Near-Optimal Algorithms for Maximum Constraint Satisfaction Problems

Moses Charikar* Konstantin Makarychev*† Yury Makarychev*‡

Princeton University

Abstract

In this paper we present 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.

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 2-CSPs 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 [5], gave an semidefinite programming based algorithm for MAX CUT, a special case of MAX 2CSP. If the optimal 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

*http://www.cs.princeton.edu/~moses/ Supported by NSF ITR grant CCR-0205594, NSF CAREER award CCR-0237113, MSPA-MCS award 0528414, and an Alfred P. Sloan Fellowship.

†http://www.cs.princeton.edu/~kmakaryc/ Supported by a Gordon Wu fellowship.

‡http://www.cs.princeton.edu/~ymakaryc/ Supported by a Gordon Wu fellowship.

	MAX CUT		MAX 2CSP	
Approximation ratio	0.878	[5]	0.874	[10]
Almost satisfiable instances				
• $\varepsilon > 1/\log n$	$1 - O(\sqrt{\varepsilon})$	[5]	$1 - O(\varepsilon^{1/3})$	[17]
• $\varepsilon < 1/\log n$	$1 - O(\sqrt{\varepsilon \log n})$	[1]	$1 - O(\sqrt{\varepsilon \log n})$	[1]

Table 1: 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 $\varepsilon > 1/\log n$.

$OPT = 1 - \varepsilon$), the algorithm finds an assignment of variables that satisfies a $(1 - O(\sqrt{\varepsilon}))$ fraction of all constraints.

In the same paper [5], Goemans and Williamson also gave a 0.796 approximation algorithm for MAX DICUT and a 0.758 approximation algorithm for MAX 2SAT. These results were improved in several follow-up papers: Feige and Goemans [4], Zwick [16], Matuura and Matsui [11], and Lewin, Livnat and Zwick [10]. The approximation ratios obtained by Lewin, Livnat and Zwick [10] are 0.874 for MAX DICUT and 0.94 for MAX 2SAT. There is a simple approximation preserving reduction from MAX 2CSP to MAX DICUT. Therefore, their algorithm for MAX DICUT can 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 [5] for MAX CUT.

Khot, Kindler, Mossel, and O'Donnell [9] recently showed that both results of Goemans and Williamson [5] for MAX CUT are optimal and the results of Lewin, Livnat and Zwick [10] are almost optimal[1] assuming Khot's Unique Games Conjecture [8]. The MAX 2SAT hardness result was further improved by Austrin [2], who showed that the MAX 2SAT algorithm of Lewin, Livnat and Zwick [10] 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 [17] developed an approximation algorithm that satisfies a $1 - O(\varepsilon^{1/3})$ fraction of all constraints[2]. However the best known hardness result [9] (assuming the Unique Games Conjecture) is that it is hard to satisfy $1 - \Omega(\sqrt{\varepsilon})$ 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 [5] for MAX CUT. Table 1 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 [7], if $(1 - \varepsilon)$ fraction of all constraints

[1] Khot, Kindler, Mossel, and O'Donnell [9] proved 0.943 hardness result for MAX 2SAT and 0.878 hardness result for MAX 2CSP.

[2] He 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).

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 [15] 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 [6] proposed an algorithm with an asymptotically better approximation guarantee $\Omega\left(k/(2^k \log k)\right)$. Also, Samorodnitsky and Trevisan [14] proved that it is 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 [8]. We close the gap between the upper and lower bounds for k-CSP by giving an algorithm with approximation ratio $\Omega\left(k/2^k\right)$. By the results of [14], 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 [6]: 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 [13] and Nesterov [12]. 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 [3]. 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[3]: it gives a $\Omega\left(k \log d/d^k\right)$ approximation for instances with domain size d.

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.

2 Approximation Algorithm for MAX 2CSP

2.1 SDP Relaxation

In this section we describe the vector program (SDP) for MAX 2CSP/MAX 2SAT. For convenience we replace each negation $\bar{x}_i$ with a new variable x_{-i} that is equal by the definition to $\bar{x}_i$. First, we transform our problem 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 $x_{-i} \vee x_{-j}$;

[3]To apply the result to an instance with a larger domain, we just encode each domain value with $\log d$ bits.

- 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 now rewrite all clauses in the form $x_i \to x_j$, 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 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; v_i and v_{-i} are opposite; and some ℓ_2^2-triangle inequalities.

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

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

to the objective function. In the intended solution this expression equals to 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). Note that each term in the SDP is positive due to the triangle inequality constraints.

We get an SDP relaxation for MAX 2SAT:

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

subject to

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

In a slightly different form, this semidefinite program was introduced by Feige and Goemans [4]. Later, Zwick [17] 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.

Figure 1: Approximation Algorithm for MAX 2CSP

1. Solve the SDP for MAX 2SAT. Denote by SDP the objective value of the solution and by ε the fraction of the constraints "unsatisfied" by the vector solution, that is,
$$\varepsilon = \frac{SDP}{\#\text{constraints}}.$$
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:
   $$t_i = -\langle v_i, v_0 \rangle \big/ \sqrt{\varepsilon}\ .$$
   (c) If $\xi_i \geq t_i$, set $x_i = 1$, otherwise set $x_i = 0$.

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 2.1. *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 $t \to \infty$ 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 2.2. *There exists a constant C such that for every positive t, the following inequality holds $\tilde{\Phi}(t) \le C\frac{1}{\sqrt{2\pi}}e^{-\frac{t^2}{2}}$.*

A clause $x_i \to x_j$ is not satisfied by the algorithm if $\xi_i \ge t_i$ and $\xi_j \le t_j$ (*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 2.3. *Let ξ_i and ξ_j be two standard normal random variables with covariance $1-2\Delta^2$ (where $\Delta \ge 0$). For all real numbers t_i, t_j and $\delta = (t_j - t_i)/2$ we have (for some absolute constant C)*

1. *If $t_j \le t_i$,*
$$\Pr(\xi_i \ge t_i \text{ and } \xi_j \le t_j) \le C\min(\Delta^2/|\delta|, \Delta).$$
2. *If $t_j \ge t_i$,*
$$\Pr(\xi_i \ge t_i \text{ and } \xi_j \le t_j) \le C(\Delta + 2\delta).$$

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

Let $\xi = (\xi_i + \xi_j)/2$ and $\eta = (\xi_i - \xi_j)/2$. Notice that $\mathrm{Var}[\xi] = 1 - \Delta^2$, $\mathrm{Var}[\eta] = \Delta^2$; and random variables ξ and η are independent. We estimate the desired probability as follows:

$$\begin{aligned}\Pr(\xi_j \le t_j \text{ and } \xi_i \ge t_i) &= \Pr\left(\eta \ge \left|\xi - \frac{t_i+t_j}{2}\right| + \frac{t_i - t_j}{2}\right)\\ &= \int_{-\infty}^{+\infty} \Pr\left(\eta \ge \left|\xi - \frac{t_i+t_j}{2}\right| + \frac{t_i - t_j}{2} \ \mid\ \xi = t\right) dF_\xi(t).\end{aligned}$$

Note that the density of the normal distribution with variance $1-\Delta^2$ is always less than $1/\sqrt{2\pi(1-\Delta^2)} < 1$, thus we can replace $dF_\xi(t)$ with dt.

$$\begin{aligned}\Pr(\xi_j \le t_j \text{ and } \xi_i \ge t_i) &\le \int_{-\infty}^{+\infty} \tilde{\Phi}\left(\frac{\left|t - \frac{t_i+t_j}{2}\right| + \frac{t_i-t_j}{2}}{\Delta}\right) dt\\ &= \int_{-\infty}^{+\infty} \tilde{\Phi}\left(\frac{|t| + |\delta|}{\Delta}\right) dt\\ &= \Delta \int_{-\infty}^{+\infty} \tilde{\Phi}\left(|s| + |\delta|/\Delta\right) ds \quad \text{(by Corollary 2.2)}\\ &\le C'\Delta \cdot \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} e^{-\frac{(|s|+|\delta|/\Delta)^2}{2}} ds\\ &= 2C'\Delta \cdot \tilde{\Phi}(|\delta|/\Delta) \quad \text{(by Lemma 2.1)}\\ &\le 2C' \min(\Delta^2/|\delta|, \Delta).\end{aligned}$$

2. We have

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

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

Theorem 2.4. *The approximation algorithm finds an assignment satisfying a $1 - O(\sqrt{\varepsilon})$ fraction of all constraints, if a $1-\varepsilon$ fraction of all constraints is satisfied in the optimal solution.*

Proof. We shall estimate the probability of satisfying a clause $x_i \to x_j$. Set $\Delta_{ij} = \|v_j - v_i\|/2$ (so that $\mathrm{cov}(\xi_i, \xi_j) = \langle v_i, v_j\rangle = 1 - 2\Delta_{ij}^2$) 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 2.3 in all of them):

1. If $\delta_{ij} \ge 0$, then the probability that the clause is not satisfied (*i.e.* $\xi_i \ge t_i$ and $x_j \le t_j$) is at most
$$C(\Delta_{ij} + 2\delta_{ij}) \le C(\sqrt{2c_{ij}} + 4c_{ij}/\sqrt{\varepsilon}).$$

2. If $\delta_{ij} < 0$ and $\Delta_{ij}^2 \le 4c_{ij}$, then the probability that the clause is not satisfied is at most
$$C\Delta_{ij} \le 2C\sqrt{c_{ij}}.$$

3. If $\delta_{ij} < 0$ and $\Delta_{ij}^2 > 4c_{ij}$, then the probability that the constraint is not satisfied is at most
$$\frac{C\Delta_{ij}^2}{|\delta_{ij}|} = \frac{C\Delta_{ij}^2}{(\Delta_{ij}^2 - 2c_{ij})/\sqrt{\varepsilon}} \le \frac{C\sqrt{\varepsilon}\,\Delta_{ij}^2}{\Delta_{ij}^2 - \Delta_{ij}^2/2} = 2C\sqrt{\varepsilon}.$$

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 the 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 $\sqrt{\cdot}$). □

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 $l_1 \equiv l_2 \equiv \cdots \equiv l_k$, 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. 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. Finally, we replace each conjunction $l_1 \wedge l_2 \wedge \dots \wedge l_k$ with the constraint $l_1 \equiv l_2 \equiv \cdots \equiv l_k$. 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 then 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

For brevity, 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 Max k-AllEqual problem:

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

subject to

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

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 [13] and Nesterov [12].

Theorem 3.2 (Rietz [13], Nesterov [12]). *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.*

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

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.*

Observe that the quadratic form

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

is positive semidefinite. 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.

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

(Formally, v_{-i} is a shortcut 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 [5] to get a better approximation[4].

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. Denote $Z_C = \frac{1}{k} \sum_{i \in C} z_i$. Then Theorem 3.2 guarantees that

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

where SDP is the SDP value.

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

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)

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

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

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

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):

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

Therefore,

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

So the expected number of satisfied clauses is

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

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. □

Acknowledgment

We thank Noga Alon for pointing out that a non-algorithmic version of the result of Nesterov [12] was proved by Rietz [13].

References

[1] A. Agarwal, M. Charikar, K. Makarychev, and Y. Makarychev. $O(\sqrt{\log n})$ *approximation algorithms for Min UnCut, Min 2CNF Deletion, and directed cut problems.* In Proceedings of the 37th Annual ACM Symposium on Theory of Computing, pp. 573–581, 2005.

[2] P. Austrin *Balanced Max 2-Sat might not be the hardest.* ECCC Report TR06-088, 2006.

[3] M. Charikar and A. Wirth, *Maximizing quadratic programs: extending Grothendieck's Inequality.* In Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, pp. 54–60, 2004.

[4] U. Feige and M. Goemans. *Approximating the value of two prover proof systems, with applications to MAX 2SAT and MAX DICUT.* In Proceedings of the 3rd IEEE Israel Symposium on the Theory of Computing and Systems, pp. 182–189, 1995.

[5] M. Goemans and D. Williamson. *Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming.* Journal of the ACM, vol. 42, no. 6, pp. 1115–1145, Nov. 1995.

[6] G. Hast. *Approximating Max kCSP — Outperforming a Random Assignment with Almost a Linear Factor.* In Proceedings of the 32nd International Colloquium on Automata, Languages and Programming, pp. 956–968, 2005.

[7] J. Håstad. *Some optimal inapproximability results.* Journal of the ACM, vol. 48, no. 4, pp. 798-859, 2001.

[8] S. Khot. *On the power of unique 2-prover 1-round games.* In Proceedings of the 34th ACM Symposium on Theory of Computing, pp. 767–775, 2002.

[9] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. *Optimal inapproximability results for MAX-CUT and other two-variable CSPs?* In Proceedings of the 45th IEEE Symposium on Foundations of Computer Science, pp. 146–154, 2004.

[10] M. Lewin, D. Livnat, and U. Zwick. *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, pp. 67–82, 2002.

[11] S. Matuura and T. Matsui. *New approximation algorithms for MAX 2SAT and MAX DICUT,* Journal of Operations Research Society of Japan, vol. 46, pp. 178–188, 2003.

[12] Y. Nesterov. *Quality of semidefinite relaxation for nonconvex quadratic optimization.* CORE Discussion Paper 9719, March 1997.

[13] E. Rietz. *A proof of the Grothendieck inequality,* Israel J. Math, vol. 19, pp. 271–276, 1974.

[14] A. Samorodnitsky and L. Trevisan. *Gowers Uniformity, Influence of Variables, and PCPs.* In Proceedings of the 38th ACM symposium on Theory of computing, pp. 11–20, 2006.

[15] L. Trevisan. *Parallel Approximation Algorithms by Positive Linear Programming.* Algorithmica, vol. 21, no. 1, pp. 72-88, 1998.

[16] U. Zwick. *Analyzing the MAX 2-SAT and MAX DI-CUT approximation algorithms of Feige and Goemans.* Manuscript. Available at `www.cs.tau.ac.il/~zwick/my-online-papers.html`.

[17] U. Zwick. *Finding almost satisfying assignments.* In Proceedings of the 30th Annual ACM Symposium on Theory of Computing, pp. 551–560, 1998.

A Proof of Inequality (2)

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

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

Let us first simplify this expression

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

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

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

and

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

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.