

Near-Optimal Algorithms for Maximum Constraint Satisfaction Problems

Moses Charikar

Princeton University

and

Konstantin Makarychev

IBM T.J. Watson Research Center

and

Yury Makarychev

Microsoft Research New England

In this paper, we present two approximation algorithms for the maximum constraint satisfaction problem with k variables in each constraint (MAX k -CSP).

Given a $(1 - \varepsilon)$ satisfiable 2CSP our first algorithm finds an assignment of variables satisfying a $1 - O(\sqrt{\varepsilon})$ fraction of all constraints. The best previously known result, due to Zwick, was $1 - O(\varepsilon^{1/3})$.

The second algorithm finds a $ck/2^k$ approximation for the MAX k -CSP problem (where $c > 0.44$ is an absolute constant). This result improves the previously best known algorithm by Hast, which had an approximation guarantee of $\Omega(k/(2^k \log k))$.

Both results are optimal assuming the Unique Games Conjecture and are based on rounding natural semidefinite programming relaxations. We also believe that our algorithms and their analysis are simpler than those previously known.

Categories and Subject Descriptors: F.2.0 [Analysis of Algorithms and Problem Complexity]: General

General Terms: Algorithms, Theory

Additional Key Words and Phrases: MAX k -CSP, MAX 2CSP, SDP

1. INTRODUCTION

In this paper we study the maximum constraint satisfaction problem with k variables in each constraint (MAX k -CSP): Given a set of boolean variables and constraints, where each constraint depends on k variables, our goal is to find an assignment so as to maximize the number of satisfied constraints.

Several instances of 2CSPs have been well studied in the literature and semidefinite programming approaches have been very successful for these problems. In their seminal paper, Goemans and Williamson [1995] gave a semidefinite programming based algorithm for MAX CUT, a special case of MAX 2CSP. If the opti-

A preliminary version of this article appeared in Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms, pages 62–68.

This work was done while K. Makarychev and Y. Makarychev were graduate students at Princeton University.

M. Charikar was supported by NSF ITR grant CCR-0205594, NSF CAREER award CCR-0237113, MSPA-MCS award 0528414, and an Alfred P. Sloan Fellowship. K. Makarychev was supported by a Gordon Wu fellowship. Y. Makarychev was supported by a Gordon Wu fellowship.

Authors addresses: M. Charikar, Princeton University, Department of Computer Science, 35 Olden Street, Princeton, NJ 08540, email: moses@cs.princeton.edu. K. Makarychev, IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598, email: konstantin@us.ibm.com. Y. Makarychev, Microsoft Research New England One Memorial Dr., Cambridge, MA 02142, email: yurym@microsoft.com.

Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© 20YY ACM 0000-0000/20YY/0000-0001 \$5.00

	MAX CUT	MAX 2CSP
Approximation ratio	0.878 [Goemans and Williamson 1995]	0.874 [Lewin et al. 2002]
Almost satisfiable instances		
• $\varepsilon > 1/\log n$	$1 - O(\sqrt{\varepsilon})$ [Goemans and Williamson 1995]	$1 - O(\varepsilon^{1/3})$ [Zwick 1998]
• $\varepsilon < 1/\log n$	$1 - O(\sqrt{\varepsilon \log n})$ [Agarwal et al. 2005]	$1 - O(\sqrt{\varepsilon \log n})$ [Agarwal et al. 2005]

Table I. Note that the approximation ratios were almost the same for MAX CUT and MAX 2CSP; and in the case of almost satisfiable instances the approximation guarantees were the same for $\varepsilon < 1/\log n$, but not for $\varepsilon > 1/\log n$.

mal solution satisfies OPT constraints (in this problem satisfied constraints are cut edges), their algorithm finds a solution satisfying at least $\alpha_{GW} \cdot OPT$ constraints, where $\alpha_{GW} \approx 0.878$. Given an almost satisfiable instance (where $OPT = 1 - \varepsilon$), the algorithm finds an assignment of variables that satisfies a $(1 - O(\sqrt{\varepsilon}))$ fraction of all constraints.

In the same paper, Goemans and Williamson [1995] also gave a 0.796 approximation algorithm for MAX DICUT and a 0.878 approximation algorithm for MAX 2SAT. These results were improved in several follow-up papers: Feige and Goemans [1995], Zwick [2000], Matuura and Matsui [2003], and Lewin, Livnat and Zwick [2002]. The approximation ratios obtained by Lewin, Livnat and Zwick [2002] are 0.874 for MAX DICUT and 0.940 for MAX 2SAT. The algorithm of Lewin, Livnat and Zwick [2002] can also be used for solving MAX 2CSP. Note that their approximation guarantee for an arbitrary MAX 2CSP almost matches the approximation guarantee of Goemans and Williamson [1995] for MAX CUT.

Khot, Kindler, Mossel, and O’Donnell [2007] recently showed that both results of Goemans and Williamson [1995] for MAX CUT are optimal and the results of Lewin, Livnat and Zwick [2002] are almost optimal¹ assuming Khot’s Unique Games Conjecture [Khot 2002]. The MAX 2SAT hardness result was further improved by Austrin [2007a], who showed that the MAX 2SAT algorithm of Lewin, Livnat and Zwick [2002] is optimal assuming the Unique Games Conjecture.

An interesting gap remained for almost satisfiable instances of MAX 2CSP (*i.e.* where $OPT = 1 - \varepsilon$). On the positive side, Zwick [1998] developed an approximation algorithm that satisfies a $1 - O(\varepsilon^{1/3})$ fraction of all constraints². However the best known hardness result [Khot et al. 2007] (assuming the Unique Games Conjecture) is that it is hard to satisfy $1 - O(\sqrt{\varepsilon})$ fraction of constraints. In this paper, we close the gap by presenting a new approximation algorithm that satisfies $1 - O(\sqrt{\varepsilon})$ fraction of all constraints. Our approximation guarantee for arbitrary MAX 2CSP matches the guarantee of Goemans and Williamson [1995] for MAX CUT. Table I compares the previous best known results for the two problems.

So far, we have discussed MAX k -CSP for $k = 2$. The problem becomes much harder for $k \geq 3$. In contrast to the $k = 2$ case, it is NP-hard to find a satisfying assignment for 3CSP. Moreover, according to Håstad’s 3-bit PCP Theorem [Håstad 2001], if $(1 - \varepsilon)$ fraction of all constraints is satisfied in the optimal solution, we cannot find a solution satisfying more than $(1/2 + \varepsilon)$ fraction of constraints.

The approximation factor for MAX k -CSP is of interest in complexity theory since it is closely tied to the relationship between the completeness and soundness of k -bit PCPs. A trivial algorithm for k -CSP is to pick a random assignment. It satisfies each constraint with probability at least $1/2^k$ (except those constraints which cannot be satisfied). Therefore, its approximation ratio is $1/2^k$. Trevisan [1998] improved on this slightly by giving an algorithm with approximation ratio $2/2^k$. Until recently, this was the best approximation ratio for the problem. Recently, Hast [2005] proposed an algorithm with an asymptotically

¹Khot, Kindler, Mossel, and O’Donnell [2007] proved 0.943 hardness result for MAX 2SAT and 0.878 hardness result for MAX 2CSP.

²He developed an algorithm for MAX 2SAT, but it is easy to see that in the case of almost satisfiable instances MAX 2SAT is equivalent to MAX 2CSP (see Section 2.1 for more details).

better approximation guarantee $\Omega(k/(2^k \log k))$. Also, Samorodnitsky and Trevisan [2006] proved that it is NP-hard to approximate MAX k -CSP within $2k/2^k$ for every $k \geq 3$, and within $(k+1)/2^k$ for infinitely many k assuming the Unique Games Conjecture of Khot [2002]. We close the gap between the upper and lower bounds for k -CSP by giving an algorithm with approximation ratio $\Omega(k/2^k)$. By the results of [Samorodnitsky and Trevisan 2006], our algorithm is asymptotically optimal within a factor of approximately $1/0.44 \approx 2.27$ (assuming the Unique Games Conjecture).

In our algorithm, we use the approach of Hast [2005]: we first obtain a “preliminary” solution $z_1, \dots, z_n \in \{-1, 1\}$ and then independently flip the values of z_i using a slightly biased distribution (*i.e.* we keep the old value of z_i with probability slightly larger than $1/2$). In this paper, we improve and simplify the first step in this scheme. Namely, we present a new method of finding z_1, \dots, z_n , based on solving a certain semidefinite program (SDP) and then rounding the solution to ± 1 using the result of Rietz [1974] and Nesterov [1997]. Note, that Hast obtains z_1, \dots, z_n by maximizing a quadratic form (which differs from our SDP) over the domain $\{-1, 1\}$ using the algorithm of Charikar and Wirth [2004]. The second step of our algorithm is essentially the same as in Hast’s algorithm.

Our result is also applicable to MAX k -CSP with a larger domain. It gives a $\Omega(k \log d/d^k)$ approximation for instances with domain size d . To apply the result to an instance with a larger domain, we just encode each domain value with $\log d$ bits. We describe the details in Section 3.4.

In Section 2, we describe our algorithm for MAX 2CSP and in Section 3, we describe our results for MAX k -CSP. Both algorithms are based on exploiting information from solutions to natural SDP relaxations for the problems.

Added in proof. After this paper appeared at SODA 2007, Austrin and Mossel [2008], and Guruswami and Raghavendra [2008] showed that assuming the Unique Games Conjecture, it is NP-hard to approximate MAX k -CSP with domain size d within factor $\Omega(kd^2/d^k)$ (see [Austrin and Mossel 2008; Guruswami and Raghavendra 2008] for more details). Guruswami and Raghavendra [2008] also proposed an alternative algorithm for MAX k -CSP with non-boolean domain. However, our algorithm for that problem gives a better approximation guarantee (of $\Omega(k \log d/d^k)$).

Austrin [2007b] proved that assuming the Unique Games Conjecture and a certain geometric conjecture it is NP-hard to approximate any 2CSP problem with the ratio better than the integrality gap of the SDP relaxation. Raghavendra [2008] proved that statement for arbitrary k -CSP assuming only the Unique Games Conjecture.

2. APPROXIMATION ALGORITHM FOR MAX 2CSP

2.1 SDP Relaxation

In this section we describe the vector program (SDP) for MAX 2CSP/MAX 2SAT.

Definition 2.1 MAX k -AllEqual Problem. Given a set of variables x_1, \dots, x_n and a set of constraints $p_k(x_i, x_j)$, where p_k is an arbitrary boolean predicates. The goal is to find an assignment to the variables x_i so as to maximize the number of satisfied constraints.

For convenience we replace each negation \bar{x}_i with a new variable x_{-i} that is equal by definition to \bar{x}_i . First, we transform our instance to a MAX 2SAT formula: we replace

- each constraint of the form $x_i \wedge x_j$ with two clauses x_i and x_j ;
- each constraint of the form $x_i \oplus x_j$ with two clauses $x_i \vee x_j$ and $x_{-i} \vee x_{-j}$;
- finally, each constraint x_i with $x_i \vee x_i$.

It is easy to see that the fraction of *unsatisfied* constraints in the formula is equal, up to a factor of 2, to the number of unsatisfied constraints in the original MAX 2CSP instance. Therefore, if we satisfy $1 - O(\sqrt{\varepsilon})$ fraction of all constraints in the 2SAT formula, we will also satisfy $1 - O(\sqrt{\varepsilon})$ fraction of all constraints in MAX 2CSP. In what follows, we will consider only 2SAT formulas. To avoid confusion between 2SAT and SDP constraints we will refer to them as clauses and constraints respectively. We denote the number of clauses by m .

We now rewrite all clauses in the form $x_i \rightarrow x_j$, where $i, j \in \{\pm 1, \pm 2, \dots, \pm n\}$. For each x_i , we introduce a vector variable v_i in the SDP. We also define a special unit vector v_0 that corresponds to the value 1: in

Figure 1: Approximation Algorithm for MAX 2CSP

- (1) Solve the SDP for MAX 2SAT. Denote by OPT_{SDP} the objective value of the solution and by ε the fraction of the clauses “unsatisfied” by the vector solution, that is,

$$\varepsilon = \frac{SDP}{m},$$

where m is the total number of clauses.

- (2) Pick a random Gaussian vector g with independent components distributed as $\mathcal{N}(0, 1)$.
(3) For every i ,
(a) Project the vector g to v_i :

$$\xi_i = \langle g, v_i \rangle.$$

Note, that ξ_i is a standard normal random variable, since v_i is a unit vector.

- (b) Pick a threshold t_i as follows:

$$t_i = -\langle v_i, v_0 \rangle / \sqrt{\varepsilon}.$$

- (c) If $\xi_i \geq t_i$, set $x_i = 1$, otherwise set $x_i = 0$.
-
-

the intended (integral) solution $v_i = v_0$, if $x_i = 1$; and $v_i = -v_0$, if $x_i = 0$. The SDP contains the constraints that all vectors are unit vectors and $v_i = -v_{-i}$.

For each clause $x_i \rightarrow x_j$ we add the term

$$\frac{1}{8} (\|v_j - v_i\|^2 - 2\langle v_j - v_i, v_0 \rangle)$$

to the objective function. In the intended solution this expression equals 1, if the clause is not satisfied; and 0, if it is satisfied. Therefore, our SDP is a relaxation of MAX 2SAT (the objective function measures how many clauses are not satisfied). Finally, we add constraints that ensure that all terms in the SDP objective function are non-negative.

We get an SDP relaxation for MAX 2SAT:

$$\text{minimize } \frac{1}{8} \sum_{\text{clauses } x_i \rightarrow x_j} \|v_j - v_i\|^2 - 2\langle v_j - v_i, v_0 \rangle$$

subject to

$$\begin{array}{ll} \|v_j - v_i\|^2 - 2\langle v_j - v_i, v_0 \rangle \geq 0 & \text{for all clauses } v_i \rightarrow v_j \\ \|v_i\|^2 = 1 & \text{for all } i \in \{0, \pm 1, \dots, \pm n\} \\ v_i = -v_{-i} & \text{for all } i \in \{\pm 1, \dots, \pm n\} \end{array}$$

In a slightly different form, this semidefinite program was introduced by Feige and Goemans [1995]. Later, Zwick [1998] used this SDP in his algorithm.

2.2 Algorithm and Analysis

The approximation algorithm is shown in Figure 1. We interpret the inner product $\langle v_i, v_0 \rangle$ as the bias towards rounding v_i to 1. The algorithm rounds vectors orthogonal to v_0 (“unbiased” vectors) using the random hyperplane technique. If, however, the inner product $\langle v_i, v_0 \rangle$ is positive, the algorithm shifts the random hyperplane; and it is more likely to round v_i to 1 than to 0.

It is easy to see that the algorithm always obtains a valid assignment to variables: if $x_i = 1$, then $x_{-i} = 0$ and vice versa. We will need several facts about normal random variables. Denote the probability that a

standard normal random variable is greater than $t \in \mathbb{R}$ by $\tilde{\Phi}(t)$, in other words

$$\tilde{\Phi}(t) \equiv 1 - \Phi_{0,1}(t) = \Phi_{0,1}(-t),$$

where $\Phi_{0,1}$ is the normal distribution function. The following lemma gives well-known lower and upper bounds on $\tilde{\Phi}(t)$.

LEMMA 2.2. *For every positive t ,*

$$\frac{t}{\sqrt{2\pi}(t^2 + 1)} e^{-\frac{t^2}{2}} < \tilde{\Phi}(t) < \frac{1}{\sqrt{2\pi}t} e^{-\frac{t^2}{2}}.$$

PROOF. Observe, that in the limit $t \rightarrow \infty$ all three expressions are equal to 0. Hence the lemma follows from the following inequality on the derivatives:

$$\left(\frac{t}{\sqrt{2\pi}(t^2 + 1)} e^{-\frac{t^2}{2}} \right)' > -\frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} > \left(\frac{1}{\sqrt{2\pi}t} e^{-\frac{t^2}{2}} \right)'$$

□

COROLLARY 2.3. *The following inequalities hold for³ $t \geq 0$: $\tilde{\Phi}(t) \leq e^{-\frac{t^2}{2}}$; and $\tilde{\Phi}(t) \leq \frac{1}{\sqrt{2\pi}t}$.*

LEMMA 2.4. *Let η be a random normal variable with variance Δ^2 and mean 0; and let t be a positive real number. Denote*

$$(\eta - t)^+ = \begin{cases} \eta - t, & \text{if } \eta - t \geq 0; \\ 0, & \text{otherwise.} \end{cases}$$

Then,

$$\mathbb{E} [(\eta - t)^+] \leq \sqrt{2\pi}\Delta \tilde{\Phi}(t/\Delta).$$

PROOF. First, we prove this statement for a random normal variable η with variance 1:

$$\begin{aligned} \mathbb{E} [(\eta - t)^+] &= \int_t^\infty \Pr(\eta \geq s) ds = \int_t^\infty \tilde{\Phi}(s) ds \\ &\leq \int_t^\infty e^{-s^2/2} ds = \sqrt{2\pi}\tilde{\Phi}(t). \end{aligned}$$

Now, for arbitrary variance Δ^2 , we have

$$\mathbb{E} [(\eta - t)^+] = \Delta \mathbb{E} \left[\left(\frac{\eta}{\Delta} - \frac{t}{\Delta} \right)^+ \right] \leq \sqrt{2\pi}\Delta \tilde{\Phi}(t/\Delta).$$

□

A clause $x_i \rightarrow x_j$ is not satisfied by the algorithm if $\xi_i \geq t_i$ and $\xi_j \leq t_j$ (i.e. x_i is set to 1; and x_j is set to 0). The following lemma bounds the probability of this event.

LEMMA 2.5. *Let ξ_i and ξ_j be two standard normal random variables with covariance $1 - 2\Delta^2$ (where $\Delta \geq 0$). For all real numbers t_i, t_j and $\delta = (t_j - t_i)/2$ we have (for some absolute constant C)*

(1) *If $t_j \leq t_i$,*

$$\Pr(\xi_i \geq t_i \text{ and } \xi_j \leq t_j) \leq C \min(\Delta^2/|\delta|, \Delta).$$

(2) *If $t_j \geq t_i$,*

$$\Pr(\xi_i \geq t_i \text{ and } \xi_j \leq t_j) \leq C(\Delta + 2\delta).$$

³Observe, that $e^{-\frac{1}{2}} > 1/2 = \tilde{\Phi}(0)$.

PROOF. 1. First note that if $\Delta = 0$, then the above inequality holds, since $\xi_i = \xi_j$ almost surely. If $\Delta \geq 1/2$, then the right hand side of the inequality becomes $\Omega(1) \times \min(1/|\delta|, 1)$. Since $\max(t_i, -t_j) \geq |\delta|/2$, the inequality follows from the bound $\tilde{\Phi}(|\delta|/2) \leq O(1/|\delta|)$. So we assume $0 < \Delta < 1/2$.

Let $\xi = (\xi_i + \xi_j)/2$ and $\eta = (\xi_i - \xi_j)/2$. Then ξ and η are independent normal random variables with variance $1 - \Delta^2$ and Δ^2 correspondingly. Write

$$\Pr(\xi_j \leq t_j \text{ and } \xi_i \geq t_i) = \Pr(\xi - \eta \leq t_j \text{ and } \xi + \eta \geq t_i) = \Pr(t_i - \eta \leq \xi \leq \eta + t_j).$$

The density of the normal distribution with variance $1 - \Delta^2$ does not exceed $1/\sqrt{2\pi(1 - \Delta^2)} < 1$. Hence, for a fixed η the probability above can be bounded by $((\eta + t_j) - (t_i - \eta))^+$; and

$$\Pr(\xi_j \leq t_j \text{ and } \xi_i \geq t_i) = \mathbb{E}[\Pr(t_i - \eta \leq \xi \leq \eta + t_j \mid \eta)] \leq \mathbb{E}[(2\eta - (t_i - t_j))^+].$$

By Lemma 2.4 and Corollary 2.3,

$$\Pr(\xi_j \leq t_j \text{ and } \xi_i \geq t_i) \leq C\Delta\tilde{\Phi}\left(\frac{t_i - t_j}{2\Delta}\right) \leq C\Delta \min(1, \Delta/|\delta|).$$

2. We have

$$\begin{aligned} \Pr(\xi_j \leq t_j \text{ and } \xi_i \geq t_i) &\leq \Pr(\xi_j \leq t_j \text{ and } \xi_i \geq t_j) + \Pr(t_i \leq \xi_i \leq t_j) \\ &\leq C(\Delta + 2\delta). \end{aligned}$$

For estimating the probability $\Pr(\xi_j \leq t_j \text{ and } \xi_i \geq t_j)$ we used part 1 with $t_i = t_j$. \square

THEOREM 2.6. *The approximation algorithm finds an assignment satisfying a $1 - O(\sqrt{\varepsilon})$ fraction of all constraints, where $\varepsilon = OPT_{SDP}/m$.*

PROOF. We shall estimate the probability of satisfying a clause $x_i \rightarrow x_j$. Set $\Delta_{ij} = \|v_j - v_i\|/2$ (so that $\text{cov}(\xi_i, \xi_j) = \langle v_i, v_j \rangle = 1 - 2\Delta_{ij}^2$) and $\delta_{ij} = (t_j - t_i)/2 \equiv -\langle v_j - v_i, v_0 \rangle / (2\sqrt{\varepsilon})$. The contribution of the term to the SDP is equal to $c_{ij} = (\Delta_{ij}^2 + \delta_{ij}\sqrt{\varepsilon})/2$.

Consider the following cases (we use Lemma 2.5 in all of them):

(1) If $\delta_{ij} \geq 0$, then the probability that the clause is not satisfied (*i.e.* $\xi_i \geq t_i$ and $x_j \leq t_j$) is at most

$$C(\Delta_{ij} + 2\delta_{ij}) \leq C(\sqrt{2c_{ij}} + 4c_{ij}/\sqrt{\varepsilon}).$$

(2) If $\delta_{ij} < 0$ and $\Delta_{ij}^2 \leq 4c_{ij}$, then the probability that the clause is not satisfied is at most

$$C\Delta_{ij} \leq 2C\sqrt{c_{ij}}.$$

(3) If $\delta_{ij} < 0$ and $\Delta_{ij}^2 > 4c_{ij}$, then the probability that the clause is not satisfied is at most

$$\frac{C\Delta_{ij}^2}{|\delta_{ij}|} = \frac{C\Delta_{ij}^2}{(\Delta_{ij}^2 - 2c_{ij})/\sqrt{\varepsilon}} \leq \frac{C\sqrt{\varepsilon}\Delta_{ij}^2}{\Delta_{ij}^2 - \Delta_{ij}^2/2} = 2C\sqrt{\varepsilon}.$$

Combining these cases we get that the probability that the clause is not satisfied is at most

$$4C(\sqrt{c_{ij}} + c_{ij}/\sqrt{\varepsilon} + \sqrt{\varepsilon}).$$

The expected fraction of unsatisfied clauses is equal to the average of such probabilities over all clauses. Recall, that ε is equal, by definition, to the average value of c_{ij} . Therefore, the expected number of unsatisfied constraints is $O(\sqrt{\varepsilon} + \varepsilon/\sqrt{\varepsilon} + \sqrt{\varepsilon})$ (here we used Jensen's inequality for the function $\sqrt{\cdot}$). \square

THEOREM 2.7. *The approximation algorithm for MAX 2CSP finds an assignment satisfying a $1 - O(\sqrt{\varepsilon})$ fraction of all constraints, if the optimal solution satisfies $1 - \varepsilon$ fraction of all constraints.*

This is an immediate corollary of the Theorem 2.6.

3. APPROXIMATION ALGORITHM FOR MAX K-CSP

3.1 Reduction to MAX k -AllEqual

We use Hast's reduction of the MAX k -CSP problem to the MAX k -AllEqual problem.

Definition 3.1 MAX k -AllEqual Problem. Given a set S of clauses of the form $l_1 \equiv l_2 \equiv \dots \equiv l_k$, where each literal l_i is either a boolean variable x_j or its negation \bar{x}_j . The goal is to find an assignment to the variables x_i so as to maximize the number of satisfied clauses.

The reduction works as follows. First, we write each constraint $f(x_{i_1}, x_{i_2}, \dots, x_{i_k})$ as a CNF formula: the CNF formula has a clause for every satisfying assignment of the constraint. Then we consider each clause in the CNF formula as a separate constraint; we get an instance of the MAX k -CSP problem, where each clause is a conjunction. The new problem is equivalent to the original problem: each assignment satisfies exactly the same number of clauses in the new problem as in the original problem (note that unlike the MAX 2CSP problem, the objective function is the fraction of satisfied clauses, not the fraction of unsatisfied clauses). Finally, we replace each conjunction $l_1 \wedge l_2 \wedge \dots \wedge l_k$ with the constraint $l_1 \equiv l_2 \equiv \dots \equiv l_k$. Clearly, the value of this instance of MAX k -AllEqual is at least the value of the original problem. Moreover, it is at most two times greater than the value of the original problem: if an assignment $\{x_i\}$ satisfies a constraint in the new problem, then either the assignment $\{x_i\}$ or the assignment $\{\bar{x}_i\}$ satisfies the corresponding constraint in the original problem. Therefore, a ρ approximation guarantee for MAX k -AllEqual translates to a $\rho/2$ approximation guarantee for the MAX k -CSP.

Note that this reduction may increase the number of constraints by a factor of $O(2^k)$. However, our approximation algorithm gives a nontrivial approximation only when $k/2^k \geq 1/m$ where m is the number of constraints, that is, when $2^k \leq O(m \log m)$ is polynomial in m .

Below we consider only the MAX k -AllEqual problem.

3.2 SDP Relaxation

As before, we denote \bar{x}_i by x_{-i} . We think of each clause C as a set of indices: the clause C defines the constraint “(for all $i \in C$, x_i is true) or (for all $i \in C$, x_i is false)”. Without loss of generality we assume that there are no unsatisfiable clauses in S , *i.e.* there are no clauses that have both literals x_i and \bar{x}_i .

We consider the following SDP relaxation of the MAX k -AllEqual problem:

$$\text{maximize } \frac{1}{k^2} \sum_{C \in S} \left\| \sum_{i \in C} v_i \right\|^2$$

subject to

$$\begin{aligned} \|v_i\|^2 &= 1 && \text{for all } i \in \{\pm 1, \dots, \pm n\} \\ v_i &= -v_{-i} && \text{for all } i \in \{\pm 1, \dots, \pm n\} \end{aligned}$$

This is indeed a relaxation of the problem: in the intended solution $v_i = v_0$ if x_i is true, and $v_i = -v_0$ if x_i is false (where v_0 is a fixed unit vector). Then each satisfied clause contributes 1 to the SDP value. Hence the value of the SDP is greater than or equal to the value of the MAX k -AllEqual problem. We use the following theorem of Rietz [1974] and Nesterov [1997].

THEOREM 3.2 RIETZ [1974], NESTEROV [1997]. *There exists an efficient algorithm that given a positive semidefinite matrix $A = (a_{ij})$, and a set of unit vectors v_i , assigns ± 1 to variables z_i , s.t.*

$$\sum_{i,j} a_{ij} z_i z_j \geq \frac{2}{\pi} \sum_{i,j} a_{ij} \langle v_i, v_j \rangle. \quad (1)$$

Remark 3.3. Rietz proved that for every positive semidefinite matrix A and unit vectors v_i there exist $z_i \in \{\pm 1\}$ s.t. inequality (1) holds. Nesterov presented a polynomial time algorithm that finds such values of z_i .

Figure 2: Approximation Algorithm for the MAX k -AllEqual Problem

- (1) Solve the semidefinite relaxation for MAX k -AllEqual. Get vectors v_i .
- (2) Apply Theorem 3.2 to vectors v_i as described above. Get values z_i .
- (3) Let $\delta = \sqrt{\frac{2}{k}}$.
- (4) For each $i \geq 1$ assign (independently)

$$x_i = \begin{cases} \text{true}, & \text{with probability } \frac{1+\delta z_i}{2}; \\ \text{false}, & \text{with probability } \frac{1-\delta z_i}{2}. \end{cases}$$

Observe that the quadratic form

$$\frac{1}{k^2} \sum_{C \in \mathcal{S}} \left(\sum_{i \in C} z_i \right)^2$$

is positive semidefinite (since it is positive for arbitrary values z_1, \dots, z_n). Therefore we can use the algorithm from Theorem 3.2. Given vectors v_i as in the SDP relaxation, it yields numbers z_i s.t.

$$\begin{aligned} \frac{1}{k^2} \sum_{C \in \mathcal{S}} \left(\sum_{i \in C} z_i \right)^2 &\geq \frac{2}{\pi} \frac{1}{k^2} \sum_{C \in \mathcal{S}} \left\| \sum_{i \in C} v_i \right\|^2 \\ z_i &\in \{\pm 1\} \\ z_i &= -z_{-i} \end{aligned}$$

(Formally, v_{-i} is an abbreviation for $-v_i$; z_{-i} is a shortcut for $-z_i$).

In what follows, we assume that $k \geq 3$ — for $k = 2$ we can use the MAX CUT algorithm by Goemans and Williamson [1995] to get a better approximation⁴.

The approximation algorithm is shown in Figure 2.

3.3 Analysis

THEOREM 3.4. *The approximation algorithm finds an assignment satisfying at least $ck/2^k \cdot OPT$ clauses (where $c > 0.88$ is an absolute constant), given that OPT clauses are satisfied in the optimal solution.*

PROOF. For a constraint C , denote $Z_C = \frac{1}{k} \sum_{i \in C} z_i$. Then Theorem 3.2 guarantees that

$$\sum_{C \in \mathcal{S}} Z_C^2 = \frac{1}{k^2} \sum_{C \in \mathcal{S}} \left(\sum_{i \in C} z_i \right)^2 \geq \frac{2}{\pi} \frac{1}{k^2} \sum_{C \in \mathcal{S}} \left\| \sum_{i \in C} v_i \right\|^2 = \frac{2}{\pi} OPT_{SDP} \geq \frac{2}{\pi} OPT,$$

where OPT_{SDP} is the SDP value.

Note that the number of $z_i \in C$ equal to 1 is $\frac{1+Z_C}{2}k$, and the number of $z_i \in C$ equal to -1 is $\frac{1-Z_C}{2}k$.

⁴Our algorithm works for $k = 2$ with a slight modification: δ should be less than 1.

The probability that a constraint C is satisfied equals

$$\begin{aligned}
\Pr(C \text{ is satisfied}) &= \Pr(\forall i \in C x_i = 1) + \Pr(\forall i \in C x_i = -1) \\
&= \prod_{i \in C} \frac{1 + \delta z_i}{2} + \prod_{i \in C} \frac{1 - \delta z_i}{2} \\
&= \frac{1}{2^k} \left((1 + \delta)^{(1+Z_C)k/2} \cdot (1 - \delta)^{(1-Z_C)k/2} + (1 - \delta)^{(1+Z_C)k/2} \cdot (1 + \delta)^{(1-Z_C)k/2} \right) \\
&= \frac{(1 - \delta^2)^{k/2}}{2^k} \left(\left(\frac{1 + \delta}{1 - \delta} \right)^{Z_C k/2} + \left(\frac{1 - \delta}{1 + \delta} \right)^{Z_C k/2} \right) \\
&= \frac{1}{2^k} (1 - \delta^2)^{k/2} \cdot 2 \cosh \left(\frac{1}{2} \ln \frac{1 + \delta}{1 - \delta} \cdot Z_C k \right).
\end{aligned}$$

Here, $\cosh t \equiv (e^t + e^{-t})/2$. Let α be the minimum of the function $\cosh t/t^2$. Numerical computations show that $\alpha > 0.93945$. We have,

$$\cosh \left(\frac{1}{2} \ln \frac{1 + \delta}{1 - \delta} \cdot Z_C k \right) \geq \alpha \left(\frac{1}{2} \ln \frac{1 + \delta}{1 - \delta} \cdot Z_C k \right)^2 \geq \alpha (\delta \cdot Z_C k)^2 = 2\alpha Z_C^2 k.$$

Recall that $\delta = \sqrt{2/k}$ and $k \geq 3$. Hence

$$(1 - \delta^2)^{k/2} = \left(1 - \frac{2}{k} \right)^{k/2} \geq \left(1 - \frac{2}{k} \right) \cdot \frac{1}{e}.$$

Combining these bounds we get,

$$\Pr(C \text{ is satisfied}) \geq \frac{4\alpha}{e} \cdot \frac{k}{2^k} \cdot \left(1 - \frac{2}{k} \right) \cdot Z_C^2.$$

However, a more careful analysis shows that the factor $1 - 2/k$ is not necessary, and the following bound holds (we give a proof in the Appendix):

$$2\alpha(1 - \delta^2)^{k/2} \left(\frac{1}{2} \ln \frac{1 + \delta}{1 - \delta} \cdot Z_C k \right)^2 \geq \frac{4\alpha}{e} Z_C^2 k. \quad (2)$$

Therefore,

$$\Pr(C \text{ is satisfied}) \geq \frac{4\alpha}{e} \cdot \frac{k}{2^k} \cdot Z_C^2.$$

So the expected number of satisfied clauses is

$$\sum_{C \in S} \Pr(C \text{ is satisfied}) \geq \frac{4\alpha}{e} \frac{k}{2^k} \sum_{C \in S} Z_C^2 \geq \frac{4\alpha}{e} \frac{k}{2^k} \cdot \frac{2}{\pi} OPT.$$

We conclude that the algorithm finds an

$$\frac{8\alpha}{\pi e} \frac{k}{2^k} > 0.88 \frac{k}{2^k}$$

approximation with high probability. \square

3.4 k -CSPs with non-binary domain

We now describe our algorithm for the k -CSP problem with a non-boolean domain. If the domain size d is a power of 2 i.e. $d = 2^t$, then we just encode every value in the domain by t bits and replace all original predicates with new predicates depending on $t \times k$ boolean variables. The new problem is exactly equivalent to the original one. Our algorithm for the boolean k -CSP problem gives a

$$0.44 \frac{t \times k}{2^{t \times k}} = 0.44 \frac{k \log_2 d}{2^k}$$

approximation.

We now consider the case when the domain size is not a power of 2. Let D be the domain of the problem; and let OPT be the optimal value. Fix $d' = 2^{\lfloor \log_2 d \rfloor}$, and pick random subsets $D_i \subset D$ of size d' for every $i = 1, \dots, n$. Consider a new problem in which every variable x_i may take values only in the set D_i . Note that this restriction may make some constraints unsatisfiable. The domain size of the new problem d' is a power of two, thus the previous algorithm finds a solution of cost $0.44(k \log d' / d'^k) \times OPT'$, where OPT' is the cost of the optimal solution of the new problem. We prove below that the expected value of OPT' is at least $(d'/d)^k OPT$. Thus, the expected cost of the solution returned by the algorithm is at least

$$0.44 \frac{k \log d'}{d'^k} \times \frac{d'^k}{d^k} OPT = 0.44 \frac{k \lfloor \log d \rfloor}{d^k} OPT.$$

Let $\hat{x}_1, \dots, \hat{x}_n$ be an optimal solution to the original problem. Consider the following solution to the new problem

$$\tilde{x}_i = \begin{cases} \hat{x}_i, & \text{if } \hat{x}_i \in D_i; \\ \text{arbitrary value in } D_i, & \text{otherwise.} \end{cases}$$

Clearly, $\hat{x}_i = \tilde{x}_i$ with probability d'/d . Hence, every predicate satisfied by $\{\hat{x}\}_i$ is satisfied by $\{\tilde{x}\}_i$ with probability at least $(d'/d)^k$. Thus the new solution satisfies $(d'/d)^k OPT$ constraints on average.

We note that this reduction can be easily derandomized using the k -wise δ -dependent distribution of Azar, Motwani, and Naor [1998]. Assume $D = \mathbb{Z}_d$. Instead of sampling sets D_i uniformly at random, we let $D_i = \{r_i, r_i + 1, \dots, r_i + (d' - 1)\}$, where r_i -s are drawn from a k -wise $(\varepsilon(d'/d)^k)$ -dependent distribution \mathcal{D} on \mathbb{Z}_d^n with polynomial size sample space. For every predicate p satisfied by $\{\hat{x}\}_i$, we have

$$\Pr_{r \in \mathcal{D}} (\{\tilde{x}\}_i \text{ satisfies } p) \geq \Pr_{r \in \mathcal{U}} (\{\tilde{x}\}_i \text{ satisfies } p) - \varepsilon(d'/d)^k \geq (1 - \varepsilon)(d'/d)^k.$$

Therefore, if we run the algorithm on all samples $\{r_i\}_i$ in the sample space, we will find a $0.44(1 - \varepsilon)k \lfloor \log_2 d \rfloor / d^k$ approximation.

Acknowledgment

We thank Noga Alon for pointing out that a non-algorithmic version of the result of Nesterov [1997] was proved by Rietz [1974].

REFERENCES

- AGARWAL, A., CHARIKAR, M., MAKARYCHEV, K., AND MAKARYCHEV, Y. 2005. $O(\sqrt{\log n})$ approximation algorithms for MIN Uncut, MIN 2CNF deletion, and directed cut problems. In *Proceedings of the 37th annual ACM symposium on Theory of computing*. ACM, New York, NY, USA, 573–581.
- AUSTRIN, P. 2007a. Balanced MAX 2-SAT might not be the hardest. In *Proceedings of the 39th annual ACM symposium on Theory of computing*. ACM, New York, NY, USA, 189–197.
- AUSTRIN, P. 2007b. Towards sharp inapproximability for any 2-CSP. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society, Washington, DC, USA, 307–317.
- AUSTRIN, P. AND MOSSEL, E. 2008. Approximation resistant predicates from pairwise independence. In *Proceedings of the 2008 IEEE 23rd Annual Conference on Computational Complexity*. IEEE Computer Society, Washington, DC, USA, 249–258.
- AZAR, Y., MOTWANI, R., AND NAOR, J. 1998. Approximating probability distributions using small sample spaces. *Combinatorica* 18, 2, 151–171.
- CHARIKAR, M. AND WIRTH, A. 2004. Maximizing quadratic programs: Extending Grothendieck’s inequality. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society, Washington, DC, USA, 54–60.
- FEIGE, U. AND GOEMANS, M. X. 1995. Approximating the value of two prover proof systems, with applications to MAX-2SAT and MAX DICUT. In *Proceedings of the 3rd Israel Symposium on Theory of Computing and Systems*. 182–189.
- GOEMANS, M. X. AND WILLIAMSON, D. P. 1995. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM* 42, 6, 1115–1145.
- GURUSWAMI, V. AND RAGHAVENDRA, P. 2008. Constraint satisfaction over a non-boolean domain: Approximation algorithms and unique-games hardness. In *Proceedings of APPROX-RANDOM*. 77–90.
- HAST, G. 2005. Approximating MAX kCSP — outperforming a random assignment with almost a linear factor. In *Proceedings of ICALP*. 956–968.
- HÅSTAD, J. 2001. Some optimal inapproximability results. *J. ACM* 48, 4, 798–859.

- KHOT, S. 2002. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th annual ACM symposium on Theory of computing*. ACM, New York, NY, USA, 767–775.
- KHOT, S., KINDLER, G., MOSSEL, E., AND O'DONNELL, R. 2007. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM J. Comput.* *37*, 1, 319–357.
- LEWIN, M., LIVNAT, D., AND ZWICK, U. 2002. Improved rounding techniques for the MAX 2-SAT and MAX DI-CUT problems. In *Proceedings of the 9th International IPCO Conference on Integer Programming and Combinatorial Optimization*. Springer-Verlag, London, UK, 67–82.
- MATUURA, S. AND MATSUI, T. 2003. New approximation algorithms for MAX 2SAT and MAX DICUT. *Journal of the Operations Research Society of Japan* *46*, 178–188.
- NESTEROV, Y. 1997. Quality of semidefinite relaxation for nonconvex quadratic optimization. CORE Discussion Paper 9719. March.
- RAGHAVENDRA, P. 2008. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the 40th annual ACM symposium on Theory of computing*. ACM, New York, NY, USA, 245–254.
- RIETZ, R. E. 1974. A proof of the Grothendieck inequality. *Israel J. Math* *19*, 3, 271–276.
- SAMORODNITSKY, A. AND TREVISAN, L. 2006. Gowers uniformity, influence of variables, and PCPs. In *Proceedings of the 38th annual ACM symposium on Theory of computing*. ACM, New York, NY, USA, 11–20.
- TREVISAN, L. 1998. Parallel approximation algorithms by positive linear programming. *Algorithmica* *21*, 1, 72–88.
- ZWICK, U. 1998. Finding almost-satisfying assignments. In *Proceedings of the 30th annual ACM symposium on Theory of computing*. ACM, New York, NY, USA, 551–560.
- ZWICK, U. 2000. Analyzing the MAX 2-SAT and MAX DI-CUT approximation algorithms of Feige and Goemans. Available at www.cs.tau.ac.il/~zwick/my-online-papers.html.

A. PROOF OF INEQUALITY (2)

In this section, we will prove inequality (2):

$$2\alpha(1 - \delta^2)^{k/2} \left(\frac{1}{2} \ln \frac{1 + \delta}{1 - \delta} \cdot Z_C k \right)^2 \geq \frac{4\alpha}{e} Z_C^2 k. \quad (2)$$

Write this inequality as follows.

$$(1 - \delta^2)^{k/2} \left(\sqrt{\frac{k}{2}} \cdot \frac{\ln(1 + \delta) - \ln(1 - \delta)}{2} \right)^2 \geq e^{-1}.$$

Note that this inequality holds for $3 \leq k \leq 7$, which can be verified by direct computation. So assume that $k \geq 8$. Denote $t = 2/k$; and replace k with $2/t$ and δ with \sqrt{t} . We get

$$(1 - t)^{1/t} \left(\frac{1}{\sqrt{t}} \cdot \frac{\ln(1 + \sqrt{t}) - \ln(1 - \sqrt{t})}{2} \right)^2 \geq e^{-1}.$$

Take the logarithm of both sides:

$$\frac{\ln(1 - t)}{t} + 2 \ln \left(\frac{1}{\sqrt{t}} \cdot \frac{\ln(1 + \sqrt{t}) - \ln(1 - \sqrt{t})}{2} \right) \geq -1.$$

Observe that

$$\frac{1}{\sqrt{t}} \cdot \frac{\ln(1 + \sqrt{t}) - \ln(1 - \sqrt{t})}{2} = 1 + \frac{t}{3} + \frac{t^2}{5} + \frac{t^3}{7} + \dots \geq 1 + \frac{t}{3};$$

and

$$\begin{aligned} \frac{\ln(1 - t)}{t} &= -1 - \frac{t}{2} - \frac{t^2}{3} - \dots \geq -1 - \frac{t}{2} - \frac{t^2}{3} \times \sum_{i=0}^{\infty} t^i \\ &\geq -1 - \frac{t}{2} - \frac{4t^2}{9}. \end{aligned}$$

In the last inequality we used our assumption that $t \equiv 2/k \leq 1/4$. Now,

$$\begin{aligned}
\frac{\ln(1-t)}{t} + 2 \ln \left(\frac{1}{\sqrt{t}} \cdot \frac{\ln(1+\sqrt{t}) - \ln(1-\sqrt{t})}{2} \right) &\geq \left(-1 - \frac{t}{2} - \frac{4t^2}{9} \right) + 2 \ln \left(1 + \frac{t}{3} \right) \\
&\geq \left(-1 - \frac{t}{2} - \frac{4t^2}{9} \right) + 2 \left(\frac{t}{3} - \frac{t^2}{18} \right) \\
&\geq -1 + \frac{t}{6} - \frac{5t^2}{9} \geq -1.
\end{aligned}$$

Here $(t/6 - 5t^2/9)$ is positive, since $t \in (0, 1/4]$. This concludes the proof.