Electronic Colloquium on Computational Complexity, Report No. 66 (2014)

Robust Approximation of Temporal CSP

Suguru Tamaki
Kyoto University
tamak@kuis.kyoto-u.ac.jp

Yuichi Yoshida
National Institute of Informatics, and
Preferred Infrastructure, Inc.
yyoshida@nii.ac.jp

Abstract

A temporal constraint language Γ is a set of relations with first-order definitions in $(\mathbb{Q}; <)$. Let $\mathsf{CSP}(\Gamma)$ denote the set of constraint satisfaction problem instances with relations from Γ. $\mathsf{CSP}(\Gamma)$ admits robust approximation if, for any $\varepsilon \geq 0$, given a $(1-\varepsilon)$-satisfiable instance of $\mathsf{CSP}(\Gamma)$, we can compute an assignment that satisfies at least a $(1-f(\varepsilon))$-fraction of constraints in polynomial time. Here, $f(\varepsilon)$ is some function satisfying $f(0) = 0$ and $\lim_{\varepsilon \to 0} f(\varepsilon) = 0$.

Firstly, we give a qualitative characterization of robust approximability: Assuming the Unique Games Conjecture, we give a necessary and sufficient condition on Γ under which $\mathsf{CSP}(\Gamma)$ admits robust approximation. Secondly, we give a quantitative characterization of robust approximability: Assuming the Unique Games Conjecture, we precisely characterize how $f(\varepsilon)$ depends on ε for each Γ. We show that our robust approximation algorithms can be run in almost linear time.

1 Introduction

In the Constraint Satisfaction Problem (CSP), we are given a set of constraints over a set of variables, and the task is to decide whether there exists an assignment of values to the variables that satisfies all the constraints. CSP can express general combinatorial and temporal problems in artificial intelligence, computer science, discrete mathematics, operations research, and elsewhere [11, 23].

In this paper, we consider the Temporal CSP (TCSP), a particular class of CSP where variables represent times and constraints represent sets of allowed temporal relations among them. Formally, a *temporal relation* is a relation with a first-order definition in $(\mathbb{Q}; <)$. TCSP forms a fundamental and important class of CSP over infinite domains [4]. Since TCSP is NP-hard in general, one of the major line of research is to identify tractable subclasses and develop efficient algorithms for them. One of the standard way to define subclasses of TCSP is restricting constraint languages.

A *temporal constraint language*, denoted by Γ, is a set of temporal relations. $\mathsf{CSP}(\Gamma)$ denotes the set of TCSP instances where each instance consists of constraints from Γ. Polynomial-time algorithms have been developed for larger and larger classes of constraint languages, see, e.g., [27, 26, 21], whereas TCSP for several specific constraint languages are known to be NP-complete [13]. Building on previous works, Bodirsky and Kára [7] finally showed the complete complexity classification of TCSP. Namely, they obtain a necessary and sufficient condition on Γ under which $\mathsf{CSP}(\Gamma)$ is tractable. The proof technique relies on a machinery from universal algebra, which plays an important role when we investigate the computational complexity of CSP in various settings.

In this paper, we study the complexity of Max-TCSP, instead of satisfiability of TCSP. In particular, we are interested in *robust approximability* of TCSP. An algorithm is called a (c, s)-*approximation algorithm* for $\mathsf{CSP}(\Gamma)$ if, given any c-satisfiable instance (some assignment satisfies at least a c-fraction of constraints) of $\mathsf{CSP}(\Gamma)$, it outputs an assignment that satisfies at least an s-fraction of constraints. An algorithm is called a *robust approximation algorithm* for $\mathsf{CSP}(\Gamma)$ if it is $(1-\varepsilon, 1-f(\varepsilon))$-approximation algorithm for any $\varepsilon \geq 0$, where f is some function satisfying $f(0) = 0$ and $\lim_{\varepsilon \to 0} f(\varepsilon) = 0$. When we want to specify $f(\varepsilon)$, we call it a $f(\varepsilon)$-*robust approximation algorithm*. Note that if $\mathsf{CSP}(\Gamma)$ admits polynomial-time robust approximation, then satisfiability of $\mathsf{CSP}(\Gamma)$ is solvable in polynomial-time. However, the reverse statement does not hold in general.

ISSN 1433-8092

For example, $\mathsf{CSP}(\{<\})$ (also known as the Acyclic Graph Problem) is solvable in polynomial-time, but $(1-\varepsilon, 1/2+\varepsilon)$-approximation is known to be *UG-Hard* [14], i.e., NP-Hard under Khot's Unique Games Conjecture (UGC) [17].

The notion of robust approximation is natural and useful, e.g., let us consider the Correlation Clustering Problem [1], which is equivalent to $\mathsf{CSP}(\{=,\neq\})$. Here, a variable stands for a datum and a constraint $u = v$ (resp., $u \neq v$) means u and v is similar (resp., dissimilar). The objective is to find a partition of the data into groups that agrees as much as possible with the constraints. If we are given a data set with a perfect (satisfiable) partition, then we can find it easily. However, if a small fraction of constraints are wrongly given by some reason, e.g., measurement error, then recovering the optimal partition may become much harder. Motivated by such practical applications, it is natural to ask what class of constraint languages admits robust approximation.

Our Contribution. In this paper, we give a complete complexity classification of robust approximability of TCSP:

Theorem 1. *Let Γ be a temporal constraint language. Then, $\mathsf{CSP}(\Gamma)$ admits polynomial-time robust approximation if either Γ is trivial or a Horn equality constraint language. Otherwise, it is UG-Hard to robustly approximate $\mathsf{CSP}(\Gamma)$.*

We say that a constraint language Γ is *trivial* if every instance of $\mathsf{CSP}(\Gamma)$ is satisfiable unless it contains an individual constraint that is unsatisfiable such as $x_i \neq x_i$. Informally, Γ is a *Horn equality constraint language* if each relation in Γ can be defined as a Horn formula whose atoms are of the form $x = y$. See Preliminaries for the more detailed definition.

We also show a more fine-grained classification that almost tightly (up to logarithmic factor) characterizes how $f(\varepsilon)$ depends on ε:

Theorem 2. *Let Γ be a Horn equality constraint language.*

1. *If Γ is not trivial, it is UG-Hard to compute $o(\sqrt{\varepsilon})$-robust approximation of $\mathsf{CSP}(\Gamma)$.*
2. *If Γ is negative, there is a polynomial-time $O(\sqrt{\varepsilon}\log(1/\varepsilon))$-robust approximation algorithm for $\mathsf{CSP}(\Gamma)$.*
3. *If Γ is not negative, it is UG-Hard to compute $o(1/\log(1/\varepsilon))$-robust approximation of $\mathsf{CSP}(\Gamma)$.*
4. *There is a polynomial-time $O(\log\log(1/\varepsilon)/\log(1/\varepsilon))$-robust approximation algorithm for $\mathsf{CSP}(\Gamma)$.*

Informally, Γ is a *negative equality constraint language* if each relation in Γ can be defined as a disjunction of negative literals or a single positive literal whose atoms are of the form $x = y$. See Preliminaries for the more detailed definition.

Furthermore, we give almost linear time algorithms for the above mentioned robust approximation results.

Theorem 3. *There exist algorithms that achieve the approximation guarantee mentioned in Items 2 and 4 of Theorem 2 in $O(m \cdot \operatorname{poly}\log n \cdot \exp(1/\varepsilon))$ time, where n is the number of variables and m is the number of constraints.*

Related Works. Motivated by obvious applications, CSP over *finite* domains has been a central problem in a lot of research areas. In their seminal paper [12], Feder and Vardi posed a famous dichotomy conjecture; "for any constraint language Γ over a finite domain, $\mathsf{CSP}(\Gamma)$ is either in P or NP-complete." The conjecture has been a driving force of the theoretical study of CSP and although it still remains open, we have developed deep mathematical insights on the structure of CSP, see, e.g., [9].

A systematic study of robust approximation algorithms was initiated by Zwick [28]. He gave polynomial-time robust approximation algorithms for 2SAT and Horn-SAT, which, combined with previous works [16, 24], implies a complete complexity classification of robust approximability of Boolean CSP. Later Dalmau and Krokhin [10] gave a more fine-grained classification which determines how $f(\varepsilon)$ depends on ε for each constraint language.

For CSP over general finite domains, Guruswami and Zhou conjectured that $\mathsf{CSP}(\Gamma)$ admits polynomial-time robust approximation if and only if $\mathsf{CSP}(\Gamma)$ has "bounded-width," which informally means that $\mathsf{CSP}(\Gamma)$ is solvable by a local consistency method. Dalmau and Krokhin [10], and Kun et al. [20] obtained robust approximation algorithms for the special case of width-1 and finally Barto and Kozik [2] confirmed the conjecture. Unlike Boolean CSP, a quantitative version of the classification has not been obtained so far, see [10].

As far as the authors know, there is only one paper that systematically studies the robust approximablity of CSP over infinite domains. Ordering CSP (OCSP) is TCSP with additional hard constraints that the variables need to be given different values. Guruswami et al. [14] showed that for any constraint language Γ, the best approximation algorithm for $\mathsf{CSP}(\Gamma)$ as OCSP is random assignment algorithms, assuming UGC. In particular, this implies that if Γ is nontrivial, then it is UG-hard to robustly approximate $\mathsf{CSP}(\Gamma)$ as OCSP. We notice that our results do not follow easily from [14] since the existence of hard constrains in OCSP affects the approximability of CSP.

As for specific CSP over infinite domains, we are only aware of the result for $\mathsf{CSP}(\{=, \neq\})$; Charikar et al. [8] gave a polynomial-time $O(\sqrt{\varepsilon}\log(1/\varepsilon))$-robust algorithm for it.

Our Technique. First we would like to emphasize that our contribution is the results themselves and not the techniques to prove them. Each technical proof is non-trivial but not too difficult to come up with for experts on each topic such as universal algebra, approximation algorithms based on SDP, and connection between hardness of approximation and integrality gap. We briefly describe the overall proof structure below.

To prove Theorem 1, first we must identify the borderline which separates tractable and intractable cases. By the results of Bodirsky and Kára [6, 7] and Guruswami et al. [14], we see that if $\mathsf{CSP}(\Gamma)$ admits robust approximation, then Γ must be a *Horn equality constraint language*. Then, we show that Γ is a *Horn equality constraint language* is sufficient by giving robust approximation algorithms.

To prove Theorem 2, first we show that the "easiest" non-trivial TCSP is $\mathsf{CSP}(\{=, \neq\})$. The approximation hardness of $\mathsf{CSP}(\{=, \neq\})$ follows a simple reduction from Max-CUT. Next we extend the robust approximation algorithm for $\mathsf{CSP}(\{=, \neq\})$ due to Charikar et al. [8] and obtain an algorithm with the same approximation guarantee when Γ is negative. If Γ is not negative, we can show $\mathsf{CSP}(\Gamma)$ is as hard as $\mathsf{CSP}(\{\mathrm{ODD}_3, \neq\})$. The approximation hardness of $\mathsf{CSP}(\{\mathrm{ODD}_3, \neq\})$ follows by modifying the approximation hardness of Horn SAT due to Guruswami and Zhou [15].

Our algorithms are based on semidefinite programming (SDP) relaxation. One might think Raghavendra's canonical SDP relaxation for CSP over finite domains [22] can be extended to handle TCSP. This is the case in the sense that its integrality gap turns out to match UG-Hardness [14]. However, it is hard to explicitly analyze its approximation guarantee, and existing rounding techniques introduce errors depending on the domain size, which is too huge for TCSP. Thus, we use an equivalent SDP relaxation tailored to equality constraint languages so as not to be affected by the domain size.

Our inapproximability results rely on UGC, which states that for any $\varepsilon > 0$, there exists an integer $q > 0$ such that it is NP-hard to compute $(1 - \varepsilon, \varepsilon)$-approximation of CSP where each constraint is a two-variable linear equation over $\mathbb{Z}_q$. This complexity theoretic assumption enables us to prove *optimal* inapproximability results for various optimization problems such as Max-CUT, Vertex Cover etc., though proving them under P$\neq$NP seems beyond our current proof techniques. See, e.g., [18] for discussion on UGC. To show inapproximability results in Theorem 2, we use the fact that the integrality gap matches UG-Hardness and explicitly give bad integrality gap instances.

Organization In the next section, we introduce notion and standard tools to analyze TCSP. Then, we prove Theorem 1, which is a "qualitative" characterization of robust approximability. Next, we prove Theorem 2, which is a "quantitative" characterization of robust approximability.

Finally, we prove Theorem 3, which gives almost linear time algorithms for the robust approximability results in Theorem 2.

2 Preliminaries

For an integer n, $[n]$ denotes the set $\{1, \ldots, n\}$. We often use n and m to denote the number of variables and constraints of the instance we are concerned with, respectively.

For two real vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, $\angle(\boldsymbol{x}, \boldsymbol{y})$ denotes the angle between them, i.e., $\arccos(\langle \boldsymbol{x}, \boldsymbol{y} \rangle / (\|\boldsymbol{x}\| \cdot \|\boldsymbol{y}\|))$.

Temporal Constraint Language. A *temporal constraint language* Γ is a finite relational structure $(\mathbb{Q}; R_1, R_2, \ldots)$ with a first-order definition in $(\mathbb{Q}; <)$, the rational numbers with the dense linear order. Each R_i is a *temporal relation*, i.e., $R_i \subseteq \mathbb{Q}^{k_i}$ for some finite k_i such that there is a first-order formula ϕ_i with k_i free variables that defines R_i over $(\mathbb{Q}; <)$.

An *instance* of the problem $\mathsf{CSP}(\Gamma)$ is $\mathcal{I} = (V, \mathcal{C})$, where V is a set of variables and $\mathcal{C}$ is a set of constraints. Each constraint $C \in \mathcal{C}$ is of the form $(x_1, \ldots, x_k; R)$, where $x_1, \ldots, x_k \in V$ are variables and $R \in \Gamma$ is a k-ary relation. We say that $\beta : V \to \mathbb{Q}$ *satisfies* a constraint $(x_1, \ldots, x_k; R) \in \mathcal{C}$ if the tuple $(\beta(x_1), \ldots, \beta(x_k))$ is in R. We say that β *satisfies* $\mathcal{I}$ if it satisfies all the constraints. When β satisfies a constraint C (resp., instance $\mathcal{I}$), we write $\beta \models C$ (resp., $\beta \models \mathcal{I}$). We denote by $\mathbf{opt}(\mathcal{I})$ the maximum fraction of constraints of $\mathcal{I}$ simultaneously satisfiable by some assignment.

An *equality constraint language* Γ is a temporal constraint language such that each relation can be defined with a $=$-formula, i.e., a quantifier-free first-order formula whose atoms are of the form $x = y$.

For each relation R from an equality constraint language, we can find a formula ϕ_R of the equality relation that defines R. In particular, we can assume that ϕ_R is represented in conjunctive normal form. We say that R is *Horn* if each clause in ϕ_R contains at most one positive literal. We say that R is *negative* if each clause in ϕ_R consists of a single positive literal or a disjunction of negative literals. We say that an equality constraint language Γ is *Horn* (resp., negative) if every relation in Γ is Horn (resp., negative). The problem $\mathsf{CSP}(\Gamma)$ is called Horn $=$-SAT (resp., $\mathsf{Negative}$ $=$-SAT) if Γ is a Horn (resp., negative) equality constraint language Γ.

Universal Algebra. We introduce several definitions from universal algebra, which is a standard tool to investigate computational complexity of CSP.

An l-ary operation f *preserves* (or is a *polymorphism* of) a k-ary relation R if for any tuples $(a^i_1, \ldots, a^i_k) \in R$ ($i \in [l]$), the tuple $(f(a^1_1, \ldots, a^l_1), \ldots, f(a^1_k, \ldots, a^l_k))$ belongs to R as well. We say that f *preserves* (or is a *polymorphism* of) a constraint language Γ if f preserves all relations in Γ.

Let Γ be a constraint language and R be a relation. Then, R is *pp-definable* in Γ if R can be defined as $R(x_1, \ldots, x_k) = \exists y_1, \ldots, y_l(\psi(x_1, \ldots, x_k, y_1, \ldots, y_l))$, where ψ is a conjunction of atomic formulas with relations in Γ and the equality $=$. If ψ does not contain the equality $=$ then we say that R is *pp-definable in* Γ *without equality*. It is known that the set of relations pp-definable in Γ is exactly the set of relations whose polymorphisms are the same as Γ [3].

We introduce the notation $\mathsf{CSP}(\Gamma) \leq_{\mathrm{RA}} \mathsf{CSP}(\Gamma')$ as a shorthand for the following. For any error function f with $\lim_{\varepsilon \to 0} f(\varepsilon) = 0$ and $f(0) = 0$, if some polynomial-time algorithm $f(\varepsilon)$-robustly approximates $\mathsf{CSP}(\Gamma')$, then there is a polynomial-time algorithm that $O(f(\varepsilon))$-robustly approximates $\mathsf{CSP}(\Gamma)$.

Though the following lemma is originally proved for Boolean CSP, the proof is also valid for TCSP.

Lemma 4 ([10])**.** *Let Γ be a constraint language and let R be a relation pp-definable in Γ without equality. Then $\mathsf{CSP}(\Gamma \cup \{R\}) \leq_{\mathrm{RA}} \mathsf{CSP}(\Gamma)$.*

Thus, if Γ itself contains the equality relation, robust approximability of $\mathsf{CSP}(\Gamma)$ is determined by polymorphisms. Indeed, any non-trivial equality constraint language turns out to contain the equality relation. To show this, we use the following fact.

Lemma 5 ([6])**.** *Let Γ be an equality constraint language that is not preserved by any constant operation. Then, $\neq$ is pp-definable in Γ.*

Lemma 6. *Let Γ be a non-trivial equality constraint language. Then, Γ pp-defines $=$ and $\neq$.*

Proof. Since Γ is non-trivial, in particular, Γ is not preserved by any constant operation. Thus, $\neq$ is pp-definable in Γ from Lemma 5.

Since Γ is non-trivial, there exists a satisfiable relation $R(x_1, \ldots, x_k)$ pp-definable in Γ such that it is not satisfied by any *all-different* assignment β, where $\beta(x_i) \neq \beta(x_j)$ holds for every $i \neq j$. This means that for any satisfying assignment β for $R(x_1, \ldots, x_k)$, $\beta(x_i) = \beta(x_j)$ holds for some $i \neq j$. As long as there is a pair of arguments (x_i, x_j) such that there is a satisfying assignment β with $\beta(x_i) \neq \beta(x_j)$, we add a constraint $(x_i \neq x_j)$ to R. Let R' be the resulting constraint. Note that R' is pp-definable in Γ as $\neq$ is pp-definable in Γ. Since R is not satisfied by the all-different assignment, R' must have some pair (x_i, x_j) such that we have not added the constraint $(x_i \neq x_j)$. Since R' becomes unsatisfiable if we add a constraint $(x_i \neq x_j)$ to R', x_i must be equal to x_j in any satisfying assignment of R'. Thus, the projection of R' to $\{x_i, x_j\}$ is the equality constraint. $\square$

Combining Lemmas 4 and 6, the following holds.

Corollary 7. *Let Γ be a non-trivial equality constraint language. Let R be a relation pp-definable in Γ. Then, $\mathsf{CSP}(\Gamma \cup \{R\}) \leq_{\mathrm{RA}} \mathsf{CSP}(\Gamma)$.*

Semidefinite Programming. We introduce an SDP relaxation $\mathsf{BasicSDP}$. For an instance $\mathcal{I} = (V, \mathcal{C})$ of a standard CSP over the domain $[q]$, we want to find a collection of vectors $\{\boldsymbol{x}_{u,a}\}_{u \in V, a \in [q]}$ and a collection of probability distributions $\{\boldsymbol{\mu}_C\}_{C \in \mathcal{C}}$:

$$\begin{aligned}
&\max \mathop{\mathbf{E}}_{C \in \mathcal{C}} \Pr_{\beta \sim \boldsymbol{\mu}_C} [\beta \models C] \\
&\text{s.t.} \Pr_{\beta \sim \boldsymbol{\mu}_C} [\beta(u) = a, \beta(v) = b] = \langle \boldsymbol{x}_{u,a}, \boldsymbol{x}_{v,b} \rangle \qquad \forall C \in \mathcal{C}, u, v \in V, a, b \in [q], \\
&\qquad \Pr_{\beta \sim \boldsymbol{\mu}_C} [\beta(u) = a] = \langle \boldsymbol{x}_{u,a}, \boldsymbol{I} \rangle \qquad \forall C \in \mathcal{C}, u \in V, a \in [q].
\end{aligned}$$

Here, $\boldsymbol{I}$ is any unit vector. Since $\boldsymbol{\mu}_C$ is a probability distribution, we implicitly impose $\langle \boldsymbol{x}_{u,a}, \boldsymbol{x}_{v,b} \rangle \geq 0$ and $\sum_a \boldsymbol{x}_{u,a} = \boldsymbol{I}$. See [22] for detailed explanation of $\mathsf{BasicSDP}$. We define $\mathbf{sdp}(\mathcal{I})$ as the optimal SDP value of $\mathsf{BasicSDP}$ for $\mathcal{I}$. For TCSP, since we only need n values though the domain is $\mathbb{Q}$, we can write down $\mathsf{BasicSDP}$ as well. Guruswami et al. showed that, assuming UGC, $\mathsf{BasicSDP}$ gives a tight approximation ratio to $\mathsf{Ordering\ CSP}$, which is a large subset of TCSP. The difference is that, in $\mathsf{Ordering\ CSP}$, we only consider constraints that can be satisfied only when all variables have different values. However, it is almost direct to modify the argument to cover the whole TCSP:

Lemma 8 ([14])**.** *Let Γ be a temporal constraint language. Suppose that there is an instance $\mathcal{I}$ of $\mathsf{CSP}(\Gamma)$ with $\mathbf{sdp}(\mathcal{I}) = c$ and $\mathbf{opt}(\mathcal{I}) = s$. Then, it is UG-Hard to compute $(c - \varepsilon, s + \varepsilon)$-approximation for $\mathsf{CSP}(\Gamma)$ for any $\varepsilon > 0$.*

Let Γ be an equality constraint language and $\mathcal{I}$ be an instance of $\mathsf{CSP}(\Gamma)$. Then, $\mathbf{sdp}(\mathcal{I})$ is determined by $\sum_{a \in [q]} \langle \boldsymbol{x}_{u,a}, \boldsymbol{x}_{v,a} \rangle$ for $u, v \in V$. Thus, by letting $\boldsymbol{x}_u = \oplus_{a=1}^{q} \boldsymbol{x}_{u,a} := (\boldsymbol{x}_{u,1}, \ldots, \boldsymbol{x}_{u,q})$, we can transform $\mathsf{BasicSDP}$ to the following SDP relaxation.

$$\begin{aligned}
&\max \mathop{\mathbf{E}}_{C \in \mathcal{C}} \Pr_{\beta \sim \boldsymbol{\mu}_C} [\beta \models C] \\
&\text{s.t.} \Pr_{\beta \sim \boldsymbol{\mu}_C} [\beta(u) = \beta(v)] = \langle \boldsymbol{x}_u, \boldsymbol{x}_v \rangle \quad \forall C \in \mathcal{C}, u, v \in V.
\end{aligned}$$

Again, we implicitly impose $\langle \boldsymbol{x}_u, \boldsymbol{x}_v \rangle \geq 0$ and $\|\boldsymbol{x}_u\|^2 = 1$. (Strictly speaking, the above formulation might be weaker than the original $\mathsf{BasicSDP}$ but suffices for our purpose.) Note that semidefinite programs can be solved within an additive error δ for any $\delta > 0$ in time polynomial in the size of an instance and $\log(1/\delta)$.

3 Qualitative Characterization

In this section, we prove Theorem 1, which is a "qualitative" characterization of robust approximability.

First we introduce well-known relations (See [7]).

- Betw is the ternary relation $\{(x, y, z) \in \mathbb{Q}^3 \mid (x < y < z) \vee (z < y < x)\}$.
- Cycl is the ternary relation $\{(x, y, z) \in \mathbb{Q}^3 \mid (x < y < z) \vee (y < z < x) \vee (z < x < y)\}$.
- Sep is the 4-ary relation $\{(x_1, y_1, x_2, y_2) \in \mathbb{Q}^4 \mid$ all distinct and the interval $[\min\{x_1, y_1\}, \max\{x_1, y_1\}]$ and the interval $[\min\{x_2, y_2\}, \max\{x_2, y_2\}]$ overlap$\}$.

Then, we use the following classification result.

Lemma 9 (Theorem 20 (and proof of Theorem 50) in [7])**.** *A temporal constraint language* Γ *satisfies at least one of the following:*

1. Γ *is trivial,*
2. *There is a pp-definition of* $<$, Cycl, Betw, *or* Sep *in* Γ, *or*
3. Γ *is an equality constraint language.*

For the first case, robust approximation is meaningless since every instance is satisfiable. As for the second case, robust approximation is hard from the following and Corollary 7.

Lemma 10 ([13, 7, 14])**.** *It is NP-Complete to solve* $\mathsf{CSP}(\{\mathsf{Betw}\})$, $\mathsf{CSP}(\{\mathsf{Cycl}\})$, *and* $\mathsf{CSP}(\{\mathsf{Sep}\})$, *and it is UG-Hard to compute* $(1 - \varepsilon, 1/2 + \varepsilon)$*-approximation of* $\mathsf{CSP}(\{<\})$ *for any* $\varepsilon > 0$.

Now we focus on the third case, i.e., Γ is an equality constraint language. The following lemma gives the condition under which $\mathsf{CSP}(\Gamma)$ is solvable.

Lemma 11 (Theorem 1 and Lemma 8 in [6])**.** *Let* Γ *be a non-trivial trivial equality constraint language. The problem* $\mathsf{CSP}(\Gamma)$ *is polynomial-time solvable if* Γ *is Horn and NP-complete otherwise.*

We show the following robust approximation algorithm for Horn $=$-SAT in the next section.

Lemma 12. *There is an* $O(\frac{\log\log 1/\varepsilon}{\log 1/\varepsilon})$*-robust approximation algorithm for* Horn $=$-SAT.

We finish the proof of Theorem 1 by combining Lemmas 9, 10, 11 and 12.

3.1 Approximability of Horn =-SAT

Now we prove Lemma 12. Let Γ_k be the equality constraint language that consists of Horn clauses of at most k literals. Note that every Horn formula is pp-definable in Γ_3 and Γ_3 contains the equality relation. Thus from Lemma 4, it suffices to consider $\mathsf{CSP}(\Gamma_3)$ to prove Lemma 12. In this section, however, we give an $O(\frac{\log(k \log 1/\varepsilon)}{\log 1/\varepsilon})$-approximation algorithm for $\mathsf{CSP}(\Gamma_k)$ to see the dependency on k.

Let $\mathcal{I} = (V, \mathcal{C})$ be an instance of $\mathsf{CSP}(\Gamma_k)$. We let $\boldsymbol{y}_C = \Pr_{\beta \sim \boldsymbol{\mu}_C}[\beta \models C]$ in the $\mathsf{BasicSDP}$. Then for each constraint $C \in \mathcal{C}$, we have a constraint of the form:

$$\boldsymbol{y}_C \leq \sum_{(u \neq v) \in C} (1 - \langle \boldsymbol{x}_u, \boldsymbol{x}_v \rangle) + \sum_{(u = v) \in C} \langle \boldsymbol{x}_u, \boldsymbol{x}_v \rangle.$$

Note that the latter sum contains at most one summand.

Let $\mathcal{I}$ be an instance with $\mathbf{opt}(\mathcal{I}) \geq 1 - \varepsilon$. Clearly $\mathbf{sdp}(\mathcal{I}) \geq 1 - \varepsilon$ holds, and it follows that $\boldsymbol{y}_C \geq 1 - \sqrt{\varepsilon}$ for at least a $(1 - \sqrt{\varepsilon})$-fraction of constraints. Then, we discard constraints C with $\boldsymbol{y}_C < 1 - \sqrt{\varepsilon}$. For simplicity of exposition, we assume that every constraint C satisfies $\boldsymbol{y}_C \geq 1 - \varepsilon$. This does not affect the final result since $O(\frac{\log(k \log 1/\varepsilon)}{\log 1/\varepsilon})$ remains the same by replacing ε with $\sqrt{\varepsilon}$. We also assume that $\varepsilon < 1/2$.

Our rounding scheme is as follows. Let $s \geq 1$ and $\delta = \delta(k, \varepsilon) \ll \varepsilon$ be parameters determined later. Let $h = \frac{2\sqrt{k}}{\delta}\log\frac{1}{\delta}$. We pick t from $\{h^0, h^1, h^2, \ldots, h^s\}$ uniformly at random. Then, we choose t random hyperplanes, which divides the entire space into 2^t cells. For each cell, we introduce a new value and assign the value to all variables in the cell. Note that the resulting assignment β only uses at most 2^t different values.

The following lemma is useful to analyze the performance of our algorithm.

Lemma 13. *Let $\boldsymbol{x}, \boldsymbol{y}$ be unit vectors. The probability that two unit vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are in the same cell given by t random hyperplanes is $(1 - \frac{\angle(\boldsymbol{x},\boldsymbol{y})}{\pi})^t$. In particular, the following hold.*

- *If $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \geq 1 - \varepsilon$, then the probability that $\boldsymbol{x}$ and $\boldsymbol{y}$ are in the same cell is $1 - O(t\sqrt{\varepsilon})$.*
- *If $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \leq 1 - \varepsilon$, then the probability that $\boldsymbol{x}$ and $\boldsymbol{y}$ are in the same cell is $\exp(-\Omega(t\sqrt{\varepsilon}))$.*

Proof. The first claim is obvious. If $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \geq 1 - \varepsilon$, then $\angle(\boldsymbol{x}, \boldsymbol{y}) \leq 2\sqrt{\varepsilon}$ holds, and it follows that $(1 - \frac{\angle(\boldsymbol{x},\boldsymbol{y})}{\pi})^t \geq (1 - \frac{2\sqrt{\varepsilon}}{\pi})^t \geq 1 - \frac{2t\sqrt{\varepsilon}}{\pi}$. If $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \leq 1 - \varepsilon$, then $\angle(\boldsymbol{x}, \boldsymbol{y}) \geq \sqrt{2\varepsilon}$ holds, and it follows that $(1 - \frac{\angle(\boldsymbol{x},\boldsymbol{y})}{\pi})^t \leq (1 - \frac{\sqrt{2\varepsilon}}{\pi})^t \leq \exp(-\frac{t\sqrt{2\varepsilon}}{\pi})$. $\square$

The following three lemmas show that each kind of constraints is satisfied with high probability.

Lemma 14. *Let C be a constraint of the form $(u = v)$. If $\boldsymbol{y}_C \geq 1-\varepsilon$, then $\Pr[\beta \models C] = 1-O(h^s\sqrt{\varepsilon})$.*

Proof. Since $\langle \boldsymbol{x}_u, \boldsymbol{x}_v \rangle \geq 1 - \varepsilon$, from Lemma 13, we have $\Pr[\beta \models C] = \mathbf{E}_t[1 - O(t\sqrt{\varepsilon})] = 1 - O(h^s\sqrt{\varepsilon})$. $\square$

Lemma 15. *Let C be a constraint of the form $(u_1 \neq v_1) \vee \cdots \vee (u_l \neq v_l)$. If $\boldsymbol{y}_C \geq 1 - \varepsilon$, then $\Pr[\beta \not\models C] = 1/s + \exp(-\Omega(h/\sqrt{2l}))$.*

Proof. We have $\sum_{i=1}^{l} \langle \boldsymbol{x}_{u_i}, \boldsymbol{x}_{v_i} \rangle \leq l - 1 + \varepsilon$. Thus, there exists some $i \in [l]$ with $\langle \boldsymbol{x}_{u_i}, \boldsymbol{x}_{v_i} \rangle \leq 1 - \frac{1-\varepsilon}{l}$. From Lemma 13, we have $\Pr[\beta \not\models C] = \mathbf{E}_t[\exp(-\Omega(t\sqrt{(1-\varepsilon)/l}))] = 1/s + \exp(-\Omega(h/\sqrt{2l}))$ (We used $\varepsilon < 1/2$). $\square$

Lemma 16. *Let C be a constraint of the form $(u_1 \neq u_1) \vee \cdots \vee (u_{l-1} \neq v_{l-1}) \vee (u_l = v_l)$. If $\boldsymbol{y}_C \geq 1 - \varepsilon$, then $\Pr[\beta \models C] = 1 - O(h^s\sqrt{\varepsilon}) - \delta - 1/s$.*

Proof. Let $\eta = 1 - \langle \boldsymbol{x}_{u_l}, \boldsymbol{x}_{v_l} \rangle$. Suppose that $\eta < 2\varepsilon$. Then, from Lemma 13, $\Pr[\beta \models C] \geq \Pr[\beta_{u_l} = \beta_{v_l}] = \mathbf{E}_t[1 - O(t\sqrt{\varepsilon})] = 1 - O(h^s\sqrt{\varepsilon})$

Suppose that $\eta \geq 2\varepsilon$. Then, there exists some $i \in [l-1]$ such that $\langle \boldsymbol{x}_{u_i}, \boldsymbol{x}_{v_i} \rangle \leq 1 - \frac{\eta-\varepsilon}{l-1} \leq 1 - \frac{\eta}{2l}$. Let $p_t^+ = \Pr[\beta \models (u_l = v_l) \mid t]$ and $p_t^- = \Pr[\beta \not\models (u_i \neq v_i) \mid t]$. We want to bound from above the number of t such that neither $p_t^+ \geq 1 - \delta$ nor $p_t^- \leq \delta$. We will choose δ so that $p_1^+ \geq 1 - \delta$. Let $t^* \in \{h^i\}_{i=0}^s$ be the smallest value such that $p_{t^*}^+ < 1 - \delta$. If $t^* \geq h^s$, then we always have $p_t^+ \geq 1 - \delta$ and we are done. Suppose $t^* < h^s$. Then, by choosing $s = \frac{1}{\delta}$ and $\delta = \frac{\log(k \log 1/\varepsilon)}{\log 1/\varepsilon}$, we have $p_{ht^*}^- \leq \delta$ as follows. From Lemma 13, $1 - \delta > p_{t^*}^+ \geq 1 - \frac{2t^*\sqrt{\eta}}{\pi}$, hence $\delta < \frac{2t^*\sqrt{\eta}}{\pi}$. Multiplying h both sides and using the definition of h, we have $\log\frac{1}{\delta} < \frac{ht^*\sqrt{\eta/k}}{\pi}$. Again from Lemma 13,

$$p_{ht^*}^- \leq \exp(-\frac{ht^*\sqrt{\eta/l}}{\pi}) \leq \exp(-\frac{ht^*\sqrt{\eta/k}}{\pi}) < \exp(-\log\frac{1}{\delta}) = \delta.$$

Thus, all but one choice of t, $p_t^+ \geq 1 - \delta$ or $p_t^- \leq \delta$ holds. Thus, $\Pr[\beta \models C] \geq \frac{1}{s} \cdot 0 + (1 - \frac{1}{s})(1 - \delta) \geq 1 - \delta - \frac{1}{s}$. $\square$

Proof of Theorem 12. From Lemmas 14, 15, and 16, the probability that β does not satisfy a constraint is at most $O(h^s\sqrt{\varepsilon}) + 1/s + \exp(-\Omega(h/\sqrt{2k})) + \delta$. From the choice of s, δ, we have $\Pr[\beta \models C] = 1 - O(\delta)$. $\square$

4 Quantitative Characterization

In this section, we prove Theorem 2, which is a "quantitative" characterization of robust approximability. Item 4 is already proved in Lemma 12. Items 1, 2 and 3 will be proved in the following sections.

4.1 Inapproximability of Correlation Clustering

In this section, we prove Item 1 of Theorem 2. Since any non-trivial Γ pp-defines $=$ and $\neq$ from Lemma 6, it suffices to show the following.

Lemma 17. *It is UG-Hard to compute $o(\sqrt{\varepsilon})$-robust approximation of $\mathsf{CSP}(\{=,\neq\})$.*

We show a reduction from Max-CUT to $\mathsf{CSP}(\{=,\neq\})$, then apply the following theorem.

Theorem 18 ([19])**.** *It is UG-Hard to compute $o(\sqrt{\varepsilon})$-robust approximation for Max-CUT.*

The reduction is as follows. Let a graph $G=(V,E)$ be an instance of Max-CUT. We construct a weighted graph $\widehat{G}=(\widehat{V},E_=\cup E_{\neq},W)$ as: (i) $\widehat{V}:=\{v_i \mid v\in V, i\in\{0,1\}\}$. (ii) $E_=:=\{(u_i,v_{1-i})\mid (u,v)\in E, i\in\{0,1\}\}$. (iii) $E_{\neq}:=\{(v_0,v_1)\mid v\in V\}$. (iv) $W: E_=\cup E_{\neq}\to[0,1]$ as $W(e)=\frac{1}{4|E|}$ if $e\in E_=$, $W(e)=\frac{d(v)}{4|E|}$ if $e=(v_0,v_1)\in E_{\neq}$. Here $d(v)$ denotes the degree of v in G. Note that $\sum_{e\in E_=}W(e)=\sum_{e\in E_{\neq}}W(e)=\frac{1}{2}$. We can regard $\widehat{G}$ as an instance of $\mathsf{CSP}(\{=,\neq\})$, and the following two lemmas hold.

Lemma 19. *If* $\mathbf{opt}(G)\geq 1-\varepsilon$, *then* $\mathbf{opt}(\widehat{G})\geq 1-\varepsilon/2$.

Proof. Let $l: V\to\{0,1\}$ be a labeling for G with $\mathbf{opt}(G)\geq 1-\varepsilon$. Define $\widehat{l}:\widehat{V}\to\{0,1\}$, a labeling of $\widehat{G}$, as: $\widehat{l}(v_0)=l(v)$ and $\widehat{l}(v_1)=1-l(v)$. Then, $\widehat{l}$ satisfies a $1-\varepsilon$ fraction of edges in $E_=$ and every edge in $E_{\neq}$. Therefore, $\mathbf{opt}(\widehat{G})\geq(1-\varepsilon)\times 1/2+1/2=1-\varepsilon/2$. □

Lemma 20. *If* $\mathbf{opt}(\widehat{G})\geq 1-\varepsilon$, *then* $\mathbf{opt}(G)\geq 1-2\varepsilon$.

Proof. First we show that if $\mathbf{opt}(\widehat{G})=1$, then $\mathbf{opt}(G)=1$. Without loss of generality, we can assume that G is connected. An optimal cut $l: V\to\{0,1\}$ is defined as follows. Pick an arbitrary vertex $v_0^*\in\widehat{V}$ and define $V_0:=\{v_0\in\widehat{V}\mid v_0 \text{ is reachable from } v_0^* \text{ using only edges in } E_=\}$, and $l(v)=0$ iff $v_0\in V_0$. Note that if $(u,v)\in E$, then exactly one of u_0,v_0 is in V_0, thus, l is an optimal cut.

Now we assume $\mathbf{opt}(\widehat{G})\geq 1-\varepsilon$ and a labeling $\widehat{l}:\widehat{V}\to\{1,2,\ldots,2|V|\}$ is optimal. We say a pair of vertices (v_0,v_1) is *good* if $\widehat{l}(v_0)\neq\widehat{l}(v_1)$. Consider a subgraph $\widehat{G}'$ induced by good vertices from $\widehat{G}$. To obtain $\widehat{G}'$, we need to remove at most an ε fraction of edges from $\widehat{G}$. Thus, the total weight of satisfied edges is at least $1-2\varepsilon$ in $\widehat{G}'$. Let $\widehat{G}''$ be a subgraph obtained from $\widehat{G}'$ by deleting all unsatisfied edges. Then, we can construct a cut from the labeling of $\widehat{G}''$ so that $\mathbf{opt}(G)\geq 1-2\varepsilon$, by similar reasoning for the case of $\mathbf{opt}(\widehat{G})=1$. □

Combining Theorem 18 and Lemmas 19, 20, we complete the proof of Lemma 17.

4.2 Approximability of Negative =-SAT

In this section, we prove Item 2 of Theorem 2. Let Γ_k be the equality constraint language consisting of negative clauses of at most k literals. Since every negative formula is pp-definable in Γ_k for some k, we consider $\mathsf{CSP}(\Gamma_k)$.

Given an instance $\mathcal{I}=(V,\mathcal{C})$ of $\mathsf{CSP}(\Gamma_k)$, let $\mathcal{C}_=$ be the set of constraints of the form $(u=v)$ and $\mathcal{C}_{\neq}=\mathcal{C}\setminus\mathcal{C}_=$. Then, we solve BasicSDP. For each constraint $C\in\mathcal{C}$, we let $\boldsymbol{y}_C=\Pr_{\beta\sim\boldsymbol{\mu}_C}[\beta\models C]$. Then, we have:

$$\begin{aligned}
\boldsymbol{y}_C &\leq \langle \boldsymbol{x}_u,\boldsymbol{x}_v\rangle \qquad \text{if } C\in\mathcal{C}_=,\\
\boldsymbol{y}_C &\leq \sum_{(u\neq v)\in C}(1-\langle \boldsymbol{x}_u,\boldsymbol{x}_v\rangle) \qquad \text{if } C\in\mathcal{C}_{\neq}.
\end{aligned}$$

Our rounding scheme uses t random hyperplanes to define an assignment β as was the case for Horn $=$-SAT, but here we fix $t = 10\sqrt{k}\log(1/\varepsilon)$.

Proof of Item 2 of Theorem 2. We can safely assume that each constraint C satisfies $\boldsymbol{y}_C \geq 1/2$ (At most an $O(\varepsilon)$-fraction of constraints can satisfy $\boldsymbol{y}_C < 1/2$). For a constraint $C \in \mathcal{C}$, we set $\varepsilon_C = 1 - \boldsymbol{y}_C$.

We consider the loss caused by $\mathcal{C}_=$. From Lemma 13, if $\boldsymbol{y}_C \geq 1-\delta$ for $C \in \mathcal{C}_=$, then $\Pr[\beta \models C] = 1 - O(\sqrt{k\delta}\log(1/\varepsilon))$. Thus, the total loss is proportional to

$$\frac{1}{m}\sum_{C \in \mathcal{C}_=} \sqrt{k\varepsilon_C}\log(1/\varepsilon) \leq \frac{\sqrt{k}\log(1/\varepsilon)}{m}\sqrt{|\mathcal{C}_=|\sum_{C \in \mathcal{C}_=}\varepsilon_C} \leq \sqrt{k}\log(1/\varepsilon)\sqrt{\frac{1}{m}\sum_{C \in \mathcal{C}_=}\varepsilon_C} \leq \sqrt{k\varepsilon}\log(1/\varepsilon).$$

The first inequality is by Cauchy-Schwartz.

We now turn to $\mathcal{C}_{\neq}$. Let $C \in \mathcal{C}_{\neq}$ be a constraint of l literals. Then, We have $\sum_{i=1}^{l}\langle \boldsymbol{x}_{u_i}, \boldsymbol{x}_{v_i}\rangle \leq l - 1 + 1/2 = l - 1/2$ from $\varepsilon_C \leq 1/2$. Thus, there exists some $i \in [l]$ with $\langle \boldsymbol{x}_{u_i}, \boldsymbol{x}_{v_i}\rangle \leq 1 - \frac{1}{2l}$. From Lemma 13, we have $\Pr[\beta \not\models C] = \exp(-\frac{t\sqrt{1/l}}{\pi}) = O(\sqrt{\varepsilon})$. Thus, the total loss is at most $\frac{1}{m}\sum_{C \in \mathcal{C}_{\neq}} O(\sqrt{\varepsilon}) = O(\sqrt{\varepsilon})$.

In summary, the total loss is at most $O(\sqrt{k\varepsilon}\log(1/\varepsilon)) + O(\sqrt{\varepsilon}) = O(\sqrt{k\varepsilon}\log(1/\varepsilon))$. $\square$

4.3 Inapproximability of Non-Negative $=$-SAT

In this section, we prove Item 3 of Theorem 2. We introduce a relation $\mathrm{ODD}_3(x,y,z) = \{(x,y,z) \in \mathbb{Q}^3 \mid |\{x,y,z\}| = 1 \text{ or } 3\}$. We use the following fact.

Lemma 21 ([5])**.** *Let Γ be an equality constraint language such that Γ is not preserved by a constant operation and some relation $R \in \Gamma$ is not negative. Then,* ODD_3 *is pp-definable in* Γ*.*

From Corollary 7, Lemmas 5 and 21, it suffices to show the following inapproximability result.

Lemma 22. *It is UG-Hard to compute $o(\frac{1}{\log 1/\varepsilon})$-robust approximation of CSP$(\{\mathrm{ODD}_3, \neq\})$.*

We will give an instance $\mathcal{I}$ with $\mathbf{sdp}(\mathcal{I}) = 1-\varepsilon$ and $\mathbf{opt}(\mathcal{I}) = 1 - O(\frac{1}{\log 1/\varepsilon})$. Then, we have the desired result from Lemma 8. We borrow several ideas from [15], which shows that computing $o(\frac{1}{\log 1/\varepsilon})$-robust approximation of Horn SAT (over the Boolean domain) is UG-Hard.

Given a parameter k, our integrality gap instance $\mathcal{I} = (V, \mathcal{C})$ looks as follows.

$$\begin{aligned}
\text{Variables} &: u_1, \ldots, u_k, v_1, \ldots, v_k \\
\text{Initial constraint} &: \mathrm{ODD}_3(u_1, u_1, v_1) \\
\text{Block } i\ (1 \leq i \leq k-1) &: \begin{cases} \mathrm{ODD}_3(u_i, v_i, u_{i+1}), \\ \mathrm{ODD}_3(u_i, v_i, v_{i+1}) \end{cases} \\
\text{Final constraint} &: (u_k \neq v_k)
\end{aligned}$$

We intend to set $u_1 = v_1$ using the initial constraint and to set $u_i = v_i = u_{i+1} = v_{i+1}$ using Block i. Because of the final constraint, the instance $\mathcal{I}$ is unsatisfiable. Since $\mathcal{I}$ has $2k$ constraints, we have $\mathbf{opt}(\mathcal{I}) \leq 1 - \frac{1}{2k}$.

Now we show that $\mathbf{sdp}(\mathcal{I}) \geq 1 - \frac{1}{\exp(k)}$. Suppose that we have fixed SDP vectors $\boldsymbol{x} = \{\boldsymbol{x}_v\}_{v \in V}$ in BasicSDP. Then, for each constraint $C \in \mathcal{C}$, the optimal probability distribution $\boldsymbol{\mu}_C$ is locally determined from $\boldsymbol{x}$. Thus, to construct a good SDP solution, we can concentrate on constructing good SDP vectors $\boldsymbol{x}$. We say that $\boldsymbol{x}$ *satisfies* a constraint C if there is a probability distribution $\boldsymbol{\mu}_C$ that is consistent with $\boldsymbol{x}$ such that $\Pr_{\beta \sim \boldsymbol{\mu}_C}[\beta \models C] = 1$.

For $\delta = \frac{1}{\exp(k)}$, our SDP vectors $\boldsymbol{x}$ will satisfy the initial constraint up to $1-\delta$ and completely satisfy Block i $(1 \leq i \leq k-1)$ and the final constraint. Since it is hard to construct all the SDP vectors at once, we make SDP vectors for each block first so that they agree with each other on some interface, and then we coalesce them together. The following definition and claim help us bring down the difficulty.

Definition 23 (partial SDP solution)**.** *Let $\mathcal{C}' \subseteq \mathcal{C}$ be a set of constraints. Then, SDP vectors $\{\boldsymbol{x}_v\}_{v \in V'}$ for $V' \subseteq V$ is said to be a partial SDP solution on $\mathcal{C}'$ if every constraint in $\mathcal{C}'$ is satisfied by $\boldsymbol{x}$. (In particular, $\boldsymbol{x}_v$ must be defined for every variable v that appears in $\mathcal{C}'$.)*

An easy modification of Claim 7 of [15] gives the following.

Lemma 24 ([15])**.** *Let $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathcal{C}$ be two disjoint set of constraints. Let $\boldsymbol{x}^1 = \{\boldsymbol{x}^1_v\}_{v \in V_1}$ and $\boldsymbol{x}^2 = \{\boldsymbol{x}^2_v\}_{v \in V_2}$ be partial SDP solutions on $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. Suppose that, for all $u, v \in V_1 \cap V_2$, it holds that $\langle \boldsymbol{x}^1_u, \boldsymbol{x}^1_v \rangle = \langle \boldsymbol{x}^2_u, \boldsymbol{x}^2_v \rangle$. Then, there exists a partial SDP solution $\boldsymbol{y}$ on $\mathcal{C}_1 \cup \mathcal{C}_2$ that preserves inner products between vectors corresponding to variables in $\mathcal{C}_1 \cap \mathcal{C}_2$.*

Now we construct a partial SDP solution for each block.

Lemma 25. *For any $0 \leq \delta \leq 1/2$ and $1 \leq i \leq k-1$, there exists a partial SDP solution $\{\boldsymbol{x}_{u_i}, \boldsymbol{x}_{v_i}, \boldsymbol{x}_{u_{i+1}}, \boldsymbol{x}_{v_{i+1}}\}$ to Block i such that*

$$\langle \boldsymbol{x}_{u_i}, \boldsymbol{x}_{v_i} \rangle = 1 - \delta \text{ and } \langle \boldsymbol{x}_{u_{i+1}}, \boldsymbol{x}_{v_{i+1}} \rangle = 1 - 2\delta.$$

Proof. Consider the following matrix whose columns and rows correspond to $\boldsymbol{x}_{u_i}$, $\boldsymbol{x}_{v_i}$, $\boldsymbol{x}_{u_{i+1}}$, $\boldsymbol{x}_{v_{i+1}}$ in this order and each element represents the inner product between corresponding vectors.

$$A = \begin{pmatrix} 1 & 1-\delta & 1-\delta & 1-\delta \\ 1-\delta & 1 & 1-\delta & 1-\delta \\ 1-\delta & 1-\delta & 1 & 1-2\delta \\ 1-\delta & 1-\delta & 1-2\delta & 1 \end{pmatrix}.$$

This matrix A satisfies the condition of the lemma, and we can construct a probability distribution satisfying Block i that is consistent to inner products determined by A. For example, for the constraint $\mathrm{ODD}_3(u_i, v_i, u_{i+1})$, we can use the probability distribution for which $|\{u_i, v_i, u_{i+1}\}| = 1$ with probability $1-\delta$ and $|\{u_i, v_i, u_{i+1}\}| = 3$ with probability δ.

To ensure there are vectors realizing the matrix A, we need to show that A is positive semidefinite. Let J be the all-one matrix. Then,

$$A = (1-2\delta)J + \delta \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 0 \\ 1 & 1 & 0 & 2 \end{pmatrix}.$$

We can check that last matrix is positive semidefinite. Thus, A is a sum of semidefinite matrices and hence A is also positive semidefinite. $\square$

Lemma 26. $\mathbf{sdp}(\mathcal{I}) \geq 1 - \frac{1}{k2^{k+1}}$.

Proof. Let $\delta > 0$ be a sufficiently small value. By combining partial SDP solutions given by Lemma 25 iteratively using Lemma 24, we have an SDP solution $\boldsymbol{x} = \{\boldsymbol{x}_v\}_{v \in V}$ with the following property: it is a partial SDP solution for all constraints in Blocks 1 to $k-1$, and

$$\langle \boldsymbol{x}_{u_1}, \boldsymbol{x}_{v_1} \rangle = 1 - \delta, \quad \langle \boldsymbol{x}_{u_k}, \boldsymbol{x}_{v_k} \rangle = 1 - 2^k \delta.$$

Then, the loss from the initial constraint is δ, and the loss from the final constraint is $1 - 2^k\delta$. By choosing $\delta = 1/2^k$, the optimal SDP value is at least $1 - \frac{\delta}{2k} = 1 - \frac{1}{k2^{k+1}}$. $\square$

Since $\mathbf{opt}(\mathcal{I}) \leq 1 - \frac{1}{2k}$ whereas $\mathbf{sdp}(\mathcal{I}) \geq 1 - \frac{1}{k2^{k+1}}$, we have Lemma 22 from Lemma 8, which gives Item 3 of Theorem 2.

5 Robust Approximation of Horn =-SAT in Almost Liner Time

In this section, we show that we can solve BasicSDP for Horn =-SAT in almost linear time. Since rounding can be done in linear time, we can obtain an $O(\log\log(1/\varepsilon)/\log(1/\varepsilon))$-robust approximation for Horn =-SAT as well as an $O(\sqrt{\varepsilon}\log(1/\varepsilon))$-robust approximation for Negative =-SAT in almost linear time. Recall that Negative =-SAT is a special case of Horn =-SAT.

For a TCSP instance $\mathcal{I}$, let $\mathcal{I}_q$ be the instance whose domain is restricted to $[q]$ instead of $\mathbb{Q}$. The following lemma says that, if q is large enough, then the optimal value does not decrease much by using only q values.

Lemma 27. *Let $\mathcal{I}$ be an instance of Horn =-SAT. Then, $\mathbf{opt}(\mathcal{I}_q) \geq (1-\frac{1}{q})\mathbf{opt}(\mathcal{I})$ holds.*

Proof. Let $\beta^* : V \to \mathbb{Q}$ be the optimal assignment for $\mathcal{I}$. Let $\phi : \mathbb{Q} \to [q]$ be a random mapping. (We do not have to define the whole mapping explicitly as the size of the range of β^* is bounded by $|V|$.) Then, we construct q-valued β from β^* by setting $\beta_v = \phi(\beta^*_v)$. Let C be a constraint satisfied by β^*. If C is of the form $(u = v)$, then $\Pr[\beta \models C] = 1$. If C is of the form $\wedge_{i=1}^{k}(u_i = v_i) \to \mathbf{false}$ for some $k \geq 1$, then there exists some $i \in [k]$ such that $\beta^*(u_i) \neq \beta^*(v_i)$. Thus, $\Pr[\beta \models C] \geq 1 - \frac{1}{q}$. Finally, suppose C is of the form $\wedge_{i=1}^{k-1}(u_i = v_i) \to (u_k = v_k)$ for some $k \geq 1$. Then, $\beta^*(u_k) = \beta^*(v_k)$ holds or there exists some $i \in [k]$ such that $\beta^*(u_i) \neq \beta^*(v_i)$. From the same reasoning, $\Pr[\beta \models C] \geq 1 - \frac{1}{q}$ holds. □

For CSP over finite domains, it is known that an almost optimal SDP solution can be obtained in almost linear time as follows.

Lemma 28 ([25])**.** *Let $\mathcal{I} = (V, \mathcal{C})$ be a CSP instance on n variables over the domain $[q]$ with m constraints and maximum arity k. Suppose $\mathbf{sdp}(\mathcal{I}) \geq \alpha$. Then for every $\varepsilon > 0$, we can compute in time $m \cdot \mathrm{poly}(k^q/\varepsilon) \cdot \mathrm{poly}\log n$ an SDP solution of value at least $\alpha - \varepsilon$ that is feasible for a CSP instance $\mathcal{I}'$ obtained from $\mathcal{I}$ by discarding at most an ε-fraction of constraints.*

Lemma 29. *Let $\mathcal{I} = (V, \mathcal{C})$ be an instance of Horn =-SAT of maximum arity k with $\mathbf{opt}(\mathcal{I}) \geq \alpha$. Then, we can compute in time $m \cdot \mathrm{poly}(k^{1/\varepsilon}/\varepsilon) \cdot \mathrm{poly}\log n$ an SDP solution of value at least $\alpha - O(\varepsilon)$.*

Proof. We set $q = 1/\varepsilon$. From Lemma 27, $\mathbf{opt}(\mathcal{I}_q) \geq \alpha - \varepsilon$. Using Lemma 28, we obtain a feasible SDP solution $\{\boldsymbol{x}_{u,a}\}_{u \in V, a \in [q]}$ of value at least $\alpha - O(\varepsilon)$. Here, the $O(\cdot)$ notation arises since we have discarded an ε-fraction of constraints from $\mathcal{I}_q$,

Now, we define $\boldsymbol{x}_u$ as $\oplus_{i=1}^{q}\boldsymbol{x}_{u,i}$ and claim $\{\boldsymbol{x}_u\}_{u \in V}$ is a good SDP solution for $\mathcal{I}$. The objective value does not change since $\langle \boldsymbol{x}_u, \boldsymbol{x}_v\rangle = \sum_{i \in [q]}\langle \boldsymbol{x}_{u,i}, \boldsymbol{x}_{v,i}\rangle$ and the objective value is only determined by these inner products. Moreover, constraints in BasicSDP are satisfied since $\|\boldsymbol{x}_u\|^2 = \sum_{i=1}^{q}\|\boldsymbol{x}_{u,i}\|^2 = 1$ and $\langle \boldsymbol{x}_u, \boldsymbol{x}_v\rangle = \sum_i \langle \boldsymbol{x}_{u,i}, \boldsymbol{x}_{v,i}\rangle \geq 0$. □

Combining the rounding method given in Sections 3 and 4, we have the following.

Corollary 30. *For Horn =-SAT (resp., Negative =-SAT) of maximum arity k, In $m \cdot \mathrm{poly}(k^{1/\varepsilon}/\varepsilon) \cdot \mathrm{poly}\log n$, we can compute an $O(\frac{\log(k\log 1/\varepsilon)}{\log 1/\varepsilon})$-robust approximation (resp., an $O(\sqrt{\varepsilon}\log(1/\varepsilon)$-robust approximation).*

References

[1] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004.

[2] L. Barto and M. Kozik. Robust satisfiability of constraint satisfaction problems. In *Proceedings of the 44th Symposium on Theory of Computing Conference (STOC)*, pages 931–940, 2012.

[3] M. Bodirsky. *Constraint satisfaction with infinite domains*. PhD thesis, Humboldt-Universität zu Berlin, 2004.

[4] M. Bodirsky. Complexity classification in infinite-domain constraint satisfaction. *CoRR*, abs/1201.0856, 2012.

[5] M. Bodirsky, H. Chen, and M. Pinsker. The reducts of equality up to primitive positive interdefinability. *J. Symb. Log.*, 75(4):1249–1292, 2010.

[6] M. Bodirsky and J. Kára. The complexity of equality constraint languages. *Theory Comput. Syst.*, 43(2):136–158, 2008.

[7] M. Bodirsky and J. Kára. The complexity of temporal constraint satisfaction problems. *J. ACM*, 57(2), 2010.

[8] M. Charikar, V. Guruswami, and A. Wirth. Clustering with qualitative information. *J. Comput. Syst. Sci.*, 71(3):360–383, 2005.

[9] N. Creignou, P. G. Kolaitis, and H. Vollmer, editors. *Complexity of Constraints - An Overview of Current Research Themes [Result of a Dagstuhl Seminar]*, volume 5250 of *Lecture Notes in Computer Science*. Springer, 2008.

[10] V. Dalmau and A. A. Krokhin. Robust satisfiability for csps: Hardness and algorithmic results. *TOCT*, 5(4):15, 2013.

[11] R. Dechter. *Constraint Processing*. Morgan Kaufmann, 2003.

[12] T. Feder and M. Y. Vardi. The computational structure of monotone monadic snp and constraint satisfaction: A study through datalog and group theory. *SIAM J. Comput.*, 28(1):57–104, 1998.

[13] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman and Company, 1979.

[14] V. Guruswami, J. Håstad, R. Manokaran, P. Raghavendra, and M. Charikar. Beating the random ordering is hard: Every ordering csp is approximation resistant. *SIAM J. Comput.*, 40(3):878–914, 2011.

[15] V. Guruswami and Y. Zhou. Tight bounds on the approximability of almost-satisfiable Horn SAT and Exact Hitting Set. *Theory of Computing*, 8(1):239–267, 2012.

[16] J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

[17] S. Khot. On the power of unique 2-prover 1-round games. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 767–775, 2002.

[18] S. Khot. On the unique games conjecture (invited survey). In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity (CCC)*, pages 99–121, 2010.

[19] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM J. Comput.*, 37(1):319–357, 2007.

[20] G. Kun, R. O'Donnell, S. Tamaki, Y. Yoshida, and Y. Zhou. Linear programming, width-1 CSPs, and robust satisfaction. In *Proceedings of the 3rd Innovations in Theoretical Computer Science (ITCS) conference*, pages 484–495, 2012.

[21] B. Nebel and H.-J. Bürckert. Reasoning about temporal relations: A maximal tractable subclass of allen's interval algebra. *J. ACM*, 42(1):43–66, 1995.

[22] P. Raghavendra. *Approximating NP-hard problems: efficient algorithms and their limits*. PhD thesis, University of Washington, 2009.

[23] F. Rossi, P. van Beek, and T. Walsh, editors. *Handbook of Constraint Programming*. Foundations of Artificial Intelligence. Elsevier Science, 2006.

[24] T. J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC)*, pages 216–226, 1978.

[25] D. Steurer. Fast sdp algorithms for constraint satisfaction problems. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 684–697, 2010.

[26] P. van Beek. Reasoning about qualitative temporal information. *Artif. Intell.*, 58(1-3):297–326, 1992.

[27] M. Vilain, H. Kautz, and P. van Beek. Constraint propagation algorithms for temporal reasoning: A revised report. In D. S. Weld and J. de Kleer, editors, *Readings in Qualitative Reasoning about Physical Systems*, pages 373–381, 1989.

[28] U. Zwick. Finding almost-satisfying assignments. In *Proceedings of the 30th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 551–560, 1998.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de