., $\mathrm{rank}_{\mathrm{LP}} \geq \mathrm{fc}_+(\mathcal{P}, C, S)$, let $M = TU + \mu\mathbb{1}$ be a size-r LP factorization. We shall construct an LP formulation of size r. Let T_f denote the f-row of T for $f \in \mathcal{F}^S$, and U_s denote the s-column of U for $s \in \mathcal{S}$. We claim that the linear system $x \geq 0$ with representations

$$w^f(x) := C(f) - \mu(f) - T_f x \qquad \forall f \in \mathcal{F}^S \qquad \text{and} \qquad x^s := U_s \qquad \forall s \in \mathcal{S}$$

satisfies the requirements of Definition 2.5. Condition (2) is implied by the factorization $M = TU + \mu\mathbb{1}$:

$$w^f(x^s) = C(f) - \mu(f) - T_f U_s = C(f) - M(f, s) = \mathrm{val}_f(s).$$

Moreover, $x^s \geq 0$, because U is nonnegative, so that Condition (1) is fulfilled. Finally, Condition (3) also follows readily:

$$\max\left\{w^f(x) \mid x \geq 0\right\} = \max\left\{C(f) - \mu(f) - T_f x \mid x \geq 0\right\} = C(f) - \mu(f) \leq C(f),$$

as the nonnegativity of T implies $T_f x \geq 0$; equality holds e.g., for $x = 0$. Recall also that $\mu(f) \geq 0$. Thus we have constructed an LP formulation with r inequalities, as claimed. □

Remark 3.6 It is counter-intuitive that 0 is always a maximizer, and, actually, it is an artifact of the construction. At a conceptual level, the polyhedron $Ax \leq b$ containing $\operatorname{conv} x^s \mid s \in \mathcal{S}$ is represented as the intersection of the nonnegative cone with an affine subspace in the slack space. The affine functions w^f are extended to attain their optimum value on this intersection in the nonnegative cone, and thus also at 0, the apex of the cone. In particular, intersecting with the affine subspace is no longer needed. See Fig. 1 for an illustration.

Proof of Theorem 3.5—the semidefinite case As before we confine ourselves to the case of $\mathcal{P}$ being a maximization problem. The proof is analogous to the linear case, but for the sake of completeness, we provide a full proof.

To prove $\mathrm{rank}_{\mathrm{SDP}}\, M \leq \mathrm{fc}_{\oplus}(\mathcal{P}, C, S)$, let $\mathcal{A}(X) = b$, $X \in \mathbb{S}^r_+$ be an arbitrary size-r SDP formulation of $\mathcal{P}$, with realizations $\left\{w^f \mid f \in \mathcal{F}^S\right\}$ of instances and $\{X^s \mid s \in \mathcal{S}\}$ of feasible solutions. To apply strong duality, we may assume that the convex set $\left\{X \in \mathbb{S}^r_+ \mid \mathcal{A}(X) = b\right\}$ has an interior point (i.e., Slater's condition is satisfied), because otherwise it would be contained in a proper face of $\mathbb{S}^r_+$, which is an SDP cone of smaller size, resulting in a smaller SDP factorization.

We shall construct a size-r SDP factorization of M. As $\max_{X \in \mathbb{S}^r_+ :\, \mathcal{A}(X)=b} w^f(X) \leq C(f)$ by Condition (6), via strong duality using Slater's condition, we have

$$C(f) - w^f(X) = \langle T_f, X\rangle + \langle y^f, b - \mathcal{A}(X)\rangle + \lambda_f$$

for all $f \in \mathcal{F}$ with some $T_f \in \mathbb{S}^r_+$, $y^f \in \mathbb{R}^k$ and $\lambda_f \in \mathbb{R}_+$. By substituting X^s into X, we obtain

$$M(f, s) = C(f) - w^f(X^s) = \langle T_f, X^s\rangle + \langle y^f, b - \mathcal{A}(X^s)\rangle + \lambda_f = \langle T_f, X^s\rangle + \lambda_f,$$

which is an SDP factorization of size r.

For the converse, i.e., $\mathrm{fc}_{\oplus}(\mathcal{P}, C, S) \leq \mathrm{rank}_{\mathrm{SDP}}\, M$, let $M(f, s) = \langle T_f, U_s\rangle + \mu(f)$ be a size-r SDP factorization. We shall construct an SDP formulation of size r. We claim that the SDP formulation:

$$X \in \mathbb{S}^r_+ \tag{7}$$

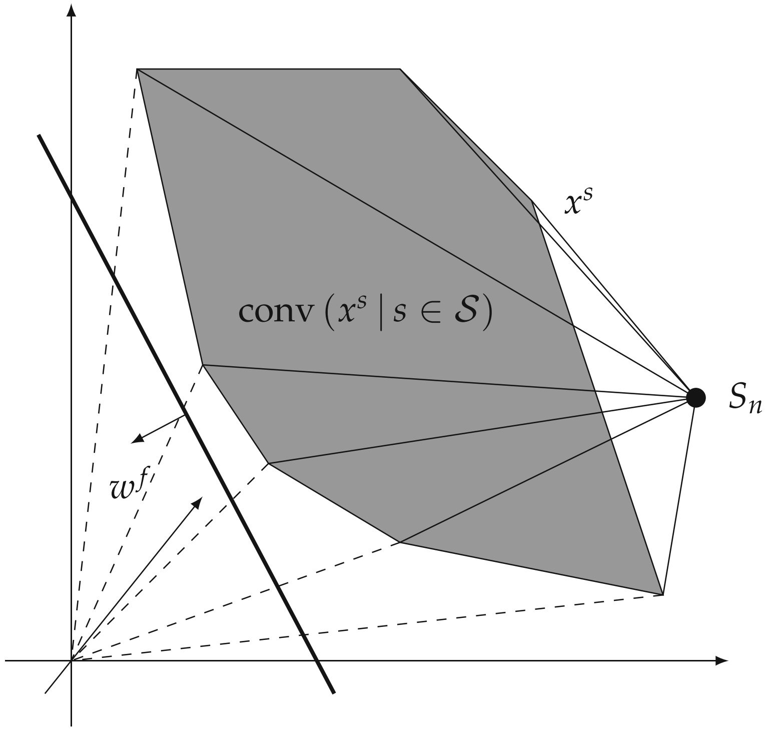

Fig. 1 Linear program from an LP factorization. The LP is the positive orthant $x \geq 0$. The point 0 is a maximizer for all linearizations $w^f = C(f) - \mu(f) - T_f x$ of objective functions val_f for all instances f. The normals of all objective functions point in nonpositive direction as $T_f \geq 0$

with representations

$$w^f(X) := C(f) - \mu(f) - \langle T_f, X\rangle \qquad \forall f \in \mathcal{F}^S \quad \text{and} \quad X^s := U_s \qquad \forall s \in \mathcal{S}$$

satisfies the requirements of Definition 2.8. Condition (5) follows by:

$$w^f(X^s) = C(f) - \mu(f) - \langle T_f, U_s\rangle = C(f) - M(f, s) = \mathrm{val}_f(s).$$

Moreover, the $X^s = U_s$ are psd, hence clearly satisfy the system (7), so that Condition (4) is fulfilled. Finally, Condition (6) also follows readily.

$$\begin{aligned}\max\left\{w^f(X) \mid X \in \mathbb{S}^r_+\right\} &= \max\left\{C(f) - \mu(f) - \langle T_f, X\rangle \mid X \in \mathbb{S}^r_+\right\} \\ &= C(f) - \mu(f) \leq C(f),\end{aligned}$$

as T_f and X being psd implies $\langle T_f, X\rangle \geq 0$; equality holds e.g., for $X = 0$. Thus we have constructed an SDP formulation of size r as claimed. $\square$

3.1 Examples

We will now briefly introduce examples to exemplify the notion of formulation complexity. The lower bounds in [25,42] are concerned with specific polytopes, namely the TSP polytope as well as the matching polytope. In the case of the *matching problem* a lower bound $\mathrm{fc}_+(\mathcal{P}_{\mathrm{Match}}) = 2^{\Omega(n)}$ for the formulation complexity follows immediately and this also extends to the approximate case via [4]. The situation in the case of the TSP polytope is slightly different though as the polyhedral reduction used in Yannakakis's original argument [48] is *not (a priori)* a reduction compatible with our setup; we will come back to this in Remark 4.11.

Next, we provide an example for maximum independent sets in a uniform model; see [12] for more details as well as an average case analysis. Here there is no bound on the maximum degree of graphs, unlike in Theorem 5.3.

Example 3.7 (*Maximum independent set problem* (*uniform model*)) Let us consider the maximum independent set problem $\mathcal{P}$ over some family $\mathcal{G}$ of graphs G where $V(G) \subseteq [n]$ with aim to estimate the maximum size $\alpha(G)$ of independent sets in each $G \in \mathcal{G}$.

A natural choice is to let the *feasible solutions* be all subsets S of $[n]$, and the *instances* be all $G \in \mathcal{G}$. The objective function is

$$\mathrm{val}_G(S) \coloneqq |V(G) \cap S| - |E(G(S))| .$$

Here $\mathrm{val}_G(S)$ can be easily seen to lower bound the size of an independent set, obtained from S by removing vertices not in G, and also removing one end point of every edge with both end points in S. Clearly, $\mathrm{val}_G(S) = |S|$ for independent sets S of G, i.e., in this case our choice is exact. Thus $\alpha(G) = \max_{S\subseteq[n]} \mathrm{val}_G(S)$.

Let us consider the special case when $\mathcal{G}$ is the set of all simple graphs with $V(G) \subseteq [n]$. We shall use guarantees $S(G) \coloneqq \max \mathrm{val}_G = \alpha(G)$ and $C(G) \coloneqq \rho^{-1} \mathrm{val}_G$ for an approximation factor $0 < \rho \leq 1$. Restricting to complete graphs K_U with $U \subseteq [n]$, the obtained slack matrix is a $(\rho^{-1} - 1)$-shift of the (partial) unique disjointness matrix, hence for approximations within a factor of ρ, we obtain the lower bound $\mathrm{fc}_+(\mathcal{P}, C, \max) \geq 2^{\frac{n\rho}{8}}$ with [5,14]. See [12] for other choices of $\mathcal{G}$, such as e.g., randomly choosing the graphs.

Finally, we would like to show how our framework can be applied to function classes of interest where a polyhedral formulation is not readily available. It is well-known that the level-k Sherali–Adams hierarchy captures all nonnegative k-juntas, i.e., functions $f\colon \{0,1\}^n \to \mathbb{R}_+$ that depend only on k coordinates of the input (see e.g., [16]) and it can be written as a linear program using $O(n^k)$ inequalities. We will now show that this is essentially optimal for k small.

Example 3.8 (*k-juntas*) We consider the problem of maximizing nonnegative k-juntas over the n-dimensional hypercube. Let the set of instances $\mathcal{F}$ be the family of all nonnegative k-juntas and let the set of feasible solutions be $\mathcal{S} = \{0,1\}^n$, with $\mathrm{val}_f(s) \coloneqq f(s)$. We put $C(f) = S(f) = \max_{s\in\mathcal{S}} \mathrm{val}_f(s)$.

As we are interested in a lower bound we will confine ourselves to a specific subfamily of functions $\mathcal{F}' := \{f_a \mid a \in \{0,1\}^n, |a| = k\} \subseteq \mathcal{F}$ with $f_a(b) := a^\intercal b - 2\binom{a^\intercal b}{2}$, and hence $C(f_a) = 1$. Clearly $\left|\mathcal{F}'\right| = \binom{n}{k}$, so that the nonnegative rank of the slack matrix $S_{a,b} := C(f_a) - f_a(b) = 1 - a^\intercal b + 2\binom{a^\intercal b}{2} = (1 - a^\intercal b)^2$, with $a, b \in \{0,1\}^n$ and $|a| = k$ is at most $\binom{n}{k}$.

Now for each $f_a \in \mathcal{F}'$ we have that $C(f_a) - f_a(b) = (1 - a^\intercal b)^2 = 1$ if $a \cap b = \emptyset$ and there are 2^{n-k} such choices for b for a given a. Thus the matrix S has $\binom{n}{k}2^{n-k}$ entries 1 arising from disjoint pairs a, b. However in [31] it was shown that any nonnegative rank-1 matrix can cover at most 2^n of such pairs. Thus the nonnegative rank of S is at least $\frac{\binom{n}{k}2^{n-k}}{2^n} = \frac{\binom{n}{k}}{2^k}$. The latter is $\Omega(n^k)$ for k constant and at least $\Omega(n^{k-\alpha})$ for $k = \alpha \log n$ with $\alpha \in \mathbb{N}$ constant and $k > \alpha$. Thus the LP formulation for k-juntas derived from the level-k Sherali–Adams hierarchy is essentially optimal for small k.

4 Affine reductions for LPs and SDPs

We will now introduce natural reductions between problems, with control on approximation guarantees that translate to the underlying LP and SDP level.

For a set X we use cone X to denote all formal nonnegative combinations of elements of X, i.e., cone $X := \left\{\sum_{x \in X} a_x x \mid a_x \geq 0\right\}$. Note that if X is not equipped with a sum or a multiplication with scalars then a formal nonnegative combination is not necessarily an element of X. Similarly let conv X be the set of all formal convex combinations, i.e., conv $X := \left\{\sum_{x \in X} a_x x \mid a_x \geq 0 \ \forall x \in X, \sum_{x \in X} a_x = 1\right\}$.

Definition 4.1 (*Reductions between problems*) Let $\mathcal{P}_1 = (\mathcal{S}_1, \mathcal{F}_1, \mathrm{val})$ and $\mathcal{P}_2 = (\mathcal{S}_2, \mathcal{F}_2, \mathrm{val})$ be maximization problems. Let C_1, S_1 and C_2, S_2 be guarantees for $\mathcal{P}_1$ and $\mathcal{P}_2$ respectively. A *reduction* from $\mathcal{P}_1$ to $\mathcal{P}_2$ respecting these guarantees consist of two maps:

(i) $\beta\colon \mathcal{F}_1^{S_1} \to \operatorname{cone} \mathcal{F}_2^{S_2} + \mathbb{R}$ rewriting instances as formal nonnegative combinations: $\beta(f_1) := \sum_{f \in \mathcal{F}_2^{S_2}} b_{f_1,f} \cdot f + \mu(f_1)$ with $b_{f_1,f} \geq 0$ for all $f \in \mathcal{F}_2^{S_2}$; the term $\mu(f_1)$ is called the *affine shift*

(ii) $\gamma\colon \mathcal{S}_1 \to \operatorname{conv} \mathcal{S}_2$ rewriting solutions as formal convex combination of $\mathcal{S}_2$: $\gamma(s_1) := \sum_{s \in \mathcal{S}_2} a_{s_1,s} \cdot s$ with $a_{s_1,s} \geq 0$ for all $s \in \mathcal{S}_2$ and $\sum_{s \in \mathcal{S}_2} a_{s_1,s} = 1$;

subject to

$$\mathrm{val}_{f_1}(s_1) = \sum_{\substack{f \in \mathcal{F}_2^{S_2} \\ s \in \mathcal{S}_2}} b_{f_1,f} a_{s_1,s} \cdot \mathrm{val}_f(s) + \mu(f_1), \qquad s_1 \in \mathcal{S}_1,\ f_1 \in \mathcal{F}_1^{S_1}, \tag{8}$$

expressing representation of the objective function of $\mathcal{P}_1$ by that of $\mathcal{P}_2$, and additionally

$$C_1(f_1) \geq \sum_{f \in \mathcal{F}_2^{S_2}} b_{f_1,f} \cdot C_2(f) + \mu(f_1), \qquad f_1 \in \mathcal{F}_1^{S_1}, \tag{9}$$

ensuring feasibility of the completeness guarantee.

Observe that the role of soundness guarantees of $\mathcal{P}_1$ and $\mathcal{P}_2$ in the definition is to restrict the instances considered: the map β involves only the instances whose optimum value is bounded by these guarantees. One can analogously define reductions involving minimization problems. E.g., for a reduction from a maximization problem $\mathcal{P}_1$ to a minimization problem $\mathcal{P}_2$, the formulas are

$$\beta(f_1) \coloneqq \mu(f_1) - \sum_{f \in \mathcal{F}_2^{S_2}} b_{f_1,f} \cdot f$$
$$\operatorname{val}_{f_1}(s_1) = \mu(f_1) - \sum_{\substack{f \in \mathcal{F}_2^{S_2} \\ s \in \mathcal{S}_2}} b_{f_1,f} a_{s_1,s} \cdot \operatorname{val}_f(s)$$
$$C_1(f_1) \geq \mu(f_1) - \sum_{\operatorname{val}_f \in \mathcal{F}_2^{S_2}} b_{f_1,f} \cdot C(f).$$

Note that elements in $\mathcal{S}_2$ are obtained as convex combinations, while elements in $\mathcal{F}_2$ are obtained as nonnegative combinations and a shift. The additional freedom for instances allows scaling and shifting the function values.

In order to use reductions as a lower bounding techniques we show that a reduction between two optimization problems results in an inequality of the corresponding LP and SDP formulation complexities.

Proposition 4.2 (Reductions of formulations) *Consider a reduction from an optimization problem $\mathcal{P}_1$ to another one $\mathcal{P}_2$ respecting completeness and soundness guarantees C_1, S_1 and C_2, S_2. Then* $\operatorname{fc}_+(\mathcal{P}_1, C_1, S_1) \leq \operatorname{fc}_+(\mathcal{P}_2, C_2, S_2)$ *and* $\operatorname{fc}_\oplus(\mathcal{P}_1, C_1, S_1) \leq \operatorname{fc}_\oplus(\mathcal{P}_2, C_2, S_2)$.

To provide some intuition, on a slack matrix level a reduction expresses the slack matrix M_1 of $\mathcal{P}_1$ in the form $M_1 = R \cdot M_2 \cdot C + t\mathbb{1}$, where M_2 is the slack matrix of problem $\mathcal{P}_2$.

Proof We will use the notation from Definition 4.1 for the reduction. We only prove the claim for LP formulations and for two maximization problems, as the proof is analogous for SDP formulations and when either or both problems are minimization problems. Let us choose an LP formulation $Ax \leq b$ of $\mathcal{P}_2$ with x^s realizing $s \in \mathcal{S}_2$ and w^f realizing $f \in \mathcal{F}_2^{S_2}$. For $\mathcal{P}_1$ we shall use the same linear program $Ax \leq b$ with the following realizations y^{s_1} of feasible solutions $s_1 \in \mathcal{S}_1$, and u^{f_1} of instances $f_1 \in \mathcal{F}_1^{S_1}$, where

$$y^{s_1} \coloneqq \sum_{s \in \mathcal{S}_2} a_{s_1,s} \cdot x^s, \qquad u^{f_1}(x) \coloneqq \sum_{f \in \mathcal{F}_2^{S_2}} b_{f_1,f} \cdot w^f(x) + \mu(f_1).$$

As y^{s_1} is a convex combination of the x^s, obviously $Ay^{s_1} \leq b$. The u^{f_1} are clearly affine functions with

$$\begin{aligned} u^{f_1}(y^{s_1}) &= \sum_{f \in \mathcal{F}_2^{S_2}} b_{f_1,f} \cdot w^f \left(\sum_{s \in \mathcal{S}_2} a_{s_1,s} \cdot x^s \right) + \mu(f_1) \\ &= \sum_{f \in \mathcal{F}_2^{S_2}} b_{f_1,f} \sum_{s \in \mathcal{S}_2} a_{s_1,s} \cdot w^f(x^s) + \mu(f_1) = \mathrm{val}_{f_1}(s_1) \end{aligned}$$

by Eq. (8). Moreover, by Eq. (9).

$$\begin{aligned} \max_{Ax \leq b} u^{f_1}(x) &\leq \sum_{f \in \mathcal{F}_2^{S_2}} b_{f_1,f} \cdot \max_{Ax \leq b} w^f(x) + \mu(f_1) \\ &\leq \sum_{f \in \mathcal{F}_2^{S_2}} b_{f_1,f} \cdot C_2(f) + \mu(f_1) \leq C_1(f_1). \end{aligned}$$

□

We briefly explain the equivalent formulation of Proposition 4.2 on the slack matrix level. First observe that whenever $M_1 = R \cdot M_2 \cdot C + t\mathbb{1}$ with M_1, M_2, R, C nonnegative matrices, and t a nonnegative vector, such that $\mathbb{1}C = \mathbb{1}$, then $\operatorname{rank}_{\mathrm{LP}} M_1 \leq \operatorname{rank}_{\mathrm{LP}} M_2$ and $\operatorname{rank}_{\mathrm{SDP}} M_1 \leq \operatorname{rank}_{\mathrm{SDP}} M_2$. Given a reduction of $\mathcal{P}_1$ to $\mathcal{P}_2$ with the notation as in Definition 4.1, one chooses M_1 and M_2 to be the slack matrices of $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively, together with matrices R, C and a vector t with the following entries:

$$R(f_1, f) = b_{f_1,f}, \quad C(s, s_1) = a_{s_1,s}, \quad t(f) = C_2(f_1) + \mu(f_1) - \sum_{f \in \mathcal{F}_2} b_{f_1,f} \cdot C(f),$$

all nonnegative, satisfying $M_1 = R \cdot M_2 \cdot C + t\mathbb{1}$. Now given a size-$r$ LP factorization $M_2 = \sum_{i \in [r]} u_i v_i + \mu\mathbb{1}$ of M_2, we obtain a size-r LP factorization of M_1 via linearity. The SDP case is analogous.

For most problems, there is a natural *size* $|f|$ of an instance f, which is a nonnegative number, and guarantees are often proportional to it. In many cases, as for the examples presented here, there is no need to consider nonnegative linear combinations in reductions. In these cases we can use a lightweight version of reduction of the form $\beta \colon \mathcal{F}_1 \to \mathcal{F}_2$ and $\gamma \colon \mathcal{S}_1 \to \mathcal{S}_2$. For convenience, the objective function val of $\mathcal{P}_2$ now appears on the left-hand side by rearranging.

Corollary 4.3 (Inapproximability via simple reductions) *Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be two optimization problems. Let $\mathcal{P}_1$ the completeness guarantee of $\mathcal{P}_1$ have the form*

$$C_1(f) = \tau_1 |f|, \qquad f \in \mathcal{F}_1$$

proportional to the size $|f|$ *of each instance* f *with* $|f| \geq 0$. *Furthermore, let* $\gamma : \mathcal{S}_1 \to \mathcal{S}_2$ *and* $\beta : \mathcal{F}_1^{S_1} \to \mathcal{F}_2$ *be maps satisfying for some constants* α, μ *and* η

$$\operatorname{val}_{\beta(f_1)}[\gamma(s_1)] = \alpha \operatorname{val}_{f_1}(s_1) + \mu\, |f_1| \quad \textit{and} \quad |\beta(f_1)| = \eta \cdot |f_1|\,,$$

where $\alpha > 0$ *if* $\mathcal{P}_2$ *is a maximization problem, and* $\alpha < 0$ *if* $\mathcal{P}_2$ *is a minimization problem. Let the completeness guarantee* C_2 *of* $\mathcal{P}_2$ *be given as*

$$C_2(f) := \frac{(\alpha\tau_1 + \mu)}{\eta}\, |f| \qquad\qquad f \in \mathcal{F}_2.$$

Furthermore let σ_2 *be a nonnegative number satisfying for all* $f_1 \in \mathcal{F}_1^{S_1}$

$$\begin{aligned} \max \operatorname{val}_{\beta(f_1)} &\leq \sigma_2\, |\beta(f_1)| && \textit{if } \mathcal{P}_2 \textit{ is a maximization problem} \\ \min \operatorname{val}_{\beta(f_1)} &\geq \sigma_2\, |\beta(f_1)| && \textit{if } \mathcal{P}_2 \textit{ is a minimization problem} \end{aligned}$$

and we set

$$S_2(f) = \sigma_2\, |f| \qquad\qquad f \in \mathcal{F}_2$$

Then β *and* γ *form a reduction from* $\mathcal{P}_1$ *with guarantees* C_1, S_1 *to* $\mathcal{P}_2$ *with guarantees* C_2, S_2. *In particular,* $\operatorname{fc}_+\left(\mathcal{P}_2, \frac{(\alpha\tau_1+\mu)}{\eta}\,|f|\,, \sigma_2\,|f|\right) \geq \operatorname{fc}_+(\mathcal{P}_1, \tau_1\,|f|\,, \sigma_1\,|f|)$ *and* $\operatorname{fc}_\oplus\left(\mathcal{P}_2, \frac{(\alpha\tau_1+\mu)}{\eta}\,|f|\,, \sigma_2\,|f|\right) \geq \operatorname{fc}_\oplus(\mathcal{P}_1, C_1, S_1)$, *i.e.,* $\mathcal{P}_2$ *is inapproximable within a factor of* $\sigma_2\eta/(\alpha\tau_1+\mu)$ *by LP and SDP formulations of size less than* $\operatorname{fc}_+(\mathcal{P}_1, C_1, S_1)$ *and* $\operatorname{fc}_\oplus(\mathcal{P}_1, C_1, S_1)$, *respectively.*

Proof Equation (8) follows directly from $\operatorname{val}_{\beta(f_1)}[\gamma(s_1)] = \alpha \operatorname{val}_{f_1}(s_1) + \mu\, |f_1|$. Using the parameters as given by the same relation to rewrite Eq. (9), we get

$$C_1(f_1) \geq \frac{1}{\alpha} C_2(\beta(f_1)) - \frac{\mu\, |f_1|}{\alpha}.$$

Plugging in the definition of C_2 and the relation between the sizes of the two problems, $|\beta(f_1)| = \eta \cdot |f_1|$, we get

$$C_1(f_1) \geq \frac{\alpha\tau_1 + \mu}{\alpha\eta}\, |\beta(f_1)| - \frac{\mu\, |f_1|}{\alpha} = \frac{\alpha\tau_1 + \mu}{\alpha\eta} \eta\, |f_1| - \frac{\mu\, |f_1|}{\alpha} = \tau_1\, |f_1|\,,$$

showing that Eq. (9) holds. □

Remark 4.4 (*size-proportional guarantees*) In many cases also the soundness guarantee of $\mathcal{P}_1$ is proportional to the size of f, i.e., $S_1(f) = \sigma_1\, |f|$ for all $f \in \mathcal{F}_1$. Then a common choice is $\sigma_2 = \alpha\sigma_1 + \mu$, provided that the reduction is *exact*, i.e.,

$$\operatorname{opt}_{\mathcal{P}_1} \operatorname{val}_{\beta(f_1)} = \alpha \operatorname{opt}_{\mathcal{P}_2} \operatorname{val}_{f_1} + \mu\, |f_1|\,,$$

Table 3 Summary of the hardness results for MaxCUT and Max-k-XOR for $k \geq 2$

	Completeness (τ_{MaxCUT})	Soundness (σ_{MaxCUT})	Inapprox factor	Size	Source
fc_+	$1-\varepsilon$	$1/2+\varepsilon$	$1/2+\Theta(\varepsilon)$	$2^{n^{c(\varepsilon)}}$	Theorem 4.6
$\mathrm{fc}_\oplus$	Exact case		1	$2^{\Omega\left(n^{2/13}\right)}$	Theorem 4.7
	$4/5-\varepsilon$	$3/4+\varepsilon$	$15/16+\Theta(\varepsilon)$	$n^{\Omega(\log n/\log\log n)}$	Theorem 4.8
	$1-\varepsilon$	$c_{GW}+\varepsilon$	$c_{GW}+\Theta(\varepsilon)$	Superpolynomial	Conjecture 4.9

Table 4 Summary of the hardness results for the bounded degree case MaxCUT_Δ

	Completeness ($\tau_{\mathsf{MaxCUT}_\Delta}$)	Soundness ($\sigma_{\mathsf{MaxCUT}_\Delta}$)	Inapprox factor	Size	Source
fc_+	$1-\varepsilon$	$1/2+\varepsilon$	$1/2+\Theta(\varepsilon)$	$2^{n^{c(\varepsilon)}}$	Theorem 4.6
$\mathrm{fc}_\oplus$	$1-\varepsilon$	$c_{GW}+\varepsilon$	$c_{GW}+\Theta(\varepsilon)$	Superpolynomial	Conjecture 4.9

where the operator $\mathrm{opt}_{\mathcal{P}_1}$ is max when $\mathcal{P}_1$ is a maximization problem, and the operator $\mathrm{opt}_{\mathcal{P}_1}$ is min when $\mathcal{P}_1$ is a minimization problem. The operator $\mathrm{opt}_{\mathcal{P}_2}$ is defined similarly for $\mathcal{P}_2$.

Notation 4.5 We shall write $\mathrm{fc}_+(\mathcal{P}, \tau, \sigma)$ for $\mathrm{fc}_+(\mathcal{P}, C, S)$ with $C(f) = \tau\,|f|$ and $S(f) = \sigma\,|f|$.

The base problems $\mathcal{P}_1$ from which we reduce will be the CSPs MaxCUT, MaxCUT_Δ or Max-k-XOR in our examples. For CSPs, the size of an instance, i.e., weighting $(w_1, \ldots, w_m)$ is the total weight $\sum_{i\in[m]} w_i$ of all clauses. For 0/1 weightings representing a subset L of clauses, the size is the number of elements of L.

The problems below constitute our base problems and play the same role as e.g., Max-3-XOR in Håstad's PCP theorem (see [30]). We summarize the hardness results in Table 3 and Table 4; the hardness results for MaxCUT and Max-k-XOR in Table 3 are the same, since MaxCUT is a subproblem of Max-k-XOR for all $k \geq 2$ and this is how we establish the hardness of Max-k-XOR.

Theorem 4.6 [37, Corollary 1.3] *For every $\varepsilon > 0$ there exists a constant $c(\varepsilon)$ such that for every $k \geq 2$, we have $\mathrm{fc}_+(\textsf{Max-}k\textsf{-XOR}, 1-\varepsilon, 1/2+\varepsilon)$ and $\mathrm{fc}_+(\textsf{MaxCUT}, 1-\varepsilon, 1/2+\varepsilon)$ are both at least $2^{n^{c(\varepsilon)}}$ for all n, resulting in an inapproximability factor of $1/2+\Theta(\varepsilon)$. Moreover, for the bounded degree case we have $\mathrm{fc}_+(\textsf{MaxCUT}_\Delta, 1-\varepsilon, 1/2+\varepsilon) = 2^{n^{c(\varepsilon)}}$ for all n, where Δ is large enough depending on ε and therefore the same inapproximability factor.*

Proof We first show the hardness result for MaxCUT. Although not explicitly stated in Corollary 1.3 of [37] the statement shown is that there exists a constant $c(\varepsilon)$ so that there is no LP relaxation of size less than $2^{n^{c(\varepsilon)}}$ that achieves a completeness guarantee of $1-\varepsilon$ and a soundness guarantee of $1/2+\varepsilon$, which is a reformulation of our statement. Since the argument ultimately goes back to the proof of Theorem 5.3(I) in [18] and that construction only uses bounded degree graphs, where the bound Δ depends on ε but not on n, it follows that we get the same statement for MaxCUT_Δ.

The result for Max-k-XOR follows, since MaxCUT is a subproblem of Max-2-XOR and therefore of Max-k-XOR for $k \geq 2$. $\square$

In the SDP case we have the following results.

Theorem 4.7 [38] *For the exact semidefinite formulation complexity we have* $\mathrm{fc}_\oplus(\mathit{MaxCUT}) = 2^{\Omega(n^{2/13})}$ *and* $\mathrm{fc}_\oplus(\mathit{Max}\text{-}k\text{-}\mathit{XOR}) = 2^{\Omega(n^{2/13})}$ *for* $k \geq 2$ *and for infinitely many* n.

Proof We use a result by [38] stating that the slack matrix of the cut polytope has PSD rank at least $2^{\Omega(n^{2/13})}$. Together with the factorization theorem (Theorem 3.5) we get the result for MaxCUT.

The result for Max-k-XOR follows since MaxCUT is a subproblem of Max-k-XOR for each $k \geq 2$. $\square$

In the approximate case we can use a result by [13].

Theorem 4.8 [13, Theorem 7.1] *For every* $\varepsilon > 0$ *and* $k \geq 2$ *there are infinitely many* n *such that* $\mathrm{fc}_\oplus(\mathit{MaxCUT}, 4/5 - \varepsilon, 3/4 + \varepsilon) = n^{\Omega(\log n / \log\log n)}$ *and* $\mathrm{fc}_\oplus(\mathit{Max}\text{-}k\text{-}\mathit{XOR}, 4/5-\varepsilon, 3/4+\varepsilon) = n^{\Omega(\log n / \log\log n)}$*, resulting in inapproximability factors of* $15/16 + \Theta(\varepsilon)$.

Recall from [36] that under the Unique Games Conjecture, MaxCUT cannot be approximated better than c_{GW} by a polynomial-time algorithm. This motivates the following conjecture, which provides the SDP-hard base problem with the strongest approximation guarantee. For some problems it might be possible to also reduce from Max-3-SAT, which is SDP-hard to approximate within any factor better than 7/8; see [38].

Recall that to obtain the completeness and soundness guarantees for MaxCUT, the given factors need to be multiplied with the number of edges, i.e, $C(G) = (1 - \varepsilon)\,|E(G)|$ and $S(G) = (c_{GW} + \varepsilon)\,|E(G)|$ for all graphs G.

Conjecture 4.9 (SDP inapproximability of MaxCUT) *For every* $\varepsilon > 0$*, and for every constant* Δ *large enough depending on* ε*, the formulation complexity* $\mathrm{fc}_\oplus(\mathit{MaxCUT}_\Delta, 1 - \varepsilon, c_{GW} + \varepsilon)$ *of MaxCUT is superpolynomial.*

Note however that for fixed Δ, there are algorithms achieving an approximation factor of $c_{GW} + \varepsilon$ by [24], hence in the conjecture Δ should go to infinity as ε tends to 0. Finally, we remark that by [33, Lemma 2.9] there are graphs G where the Goemans–Williamson SDP is off by a factor of $c_{GW} + \varepsilon$. The conjecture claims, in particular, that there are also such graphs with SDP optimum $(1 - \varepsilon)\,|E(G)|$.

Remark 4.10 (*Reductions are agnostic*) We want to stress that a reduction between two problems establishes a relation between the formulation complexities, both in the linear as well as in the semidefinite case. We therefore use fc as a shorthand if a statement holds for both fc_+ and $\mathrm{fc}_\oplus$.

Remark 4.11 (*Facial reductions and formulation complexity*) We would like to conclude this section with an important remark relating our reduction mechanism with *facial reductions* used in [25,48] for the polyhedral setup. Facial reductions essentially state: if Q projects to a face of P, then $\mathrm{xc}(P) \leq \mathrm{xc}(Q)$, where $\mathrm{xc}(.)$ denotes the extension complexity.

As the notion of formulation complexity does not directly deal with polytopes, there is no direct translation of monotonicity of extension complexity under faces and projections (see [25]). Thus many reductions that have been used in the context of extension complexity and polytopes do not apply, such as e.g., the one from TSP to matching in [48,49]. However, in all cases that we considered, the facial reduction underlies a reduction between the problems as defined in Definition 4.1.

5 Inapproximability of VertexCover and IndependentSet

We will now establish inapproximability results for VertexCover and IndependentSet via reduction from MaxCUT, even for bounded degree subgraphs. These two problems are of particular interest, answering a question of [16,45] as well as a weak version of sparse graph conjecture from [7]. Moreover, VertexCover is not of the CSP type, therefore the framework in [16] does not apply. Using our reduction framework, recently these results have been further improved in [3] to obtain $(2-\varepsilon)$-inapproximability for VertexCover (which is optimal) and inapproximability of IndependentSet within any constant factor. The current best PCP bound for bounded degree IndependentSet can be found in [17]. See also [2] for inapproximability results assuming the Unique Games Conjecture.

The minimization problem VertexCover(G) of a graph G asks for a minimum weighted vertex cover of G. We consider the non-uniform model with instances being the induced subgraphs of G.

Definition 5.1 (VertexCover) Given a graph G, the problem VertexCover(G) has all vertex covers S of G as feasible solutions, and instances all induced subgraphs H of G. The problem VertexCover(G) is the minimization problem with its objective function having values $\mathrm{val}_H(S) := |S \cap V(H)|$. The problem VertexCover$(G)_\Delta$ is the restriction of instances to induced subgraphs H, with maximum degree at most Δ.

Note that for every vertex cover S of G, any induced subgraph H has $S \cap V(H)$ as a vertex cover, and all vertex covers of H are of this form. In particular, $\min \mathrm{val}_H$ is the minimum size of a vertex cover of H.

The problem IndependentSet asks for maximum sized independent sets in graphs. As independent sets are exactly the complements of vertex covers, it is natural to use a formulation similar to VertexCover.

Definition 5.2 (IndependentSet) Given a graph G, the maximization problem IndependentSet(G) has all independent sets S of G as feasible solutions, and instances are all induced subgraphs H of G. The objective function is $\mathrm{val}_H(S) := |S \cap V(H)|$. The subproblem IndependentSet$(G)_\Delta$ is the restriction to all induced subgraphs H with maximum degree at most Δ.

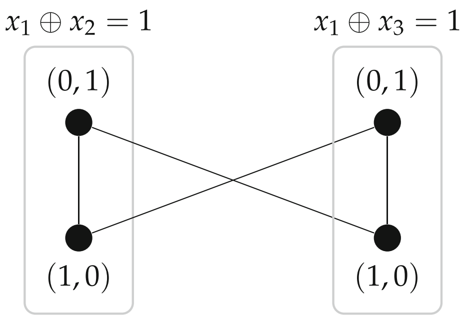

Fig. 2 Conflict graph of 2-XOR clauses. We include edges between all conflicted partial assignment to variables

For both VertexCover and IndependentSet, we shall use the following conflict graph G for a fixed n, similar to [23]; we might think of G as a *universal* graph encoding all possible instances. Let the vertices of G be all partial assignments σ of two variables x_i and x_j satisfying the 2-XOR clause $x_i \oplus x_j = 1$. Two vertices σ_1 and σ_2 are connected if and only if the assignments σ_1 and σ_2 are incompatible (i.e., assign different truth values to some variable), see Fig. 2 for an illustration. As we are considering problems for optimizing size of vertex sets, it is natural to define the size of an instance, i.e., a subgraph K, as the size of its vertex set $|V(K)|$.

The following theorem establishes the relative hardness of VertexCover and IndependentSet to MaxCUT. In order to achieve inapproximability factors as given in Tables 1 and 2 we use the base hardness of MaxCUT as given in Tables 3 and 4 together with the relative hardness results.

Theorem 5.3 *For every graph G' with $\left|V(G')\right| = n$ there is a graph G with $|V(G)| = 2\binom{n}{2}$ such that for all Δ*

$$\begin{aligned}&\operatorname{fc}\left(\mathsf{VertexCover}(G)_{2\Delta-1}, 1 - \frac{\tau_{\mathsf{MaxCUT}}}{2}, 1 - \frac{\sigma_{\mathsf{MaxCUT}}}{2}\right)\\ &\quad \geq \operatorname{fc}(\mathsf{MaxCUT}_{\Delta}(G'), \tau_{\mathsf{MaxCUT}}, \sigma_{\mathsf{MaxCUT}})\end{aligned}$$

and

$$\begin{aligned}&\operatorname{fc}\left(\mathsf{IndependentSet}(G)_{2\Delta-1}, \frac{\tau_{\mathsf{MaxCUT}}}{2}, \frac{\sigma_{\mathsf{MaxCUT}}}{2}\right)\\ &\quad \geq \operatorname{fc}(\mathsf{MaxCUT}_{\Delta}(G'), \tau_{\mathsf{MaxCUT}}, \sigma_{\mathsf{MaxCUT}}).\end{aligned}$$

Additionally in the unbounded degree case we get the same inequalities.

Proof We shall use the graph G constructed above, which has $m = 2\binom{n}{2}$ vertices. We reduce $\mathsf{MaxCUT}_{\Delta}(G')$ to $\mathsf{VertexCover}(G)_{2\Delta-1}$ using Corollary 4.3 with $\alpha = -1$, $\mu = 2$, and $\eta = 2$. We use the following guarantees for VertexCover:

$$\begin{aligned}C_{\mathsf{VertexCover}(G)}(H) &= (1 - \tau_{\mathsf{MaxCUT}}/2)\,|V(H)|\,,\\ S_{\mathsf{VertexCover}(G)}(H) &= (1 - \sigma_{\mathsf{MaxCUT}}/2)\,|V(H)|\,.\end{aligned}$$

Let $H(K)$ be the induced subgraph of G on the set $V(H(K)) := \{\sigma \mid \{i,j\} \in E(K), \operatorname{dom}\sigma = \{x_i, x_j\}\}$ of all partial assignments σ which assign values to variables x_i, x_j corresponding to an edge $\{i,j\}$ of K. In particular, $|V(H(K))| = 2\,|E(K)|$, as there are two partial assignments per each edge $\{i,j\}$.

Note that for every partial assignment σ to x_i and x_j, there are $2\Delta - 1$ partial assignments incompatible with it in $V(H(K))$: exactly one assignment for every edge of K incident to i or j. Thus the maximum degree of $H(K)$ is at most $2\Delta - 1$.

We now define the two maps providing the reduction. Let $\beta(K) := H(K)$. For a total assignment s, let $\gamma(s) := \{\sigma \mid \sigma \nsubseteq s\}$ be the set of partial assignments incompatible with s; this is clearly a vertex cover.

It remains to show that this is a reduction. For every edge $\{i,j\} \in K$, there are two partial assignments σ to x_i and x_j satisfying $x_i \oplus x_j = 1$. If s satisfies $x_i \oplus x_j = 1$, i.e., $\{i,j\}$ is in the cut induced by s, then exactly one of the σ is compatible with s, otherwise both of the assignments are incompatible. This provides

$$\mathrm{val}_{H(K)}^{\mathsf{VertexCover}}[\gamma(s)] = \left|\{\sigma \mid \sigma \nsubseteq s\}\right| = 2\,|E(K)| - \mathrm{val}_K^{\mathsf{MaxCUT}}(s).$$

To compare optimum values, note that for any vertex cover S of G, the partial assignments $\{\sigma \mid \sigma \notin S\}$ occurring in the complement of S are compatible (as the complement forms a stable set), hence there is a global assignment s of $x_1, \ldots, x_n$ compatible with all of them. In particular, $\gamma(s) \subseteq S$, hence $\mathrm{val}_{H(K)}^{\mathsf{VertexCover}}(S) \geq \mathrm{val}_K^{\mathsf{MaxCUT}}[\gamma(s)]$, so that we obtain

$$\begin{aligned}
\min \mathrm{val}_{H(K)}^{\mathsf{VertexCover}} &= \min_s \mathrm{val}_{H(K)}^{\mathsf{VertexCover}}[\gamma(s)] = 2\,|E(K)| - \max \mathrm{val}_K^{\mathsf{MaxCUT}} \\
&\geq 2\,|E(K)| - \sigma_{\mathsf{MaxCUT}}\,|E(K)| = (1 - \sigma_{\mathsf{MaxCUT}}/2)\,|V(H(K))| \\
&= S_{\mathsf{VertexCover}}(H(K)).
\end{aligned}$$

Finally, it is easy to verify that $C_{\mathsf{VertexCover}}(H(K)) = 2\,|E(K)| - C_{\mathsf{MaxCUT}}(K)$. This finishes the proof that β and γ define a reduction to $\mathsf{VertexCover}(G)_\Delta$. Hence with Corollary 4.3 and Remark 4.4 (or Proposition 4.2) we get the statements for $\mathsf{VertexCover}$.

For $\mathsf{IndependentSet}$, we apply a similar reduction from $\mathsf{MaxCUT}_\Delta(G')$ to $\mathsf{IndependentSet}(G)_{2\Delta-1}$. We define $\beta(K) := H(K)$ as above and we set $\gamma(s) := \{\sigma \mid \sigma \subseteq s\}$ to be the set of partial assignments compatible with the total assignment s, this is clearly an independent set, containing exactly one vertex per satisfied clause. In particular, $\mathrm{val}_{H(K)}[\gamma(s)] = \mathrm{val}_K(s)$. The rest of the argument is analogous to the case of $\mathsf{VertexCover}(G)_\Delta$, and hence omitted. Now the parameters for Corollary 4.3 are $\alpha = 1$, $\mu = 0$, and $\eta = 2$. □

6 Inapproximability of CSPs

In this section we present example reductions for minimum and maximum constraint satisfaction problems. The results for *binary* *Max-CSPs*, (for CSPs as defined in Definition 2.2) could also be obtained in the LP case from [16] by combination with the respective Sherali–Adams/Lasserre gap instances. For simplicity of exposition, we reduce from Max-2-XOR, or sometimes MaxCUT, however by reducing from the subproblem MaxCUT_Δ, we immediately obtain the results for bounded occurrence of literals, with Δ depending on the approximation factor.

In the following we formulate hardness of approximation relative to the respective base problem. The specific approximability factors given in Tables 1 and 2 can be computed from the relative statement by using the specific hardness results for MaxCUT.

6.1 Max-MULTI-k-CUT: a non-binary CSP

The Max-MULTI-k-CUT problem is interesting on its own being a CSP over a non-binary alphabet, thus the framework in [16] does not readily apply. Note that Max-MULTI-k-CUT is APX-hard, as it contains MaxCUT. The current best PCP inapproximability bound $1 - 1/(34k) + \varepsilon$ is given by [32]. Here we omit the definition of non-binary CSPs, where the feasible solutions are no longer two-valued assignments, and restrict to Max-MULTI-k-CUT.

Definition 6.1 (Max-MULTI-k-CUT) For fixed positive integers n and k, the problem Max-MULTI-k-CUT has

(i) **feasible solutions:** all partitions of $[n]$ into k sets;
(ii) **instances:** all graphs G with $V(G) \subseteq [n]$.
(iii) **objective function:** for a graph G and a partition p of $[n]$, let $\mathrm{val}_G(p)$ be the number of edges of G whose end points lie in different cells of p.

This differs from a binary CSP only by having a different type of feasible solutions. Hence it is still natural to define the size of an instance, i.e., graph G, as the number of clauses, i.e, number of edges $|E(G)|$.

Corollary 6.2 *Let $k \geq 3$ be a fixed integer. Then*

$$\begin{aligned} &\mathrm{fc}\left(\textsf{Max-MULTI-}k\textsf{-CUT}, 1 + \frac{\tau_{\textsf{MaxCUT}}}{c(k)}, 1 + \frac{\sigma_{\textsf{MaxCUT}}}{c(k)}\right) \\ &\quad \geq \mathrm{fc}(\textsf{MaxCUT}, \tau_{\textsf{MaxCUT}}, \sigma_{\textsf{MaxCUT}}), \end{aligned}$$

where

$$c(k) := \left(\binom{k-2}{2} + 2(k-2)\right)\left(\binom{k+2}{2} - 3\right).$$

Proof We reduce MaxCUT to Max-MULTI-k-CUT. The reduction is essentially identical to [39], and as before it suffices to verify its compatibility with our reduction mechanism, which we leave to the interested reader. ◻

6.2 Inapproximability of general 2-CSPs

First we consider general CSPs with no restrictions on constraints, for which the exact approximation factor can be easily established. We present the hardness of LP approximation here. The LP with matching factor can be found in [46].

Definition 6.3 (Max-2-CSP and Max-2-CONJSAT) The problem Max-2-CSP is the CSP on variables $x_1, \dots, x_n$ with constraint family $\mathcal{C}_{2CSP}$ consisting of all possible constraints depending on at most two variables. The problem Max-2-CONJSAT is the CSP with constraint family consisting of all possible conjunctions of two literals.

We obtain the following:

Corollary 6.4 *For every $\varepsilon > 0$ and every $n \geq 1$, we have*

$$\mathrm{fc}(\textsf{Max-2-CSP}, \tau_{\textsf{MaxCUT}}, \sigma_{\textsf{MaxCUT}}) \geq \mathrm{fc}(\textsf{MaxCUT}, \tau_{\textsf{MaxCUT}}, \sigma_{\textsf{MaxCUT}}),$$

where n is the number of variables of Max-2-CSP. Similarly,

$$\mathrm{fc}(\textsf{Max-2-CONJSAT}, \tau_{\textsf{MaxCUT}}/2, \sigma_{\textsf{MaxCUT}}/2) \geq \mathrm{fc}(\textsf{MaxCUT}, \tau_{\textsf{MaxCUT}}, \sigma_{\textsf{MaxCUT}})$$

holds.

Proof We identify Max-2-XOR and hence MaxCUT as a subproblem of Max-2-CSP: Every 2-XOR clause is evidently a boolean function of 2 variables. So restricting the instances of Max-2-CSP to 2-XOR clauses with 0/1 weights gives Max-2-XOR. Now with Theorem 4.6, the result follows.

The claim about Max-2-CONJSAT follows via the reduction from Max-2-CSP to Max-2-CONJSAT in [46]. We prefer to reduce from Max-2-XOR instead for easier control over the approximation guarantees. The idea is to write each clause C in disjunctive normal form, and replace C with the set $S(C)$ of conjunctions in its normal form, one conjunction for every assignment satisfying C. In particular, for 2-XOR clauses $S(x_i \oplus x_j = 1) = \{x_i \wedge \neg x_j, \neg x_i \wedge x_j\}$ and $S(x_i \oplus x_j = 0) = \{x_i \wedge x_j, \neg x_i \wedge \neg x_j\}$. Therefore formally, a set of clauses L is mapped to $\beta(L) = \bigcup_{C \in L} S(C)$. Every assignment of variables is mapped to themselves, i.e., γ is the identity. We have $\mathrm{val}_{\beta(L)}(s) = \mathrm{val}_L(s)$ and $|\beta(L)| = 2\,|L|$. Now the claim follows with Corollary 4.3 with the parameters $\alpha = 1$, $\mu = 0$ and $\eta = 2$. □

6.3 Max-2-SAT and Max-3-SAT inapproximability

We now establish an LP-inapproximability factor of $\frac{3}{4} + \varepsilon$ for Max-2-SAT via a direct reduction from MaxCUT and an SDP-inapproximability factor of about $\frac{35}{36} + \varepsilon \approx 0.9722 + \varepsilon$. Note that [26] show the existence of an LP that achieves a factor of $\frac{3}{4}$, so that our estimation is tight in the LP case. Moreover, in [22] it is shown that Max-2-SAT can be approximated with a small SDP within a factor of 0.931 leaving a (conditional) gap of less than 0.04.

Obviously, the same factor applies for Max-k-SAT with $k \geq 2$, too. Note that we allow clauses with less than k literals in Max-k-SAT, which is in line with the definition in [43] to maintain compatibility. In [38, Theorem 1.5] an SDP inapproximability of $7/8 + \varepsilon$ for Max-3-SAT was shown, however we are not aware of any Lassere gap instance for Max-2-SAT so that [38] cannot be invoked for Max-2-SAT in the SDP case; the Lassere gap for Max-2-SAT from [47] is 1 and in [43] it is explicitly stated that their results only apply for $k \geq 3$.

Definition 6.5 (Max-k-SAT) For fixed $n, k \in \mathbb{N}$, the problem Max-k-SAT is the CSP on the set of variables $\{x_1, \ldots, x_n\}$, where the constraint family $\mathcal{C}$ is the set of all sat clauses which consist of at most k literals.

Corollary 6.6 *For infinitely many n and $k \geq 2$ it holds that*

$$\operatorname{fc}\left(\textit{Max-}k\textit{-SAT}, \frac{1 + \tau_{\textit{MaxCUT}}}{2}, \frac{1 + \sigma_{\textit{MaxCUT}}}{2}\right) \\ \geq \operatorname{fc}(\textit{MaxCUT}, \tau_{\textit{MaxCUT}}, \sigma_{\textit{MaxCUT}}),$$

where n is the number of variables.

Proof We reduce MaxCUT to Max-2-SAT. For a 2-XOR clause $l = (x_i \oplus x_j = 1)$ with $i, j \in [n]$, we define two auxiliary constraints $C_1(l) = (x_i \vee x_j)$ and $C_2(l) = (\bar{x}_i \vee \bar{x}_j)$. Let $\beta(L) := \{C_1(l), C_2(l) \mid l \in L\}$ for a set of 2-XOR clauses L. We choose γ to be the identity map. Observe that whenever l is satisfied by a partial assignment s then both $C_1(l)$ and $C_2(l)$ are also satisfied by s, otherwise exactly one of $C_1(l)$ and $C_2(l)$ is satisfied. Hence we obtain a reduction from MaxCUT to Max-2-SAT using Corollary 4.3 with the parameters $\alpha = 1$, $\mu = 1$, and $\eta = 2$. The statements for Max-k-SAT follow, as Max-2-SAT is a subproblem of Max-k-SAT. □

Max-DICUT inapproximability

Problem Max-DICUT asks for a maximum sized cut in a directed graph G, i.e., partitioning the vertex set $V(G)$ into two parts V_0 and V_1, such that the number of directed edges $(i, j) \in E(G)$ going from V_0 to V_1, i.e., $i \in V_0$ and $j \in V_1$ are maximal. We use a formulation similar to MaxCUT.

Definition 6.7 (*Directed Cut*) For a fixed $n \in \mathbb{N}$, the problem Max-DICUT is the CSP with constraint family $\mathcal{C}_{DICUT} = \{\neg x_i \wedge x_j \mid i, j \in [n], i \neq j\}$.

We obtain $(1/2 + \varepsilon)$-inapproximability via the standard reduction from undirected graphs, by replacing every edge with two, namely, one edge in either direction. The inapproximability factor is tight as the LP in [46, p. 84, Eq. (DI)], is $\frac{1}{2}$-approximate for maximum weighted directed cut. In the SDP case we obtain $0.9375 + \varepsilon$-inapproximability and in [22] it is shown that Max-DICUT can be approximated with a small SDP within a factor of 0.859 leaving a gap of about 0.08.

Corollary 6.8 *For every ε and infinitely many n, we have*

$$\mathrm{fc}\left(\textsf{Max-DICUT}, \frac{\tau_{\textsf{MaxCUT}}}{2}, \frac{\sigma_{\textsf{MaxCUT}}}{2}\right) \geq \mathrm{fc}(\textsf{MaxCUT}, \tau_{\textsf{MaxCUT}}, \sigma_{\textsf{MaxCUT}}).$$

6.4 Minimum constraint satisfaction

In this section we examine minimum constraint satisfaction problems, a variant of constraint satisfaction problems, where the objective is not to maximize the number of *satisfied* constraints, but to minimize the number of *unsatisfied* constraints. This is equivalent to maximizing the number of satisfied constraints, however, the changed objective function yields different approximation factors due to the change in the magnitude of the optimum value; this is in analogy to the algorithmic world. We consider only Min-2-CNFDeletion and MinUnCUT from [1], which are complete in their class in the algorithmic hierarchy; our technique applies to many more problems in [39]. The problem Min-2-CNFDeletion is of particular interest here, as it is considered to be the hardest minimum CSP with nontrivial approximation guarantees (see [1]). We start with the definition of minimum CSPs.

Definition 6.9 The *minimum Constraint Satisfaction Problem* on variables $x_1, \ldots, x_n$ with constraint family $\mathcal{C} = \{C_1, \ldots, C_m\}$ is the minimization problem with

(i) **feasible solutions** all 0/1 assignments to $x_1, \ldots, x_n$;
(ii) **instances** all nonnegative weightings $w_1, \ldots, w_m$ of the constraints $C_1, \ldots, C_m$;
(iii) **objective functions** weighted sum of negated constraints, i.e. $\mathrm{val}_{w_1,\ldots,w_m}(x_1, \ldots, x_n) = \sum_i w_i[1 - C_i(x_1, \ldots, x_n)]$.

The goal is to minimize the objective function, i.e., the weight of unsatisfied constraints.

As mentioned above, we consider two examples.

Example 6.10 (*Minimum CSPs*) The problem Min-2-CNFDeletion is the minimum CSP with constraint family consisting of all disjunction of two literals, as in Max-2-SAT. The problem MinUnCUT is the minimum CSP with constraint family consisting of all equations $x_i \oplus x_j = b$ with $b \in \{0, 1\}$, as in Max-2-XOR.

We are ready to prove LP inapproximability bounds for these problems. Instead of the reductions in [20], we use direct, simpler reductions from MaxCUT and here we provide reductions for general weights. Note that the current best known algorithmic inapproximability for Min-2-CNFDeletion is $8\sqrt{5} - 15 - \varepsilon \approx 2.88854 - \varepsilon$ by [20]. Assuming the Unique Games Conjecture, [19] establish that Min-2-CNFDeletion cannot be approximated within any constant factor and our LP inapproximability factor coincides with this one. The problem MinUnCUT is known to be SNP-hard (see [39]) however the authors are not aware of the strongest known factor. We refer the reader to [35] for a classification of all minimum CSPs.

Theorem 6.11 *For infinitely many n, we have*

$$\mathrm{fc}\left(\textsf{Min-2-CNFDeletion}, \frac{1 - \tau_{\textsf{MaxCUT}}}{2}, \frac{1 - \sigma_{\textsf{MaxCUT}}}{2}\right) \geq \mathrm{fc}(\textsf{MaxCUT}, \tau_{\textsf{MaxCUT}}, \sigma_{\textsf{MaxCUT}})$$

and

$$\mathrm{fc}(\mathsf{MinUnCUT}, 1 - \tau_{\mathsf{MaxCUT}}, 1 - \sigma_{\mathsf{MaxCUT}}) \geq \mathrm{fc}(\mathsf{MaxCUT}, \tau_{\mathsf{MaxCUT}}, \sigma_{\mathsf{MaxCUT}}).$$

Proof We reduce from MaxCUT to Min-2-CNFDeletion similar to the previous reductions: assignments are mapped to themselves, i.e., γ is the identity. Under β every clause C_ℓ is replaced with two disjunctive clauses $C_\ell(1)$ and $C_\ell(2)$, both inheriting the weight w_ℓ of C_ℓ, i.e., $\mathrm{val}^{\mathsf{Min\text{-}2\text{-}CNFDeletion}}_{\beta(w_1,\ldots,w_m)}(x_1, \ldots, x_n) = \sum_\ell w_\ell[(1 - C_\ell(1)) + (1 - C_\ell(2))]$.

For $C_\ell = (x_i \oplus x_j = 1)$, we let $C_\ell(1) := x_i \vee \neg x_j$ and $C_\ell(2) := \neg x_i \vee x_j$. Note that if C_ℓ is unsatisfied, then both of $C_\ell(1)$ and $C_\ell(2)$ are satisfied, and if C_ℓ is satisfied, then exactly one of $C_\ell(1)$ and $C_\ell(2)$ is satisfied. Therefore $\mathrm{val}^{\mathsf{Min\text{-}2\text{-}CNFDeletion}}_{\beta(w_1,\ldots,w_m)} = \sum_\ell w_\ell - \mathrm{val}^{\mathsf{MaxCUT}}_{w_1,\ldots,w_m} w_\ell$. This provides the desired lower bound.

For MinUnCUT the reduction is similar but simpler, as we replace every clause C with itself. □

7 From matrix approximation to problem approximations

We will now explain how a nonnegative matrix with small nonnegative rank (or semidefinite rank) that is close to a slack matrix of a problem $\mathcal{P}$ of interest can be rounded to an actual slack matrix with a moderate increase in nonnegative rank (or semidefinite rank) and error. This argument is implicitly contained in [15,41] for the linear and semidefinite case respectively. In some sense we might want to think of this approach as an interpolation between a slack matrix (which corresponds to $\mathcal{P}$) and a close-by matrix of low nonnegative rank (or semidefinite rank) that does not correspond to any optimization problem. The result is a low nonnegative rank (or semidefinite rank) approximation of $\mathcal{P}$ with small error.

We will need the following simple lemma. Recall that the exterior algebra of a vector space V is the $\mathbb{R}$-algebra generated by V subject to the relations $v^2 = 0$ for all $v \in V$. As is customary, the product in this algebra is denoted by $\wedge$. The subspace of homogeneous degree-k elements (i.e., linear combination of elements of the form $v_1 \wedge \cdots \wedge v_k$ with $v_1, \ldots, v_k \in V$) is denoted by $\bigwedge^k V$. Recall that for $k = \dim V$, the space $\bigwedge^k V$ is one dimensional and is generated by $v_1 \wedge \cdots \wedge v_k$ for any basis $v_1, \ldots, v_k$ of V.

Lemma 7.1 *Let $M \in \mathbb{R}^{m \times n}$ be a real matrix of rank r. Then there are column vectors $a_1, \ldots, a_r \in \mathbb{R}^m$ and row vectors $b_1, \ldots, b_r \in \mathbb{R}^n$ with $M = \sum_{i \in [r]} a_i b_i$. Moreover, $\|a_i\|_\infty \leq 1$ and $\|b_i\|_\infty \leq \|M\|_\infty$ for all $1 \leq i \leq r$.*

Proof Consider the r dimensional vector space V spanned by all the rows $M_1, \ldots, M_m$ of M and identify the one dimensional exterior product $\bigwedge^r V$ with $\mathbb{R}$. Now choose r rows $M_{i_1}, \ldots, M_{i_r}$ for which $M_{i_1} \wedge \cdots \wedge M_{i_r}$ is the largest in absolute value in $\mathbb{R}$. As the M_i together span V it follows that the largest value is non-zero. Hence $M_{i_1}, \ldots, M_{i_r}$ form a basis of V. Therefore any row M_k can be uniquely written as a linear combination of the basis elements:

$$M_k = \sum_{j \in [r]} a_{k,j} M_{i_j}. \tag{10}$$

Fixing $j \in [r]$ and taking exterior products with the M_{i_l} where $l \neq j$ and both side we obtain

$$M_{i_1} \wedge \cdots \wedge M_{i_{j-1}} \wedge M_k \wedge M_{i_{j+1}} \wedge \cdots \wedge M_{i_r} = a_{k,j} \cdot M_{i_1} \wedge \cdots \wedge M_{i_r},$$

using the vanishing property of the exterior product. By maximality of $M_{i_1} \wedge \cdots \wedge M_{i_r}$, it follows that $|a_{k,j}| \leq 1$. We choose $a_k := \begin{bmatrix} a_{k,1} \\ \cdots \\ a_{k,r} \end{bmatrix}$, and thus we have $\|a_k\|_\infty \leq 1$. Moreover choose $b_j := M_{i_j}$, so that $\|b_j\|_\infty \leq \|M\|_\infty$ holds. Finally, Eq. (10) can be rewritten to $M = \sum_{j \in [r]} a_j b_j$, finishing the proof. □

For a vector a we can decompose it into its positive and negative part so that $a = a^+ - a^-$ with $a^+ a^- = 0$. Let $|a|$ denote the vector obtained from a by replacing every entry with its absolute value. Note that a^+, a^-, and $|a|$ are nonnegative vectors and $|a| = a^+ + a^-$. Furthermore their ℓ_∞-norm is at most $\|a\|_\infty$.

Theorem 7.2 *Let $\mathcal{P}$ be an optimization problem with (C, S)-approximate slack matrix M and let $\widetilde{M}$ be a nonnegative matrix. Then for the adjusted guarantee C' for $\mathcal{P}$ defined as*

$$C'(f) := C(f) + (\operatorname{rank} M + \operatorname{rank} \widetilde{M}) \|\widetilde{M} - M\|_\infty \quad \textit{if } \mathcal{P} \textit{ is a maximization problem, and}$$
$$C'(f) := C(f) - (\operatorname{rank} M + \operatorname{rank} \widetilde{M}) \|\widetilde{M} - M\|_\infty \quad \textit{if } \mathcal{P} \textit{ is a minimization problem,}$$

we have

$$\operatorname{fc}_+(\mathcal{P}, C', S) \leq \operatorname{rank}_{\mathrm{LP}} \widetilde{M} + 2(\operatorname{rank} M + \operatorname{rank} \widetilde{M}) \quad \textit{and}$$
$$\operatorname{fc}_\oplus(\mathcal{P}, C', S) \leq \operatorname{rank}_{\mathrm{SDP}} \widetilde{M} + 2(\operatorname{rank} M + \operatorname{rank} \widetilde{M}).$$

Proof We prove the statement for maximization problems; the minimization case follows similarly. The proof is based on the vector identity

$$\sum_{i \in [k]} |a_i| b - \sum_{i \in [k]} a_i b_i = \sum_{i \in [k]} a_i^+ (b - b_i) + \sum_{i \in [k]} a_i^- (b + b_i). \tag{11}$$

In our setting, the a_i, b_i with $i \in [k]$ will arise from the (not necessarily nonnegative) factorization of $\widetilde{M} - M$, obtained by applying Lemma 7.1, i.e., we have

$$\widetilde{M} - M = \sum_{i \in [k]} a_i b_i, \tag{12}$$

where $\|a_i\|_\infty \leq 1$ and $\|b_i\|_\infty \leq \|M\|_\infty$ for $i \in [k]$ with $k \leq \operatorname{rank}(\widetilde{M} - M) \leq \operatorname{rank} M + \operatorname{rank} \widetilde{M}$. Furthermore, define $b := \|\widetilde{M} - M\|_\infty \mathbb{1}$ to be the row vector with all entries equal to $\|\widetilde{M} - M\|_\infty \mathbb{1}$.

Substituting these values into (11), using Eq. (12) we obtain after rearranging

$$N := \sum_{i \in [k]} |a_i| b + M = \widetilde{M} + \sum_{i \in [k]} a_i^+ (b - b_i) + \sum_{i \in [k]} a_i^- (b + b_i),$$

so that we can conclude

$$\operatorname{rank}_{\mathrm{LP}} N \le \operatorname{rank}_{\mathrm{LP}} \widetilde{M} + 2k \le \operatorname{rank}_{\mathrm{LP}} \widetilde{M} + 2(\operatorname{rank} M + \operatorname{rank} \widetilde{M}), \tag{13}$$

and similarly

$$\operatorname{rank}_{\mathrm{SDP}} N \le \operatorname{rank}_{\mathrm{SDP}} \widetilde{M} + 2k \le \operatorname{rank}_{\mathrm{SDP}} \widetilde{M} + 2(\operatorname{rank} M + \operatorname{rank} \widetilde{M}). \tag{14}$$

It remains to relate N to the (C', S)-approximate slack matrix of $\mathcal{P}$. By definition, the entries of N are

$$N(f, s) = \underbrace{\sum_{i \in [k]} |a_i(f)| \cdot \|\widetilde{M} - M\|_\infty + C(f)}_{:= f^* \le C'(f)} - \mathrm{val}_f(s),$$

where $a_i(f)$ is the f-entry of a_i. Furthermore, as $\|a_i\|_\infty \le 1$ and $k \le \operatorname{rank} M + \operatorname{rank} \widetilde{M}$, we have $f^* \le C'(f)$. Thus the (C', S)-approximate slack matrix M' of $\mathcal{P}$ looks like

$$M'(f, s) = N(f, s) + (C'(f) - f^*),$$

and as $f^* \le C'(f)$, we have $\operatorname{rank}_{\mathrm{LP}} M' \le \operatorname{rank}_{\mathrm{LP}} N$ and $\operatorname{rank}_{\mathrm{SDP}} M' \le \operatorname{rank}_{\mathrm{SDP}} N$, establishing the claimed complexity bounds due to (13) in the LP case and (14) in the SDP case. $\square$

A possible application of Theorem 7.2 is to 'thin-out' a given factorization of a slack matrix to obtain an approximation with low nonnegative rank. The idea is that if a nonnegative matrix factorization contains a large number of factors that contribute only very little to each of the entries, then we can simply drop those factors, significantly reduce the nonnegative rank, and obtain a very good approximation of the original optimization problem. Theorem 7.2 is then used to turn the approximation of the matrix into an approximation of the original problem of interest. Also, it is possible to obtain low rank approximations of combinatorial problems sampling rank-1 factors proportional to their ℓ_1-weight as done in the context of information-theoretic approximations in [9].

Acknowledgements Research reported in this paper was partially supported by NSF Grant CMMI-1300144 and NSF CAREER Grant CMMI-1452463. The authors would like to thank James Lee for the helpful discussions regarding max-CSPs. We are indebted to Siu On Chan for some of the PCP inapproximability bounds as well as Santosh Vempala for the helpful discussions as well as the anonymous reviewers for significantly improving the presentation of the paper.

References

1. Agarwal, A., Charikar, M., Makarychev, K., Makarychev, Y.: $O(\sqrt{\log n})$ approximation algorithms for Min UnCut, Min 2CNF Deletion, and directed cut problems. In: Proceedings of STOC, pp. 573–581. ACM (2005)
2. Austrin, P., Khot, S., Safra, M.: Inapproximability of vertex cover and independent set in bounded degree graphs. In: Computational Complexity, CCC, pp. 74–80. IEEE (2009)
3. Bazzi, A., Fiorini, S., Pokutta, S., Svensson, O.: No small linear program approximates vertex cover within a factor $2 - \epsilon$. In: Proceedings of FOCS, pp. 1123–1142 (2015)
4. Braun, G., Pokutta, S.: The matching polytope does not admit fully-polynomial size relaxation schemes. In: Proceedings of SODA, pp. 837–846 (2015). https://doi.org/10.1137/1.9781611973730.57
5. Braun, G., Pokutta, S.: Common information and unique disjointness. Algorithmica **76**(3), 597–629 (2016)
6. Braun, G., Fiorini, S., Pokutta, S., Steurer, D.: Approximation limits of linear programs (beyond hierarchies). In: Proceedings of FOCS, pp. 480–489 (2012). ISBN 978-1-4673-4383-1. https://doi.org/10.1109/FOCS.2012.10
7. Braun, G., Fiorini, S., Pokutta, S.: Average case polyhedral complexity of the maximum stable set problem. In: Proceedings of RANDOM (2014). http://www.dagstuhl.de/dagpub/978-3-939897-74-3. Accessed 12 Jan 2018
8. Braun, G., Fiorini, S., Pokutta, S., Steurer, D.: Approximation limits of linear programs (beyond hierarchies). Math. Oper. Res. (2014). https://doi.org/10.1287/moor.2014.0694
9. Braun, G., Jain, R., Lee, T., Pokutta, S.: Information-theoretic approximations of the nonnegative rank. Comput. Complex. 1–51 (2014). http://eccc.hpi-web.de/report/2013/158
10. Braun, G., Pokutta, S., Zink, D.: Inapproximability of combinatorial problems via small LPs and SDPs. In: Proceeedings of STOC (2015)
11. Braun, G., Brown-Cohen, J., Huq, A., Pokutta, S., Raghavendra, P., Roy, A., Weitz, B., Zink, D.: The matching problem has no small symmetric SDP. In: Proceedings of SODA, pp. 1067–1078 (2016)
12. Braun, G., Fiorini, S., Pokutta, S.: Average case polyhedral complexity of the maximum stable set problem. Math. Program. **160**(1), 407–431 (2016)
13. Braun, G., Pokutta, S., Roy, A.: Strong reductions for extended formulations. In: International Conference on Integer Programming and Combinatorial Optimization, pp. 350–361. Springer (2016)
14. Braverman, M., Moitra, A.: An information complexity approach to extended formulations. In: Proceedings of STOC, pp. 161–170 (2013)
15. Briët, J., Dadush, D., Pokutta, S.: On the existence of 0/1 polytopes with high semidefinite extension complexity. Math. Program. **153**(1), 179–199 (2015)
16. Chan, S., Lee, J., Raghavendra, P., Steurer, D.: Approximate constraint satisfaction requires large LP relaxations. In: Proceedings of FOCS, pp. 350–359 (2013). https://doi.org/10.1109/FOCS.2013.45
17. Chan, S.O.: Approximation resistance from pairwise independent subgroups. In: Proceedings of STOC, pp. 447–456. ACM (2013). ISBN 978-1-4503-2029-0. https://doi.org/10.1145/2488608.2488665
18. Charikar, M., Makarychev, K., Makarychev, Y.: Integrality gaps for Sherali–Adams relaxations. In: Proceedings of STOC, pp. 283–292 (2009)
19. Chawla, S., Krauthgamer, R., Kumar, R., Rabani, Y., Sivakumar, D.: On the hardness of approximating multicut and sparsest-cut. Comput. Complex. **15**(2), 94–114 (2006)
20. Chlebík, M., Chlebíková, J.: On approximation hardness of the minimum 2SAT-DELETION problem. In: Mathematical Foundations of Computer Science 2004, pp. 263–273. Springer (2004)
21. Engebretsen, L., Karpinski, M.: Approximation hardness of TSP with bounded metrics. In: Automata, Languages and Programming, volume 2076 of Lecture Notes in Computer Science, pp. 201–212. Springer, Berlin (2001). ISBN 978-3-540-42287-7 (print) and 978-3-540-48224-6 (online). https://doi.org/10.1007/3-540-48224-5_17. http://theory.cs.uni-bonn.de/ftp/reports/cs-reports/2000/85220-CS.pdf. Accessed 12 Jan 2018
22. Feige, U., Goemans, M.: Approximating the value of two power proof systems, with applications to Max 2SAT and Max DICUT. In: Theory of Computing and Systems, 1995. Proceedings, Third Israel Symposium on the, pp. 182–189. IEEE (1995)
23. Feige, U., Goldwasser, S., Lovász, L., Safra, S., Szegedy, M.: Approximating clique is almost NP-complete. In: Proceedings of FOCS, pp. 2–12. IEEE Comput. Soc. Press (1991). ISBN 0-8186-2445-0. https://doi.org/10.1109/SFCS.1991.185341

24. Feige, U., Karpinski, M., Langberg, M.: Improved approximation of Max-Cut on graphs of bounded degree. J. Algorithms **43**(2), 201–219 (2002). ISSN 0196-6774. https://doi.org/10.1016/S0196-6774(02)00005-6. http://www.paradise.caltech.edu/~mikel/papers/deg_cut.ps
25. Fiorini, S., Massar, S., Pokutta, S., Tiwary, H.R., de Wolf, R.: Linear vs. semidefinite extended formulations: exponential separation and strong lower bounds. In: Proceedings of STOC (2012)
26. Goemans, M.X., Williamson, D.P.: New 3/4-approximation algorithms for the maximum satisfiability problem. SIAM J. Discrete Math. **7**(4), 656–666 (1994)
27. Goemans, M.X., Williamson, D.P.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. J. Assoc. Comput. Mach. **42**, 1115–1145 (1995). https://doi.org/10.1145/227683.227684
28. Göös, M., Jain, R., Watson, T.: Extension complexity of independent set polytopes. In: Proceedings of FOCS, pp. 565–572. IEEE (2016)
29. Gouveia, J., Parrilo, P.A., Thomas, R.R.: Lifts of convex sets and cone factorizations. Math. Oper. Res. **38**(2), 248–264 (2013)
30. Håstad, J.: Some optimal inapproximability results. J. ACM (JACM) **48**(4), 798–859 (2001)
31. Kaibel, V., Weltge, S.: A short proof that the extension complexity of the correlation polytope grows exponentially. Discrete Comput. Geom. **53**(2), 397–401 (2015)
32. Kann, V., Khanna, S., Lagergren, J., Panconesi, A.: On the hardness of approximating Max-k-CUT and its dual. Chic. J. Theor. Comput. Sci. **2**, 1–18 (1997)
33. Karloff, H.: How good is the Goemans–Williamson MAX CUT algorithm? SIAM J. Comput. **29**(1), 336–350 (1999). https://doi.org/10.1137/S0097539797321481
34. Karpinski, M., Lampis, M., Schmied, R.: New inapproximability bounds for TSP. In: Algorithms and Computation, volume 8283 of Lecture Notes in Computer Science, pp. 568–578. Springer, Berlin (2013). ISBN 978-3-642-45029-7 (print) and 978-3-642-45030-3 (online). https://doi.org/10.1007/978-3-642-45030-3_53
35. Khanna, S., Sudan, M., Trevisan, L., Williamson, D.P.: The approximability of constraint satisfaction problems. SIAM J. Comput. **30**(6), 1863–1920 (2001)
36. Khot, S., Kindler, G., Mossel, E., O'Donnell, R.: Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? SIAM J. Comput. **37**(1), 319–357 (2007). https://doi.org/10.1137/S0097539705447372
37. Kothari, P., Meka, R., Raghavendra, P.: Approximating rectangles by juntas and weakly-exponential lower bounds for LP relaxations of CSPs. arXiv preprint arXiv:1610.02704 (2016)
38. Lee, J., Raghavendra, P., Steurer, D.: Lower bounds on the size of semidefinite programming relaxations. In: Proceedings of STOC (2015)
39. Papadimitriou, C.H., Yannakakis, M.: Optimization, approximation, and complexity classes. J. Comput. Syst. Sci. **43**(3), 425–440 (1991)
40. Pashkovich, K.: Extended formulations for combinatorial polytopes. Ph.D. thesis, Magdeburg Universität (2012)
41. Rothvoß, T.: Some 0/1 polytopes need exponential size extended formulations. Math. Program. **142**(1–2), 255–268 (2013)
42. Rothvoß, T.: The matching polytope has exponential extension complexity. In: Proceedings of STOC (2014)
43. Schoenebeck, G.: Linear level Lasserre lower bounds for certain k-CSPs. In: Proceedings of FOCS, pp. 593–602 (2008). https://doi.org/10.1109/FOCS.2008.74
44. Schrijver, A.: Theory of Linear Programming. Wiley-Interscience, New York (1986)
45. Singh, M.: Bellairs workshop on approximation algorithms. Open Probl. Sess. **1**, 2 (2010)
46. Trevisan, L.: Parallel approximation algorithms by positive linear programming. Algorithmica **21**(1), 72–88 (1998). ISSN 0178-4617 (print) and 1432-0541 (online). https://doi.org/10.1007/PL00009209
47. Tulsiani, M.: CSP gaps and reductions in the Lasserre hierarchy. In: Proceedings of STOC, pp. 303–312. ACM (2009)
48. Yannakakis, M.: Expressing combinatorial optimization problems by linear programs (extended abstract). In: Proceedings of STOC, pp. 223–228 (1988)
49. Yannakakis, M.: Expressing combinatorial optimization problems by linear programs. J. Comput. Syst. Sci. **43**(3), 441–466 (1991). https://doi.org/10.1016/0022-0000(91)90024-Y