How to Round Any CSP

Prasad Raghavendra*, David Steurer†,
*Microsoft Research New England
Cambridge, MA 02142
†Department of Computer Science
Princeton University, Princeton, NJ 08540-5233

Abstract— A large number of interesting combinatorial optimization problems like Max Cut, Max k-Sat, and Unique Games fall under the class of constraint satisfaction problems (CSPs). Recent work [32] by one of the authors identifies a semidefinite programming (SDP) relaxation that yields the optimal approximation ratio for every CSP, under the Unique Games Conjecture (UGC). Very recently [33], the authors also showed unconditionally that the integrality gap of this basic SDP relaxation cannot be reduced by adding large classes of valid inequalities (e.g., in the fashion of Sherali–Adams LP hierarchies).

In this work, we present an efficient rounding scheme that achieves the integrality gap of this basic SDP relaxation for every CSP (and by [33] it also achieves the gap of much stronger SDP relaxations). The SDP relaxation we consider is stronger or equivalent to any relaxation used in literature to approximate CSPs. Thus, irrespective of the truth of the UGC, our work yields an efficient generic algorithm that for every CSP, achieves an approximation at least as good as the best known algorithm in literature.

The rounding algorithm in this paper can be summarized succinctly as follows: Reduce the dimension of SDP solution by random projection, discretize the projected vectors, and solve the resulting CSP instance by brute force! Even the proof is simple in that it avoids the use of the machinery from unique games reductions such as dictatorship tests, Fourier analysis or the invariance principle.

A common theme of this paper and the subsequent paper [33] is a *robustness lemma* for SDP relaxations which asserts that approximately feasible solutions can be made feasible by "smoothing" without changing the objective value significantly.

Keywords-semidefinite programming, approximation algorithms, rounding algorithm, integrality gap, dimension reduction, constraint satisfaction problem.

1. Introduction

A vast majority of approximation algorithms involve two distinct phases—relaxation and rounding. More precisely, given an instance $\Im$ of a combinatorial optimization problem Γ, most approximation algorithms consist of the following two stages:

Relaxation: A convex relaxation $\Im_{\text{relax}}$ (linear or semidefinite) of the instance $\Im$ is constructed. These relaxations can be optimized efficiently using interior point methods [36], [30]. Being a relaxation, the optimum to $\Im_{\text{relax}}$ is trivially at least as good (larger for maximization, smaller otherwise) as the optimum to $\Im$. Hence, the value of the optimum solution to $\Im_{\text{relax}}$ serves as a bound on the optimum value for the original instance $\Im$. The integrality gap of the relaxation is the maximum possible gap over all instances of the problem Γ, between the optimum of the original instance $\Im$ and that of the relaxation $\Im_{\text{relax}}$.

Rounding: In this step, the optimal solution to the convex relaxation $\Im_{\text{relax}}$ is used to obtain a solution to the original instance $\Im$. By exhibiting a solution to $\Im$ which is within an α factor of the solution to $\Im_{\text{relax}}$, the rounding yields an α factor approximation guarantee. Further, the rounding algorithm serves as a constructive proof that the integrality gap of the relaxation is at most α. Hence, this is easily the most difficult step requiring ingenuity and techniques from linear programming duality, metric embeddings and other areas. We will call a rounding scheme to be *optimal* for the relaxation, if it achieves an approximation guarantee equal to the integrality gap of the relaxation. In other words, an optimal rounding scheme extracts the best solution that could possibly be obtained from the relaxation.

Among the relaxation techniques, perhaps the most general and powerful is semidefinite programming. A semidefinite program consists of vector valued variables, with linear constraints on their inner products. The objective being optimized is a linear function of the inner products of the variables. Since the seminal work of Goemans and Williamson [15], SDPs have fueled some of the major advances in approximation algorithms. They have found application to problems ranging from constraint satisfaction problems [6], [27], [20], [17], [5], [10] to vertex coloring [19], [2], [9], [11], vertex ordering [7], [12] to graph decomposition [13], [3], and discrete optimization [29], [8], [1], [24].

Despite all the successes, rounding the solution of a semidefinite program remains a difficult task. Contrast this to linear programming which has seen the development of primal-dual [35] and iterative rounding techniques [18], [34], leading to simple combinatorial algorithms. Part of the problem is that the approximation ratios involved in SDP based algorithms are irrational numbers stemming from the geometry of vectors. Even for problems like Max 3-Sat where the optimal approximation ratio is a simple fraction like $\frac{7}{8}$ [20], the analysis of the rounding procedure is fairly

* Work done while the author was at University of Washington, and visiting Carnegie Mellon University. Supported by NSF grant CCF-0343672.
† Research supported by NSF grants 0830673, 0832797, 528414.

involved. In this work, we study the problem of rounding a natural SDP relaxation for general constraint satisfaction problems.

In a constraint satisfaction problem (CSP), the objective is to find an assignment to a set of variables that satisfies the maximum number of a given set of constraints on them. Formally, a CSP Λ is specified by a set of predicates over a finite domain $[q] = \{1, \ldots, q\}$. Every instance of the problem Λ consists of a set of variables V, and a set of constraints on them. Each constraint consists of a predicate from Λ applied to a subset of variables. The objective is to find an assignment to the variables that satisfies the maximum number of constraints. More generally, the predicates can be replaced by bounded real valued payoff functions while the objective is to maximize the total payoff. A large number of the fundamental optimization problems such as Max Cut and Max k-Sat are examples of CSPs.

Beginning with the work of Goemans and Williamson [15] on the Max Cut, semidefinite programs have been instrumental in approximation algorithms for several well known CSPs like Max 2-Sat [6], [27], Max 3-Sat [20], Max 4-Sat [16] , Max k-CSP [37], [17], [6], Max DiCut [28], Max Cut Gain [8] and Unique Games [5], [10]. Underlying all these works on seemingly diverse CSPs lies the simplest semidefinite relaxations for the problems. Even though the rounding techniques vary from one problem to another, the SDP relaxation involved in all these algorithms is roughly the same.

In fact, the Unique games conjecture of Khot [21] implies that indeed the simplest semidefinite relaxation yield the optimal approximation ratios for CSPs. While several UG hardness results for problems like Max Cut [23], [25], [31], Max 2-Sat [22], [4] suggested such an implication, its full generality was realized in recent works by Austrin [4] and one of the authors [32]. Specifically, it was shown in [32] that a simple generic SDP relaxation yields the optimal approximation for every CSP assuming the Unique Games Conjecture. Henceforth, we shall use $\mathrm{SDP}_{\mathrm{gen}}$ to denote this generic relaxation, which is either equivalent or stronger than every SDP relaxation used for an algorithm in literature.

Surprisingly, this pursuit for UG-hardness results has led to new rounding schemes for CSPs. The connection between rounding schemes and UG hardness was first pointed out in the work of Austrin [4] on Boolean 2-CSPs. Fleshing out this connection in its full generality, [32] designed rounding schemes that achieve the optimal approximation ratio for every CSP assuming the Unique Games Conjecture. Consequently, assuming the Unique Games conjecture, for every CSP, it is clear what the optimal SDP relaxation is, and how to round it.

Yet, the situation is not entirely satisfactory, since the optimality of these generic rounding schemes relies on the Unique games conjecture. In other words, if the Unique games conjecture were to be false, there would be no guarantee on the performance of generic rounding scheme in [32]. Towards rectifying this, the work [32] also obtains an unconditional guarantee on the rounding scheme for the case of 2-CSPs. Specifically, for every 2-CSP, irrespective of the truth of UGC, the generic rounding scheme in [32] achieves an approximation equal to the integrality gap of the SDP relaxation $\mathrm{SDP}_{\mathrm{gen}}$. In a related work, O'Donnell and Wu [31] obtained rounding schemes that achieve the integrality gap of the relaxation unconditionally for the Max Cut problem.

For general CSPs with arity greater than 2, there are no known rounding schemes that are optimal for the relaxation $\mathrm{SDP}_{\mathrm{gen}}$, unless one assumes UGC. To show unconditional guarantees, the approach of [32] relies on the Khot–Vishnoi integrality gap [26] for unique games. Extending this approach to CSPs of arity greater than 2, is tied to the problem of constructing integrality gaps for stronger SDP relaxations of unique games. Specifically, extending the unconditional result of [32] to a CSP of arity r would require an SDP integrality gap for roughly r-rounds of any of the SDP hierarchies.

Integrality gaps for such strong SDP relaxations of Unique Games were not known, when this work was concieved. In subsequent work [33], the authors used techniques from this paper to construct integrality gaps for strong SDP relaxations of Unique Games. However, the resulting rounding scheme from [32] remains very complex, with its proof relying on a web of reductions between integrality gaps, UG hardness results and dictatorship tests.

1.1. Results

In this work, we design a generic rounding scheme that achieves the integrality gap of the relaxation $\mathrm{SDP}_{\mathrm{gen}}$ for every CSP unconditionally. To state our result precisely, we need to define the SDP integrality gap curve $S_\Lambda(c)$ for a CSP Λ. Let $\mathrm{sdp}(\mathcal{P})$ denote the objective value of an optimal solution for the $\mathrm{SDP}_{\mathrm{gen}}$ relaxation of an instance $\mathcal{P}$. Let $\mathrm{opt}(\mathcal{P})$ denote the value of the optimal solution to $\mathcal{P}$. The integrality gap curve $S_\Lambda(c)$ is the minimum value of $\mathrm{opt}(\mathcal{P})$, given that $\mathrm{sdp}(\mathcal{P}) = c$ where the minimum is over all instances $\mathcal{P}$ of the problem Λ. Formally,

$$S_\Lambda(c) = \inf_{\mathcal{P}\in\Lambda, \mathrm{sdp}(\mathcal{P})=c} \mathrm{opt}(\mathcal{P}) .$$

Theorem 1.1: For every CSP Λ and for every $\varepsilon > 0$, there exists a polynomial time approximation algorithm for Λ-CSP that returns an assignment of value at least $S_\Lambda(c-\varepsilon) - \varepsilon$ on an instance Φ with SDP value c. The algorithm runs in time $\exp(\exp(\mathrm{poly}(kq/\varepsilon)))$.

The above result also holds in the more general setting where predicates are replaced by bounded real valued payoff functions. For a traditional CSP Λ consisting of predicates, the above theorem implies the following corollary.

Corollary 1.2: Given a Λ-CSP with non-negative valued payoff functions, for every $\varepsilon > 0$, there exists a polynomial time approximation algorithm for Λ-CSP with approximation ratio at most the integrality gap α_Λ defined as,

$$\alpha_\Lambda \stackrel{\text{def}}{=} \sup_{\mathcal{P}} \frac{\text{sdp}(\mathcal{P})}{\text{opt}(\mathcal{P})}\,.$$

The algorithm runs in time $\exp(\exp(\text{poly}(kq/\varepsilon)))$.

As already pointed out, the relaxation SDP_{gen} is either equivalent to or stronger than every SDP relaxation used to approximate a CSP. Hence, this work yields a generic algorithm that for every CSP, achieves an approximation ratio at least as good as the best known algorithm in literature.

On the downside, the proof of optimality of the rounding scheme is non-explicit. To show the optimality of the rounding scheme, we proceed as follows: given an instance $\mathfrak{I}$ on which the rounding scheme only achieves an α approximation, we exhibit an instance on which the integrality gap of the SDP is at least α. In particular, this yields no information on the approximation ratio α achieved by the rounding scheme. However, the techniques in this work yield an algorithm to compute the integrality gap of SDP_{gen} for any given CSP Λ.

Theorem 1.3: For every constant $\varepsilon > 0$ and every CSP Λ, the integrality gap curve $S_\Lambda(c)$ can be computed to an additive approximation of ε in time $\exp(\exp(\text{poly}(kq/\varepsilon)))$.

Not only does our approach bypass the need for strong SDP gaps for unique games, but it is in fact surprisingly simple. Underlying this work are two ideas: dimension reduction and discretization of SDP vectors. In fact, this work does not make use of any of the machinery from UG hardness results like dictatorship tests, Fourier analysis, Hermite analysis or the invariance principle.

2. Proof Overview

Underlying this work are two surprisingly simple ideas: dimension reduction and discretization of SDP vectors. In this section, we elucidate how these are employed to obtain rounding schemes for CSPs. We begin by describing the generic SDP relaxation SDP_{gen} for a well known CSP - Max3SAT. Fix a Max3SAT instance Φ consisting of variables $V = \{y_1, \ldots, y_n\}$ and clauses $\mathcal{P} = \{P_1, \ldots, P_m\}$. The variables in SDP_{gen} are as follows:

- For each variable y_i, introduce two vector variables $\boldsymbol{v}_i = \{\boldsymbol{v}_{(i,0)}, \boldsymbol{v}_{(i,1)}\}$. In the intended solution, the assignment $y_i = 1$ is represented by $\boldsymbol{v}_{(i,0)} = 0$ and $\boldsymbol{v}_{(i,1)} = 1$, while $y_i = 0$ implies $\boldsymbol{v}_{(i,0)} = 1, \boldsymbol{v}_{(i,1)} = 0$.
- For each clause we will introduce 8 variables to denote the 8 different states possible. For instance, with the clause $P = (y_1 \vee y_2 \vee y_3)$ we shall associate 8 variables $\mu_P = \{\mu_{(P,000)}, \mu_{(P,001)}, \ldots, \mu_{(P,111)}\}$. In general, the variables μ_P form a probability distribution locally over integral solutions.
- Let $\boldsymbol{v}_0$ denote a variable representing the constant 1.

The relaxation SDP_{gen} has the minimal set of constraints necessary to ensure that for every clause $P \in \mathcal{P}$, the following hold: Firstly, μ_P is a valid probability distribution over local assignments ($\{0,1\}^3$). Further, the inner products of the vectors corresponding to variables in P match the distribution μ_P. The objective value to be maximized can be written in terms of the local integral distributions μ_P as follows:

$$\sum_{P \in \mathcal{P}} \sum_{x \in \{0,1\}^3} P(x)\mu_{P,x}$$

We wish to point out that the relaxation SDP_{gen} is an extremely minimal SDP relaxation. For instance, if two variables y_i, y_j do not occur in a clause together, then SDP_{gen} does not impose any constraints on the inner products of the corresponding vectors $\{\boldsymbol{v}_{(i,0)}, \boldsymbol{v}_{(i,1)}, \boldsymbol{v}_{(j,0)}, \boldsymbol{v}_{(j,1)}\}$. Specifically, the inner product of $\boldsymbol{v}_{(i,0)}$ and $\boldsymbol{v}_{(j,0)}$ could take negative values in a feasible solution. While this might suggest that SDP_{gen} is too weak a relaxation, recall that no stronger relaxation has been used to approximate a CSP yet, and indeed no stronger relaxation helps, under the Unique games conjecture.

Given the SDP solution to the instance Φ, we construct a constant sized Max 3-Sat instance Φ' which serves as a model for Φ. More specifically, we construct a partition $S_1 \cup S_2 \cup \ldots S_m = V$ of the set of variables V in to m subsets for some constant m. The instance Φ' is over m variables $\{s_1, s_2, \ldots, s_m\}$ corresponding to subsets $S_1, \ldots, S_m$. Essentially, the instance Φ' is obtained by merging all the variables in each of the sets S_i to a corresponding variable s_i. We will refer to Φ' as a *folding* of the instance Φ.

Observe that any assignment $\mathcal{A}'$ to Φ' yields a corresponding assignment $\mathcal{A}$ to Φ by simple unfolding, i.e., assign $\mathcal{A}(y_j) = \mathcal{A}'(s_i)$ for every variable y_j in the set S_i. Clearly, the fraction of clauses satisfied by assignment $\mathcal{A}$ on Φ is exactly the same as those satisfied by $\mathcal{A}'$ on Φ'. Observe that any folding operation immediately yields a rounding scheme— "Find the optimal assignment to Φ' by brute force, and unfold it to an assignment for Φ."

To show the optimality of this scheme, the crucial property we require of the *folding* operation is that it preserves the SDP value. Clearly, any *folding* operation can only decrease the value of the optimum for the SDP relaxation, i.e, $\text{sdp}(\Phi') \leqslant \text{sdp}(\Phi)$. We will exhibit a folded instance Φ' such that $\text{sdp}(\Phi') \approx \text{sdp}(\Phi)$. More precisely, we will exhibit a folded instance Φ' with approximately the same clauses as Φ, and roughly the same SDP value. Such a folded instance Φ' will serve as a certificate for the optimality of the rounding scheme. Recall that the folded instance is an integrality gap instance with a SDP value $\text{sdp}(\Phi') \approx \text{sdp}(\Phi)$ and optimum value $\text{opt}(\Phi')$. By definition, the scheme returns an assignment of value $\text{opt}(\Phi')$ on the instance Φ with SDP value $\text{sdp}(\Phi)$. Thus the rounding scheme achieves an approximation no worse than the integrality gap of the SDP.

At this juncture, we would like to draw a parallel between this approach and the work of Frieze–Kannan [14] on approximating dense instances of NP-hard problems. Given a dense instance of Max Cut, they construct a finite model that *approximates* the instance using the Szemerédi Regularity lemma. This finite model is nothing but a *folding* of the instance that preserves the optimum value for Max Cut. In contrast, we construct a finite model for arbitrary instances that need not be dense, while preserving an arguably simpler property— the SDP optimum.

Summarizing the discussion, the problem of rounding has been reduced to finding an algorithm to merge variables in the instance in to a few clusters, while preserving the SDP value. Intuitively, the most natural way to preserve the SDP value would be to merge variables whose SDP vectors are close to each other. In other words, we would like to cluster the SDP vectors $\{\boldsymbol{v}_{(i,b)}\}$ into a constant number of clusters. A first attempt at such a clustering would be as follows: partition the ambient space in to bins of diameter at most ε, and merge all the SDP vectors that fall in to the same bin. The number of clusters created is at most the number of bins in the partition.

In general, the optimum SDP vectors $\{\boldsymbol{v}_{(i,b)}\}$ lie in a space of dimension equal to the number of variables in the SDP (say n). A partition of the n-dimensional sphere in to bins of diameter at most ε, would require roughly $(1/\varepsilon)^n$ bins, while our goal is to use a constant number of bins. Simply put, there is little chance that n vectors in a n-dimensional space are clustered in to a few clusters. To address this issue, we pursue the most natural approach: first perform a dimension reduction on the SDP vectors by using random projections, and then cluster them together.

Heuristically, for large enough constant d, when projected in to a random d-dimensional space, at least $1-\varepsilon$ fraction of the inner products would change by at most ε. Further, merging variables within the same bin of diameter ε, could affect the inner products by at most ε. Thus the SDP value of the folded instance should be within $O(\varepsilon)$ of the original SDP value. The number of variables in the folded instance would be $(1/\varepsilon)^d$— a constant.

Making the above heuristic argument precise forms the technical core of the paper. While this is easy for some 2-CSPs like Max Cut, extending it to CSPs of larger arity and alphabet size is non-trivial. The central issue to be addressed is how does one respect all the constraints of the SDP during dimension reduction. In fact, for stronger SDP relaxations such as the one in [3], it is unclear whether a dimension reduction can be carried out at all. For a subset of CSP variables involved in a constraint P, the $\mathrm{SDP}_{\mathrm{gen}}$ relaxation requires the inner products of the corresponding SDP vectors to be consistent with a local integral distribution μ_P. This translates in to the SDP vectors satisfying special constraints amongst themselves. For instance, even for a CSP of arity 3 such as Max 3-Sat, this implies the triangle inequalities on every 3-tuple of variables involved in a clause.

To make the argument precise, we use the smoothing operation defined in [32] which in some sense introduces noise to the SDP vectors. Interestingly, the smoothing operation was applied for an entirely different purpose in [32]. For every CSP instance, there is a canonical SDP solution $\{\boldsymbol{u}_{(i,b)}\}$ corresponding to the uniform distribution over all possible integral solutions. Given an arbitrary SDP solution $\{\boldsymbol{v}_{(i,b)}\}$, the ε-smoothed solution is defined by $\boldsymbol{v}^*_{(i,b)} = \sqrt{1-\varepsilon}\,\boldsymbol{v}_{(i,b)} \oplus \sqrt{\varepsilon}\,\boldsymbol{u}_{(i,b)}$, where $\oplus$ denotes the direct sum of vectors. Clearly, the SDP objective value changes by at most $O(\varepsilon)$ due to smoothing. We observe that if the vectors $\{\boldsymbol{v}_{(i,b)}\}$ are close to satisfying a valid inequality (say triangle inequality) approximately, then by smoothing, the new solution $\{\boldsymbol{v}^*_{(i,b)}\}$ satisfies the inequality. We present a separate argument to handle the equality constraints in the SDP.

In the original instance Φ, for every clause P, the inner products of the vectors involved match a local integral distribution μ_P. After random projection and discretization, for at least $1-\varepsilon$ fraction of the clauses in Φ, the corresponding inner products match a local integral distribution up to an error ε. Let us refer to these $1-\varepsilon$ fraction of the clauses as *good*. Apply the smoothing operation on the discretized SDP solution. For each *good* clause, the smoothed SDP solution is consistent with a local integral distribution. To finish the argument, we discard the ε-fraction of the *bad* clauses from the folded instance Φ'. By the definition of $\mathrm{SDP}_{\mathrm{gen}}$, once a *bad* clause P is dropped from the instance, it is no longer necessary to satisfy the SDP constraints corresponding to P. Hence, we conclude $\mathrm{sdp}(\Phi') \approx \mathrm{sdp}(\Phi)$.

3. Preliminaries

3.1. Constraint Satisfaction Problems

In this work, we consider a generalization of constraint satisfaction problems where we allow payoff functions taking values in $[-1,1]$, instead of predicates taking values in $\{0,1\}$. Formally, let Λ be a family of *payoff functions* $P\colon [q]^k \to [-1,1]$. We say Λ has *arity* k and *alphabet* $[q] \stackrel{\text{def}}{=} \{1,\dots,q\}$. A function $P'\colon [q]^V \to [-1,1]$ has *type* Λ if for some $P \in \Lambda$ and some $i_1,\dots,i_k \in V$, we have $P'(x) = P(x_{i_1},\dots,x_{i_k})$ for all $x \in [q]^V$. We define $V(P') \subseteq V$ to be the set of coordinates that P' depends on. In other words, if $P'(x) = P(x_{i_1},\dots,x_{i_k})$, then $V(P') = \{i_1,\dots,i_k\}$. In particular, $|V(P')| \leqslant k$ for any function P' of type Λ. A Λ-*CSP instance* $\mathcal{P}$ *with variable set* V is a distribution over payoff functions $P\colon [q]^V \to [-1,1]$ of type Λ.

Problem 3.1 (Λ-ConstraintSatisfactionProblem *(CSP)):* Given a variable set V and a distribution $\mathcal{P}$ over payoff functions $P\colon [q]^V \to [-1,1]$ of type Λ, the goal is to find an assignment $x \in [q]^V$ so as to maximize $\mathrm{E}_{P\sim\mathcal{P}}\, P(x)$. We define the value $\mathrm{opt}(\mathcal{P})$ as

$$\mathrm{opt}(\mathcal{P}) \stackrel{\text{def}}{=} \max_{x\in[q]^V} \mathop{\mathrm{E}}_{P\sim\mathcal{P}} P(x)\,.$$

Observe that if the payoff functions P are predicates, then maximizing the payoff amounts to maximizing the number of constraints satisfied.

3.2. SDP Relaxation

We consider the following relaxation for Λ-CSP: Given an instance $\mathcal{P}$ with variable set V, the goal is to find a collection of vectors $\{\boldsymbol{v}_{i,a}\}_{i\in V, a\in[q]} \subseteq \mathbb{R}^d$ and a collection $\{\mu_P\}_{P\in\mathrm{supp}(\mathcal{P})}$ of distributions over local assignments. Each distribution μ_P is over $[q]^{V(P)}$ (the set of assignments to the variable set $V(P)$). We will write $\Pr_{x\in\mu_P}\{E\}$ to denote the probability of an event E under the distribution μ_P.

Semidefinite Relaxation $\mathrm{SDP}_{\mathrm{gen}}$

$$\text{maximize} \quad \mathop{\mathbb{E}}_{P\sim\mathcal{P}} \mathop{\mathbb{E}}_{x\sim\mu_P} P(x) \tag{1}$$

$$\text{subject to} \quad \langle \boldsymbol{v}_{i,a}, \boldsymbol{v}_{j,b}\rangle = \Pr_{x\sim\mu_P}\left\{x_i = a,\ x_j = b\right\} \tag{2}$$

$$(P \in \mathrm{supp}(\mathcal{P})\ i,j \in V(P),\ a,b\in[q])\,,$$

$$\langle \boldsymbol{v}_{i,a}, \boldsymbol{v}_0\rangle = \Pr_{x\sim\mu_P}\left\{x_i = a\right\} \tag{3}$$

$$(P \in \mathrm{supp}(\mathcal{P}),\ i \in V(P),\ a\in[q])\,.$$

Here $\boldsymbol{v}_0$ can be any fixed unit vector in $\mathbb{R}^d$, and d can be any sufficiently large number, say $d = q|V|$.

We denote by $\mathrm{sdp}(\mathcal{P})$ the maximum value of an SDP solution for $\mathcal{P}$. Clearly, $\mathrm{sdp}(\mathcal{P}) \geqslant \mathrm{opt}(\mathcal{P})$.

We claim that the optimization problem $\mathrm{SDP}_{\mathrm{gen}}$ is (equivalent to) a semidefinite program of polynomial size and thus it can be solved in polynomial time (to arbitrary accuracy). To show this claim, let us introduce additional real-valued variables $\mu_{P,x}$ for $P\in\mathrm{supp}(\mathcal{P})$ and $x\in[q]^{V(P)}$. We add the constraints $\mu_{P,x}\geqslant 0$ and $\sum_{x\in[q]^{V(P)}}\mu_{P,x} = 1$. We can now make the following substitutions to eliminate the distributions μ_P,

$$\mathop{\mathbb{E}}_{x\sim\mu_P} P(x) = \sum_{x\in[q]^{V(P)}} P(x)\mu_{P,x}\,,$$

$$\Pr_{x\sim\mu_P}\left\{x_i = a\right\} = \sum_{\substack{x\in[q]^{V(P)}\\ x_i=a}} \mu_{P,x}\,,$$

$$\Pr_{x\sim\mu_P}\left\{x_i = a, x_j = b\right\} = \sum_{\substack{x\in[q]^{V(P)}\\ x_i=a,x_j=b}} \mu_{P,x}\,.$$

After substituting the distributions μ_P by the scalar variables $\mu_{P,x}$, we are left with a linear optimization problem over the cone of positive semidefinite matrices—an SDP—of size $\mathrm{poly}(q^k, |\mathrm{supp}(\mathcal{P})|)$.

4. Rounding General CSPs

Variable folding: Let $\mathcal{P}$ be a Λ-CSP instance with variable set $V = [n]$. For a mapping $\varphi\colon V \to W$, we define a new Λ-CSP instance $\mathcal{P}/\varphi$ on the variable set W by identifying variables of $\mathcal{P}$ that get mapped to the same variable in W. Formally, the payoff functions in $\mathcal{P}/\varphi$ are of the form $P(x_{\phi(1)},\ldots,x_{\phi(n)})$ for $x\in[q]^W$. Since any assignment for $\mathcal{P}/\phi$ corresponds to an assignment for $\mathcal{P}$, we can note the following fact.

Fact 4.1: $\mathrm{opt}(\mathcal{P}) \geqslant \mathrm{opt}(\mathcal{P}/\phi)$.

In general, the optimal value of the folded instance might be significantly lower than the optimal value of the original instance. However, we will show that we can always find a variable folding that approximately preserves the SDP value (of an instance that is close to the original instance).

Theorem 4.2: Given $\varepsilon > 0$ and a Λ-CSP instance $\mathcal{P}$, we can efficiently compute another Λ-CSP instance $\mathcal{P}'$ and a variable folding ϕ such that

1) $\mathcal{P}'$ is obtained by discarding an ε fraction of payoffs from the instance $\mathcal{P}$. Formally, $V(\mathcal{P}') = V(\mathcal{P})$ and $\|\mathcal{P} - \mathcal{P}'\|_1 \leqslant \varepsilon$,
2) $\mathrm{sdp}(\mathcal{P}'/\phi) \geqslant \mathrm{sdp}(P) - \varepsilon$,
3) the variable set of $\mathcal{P}'/\phi$ has cardinality $\exp(\mathrm{poly}(kq/\varepsilon))$.

Given the above theorem, we can immediately show the main results of the paper.

Proof of Theorem 1.1: Given a Λ-CSP instance $\mathcal{P}$ with variable set $V = [n]$, we first compute another instance $\mathcal{P}'$ and a variable folding ϕ according to Lemma 4.2. Since $\mathcal{P}'/\phi$ has only $\exp(\mathrm{poly}(kq/\varepsilon))$ variables, we can compute an optimal assignment for $\mathcal{P}'/\phi$ in time $\exp(\exp(\mathrm{poly}(kq/\varepsilon)))$. This assignment can be unfolded to an assignment $x\in[q]^n$ with the same value for $\mathcal{P}'$. Since $\|\mathcal{P}-\mathcal{P}'\|_1 \leqslant \varepsilon$, the assignment x has value at least $\mathrm{opt}(\mathcal{P}'/\phi)-\varepsilon$ for the instance $\mathcal{P}$. By definition of S_Λ, we have $\mathrm{opt}(\mathcal{P}'/\Phi) \geqslant S_\Lambda(\mathrm{sdp}(\mathcal{P}'/\Phi)) \geqslant S_\Lambda(\mathrm{sdp}(\mathcal{P})-\varepsilon)$. Hence the assignment $x\in[q]^n$ has value at least $S_\Lambda(\mathrm{sdp}(\mathcal{P})-\varepsilon)-\varepsilon$ as claimed. ■

Proof of Theorem 1.3: By Theorem 4.2, to compute the SDP integrality gap within ε, it is sufficient to go over all instances of size $\exp(\mathrm{poly}(kq/\varepsilon))$. Thus the algorithm would just discretize the space of instances with $\exp(\mathrm{poly}(kq/\varepsilon))$ many variables, and compute the SDP and optimum value for each instance. ■

The rest of this section is devoted to the proof of Theorem 4.2. The construction of $\mathcal{P}'/\phi$ is described below:

Construction of $\mathcal{P}'/\phi$

Dimension reduction: Let $\{\boldsymbol{v}_{i,a}\}_{i\in V,\ a\in[q]}$, $\{\mu_P\}_{P\in\mathrm{supp}(\mathcal{P})}$ be an SDP solution for a Λ-CSP instance $\mathcal{P}$ on the variable set $V = [n]$. Suppose $\boldsymbol{v}_{i,a}\in\mathbb{R}^D$. We apply the following procedure to reduce the dimension from D to d.

1) Sample a $d\times D$ Gaussian matrix $\boldsymbol{\Phi}$, where each entry is independently distributed according to the Gaussian distribution $N(0, 1/d)$.
2) For every vector $\boldsymbol{v}_{i,a}$, compute its image $\boldsymbol{u}_{i,a}$ under the map $\boldsymbol{\Phi}$,
$$\boldsymbol{u}_{i,a} \stackrel{\mathrm{def}}{=} \boldsymbol{\Phi}\boldsymbol{v}_{i,a}\,.$$
Furthermore, define $\boldsymbol{u}_0 := \boldsymbol{\Phi}\boldsymbol{v}_0$.

Discarding bad constraints: Let $B_\varepsilon \subseteq \mathrm{supp}(\mathcal{P})$ be the set of payoff functions P such that the vectors $\boldsymbol{u}_{i,a}$ and the distributions μ_P violate one of the SDP constraints corresponding to P by more than ε. Define the instance $\mathcal{P}'$

on the set of variables V by removing all payoff functions in B_ε from $\mathcal{P}$. Formally, $\mathcal{P}'$ is obtained by conditioning the distribution $\mathcal{P}$ on the event $P \notin B_\varepsilon$.

Folding by Discretization: Let N be an ε-net for the unit ball in $\mathbb{R}^d$. We have $|N| \leqslant (c/\varepsilon)^d$ for some absolute constant c. For every vector $\boldsymbol{u}_{i,a}$, let $\boldsymbol{w}_{i,a}$ denote its closest vector in N. We identify variables of $\mathcal{P}'$ that have the same vectors $\boldsymbol{w}_{i,a}$. Formally, we output the Λ-CSP instance $\mathcal{P}'/\phi$ where $\phi\colon V \to N^q$ is defined as

$$\phi(i) \stackrel{\text{def}}{=} (\boldsymbol{w}_{i,1}, \ldots, \boldsymbol{w}_{i,q}) .$$

4.1. Property of Dimension Reduction

The key property of the dimension reduction is that it preserves inner products between vectors approximately.

Lemma 4.3 (Inner products are preserved approximately): For any two vectors $\boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{R}^D$ in the unit ball,

$$\Pr_{\boldsymbol{\Phi}}\left\{\left|\langle \boldsymbol{\Phi}\,\boldsymbol{v}_1, \boldsymbol{\Phi}\,\boldsymbol{v}_2\rangle - \langle \boldsymbol{v}_1, \boldsymbol{v}_2\rangle\right| \geqslant \tfrac{t}{\sqrt{d}}\right\} \leqslant O\left(1/t^2\right) .$$

For the sake of completeness, we present the straightforward proof of this property in Appendix 6.1. It is clear that the dimension-reduced vectors $\boldsymbol{u}_{i,a}$ together with the distributions μ_P need not form a feasible SDP solution. However, we can deduce from Lemma 4.3 that with good probability most of the constraints will be near satisfied. It follows that not too many payoffs are discarded from $\mathcal{P}$ to construct $\mathcal{P}'$.

Lemma 4.4: For every payoff $P \in \operatorname{supp}(\mathcal{P})$,

$$\Pr_{\boldsymbol{\Phi}}\{P \in B_\varepsilon\} \leqslant O\left(\tfrac{k^2q^2}{\varepsilon^2 d}\right) .$$

4.2. Discretization

Consider two vectors $\boldsymbol{u}_{i,a}$ and $\boldsymbol{u}_{j,b}$ in the unit ball. It is clear that if we move the vectors to their closest point in N, their inner product changes by at most 2ε. (Since N is an ε-net of the unit ball, each vector is moved by at most ε.)

A minor technical issue is that some of the points $\boldsymbol{u}_{i,a}$ might be outside of the unit ball. However, vectors of norm more than $\sqrt{1+\varepsilon}$ can be ignored, because they violate the constraint $\langle \boldsymbol{u}_{i,a}, \boldsymbol{u}_{i,a}\rangle \leqslant 1$ by more than ε.

In particular, the following lemma holds.

Lemma 4.5: For small enough $\varepsilon > 0$, suppose the vectors $\boldsymbol{u}_{i,a}$ satisfy all constraints corresponding to some payoff function $P \in \operatorname{supp}(\mathcal{P}')$ up to an error of ε. Then, the vectors $\boldsymbol{w}_{i,a}$ satisfy all constraints corresponding to P up to an error of 4ε.

Here we are using the fact that for each payoff $P \in \operatorname{supp}(\mathcal{P}')$, the corresponding constraints in the relaxation $\operatorname{sdp}(\mathcal{P}')$ involve just a single inner product. We also use the fact the vectors $\boldsymbol{u}_{i,a}$ for a variable $i \in V(P)$ with $P \in \operatorname{supp}(\mathcal{P}')$ have norms at most $\sqrt{1+\varepsilon}$.

4.3. Robustness of SDP Relaxation $\mathrm{SDP}_{\mathrm{gen}}$

To finish the proof of Theorem 4.2, we need to construct a completely feasible SDP solution to $\mathcal{P}'/\Phi$ from the vectors $\boldsymbol{w}_{i,a}$ which nearly satisfy all the constraints.

We will show the following theorem in Section 5.

Theorem 4.6 (Robustness of $\mathrm{SDP}_{\mathrm{gen}}$): Let $\mathcal{P}$ be a Λ-CSP instance on variable set V. Suppose $\{\boldsymbol{v}_{i,a}\}_{i\in V, a\in[q]}$, $\{\mu_P\}_{P\in\operatorname{supp}(\mathcal{P})}$ is an ε-infeasible SDP solution for $\mathcal{P}$ of value α. Here, ε-infeasible means that all constraints (2)–(3) of the relaxation $\mathrm{SDP}_{\mathrm{gen}}$ are satisfied up to an additive error of at most ε. Then,

$$\operatorname{sdp}(\mathcal{P}) \geqslant \alpha - \sqrt{\varepsilon}\cdot \operatorname{poly}(kq) .$$

4.4. Proof of Theorem 4.2

Assuming Theorem 4.6 (Robustness of $\mathrm{SDP}_{\mathrm{gen}}$) we can now complete the proof of Theorem 4.2.

For simplicity, we assume that the SDP solution $\{\boldsymbol{v}_{i,a}\}$, $\{\mu_P\}$ that was used in the construction of $\mathcal{P}'/\phi$ has value $\operatorname{sdp}(\mathcal{P})$. (The proof also works if the value of this SDP solution is close to the optimal value.)

Recall that $B_\varepsilon \subseteq \operatorname{supp}(\mathcal{P})$ is the set of payoff functions P whose constraints are violated by more than ε by the dimension-reduced vectors $\boldsymbol{u}_{i,a}$. For $d \gg k^2q^2/\varepsilon^3$, Lemma 4.4 implies that with high probability, $\|\mathcal{P} - \mathcal{P}'\|_1 \leqslant \varepsilon$. Note that the vectors $\{\boldsymbol{u}_{i,a}\}$ together with the original local distributions $\{\mu_P\}$ form an ε-infeasible SDP solution for $\mathcal{P}'$. Hence, by Lemma 4.5, the SDP solution $\{\boldsymbol{w}_{i,a}\}$, $\{\mu_P\}$ is 4ε-infeasible. The value of this SDP solution for the instance $\mathcal{P}'$ is at least $\operatorname{sdp}(\mathcal{P}) - \|\mathcal{P} - \mathcal{P}'\|_1 \geqslant \operatorname{sdp}(\mathcal{P}) - \varepsilon$. The key observation is now that the SDP solution $\{\boldsymbol{w}_{i,a}\}$, $\{\mu_P\}$ is also a solution for the folded instance $\mathcal{P}'/\phi$. We see that $\mathcal{P}'/\phi$ has a 4ε-infeasible SDP solution of value at least $\operatorname{sdp}(\mathcal{P}) - \varepsilon$. Theorem 4.6 (Robustness of $\mathrm{SDP}_{\mathrm{gen}}$) asserts that in this situation we can conclude $\operatorname{sdp}(\mathcal{P}'/\phi) \geqslant \operatorname{sdp}(\mathcal{P}) - \sqrt{\varepsilon}\cdot\operatorname{poly}(kq)$. Finally, we observe that the cardinality of the variable set of $\mathcal{P}'/\phi$ is at most $|N|^q \leqslant (c/\varepsilon)^{dq} = 2^{\operatorname{poly}(kq/\varepsilon)}$. ∎

5. Surgery & Smoothing

Let $\{\boldsymbol{v}_{i,a}\}$, $\{\mu_P\}$ be an ε-infeasible SDP solution for a Λ-CSP instance $\mathcal{P}$ on the variable set $V = [n]$. Recall that an ε-infeasible SDP solution satisfies for all $P \in \operatorname{supp}(\mathcal{P})$, $i, j \in V(P)$, and $a, b \in [q]$,

$$\left|\langle \boldsymbol{v}_{i,a}, \boldsymbol{v}_{j,b}\rangle - \Pr_{x\sim\mu_P}\left\{x_i = a,\ x_j = b\right\}\right| \leqslant \varepsilon , \tag{4}$$

and for all $i \in V(P)$ and $a \in [q]$,

$$\left|\langle \boldsymbol{v}_{i,a}, \boldsymbol{v}_0\rangle - \Pr_{x\sim\mu_P}\left\{x_i = a\right\}\right| \leqslant \varepsilon . \tag{5}$$

We construct a feasible solution that is close to the given SDP solution in two steps.

In the first step, called “surgery”, we construct vectors $\{\boldsymbol{u}_{i,a}\}$ that satisfy the equality constraints on SDP vectors, i.e., $\langle \boldsymbol{u}_{i,a}, \boldsymbol{u}_{i,b}\rangle = 0$ for all $a \neq b \in [q]$ and all $i \in V$ and $\sum_{a\in[q]} \boldsymbol{u}_{i,a} = \boldsymbol{v}_0$ for all $i \in V$.

In the second step, called “smoothing”, we construct a feasible SDP solution $\{\boldsymbol{w}_{i,a}\}$, $\{\mu'_P\}$. In this step, the vectors and the local distributions are “smoothed” which allows us to modify the local distributions so that they match the vectors perfectly.

Lemma 5.1: The vectors $\{\boldsymbol{v}_{i,a}\}$ can be transformed to vectors $\{\boldsymbol{u}_{i,a}\}$ such that for all $a \neq b \in [q]$ and all $i \in V$,

$$\langle \boldsymbol{u}_{i,a}, \boldsymbol{u}_{i,b} \rangle = 0\,, \tag{6}$$

and for all $i \in V$,

$$\sum_{a \in [q]} \boldsymbol{u}_{i,a} = \boldsymbol{v}_0\,. \tag{7}$$

Furthermore, for $i \in V$ and $a \in [q]$,

$$\|\boldsymbol{u}_{i,a} - \boldsymbol{v}_{i,a}\| \leqslant \sqrt{\varepsilon} \cdot \mathrm{poly}(q)\,. \tag{8}$$

In particular, the SDP solution $\{\boldsymbol{u}_{i,a}\}$, $\{\mu_P\}$ is η-infeasible for $\eta = \sqrt{\varepsilon} \cdot \mathrm{poly}(q)$.

Proof: From (4) it follows that $\|\boldsymbol{v}_{i,a}\|^2 \leqslant 1 + \varepsilon$ and $|\langle \boldsymbol{v}_{i,a}, \boldsymbol{v}_{i,b} \rangle| \leqslant \varepsilon$. for all $a \neq b \in [q]$. Therefore, if we apply the Gram–Schmidt orthogonalization process on the vectors $\boldsymbol{v}_{i,1}, \ldots, \boldsymbol{v}_{i,q}$, the resulting vectors $\boldsymbol{v}'_{i,1}, \ldots, \boldsymbol{v}'_{i,q}$ satisfy $\|\boldsymbol{v}_{i,a} - \boldsymbol{v}'_{i,a}\| \leqslant O(\varepsilon \cdot q)$. For every variable $i \in V$, we compute a rescaling factor α_i such that $\boldsymbol{v}_{i,0} := \sum_{a \in [q]} \alpha_i \boldsymbol{v}'_{i,a}$ is a unit vector. Note that $\alpha_i = 1 \pm \varepsilon \cdot \mathrm{poly}(q)$. Furthermore, $\langle \boldsymbol{v}_{i,0}, \boldsymbol{v}_0 \rangle \geqslant 1 - \varepsilon \cdot \mathrm{poly}(q)$. Therefore, the angle $\angle(\boldsymbol{v}_{i,0}, \boldsymbol{v}_0) = \sqrt{\varepsilon} \cdot \mathrm{poly}(q)$. For every variable $i \in V$, we define a rotation $\boldsymbol{U}_i$ which maps the vector $\boldsymbol{v}_{i,0}$ to $\boldsymbol{v}_i$ and acts as the identity on the space orthogonal to the plane $\mathrm{span}\{\boldsymbol{v}_{i,0}, \boldsymbol{v}_i\}$. We claim that the vector $\boldsymbol{u}_{i,a} := \alpha_i \boldsymbol{U}_i \boldsymbol{v}'_{i,a}$ satisfy the conditions of the lemma. By construction, the vectors satisfy the constraints (6) and (7). Since $\boldsymbol{U}_i$ is a rotation by an angle of at most $\sqrt{\varepsilon} \cdot \mathrm{poly}(q)$, we have $\|\boldsymbol{U}_i - \boldsymbol{I}\| \leqslant \sqrt{\varepsilon} \cdot \mathrm{poly}(q)$ and therefore $\|\boldsymbol{u}_{i,a} - \alpha_i \boldsymbol{v}'_{i,a}\| \leqslant \sqrt{\varepsilon} \cdot \mathrm{poly}(q)$. Previous observations imply that $\|\alpha_i \boldsymbol{v}'_{i,a} - \boldsymbol{v}_{i,a}\| \leqslant \varepsilon \cdot \mathrm{poly}(q)$. Thus, the vectors $\{\boldsymbol{u}_{i,a}\}$ satisfy also the third condition (8). ■

The existence of a local distribution μ_P imposes constraints on the vectors corresponding in $V(P)$. Specifically, the inner products of vectors corresponding to $V(P)$ must lie in a certain polytope Q_P of constant dimension, to ensure the existence of a matching local distribution μ_P. The SDP solution $\{\boldsymbol{u}_{i,a}\}$ has local distributions that match up to an error of η. In other words, for every payoff P, the vectors corresponding to $V(P)$ are within η distance from the corresponding polytope Q_P.

The idea of smoothing is to take a convex combination of the SDP solution $\{\boldsymbol{u}_{i,a}\}$, with the SDP solution corresponding to uniform distribution over all assignments. By a suitable basis change, the local polytopes Q_P can be made full-dimensional, in that they are defined by a set of inequalities (no equations involved). The SDP solution corresponding to uniform distribution over all assignments, lies at the center of each of these local polytopes Q_P. As $\{\boldsymbol{u}_{i,a}\}$ is only η away from each of these polytopes, it moves in to the polytope on taking convex combination with the center. The above intuition is formalized in the following lemma.

Lemma 5.2 (Smoothing): The local distributions $\{\mu_P\}$ can be transformed to distributions $\{\mu'_P\}$ such that for all $P \in \mathrm{supp}(\mathcal{P})$, $i \neq j \in V(P)$, and $a, b \in [q]$,

$$\Pr_{x \sim \mu'_P} \left\{ x_i = a,\ x_j = b \right\} = (1 - \delta) \langle \boldsymbol{u}_{i,a}, \boldsymbol{u}_{j,b} \rangle + \delta \cdot \tfrac{1}{q^2}\,, \tag{9}$$

where $\delta = q^4 k^2 \eta$. Furthermore, for every $P \in \mathrm{supp}(\mathcal{P})$,

$$\|\mu_P - \mu'_P\|_1 \leqslant 3\delta$$

Proof: Let us fix a payoff function $P \in \mathrm{supp}(\mathcal{P})$. Let $S = V(P)$. We may assume $S = \{1, \ldots, k\}$. We can think of μ_P as a function $f \colon [q]^k \to \mathbb{R}$ such that $f(x)$ is the probability of the assignment x under the distribution μ_P. For the case $q = 2$, the constraint (9) translates to a condition on the degree-2 Fourier coefficients of f. For larger q, we introduce the following generalization of Fourier bases.

Let $\chi_1, \ldots, \chi_q$ be an orthonormal basis of the vector space $\{f \colon [q] \to \mathbb{R}\}$ such that $\chi_1 \equiv 1$. (Here, orthonormal means $\mathrm{E}_{a \in [q]} \chi_i(a) \chi_j(a) = \delta_{ij}$ for all $i, j \in [q]$). By tensoring this basis, we obtain the orthonormal basis $\{\chi_\sigma \mid \sigma \in [q]^k\}$ of the vector space $\{f \colon [q]^k \to \mathbb{R}\}$. For $\sigma \in [q]^k$, we have $\chi_\sigma(x) = \chi_{\sigma_1}(x_1) \cdots \chi_{\sigma_k}(x_k)$. For a function $f \colon [q]^k \to \mathbb{R}$, we denote by $\hat{f}(\sigma)$ the χ_σ-coefficient of f, i.e., $\hat{f}(\sigma) := \sum_{x \in [q]^k} f(x) \chi_\sigma(x)$. Note that $f = \mathrm{E}_{\sigma \in [q]^k} \hat{f}(\sigma) \chi_\sigma$. Therefore, if we let f again be the function corresponding to μ_P, then for all $i \neq j \in S$ and $a, b \in [q]$

$$\Pr_{x \sim \mu_P} \left\{ x_i = a,\ x_j = b \right\} = \sum_{\substack{x \in [q]^k \\ x_i = a, x_j = b}} \mathop{\mathrm{E}}_{\sigma \in [q]^k} \hat{f}(\sigma) \chi_\sigma(x) \tag{10}$$

$$= \mathop{\mathrm{E}}_{\sigma \in [q]^2} \hat{f}_{ij}(\sigma) \chi_\sigma(a, b). \tag{11}$$

Here, $\hat{f}_{ij}(s, t)$ is defined as the coefficient $\hat{f}(\sigma)$ where $\sigma_i = s$, $\sigma_j = t$ and $\sigma_r = 1$ for all $r \in [q] \setminus \{i, j\}$. In the second equality we used that for every σ with $\sigma_r \neq 1$ for some $r \in [q] \setminus \{i, j\}$, the sum over the values of χ_σ in (10) vanishes.

For every variable pair $i \neq j \in S$, let $g_{ij} \colon [q]^2 \to \mathbb{R}$ be the function $g_{ij}(a, b) = \langle \boldsymbol{u}_{i,a}, \boldsymbol{u}_{j,b} \rangle$. Similarly, we let $g_i \colon [q] \to \mathbb{R}$ be the function $g_i(a) = \langle \boldsymbol{u}_{i,a}, \boldsymbol{u}_{i,a} \rangle = \langle \boldsymbol{u}_{i,a}, \boldsymbol{v}_0 \rangle$. We define a function $f' \colon [q]^k \to \mathbb{R}$ as follows

$$\hat{f}'(\sigma) = \begin{cases} \hat{g}_i(s) & \text{if } \sigma_i = s \text{ and } \sigma_r = 1 \text{ for all } r \in [q] \setminus \{i\}, \\ \hat{g}_{ij}(s, t) & \text{if } \sigma_i = s, \sigma_j = t \text{ and } \sigma_r = 1 \text{ for } r \in [q] \setminus \{i, j\}, \\ \hat{f}(\sigma) & \text{otherwise.} \end{cases}$$

The conditions (6) and (7) imply that $\hat{g}_i(s) = \hat{g}_{ij}(s, 1)$ for all $i \neq j \in S$. We also have $\hat{f}(\mathbb{1}) = \hat{g}_i(\mathbb{1}) = \hat{g}_{ij}(\mathbb{1}) = 1$. Therefore, the identity in (10)–(11) applied to f' shows that for all $i, j \in S$ and $a, b \in [q]$,

$$\langle \boldsymbol{u}_{i,a}, \boldsymbol{u}_{j,b} \rangle = \sum_{\substack{x \in [q]^k \\ x_i = a, x_j = b}} \mathop{\mathrm{E}}_{\sigma \in [q]^k} \hat{f}'(\sigma) \chi_\sigma(x) = \sum_{\substack{x \in [q]^k \\ x_i = a, x_j = b}} f'(x)\,. \tag{12}$$

We could finish the proof at this point if the function f' would correspond to a distribution μ'_P over assignments $[q]^k$. The function f' satisfies $\sum_{x\in[q]^k} f'(x) = \hat{f}(\mathbb{1}) = 1$ However, in general, the function f' might take negative values. We will show that these values cannot be too negative and that the function can be made to a proper distribution by smoothing.

Let K be an upper bound on the values of the functions $\chi_1, \ldots, \chi_q$. From the orthonormality of the functions, it follows that $K \leqslant \sqrt{q}$. Let $f_{ij}(a,b) = \Pr_{x\sim\mu_P}\left\{x_i = a,\ x_j = b\right\}$. Recall that we computed in (11) the coefficients of f_{ij} in the basis $\{\chi_{s,t} \mid s,t \in [q]\}$. Since the SDP solution $\{\boldsymbol{u}_{i,a}\}$, $\{\mu_P\}$ is η-infeasible, we have

$$\begin{aligned}\hat{g}_{ij}(s,t) &= \sum_{a,b\in[q]} g_{ij}(a,b)\chi_{st}(a,b)\\ &= \sum_{a,b\in[q]} f_{ij}(a,b)\chi_{st}(a,b) \pm K^2q^2\eta = \hat{f}_{ij}(s,t) \pm K^2q^2\eta.\end{aligned}$$

Therefore, $|\hat{f}(\sigma) - \hat{f}'(\sigma)| \leqslant K^2q^2\eta$ for all $\sigma \in [q]^k$. Thus,

$$\begin{aligned}f'(x) = \mathop{\mathrm{E}}_{\sigma\in[q]^k} \hat{f}'(\sigma)\chi_\sigma(x) &= \mathop{\mathrm{E}}_{\sigma\in[q]^k} \hat{f}(\sigma)\chi_\sigma(x) \pm \delta/q^k\\ &= f(x) \pm \delta/q^k\,, \qquad (13)\end{aligned}$$

where $\delta := K^4k^2q^2\eta$. Hence, if we let $h = (1-\delta)\cdot f' + \delta\cdot U$, where $U\colon [q]^k \to \mathbb{R}$ is the uniform distribution $U \equiv 1/q^k$, then

$$h = (1-\delta)f' + \delta/q^k \geqslant (1-\delta)f \geqslant 0.$$

It follows that h corresponds to a distribution μ'_P over assignments $[q]^k$. Furthermore, from (12) it follows that for all $i \neq j \in S$ and $a,b \in [q]$,

$$\Pr_{x\sim\mu'_P}\left\{x_i = a,\ x_j = b\right\} = (1-\delta)\langle \boldsymbol{u}_{i,a}, \boldsymbol{u}_{j,b}\rangle + \delta\cdot\tfrac{1}{q^2}\,.$$

Finally, let us estimate the statistical distance between the distributions μ_P and μ'_P,

$$\begin{aligned}\|f-h\|_1 &= \|\delta(f-U) + (1-\delta)(f-f')\|_1\\ &\leqslant 2\delta + \|f-f'\|_1 \quad \text{(using triangle inequality)}\\ &\leqslant 3\delta \quad \text{(using (13))}\,.\end{aligned}$$

In this way, we can construct a suitable distribution μ'_P for every $P \in \mathrm{supp}(\mathcal{P})$, which proves the lemma. ■

5.1. *Proof of Theorem 4.6 (Robustness of* SDP$_{\text{gen}}$*)*

Let us consider an ε-infeasible SDP solution $\{\boldsymbol{v}_{i,a}\}$, $\{\mu_P\}$ for a Λ-CSP instance $\mathcal{P}$. Suppose that this SDP solution has value α.

First, we construct vector $\{\boldsymbol{u}_{i,a}\}$ as in Lemma 5.1. These vectors together with the original local distributions $\{\mu_P\}$ form an η-infeasible SDP solution for $\mathcal{P}$, where $\eta = \sqrt{\varepsilon}\cdot\mathrm{poly}(q)$.

Next, we construct local distributions $\{\mu'_P\}$ as in Lemma 5.2. Define new vectors

$$\boldsymbol{w}_{i,a} \stackrel{\text{def}}{=} \sqrt{1-\delta}\cdot\boldsymbol{u}_{i,a} \oplus \sqrt{\delta}\cdot\boldsymbol{u}'_{i,a}\,,$$

where $\oplus$ denotes the direct sum of vectors and $\{\boldsymbol{u}'_{i,a}\}$ are vectors corresponding to the uniform average over all feasible SDP solutions (which satisfy $\langle \boldsymbol{u}'_{i,a}, \boldsymbol{u}'_{jb}\rangle = {}^1\!/_{q^2}$ for all $i \neq j \in V$ and all $a,b \in [q]$). From Lemma 5.1 and Lemma 5.2 it follows that $\{\boldsymbol{w}_{i,a}\}$, $\{\mu'_P\}$ is a feasible SDP solution for $\mathcal{P}$.

It remains to estimate the value of this feasible SDP solution:

$$\begin{aligned}\mathop{\mathrm{E}}_{P\sim\mathcal{P}}\mathop{\mathrm{E}}_{x\sim\mu'_P} P(x) &= \alpha - \mathop{\mathrm{E}}_{P\sim\mathcal{P}} \sum_{x\in[q]^{V(P)}} P(x)\left(\mu(x) - \mu'(x)\right)\\ &\geqslant \alpha - \mathop{\mathrm{E}}_{P\sim\mathcal{P}} \|\mu - \mu'\|_1\\ &\geqslant \alpha - \eta\cdot\mathrm{poly}(kq)\,.\end{aligned}$$

For the first inequality, we used that $|P(x)| \leqslant 1$. The second inequality follows from Lemma 5.2. (In the last calculation, we just verified that the value of SDP solutions is Lipschitz in the statistical distance of the local distributions.) ■

References

[1] N. Alon and A. Naor, "Approximating the cut-norm via Grothendieck's inequality," *SIAM J. Comput.*, vol. 35, no. 4, pp. 787–803, 2006. 1

[2] S. Arora, E. Chlamtac, and M. Charikar, "New approximation guarantee for chromatic number," in *STOC*, 2006, pp. 215–224. 1

[3] S. Arora, S. Rao, and U. V. Vazirani, "Expander flows, geometric embeddings and graph partitioning," in *STOC*, 2004, pp. 222–231. 1, 4

[4] P. Austrin, "Towards sharp inapproximability for any 2-CSP," in *FOCS*, 2007, pp. 307–317. 2

[5] M. Charikar, K. Makarychev, and Y. Makarychev, "Near-optimal algorithms for unique games," in *STOC*, 2006, pp. 205–214. 1, 2

[6] ——, "Near-optimal algorithms for maximum constraint satisfaction problems," in *SODA*, 2007, pp. 62–68. [Online]. Available: http://doi.acm.org/10.1145/1283383.1283391 1, 2

[7] ——, "On the advantage over random for maximum acyclic subgraph," in *FOCS*, 2007, pp. 625–633. 1

[8] M. Charikar and A. Wirth, "Maximizing quadratic programs: Extending grothendieck's inequality," in *FOCS*, 2004, pp. 54–60. 1, 2

[9] E. Chlamtac, "Approximation algorithms using hierarchies of semidefinite programming relaxations," in *FOCS*, 2007, pp. 691–701. 1

[10] E. Chlamtac, K. Makarychev, and Y. Makarychev, "How to play unique games using embeddings," in *FOCS*, 2006, pp. 687–696. 1, 2

[11] E. Chlamtac and G. Singh, "Improved approximation guarantees through higher levels of SDP hierarchies," in *APPROX-RANDOM*, 2008, pp. 49–62. 1

[12] B. Chor and M. Sudan, "A geometric approach to betweenness," *SIAM J. Discrete Math.*, vol. 11, no. 4, pp. 511–523 (electronic), 1998. 1

[13] A. Frieze and M. Jerrum, "Improved approximation algorithms for MAX k-CUT and MAX BISECTION," *Algorithmica*, vol. 18, no. 1, pp. 67–81, 1997. 1

[14] A. M. Frieze and R. Kannan, "Quick approximation to matrices and applications," *Combinatorica*, vol. 19, no. 2, pp. 175–220, 1999. 3

[15] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J. ACM*, vol. 42, no. 6, pp. 1115–1145, 1995. 1, 2

[16] E. Halperin and U. Zwick, "Approximation algorithms for MAX 4-SAT and rounding procedures for semidefinite programs," *J. Algorithms*, vol. 40, no. 2, pp. 184–211, 2001. 2

[17] G. Hast, "Beating a random assignment," in *APPROX-RANDOM*, 2005, pp. 134–145. 1, 2

[18] K. Jain, "A factor 2 approximation algorithm for the generalized steiner network problem," *Combinatorica*, vol. 21, no. 1, pp. 39–60, 2001. 1

[19] D. R. Karger, R. Motwani, and M. Sudan, "Approximate graph coloring by semidefinite programming," *J. ACM*, vol. 45, no. 2, pp. 246–265, 1998. 1

[20] H. J. Karloff and U. Zwick, "A 7/8-approximation algorithm for MAX 3SAT?" in *FOCS*, 1997, pp. 406–415. 1, 2

[21] S. Khot, "On the power of unique 2-prover 1-round games," in *STOC*, 2002, pp. 767–775. 2

[22] ——, "On the power of unique 2-prover 1-round games," in *STOC*, 2002, pp. 767–775. 2

[23] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell, "Optimal inapproximability results for MAX-CUT and other 2-variable CSPs?" *SIAM J. Comput.*, vol. 37, no. 1, pp. 319–357, 2007. 2

[24] S. Khot and A. Naor, "Linear equations modulo 2 and the L_1 diameter of convex bodies," *SIAM J. Comput.*, vol. 38, no. 4, pp. 1448–1463, 2008. 1

[25] S. Khot and R. O'Donnell, "SDP gaps and UGC-hardness for MAXCUTGAIN," in *FOCS*, 2006, pp. 217–226. 2

[26] S. Khot and N. K. Vishnoi, "The unique games conjecture, integrality gap for cut problems and embeddability of negative type metrics into ℓ_1," in *FOCS*, 2005, pp. 53–62. 2

[27] M. Lewin, D. Livnat, and U. Zwick, "Improved rounding techniques for the MAX 2-SAT and MAX DI-CUT problems," in *IPCO*, 2002, pp. 67–82. 1, 2

[28] S. Matuura and T. Matsui, "0.863-approximation algorithm for MAX DICUT," in *RANDOM-APPROX*, 2001, pp. 138–146. 2

[29] Y. Nesterov, "Semidefinite relaxation and nonconvex quadratic optimization," Universit catholique de Louvain, Center for Operations Research and Econometrics (CORE), CORE Discussion Papers 1997044, Jun. 1997. [Online]. Available: http://ideas.repec.org/p/cor/louvco/1997044.html 1

[30] J. Nocedal and S. J. Wright, *Numerical optimization*. New York: Springer-Verlag, 1999. 1

[31] R. O'Donnell and Y. Wu, "An optimal SDP algorithm for max-cut, and equally optimal long code tests," in *STOC*, 2008, pp. 335–344. 2

[32] P. Raghavendra, "Optimal algorithms and inapproximability results for every CSP?" in *STOC*, 2008, pp. 245–254. 1, 2, 4

[33] P. Raghavendra and D. Steurer, "Integrality gaps for strong SDP relaxations of unique games," in *FOCS*, 2009, to appear. 1, 2

[34] M. Singh and L. C. Lau, "Approximating minimum bounded degree spanning trees to within one of optimal," in *STOC*, 2007, pp. 661–670. 1

[35] V. V. Vazirani, *Approximation algorithms*. Berlin: Springer-Verlag, 2001. 1

[36] S. J. Wright, *Primal-dual interior-point methods*. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM), 1997. 1

[37] U. Zwick, "Finding almost-satisfying assignments," in *STOC*, 1998, pp. 551–560. 2

6. Appendix

6.1. Property of Dimension Reduction

Lemma 4.3 (Inner products are preserved approximately, restated): For any two vectors $\boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{R}^D$ in the unit ball,

$$\Pr_{\boldsymbol{\Phi}}\left\{\left|\langle \boldsymbol{\Phi}\, \boldsymbol{v}_1, \boldsymbol{\Phi}\, \boldsymbol{v}_2\rangle - \langle \boldsymbol{v}_1, \boldsymbol{v}_2\rangle\right| \geqslant \tfrac{t}{\sqrt{d}}\right\} \leqslant O\left(1/t^2\right) .$$

Proof: Note that we may assume that both vectors are unit vectors (otherwise, we can normalize them). Suppose $\langle \boldsymbol{v}_1, \boldsymbol{v}_2\rangle = \alpha$. By rotational invariance, we can assume that $\boldsymbol{v}_1 = (1, 0)$ and $\boldsymbol{v}_2 = (\alpha, \beta)$, where $\beta = \sqrt{1-\alpha^2}$. Hence, $\langle \boldsymbol{\Phi}\, \boldsymbol{v}_1, \boldsymbol{\Phi}\, \boldsymbol{v}_2\rangle$ has the same distribution as

$$\tfrac{1}{d}\left(\sum_{i=1}^{d} \alpha\xi_i^2 + \beta\xi_i\xi_i'\right),$$

where $\xi_1, \xi_1', \ldots, \xi_d, \xi_d'$ are independent standard Gaussian variables (mean 0 and standard deviation 1).

For each i, the expectation of $\alpha\xi_i^2 + \beta\xi_i\xi_i'$ is equal to α and the variance is bounded (at most 2). Hence, the expectation of $\langle \boldsymbol{\Phi}\, \boldsymbol{v}_1, \boldsymbol{\Phi}\, \boldsymbol{v}_2\rangle$ is equal to α and the standard deviation is $O(1/\sqrt{d})$. The lemma follows from Chebychev's inequality. ■