Improved Approximation Algorithms for MAX SAT*

Takao Asano† David P. Williamson‡

Abstract

MAX SAT (the maximum satisfiability problem) is stated as follows: given a set of clauses with weights, find a truth assignment that maximizes the sum of the weights of the satisfied clauses. In this paper, we consider approximation algorithms for MAX SAT proposed by Goemans and Williamson and present a sharpened analysis of their performance guarantees. We also show that these algorithms, combined with recent approximation algorithms for MAX 2SAT, MAX 3SAT, and MAX SAT due to Feige and Goemans, Karloff and Zwick, and Zwick, respectively, lead to an improved approximation algorithm for MAX SAT. By using the MAX 2SAT and 3SAT algorithms, we obtain a performance guarantee of .7846, and by using Zwick's algorithm, we obtain a performance guarantee of .8331, which improves upon the performance guarantee of .7977 based on Zwick's conjecture. The best previous result for MAX SAT without assuming Zwick's conjecture is a .770-approximation algorithm of Asano. Our best algorithm requires a new family of 3/4-approximation algorithms that generalize a previous algorithm of Goemans and Williamson.

1 Introduction

MAX SAT (the maximum satisfiability problem) is stated as follows: given a set of clauses with weights, find a truth assignment that maximizes the sum of the weights of the satisfied clauses. More precisely, an instance of MAX SAT is defined by $(\mathcal{C}, w)$, where $\mathcal{C}$ is a set of boolean clauses such that each clause $C \in \mathcal{C}$ is a disjunction of literals with a positive weight $w(C)$. We sometimes write $\mathcal{C}$ instead of $(\mathcal{C}, w)$ if the weight function w is clear from the context. Let $X = \{x_1, \ldots, x_n\}$ be the set of boolean variables in the clauses of $\mathcal{C}$. A *literal* is a variable $x \in X$ or its negation $\bar{x}$. For simplicity we assume $x_{n+i} = \bar{x}_i$ ($x_i = \bar{x}_{n+i}$). Thus, $\bar{X} = \{\bar{x} \mid x \in X\} = \{x_{n+1}, x_{n+2}, \ldots, x_{2n}\}$ and $X \cup \bar{X} = \{x_1, \ldots, x_{2n}\}$. We assume that no literals with the same variable appear more than once in a clause in $\mathcal{C}$. For each $x_i \in X$, let $x_i = 1$ ($x_i = 0$, resp.) if x_i is true (false, resp.). Then, $x_{n+i} = \bar{x}_i = 1 - x_i$ and a clause $C_j = x_{j_1} \vee x_{j_2} \vee \cdots \vee x_{j_{k_j}} \in \mathcal{C}$ can be considered to be a function on $\boldsymbol{x} = (x_1, \ldots, x_{2n})$ as follows:

$$C_j = C_j(\boldsymbol{x}) = 1 - \prod_{i=1}^{k_j} (1 - x_{j_i}).$$

*A preliminary version of this paper appeared in the *Proceedings of the 11th ACM-SIAM Symposium of Discrete Algorithms*, pp. 96–105, 2000.

†Address: Department of Information and System Engineering, Chuo University, Bunkyo-ku, Tokyo 112-8551, Japan. Email: asano@ise.chuo-u.ac.jp.

‡Address: IBM Almaden Research Center, 650 Harry Rd, K53/B1, San Jose, CA 95120. Web: www.almaden.ibm.com/cs/people/dpw. Email: dpw@almaden.ibm.com.

Thus, $C_j = C_j(\boldsymbol{x}) = 0$ or 1 for any *truth assignment* $\boldsymbol{x} \in \{0,1\}^{2n}$ with $x_i + x_{n+i} = 1$ ($i = 1, 2, ..., n$) and C_j is *satisfied* if $C_j(\boldsymbol{x}) = 1$. The *value* of a truth assignment $\boldsymbol{x}$ is defined to be

$$F_{\mathcal{C}}(\boldsymbol{x}) = \sum_{C_j \in \mathcal{C}} w(C_j) C_j(\boldsymbol{x}).$$

That is, the value of $\boldsymbol{x}$ is the sum of the weights of the clauses in $\mathcal{C}$ satisfied by $\boldsymbol{x}$. The goal of MAX SAT is to find an optimal truth assignment; that is, a truth assignment of maximum value. We will also consider MAX kSAT, a restricted version of the problem in which each clause has at most k literals.

MAX SAT is well known to be NP-hard and thus many researchers have proposed *approximation algorithms* for MAX SAT. Let A be an algorithm for MAX SAT and let $F_{\mathcal{C}}(\boldsymbol{x}^A(\mathcal{C}))$ be the value of a truth assignment $\boldsymbol{x}^A(\mathcal{C})$ produced by A for an instance $\mathcal{C}$. If $F_{\mathcal{C}}(\boldsymbol{x}^A(\mathcal{C}))$ is at least α times the value $F_{\mathcal{C}}(\hat{\boldsymbol{x}}(\mathcal{C}))$ of an optimal truth assignment $\hat{\boldsymbol{x}}(\mathcal{C})$ for any instance $\mathcal{C}$, then A is called an approximation algorithm with *performance guarantee* α. A polynomial-time approximation algorithm A with performance guarantee α is called an α-*approximation algorithm*.

Johnson [10] gave the first approximation algorithm for MAX SAT in 1974. It is a greedy algorithm whose performance guarantee was shown to be $1/2$.[1] In the early 90s, Yannakakis [15], and then Goemans and Williamson [6], proposed .75-approximation algorithms. Later, Goemans and Williamson proposed a .878-approximation algorithm for MAX 2SAT based on semidefinite programming [5]. They showed that their .878-approximation algorithm, combined with their earlier .75-approximation algorithm for MAX SAT, leads to a .7584-approximation algorithm for MAX SAT [7]. Asano, Ono, and Hirata proposed a semidefinite programming approach to MAX SAT [2] and obtained a .765-approximation algorithm by combining it with Yannakakis' .75-approximation algorithm as well as the algorithm of Goemans and Williamson. Asano [1] later presented a refinement of Yannakakis' algorithm based on network flows, and showed that it leads to a .770-approximation algorithm. Using semidefinite programming, Karloff and Zwick [11] gave a 7/8-approximation algorithm for MAX 3SAT, and Halperin and Zwick [9] gave an approximation algorithm for MAX 4SAT that numerical evidence shows has a performance guarantee of .8721. Finally, Zwick [16] recently made a conjecture about another approximation algorithm which, if true, leads to a .7977-approximation algorithm for MAX SAT; his conjecture is supported by numerical experiments. The conjecture states that the worst-case ratio of the value of a semidefinite programming relaxation of MAX SAT to the value of a certain algorithm for MAX SAT is achieved at a particular point. We summarize these results in Figure 1. Håstad [8] has shown that no approximation algorithm for MAX 3SAT (and thus MAX SAT) can achieve performance guarantee better than 7/8 unless $P = NP$; thus the Karloff and Zwick result is tight, and the Halperin and Zwick result is nearly so.

In this paper, we present a sharpened analysis of the .75-approximation algorithms for MAX SAT proposed by Goemans and Williamson. The improved analysis allows us to give better bounds on how these algorithms perform on clauses of length k, and this leads to a better overall approximation algorithm for MAX SAT in combination with other recent results. In particular, by combining our algorithm with the Feige-Goemans algorithm for MAX 2SAT and the Karloff-Zwick algorithm for MAX 3SAT, we obtain a performance guarantee of .7846 for MAX SAT. If we use Zwick's approximation algorithm for MAX SAT, and its performance guarantee is as conjectured, we obtain a performance guarantee of .8331.[2] This last algorithm requires a new

[1] In 1997, Chen, Friesen, and Zheng [3] improved the analysis of the performance guarantee to 2/3.

[2] One could also consider an intermediate algorithm that used the Feige-Goemans, Karloff-Zwick, and Halperin-Zwick algorithms in combination with our algorithm. However, we found that such an algorithm produced a performance guarantee that was better than our .7846-approximation algorithm only by a very tiny amount.

MAX SAT		MAX 2SAT		MAX 3SAT		MAX 4SAT	
.5				Johnson74 [10]			
.618				LS81 [13]			
.666				KK89 [12]			
.75				Yannakakis94 [15], GW94 [6]			
.7584	GW95 [7]	.878	GW95 [7]	.801	TSSW96 [14]	.8721†	HZ99 [9]
.765	AOH96 [2]	.931	FG95 [4]	.875	KZ97 [11]		
.770	Asano97 [1]						
.797 ‡	Zwick99 [16]						
.7846	this paper						
.8331‡	this paper						

Figure 1: Summary of performance guarantees for MAX SAT. † indicates the result is based on numerical evidence given in [9]. ‡ indicates the result is based on a conjecture in [16].

family of .75-approximation algorithms for MAX SAT which generalizes an algorithm of Goemans and Williamson.

To explain our result in more detail, we briefly review the .75-approximation algorithms of Yannakakis and Goemans and Williamson. Both of them are based on the probabilistic method. Let $\boldsymbol{x}^p = (x_1^p, \ldots, x_{2n}^p)$ be a *random* truth assignment with $0 \le x_i^p = p_i \le 1$ ($x_{n+i}^p = 1 - x_i^p = 1 - p_i = p_{n+i}$). That is, $\boldsymbol{x}^p$ is obtained by setting independently each variable $x_i \in X$ to be true with probability p_i (and $x_{n+i} = \bar{x}_i$ to be true with probability $p_{n+i} = 1 - p_i$). Then the probability that a clause $C_j = x_{j_1} \vee x_{j_2} \vee \cdots \vee x_{j_{k_j}} \in \mathcal{C}$ is satisfied by the random truth assignment $\boldsymbol{x}^p = (x_1^p, \ldots, x_{2n}^p)$ is

$$C_j(\boldsymbol{x}^p) = 1 - \prod_{i=1}^{k_j} (1 - x_{j_i}^p). \tag{1}$$

Thus, the expected value of the random truth assignment $\boldsymbol{x}^p$ is

$$F_{\mathcal{C}}(\boldsymbol{x}^p) = \sum_{C_j \in \mathcal{C}} w(C_j) C_j(\boldsymbol{x}^p). \tag{2}$$

The probabilistic method assures that there is a truth assignment $\boldsymbol{x}^q \in \{0,1\}^{2n}$ of value at least $F_{\mathcal{C}}(\boldsymbol{x}^p)$. Such a truth assignment $\boldsymbol{x}^q$ can be obtained in polynomial time by the method of conditional probabilities [6, 15]. The .75-approximation algorithm of Yannakakis finds a vector $\boldsymbol{p}$ such that expected value $F_{\mathcal{C}}(\boldsymbol{x}^p)$ of a random truth assignment $\boldsymbol{x}^p$ is at least

$$.75\hat{W}_1 + .75\hat{W}_2 + .75\hat{W}_3 + .765\hat{W}_4 + \ .762\hat{W}_5 + .822\hat{W}_6 + \sum_{k \ge 7} (1 - (.75)^k)\hat{W}_k$$

where $\hat{W}_k = \sum_{C \in \mathcal{C}_k} w(C)C(\hat{\boldsymbol{x}})$ and $F_{\mathcal{C}}(\hat{\boldsymbol{x}}) = \sum_{k \ge 1} \hat{W}_k$ for an optimal truth assignment $\hat{\boldsymbol{x}}$ of $\mathcal{C}$ ($\mathcal{C}_k$ denotes the set of clauses in $\mathcal{C}$ with k literals throughout the paper). Using a hybrid approach of combining a randomized version of Johnson's algorithm and randomized rounding of a linear programming (LP) relaxation of MAX SAT, Goemans and Williamson [6] also obtain a .75-approximation algorithm. It finds a random truth assignment of value at least

$$.75W_1^* + .75W_2^* + .789W_3^* + .810W_4^* + .820W_5^* + .824W_6^* + \sum_{k \ge 7} \beta_k W_k^*$$

where

$$\beta_k = \frac{1}{2}\left(2 - \frac{1}{2^k} - \left(1 - \frac{1}{k}\right)^k\right) \qquad (k \geq 1), \tag{3}$$

and W_k^* is the objective function value of clauses of length k provided by the linear programming relaxation. Note that $.826 < \beta_k < 1 - (.75)^k$ for $k \geq 7$. Observe that in terms of the behavior of the algorithm on clauses of length k, the two algorithms are incomparable. Asano [1] gives an improvement of Yannakakis' algorithm for finding a random truth assignment with value at least

$$.75\hat{W}_1 + .75\hat{W}_2 + .791\hat{W}_3 + .811\hat{W}_4 + .823\hat{W}_5 + .850\hat{W}_6 + \sum_{k\geq 7}(1 - (.75)^k)\hat{W}_k.$$

He noted that this bound is better than the bounds of Goemans and Williamson and Yannakakis and that it leads to a .770-approximation algorithm if it is combined with the algorithms in Asano, Ono, and Hirata [2]. His algorithm is, however, very complicated and only experts on network algorithms can understand it without difficulty. Thus, simpler algorithms with high performance guarantees are desired.

In this paper, we show that one of the non-hybrid algorithms of Goemans and Williamson finds a random truth assignment $\boldsymbol{x}^p$ with value $F_{\mathcal{C}}(\boldsymbol{x}^p)$ at least

$$.75W_1^* + .75W_2^* + .804W_3^* + .851W_4^* + .888W_5^* + .915W_6^* + \sum_{k\geq 7}\gamma_k W_k^*,$$

where

$$\gamma_k = 1 - \frac{1}{2}\left(\frac{3}{4}\right)^{k-1}\left(1 - \frac{1}{3(k-1)}\right)^{k-1}.$$

Note that this bound is the best among the bounds described above since $\gamma_k > 1 - (.75)^k$ for $k \geq 3$. If this is combined with the other MAX kSAT algorithms as stated previously, it leads to the improved approximation algorithms for MAX SAT. Additionally, a generalization of this algorithm finds a random truth assignment with value at least

$$\begin{aligned} &.914W_1^* + .75W_2^* + .75W_3^* + .766W_4^* + .784W_5^* + .801W_6^* \\ &\quad +.817W_7^* + .832W_8^* + .845W_9^* + .857W_{10}^* + \sum_{k\geq 11}\left(1 - .914^k\left(1 - \frac{1}{k}\right)^k\right)W_k^*. \end{aligned}$$

When combined with Zwick's conjectured MAX SAT algorithm, this leads to the best overall performance guarantee.

The significance of our result is twofold. First, if $P \neq NP$ the best performance guarantee achievable for MAX SAT is $\frac{7}{8}$. Our result takes us almost halfway there from Zwick's bound of .7977, assuming his conjecture is correct. Second, while much of the recent effort on MAX SAT has been invested in algorithms and analysis for semidefinite programming formulations, we show that stronger analysis of algorithms using linear programming is useful.

The remainder of the paper is structured as follows. In Section 2 we review the algorithms of Goemans and Williamson, and then sharpen their analysis. In Section 3 we give the improved approximation algorithm for MAX SAT.

2 The MAX SAT Algorithms of Goemans and Williamson

Goemans and Williamson formulate MAX SAT as an integer programming problem as follows [6]:

$$
\begin{array}{lll}
\max & \sum_{C_j \in \mathcal{C}} w(C_j) z_j & \\
\text{s.t.} & \sum_{i=1}^{k_j} y_{j_i} \geq z_j & \forall C_j = x_{j_1} \vee x_{j_2} \vee \cdots \vee x_{j_{k_j}} \in \mathcal{C} \\
& y_i + y_{n+i} = 1 & \forall i \in \{1, 2, ..., n\} \\
& y_i \in \{0, 1\} & \forall i \in \{1, 2, ..., 2n\} \\
& z_j \in \{0, 1\} & \forall C_j \in \mathcal{C}
\end{array}
$$

In this IP (integer programming) formulation, variables $\boldsymbol{y} = (y_i)$ correspond to the literals $\{x_1, \ldots, x_{2n}\}$ and variables $\boldsymbol{z} = (z_j)$ correspond to the clauses $\mathcal{C}$. Thus, variable $y_i = 1$ if and only if $x_i = 1$. Similarly, $z_j = 1$ if and only if C_j is satisfied. The first set of constraints implies that one of the literals in a clause is true if the clause is satisfied and thus this formulation exactly corresponds to MAX SAT. If the variables $\boldsymbol{y} = (y_i)$ and variables $\boldsymbol{z} = (z_j)$ are allowed to take on any values between 0 and 1, then the following LP relaxation (GW) of MAX SAT is obtained.

$$
\begin{array}{lll}
\max & \sum_{C_j \in \mathcal{C}} w(C_j) z_j & \\
\text{s.t.} & \sum_{i=1}^{k_j} y_{j_i} \geq z_j & \forall C_j = x_{j_1} \vee x_{j_2} \vee \cdots \vee x_{j_{k_j}} \in \mathcal{C} \\
(GW) & y_i + y_{n+i} = 1 & \forall i \in \{1, 2, ..., n\} \\
& 0 \leq y_i \leq 1 & \forall i \in \{1, 2, ..., 2n\} \\
& 0 \leq z_j \leq 1 & \forall C_j \in \mathcal{C}
\end{array}
$$

Using an optimal solution $(\boldsymbol{y}^*, \boldsymbol{z}^*)$ to this LP relaxation of MAX SAT, Goemans and Williamson set each variable x_i to be true with probability y_i^*. Then they show that the probability $C_j(\boldsymbol{y}^*)$ of a clause C_j with k literals being satisfied is at least $\left(1 - \left(1 - \frac{1}{k}\right)^k\right) z_j^*$. Thus, the expected value $F(\boldsymbol{y}^*)$ of the random truth assignment $\boldsymbol{y}^*$ obtained in this way satisfies

$$F(\boldsymbol{y}^*) \geq \sum_{k \geq 1} \left(1 - \left(1 - \frac{1}{k}\right)^k\right) W_k^* \geq \left(1 - \frac{1}{e}\right) W^* \approx .632 W^*,$$

where e is the base of natural logarithm and $W^* = \sum_{C_j \in \mathcal{C}} w(C_j) z_j^*$ and $W_k^* = \sum_{C_j \in \mathcal{C}_k} w(C_j) z_j^*$ (note that $W^* = \sum_{C_j \in \mathcal{C}} w(C_j) z_j^* \geq \hat{W} = \sum_{C_j \in \mathcal{C}} w(C_j) \hat{z}_j$ for an optimal solution $(\hat{\boldsymbol{y}}, \hat{\boldsymbol{z}})$ to the IP formulation of MAX SAT). Thus, this algorithm leads to .632-approximation algorithm for MAX SAT. If this is combined with a randomized version of Johnson's algorithm, it leads to a $\frac{3}{4}$-approximation algorithm. Note that the randomized version of Johnson's algorithm uses the random truth assignment in which each variable x_i is set to be true with probability $\frac{1}{2}$ and the probability of a clause C_j with k literals being satisfied is $1 - \frac{1}{2^k}$. Thus, if we choose the better of these two random truth assignments then the expected value is at least $\sum_{k\geq 1}\beta_k W_k^* \geq \frac{3}{4}W^*$ where β_k is defined in Eq. (3).

Goemans and Williamson also considered three other non-linear randomized rounding algorithms. In these algorithms, each variable x_i is set to be true with probability $f_\ell(y_i^*)$ defined as follows ($\ell = 1, 2, 3$).

$$f_1(y) = \begin{cases} \frac{3}{4}y + \frac{1}{4} & \text{if } 0 \leq y \leq \frac{1}{3} \\ \frac{1}{2} & \text{if } \frac{1}{3} \leq y \leq \frac{2}{3} \\ \frac{3}{4}y & \text{if } \frac{2}{3} \leq y \leq 1 \end{cases}$$

$$f_2(y) = (2a-1)y + 1 - a \quad \left(\frac{3}{4} \leq a \leq \frac{3}{\sqrt[3]{4}} - 1 \approx .889881\right)$$

$$1 - 4^{-y} \leq f_3(y) \leq 4^{y-1}.$$

Note that

$$f_\ell(y_i^*) + f_\ell(y_{n+i}^*) = 1$$

holds for $\ell = 1, 2$ and that $f_3(y_i^*)$ has to be chosen to satisfy $f_3(y_i^*) + f_3(y_{n+i}^*) = 1$. They then proved that the random truth assignments $\boldsymbol{x}^p = f_\ell(\boldsymbol{y}^*) = (f_\ell(y_1^*), \ldots, f_\ell(y_{2n}^*))$ obtained in this way all have the expected values at least $\frac{3}{4}W^*$ and all lead to $\frac{3}{4}$-approximation algorithms. In particular, they show that for each k, the probability of a clause C_j with k literals being satisfied is at least $\frac{3}{4}z_j^*$, and thus the expected weight of satisfied clauses in $\mathcal{C}_k$ is at least $\frac{3}{4}W_k^*$.

To give an improved approximation algorithm for MAX SAT, we will sharpen the analysis of Goemans and Williamson to provide more precise bounds on the probability of a clause $C_j = x_{j_1} \vee x_{j_2} \vee \cdots \vee x_{j_k}$ with k literals being satisfied and on the expected weight of satisfied clauses in $\mathcal{C}_k$ by the random truth assignment $\boldsymbol{x}^p = f_\ell(\boldsymbol{y}^*)$ ($\ell = 1, 2$) for each k. By symmetry, we assume $x_{j_i} = x_i$ for each $i = 1, 2, ..., k$ since $f_\ell(x) = 1 - f_\ell(\bar{x})$ and we can set $x := \bar{x}$ if necessary. Thus, we consider clause $C_j = x_1 \vee x_2 \vee \cdots \vee x_k$ corresponding to the constraint $y_1 + y_2 + \cdots + y_k \geq z_j$ in the LP relaxation (GW) of MAX SAT, and we give a bound on the ratio of the probability of clause C_j being satisfied by the random truth assignment $\boldsymbol{x}^p = f_\ell(\boldsymbol{y}^*)$ ($\ell = 1, 2$) to z_j^*. Since we will need it later, we will instead analyze a parametrized function f_1^a, as follows, where $f_1 = f_1^{3/4}$:

$$f_1^a(y) = \begin{cases} ay + 1 - a & \text{if } 0 \leq y \leq 1 - \frac{1}{2a} \\ \frac{1}{2} & \text{if } 1 - \frac{1}{2a} \leq y \leq \frac{1}{2a} \\ ay & \text{if } \frac{1}{2a} \leq y \leq 1, \end{cases} \tag{4}$$

for $\frac{1}{2} \leq a \leq 1$. Similarly, since f_2 is a parametarized function, we denote it by f_2^a from now on by explicitly expressing parameter a, i.e.,

$$f_2^a(y) = (2a-1)y + 1 - a \quad \left(\frac{1}{2} \leq a \leq 1\right). \tag{5}$$

We will use γ_k^a and δ_k^a to bound the probabilities of clause C_j with k literals being satisfied by the random truth assignments $\boldsymbol{x}^p = f_1^a(\boldsymbol{y}^*)$ and $\boldsymbol{x}^p = f_2^a(\boldsymbol{y}^*)$ respectively, where

$$\gamma_k^a = \begin{cases} a & \text{if } k = 1 \\ \min\{\gamma_{k,1}^a, \gamma_{k,2}^a\} & \text{if } k \geq 2 \end{cases} \tag{6}$$

with

$$\gamma_{k,1}^a = 1-\frac{1}{2}a^{k-1}\left(1-\frac{1-\frac{1}{2a}}{k-1}\right)^{k-1}, \tag{7}$$

$$\gamma_{k,2}^a = 1-a^k\left(1-\frac{1}{k}\right)^k, \tag{8}$$

and

$$\delta_k^a = 1-\left(a-\frac{2a-1}{k}\right)^k = 1-a^k\left(1-\frac{2-\frac{1}{a}}{k}\right)^k. \tag{9}$$

Since we obtain no structural property of optimal solutions to the LP relaxation (GW) of MAX SAT, we assume, throughout this paper, that any feasible solution can become an optimal solution. Under this generic assumption, the probability of clause $C_j = x_1 \vee x_2 \vee \cdots \vee x_k$ with k literals being satisfied by the random truth assignment $\boldsymbol{x}^p = f_2^a(\boldsymbol{y}^*)$ is at most $\delta_k^a z_j^*$ since clause C_j corresponds to the LP constraint $y_1^* + y_2^* + \cdots + y_k^* \geq z_j^*$ and it is possible that $y_1^* = y_2^* = \cdots y_k^* = \frac{1}{k}$, $z_j^* = 1$ and, by Eqs. (1), (5) and (9),

$$C_j(f_2^a(\boldsymbol{y}^*)) = 1-\prod_{i=1}^{k}(1-f_2^a(y_i^*)) = 1-\left(a-\frac{2a-1}{k}\right)^k = 1-a^k\left(1-\frac{2-\frac{1}{a}}{k}\right)^k = \delta_k^a z_j^*.$$

For the same reason, the probability of clause $C_j = x_1 \vee x_2 \vee \cdots \vee x_k$ with k literals being satisfied by the random truth assignment $\boldsymbol{x}^p = f_1^a(\boldsymbol{y}^*)$ is at most $\gamma_k^a z_j^*$, since, if $k \geq 2$ then, for $y_1^* = y_2^* = \cdots y_k^* = \frac{1}{k} \leq 1-\frac{1}{2a}$, $z_j^* = 1$, we have

$$C_j(f_1^a(\boldsymbol{y}^*)) = 1-\prod_{i=1}^{k}(1-f_1^a(y_i^*)) = 1-\left(1-\left(\frac{a}{k}+1-a\right)\right)^k = 1-a^k\left(1-\frac{1}{k}\right)^k = \gamma_{k,2}^a z_j^*,$$

by Eqs. (1), (4) and (8), and, for $y_1^* = y_2^* = \cdots y_{k-1}^* = \frac{1-\frac{1}{2a}}{k-1}$, $y_k^* = \frac{1}{2a}$, $z_j^* = 1$, we have

$$\begin{aligned} C_j(f_1^a(\boldsymbol{y}^*)) &= 1-\prod_{i=1}^{k}(1-f_1^a(y_i^*)) = 1-\left(1-\left(\frac{a(1-\frac{1}{2a})}{k-1}+1-a\right)\right)^{k-1}\left(1-\frac{a}{2a}\right) \\ &= 1-\frac{1}{2}a^{k-1}\left(1-\frac{1-\frac{1}{2a}}{k-1}\right)^{k-1} = \gamma_{k,1}^a z_j^* \end{aligned}$$

by Eqs. (1), (4) and (7). Furthermore, if $k = 1$ then, for $\frac{1}{2a} \leq y_1^* = z_j^* \leq 1$, we have

$$C_j(f_1^a(\boldsymbol{y}^*)) = 1-(1-f_1^a(y_1^*)) = f_1^a(y_1^*) = ay_1^* = az_j^* = \gamma_1^a z_j^*.$$

Thus, the probabilities of clause C_j with k literals being satisfied by the random truth assignments $\boldsymbol{x}^p = f_1^a(\boldsymbol{y}^*)$ and $\boldsymbol{x}^p = f_1^a(\boldsymbol{y}^*)$ are at most $\gamma_k^a z_j^*$ and $\delta_k^a z_j^*$, respectively. Below we will show that the converse inqualities hold; that is, the following theorems are true.

Theorem 2.1 The probability of $C_j = x_1 \vee x_2 \vee \cdots \vee x_k \in \mathcal{C}$ being satisfied by the random truth assignment $\boldsymbol{x}^p = f_1^a(\boldsymbol{y}^*) = (f_1^a(y_1^*), \ldots, f_1^a(y_{2n}^*))$ is

$$C_j(f_1^a(\boldsymbol{y}^*)) = 1-\prod_{i=1}^{k}(1-f_1^a(y_i^*)) \geq \gamma_k^a z_j^*, \tag{10}$$

for $\frac{1}{2} \le a \le 1$, where f_1^a is the function defined in Eq. (4) and γ_k^a is the number defined in Eq. (6). Thus, the expected value $F(f_1^a(\boldsymbol{y}^*))$ of the random truth assignment $\boldsymbol{x}^p = f_1^a(\boldsymbol{y}^*)$ satisfies

$$F(f_1^a(\boldsymbol{y}^*)) \ge \sum_{k \ge 1} \gamma_k^a W_k^*.$$

Theorem 2.2 The probability of $C_j = x_1 \vee x_2 \vee \cdots \vee x_k \in \mathcal{C}$ being satisfied by the random truth assignment $\boldsymbol{x}^p = f_2^a(\boldsymbol{y}^*) = (f_2^a(y_1^*), \ldots, f_2^a(y_{2n}^*))$ is

$$C_j(f_2^a(\boldsymbol{y}^*)) = 1 - \prod_{i=1}^{k} (1 - f_2^a(y_i^*)) \ge \delta_k^a z_j^*, \tag{11}$$

for $\frac{1}{2} \le a \le 1$, where f_2^a is the function defined in Eq. (5) and δ_k^a is the number defined in Eq. (9). Thus, the expected value $F(f_2^a(\boldsymbol{y}^*))$ of the random truth assignment $\boldsymbol{x}^p = f_2^a(\boldsymbol{y}^*)$ satisfies

$$F(f_2^a(\boldsymbol{y}^*)) \ge \sum_{k \ge 1} \delta_k^a W_k^*.$$

Furthermore, under the generic assumption described above, we will show that the algorithm using f_1^a dominates the algorithm that uses f_2^a; that is, the following theorem holds.

Theorem 2.3 For γ_k^a and δ_k^a defined in Eqs. (6) and (9), $\gamma_k^a > \delta_k^a$ hold for all $k \ge 3$ and for all a with $\frac{1}{2} < a < 1$. For $k = 1, 2$, $\gamma_k^a = \delta_k^a$ ($\gamma_1^a = \delta_1^a = a$, $\gamma_2^a = \delta_2^a = \frac{3}{4}$) hold.

We first give a proof of Theorem 2.2 since it is much simpler than a proof of Theorem 2.1. Then we give a proof of Theorem 2.1, and finally, we will give a proof of Theorem 2.3.

Proof of Theorem 2.2: Noting that clause $C_j = x_1 \vee x_2 \vee \cdots \vee x_k$ corresponds to the LP constraint $y_1^* + y_2^* + \cdots + y_k^* \ge z_j^*$, we will show that

$$C_j(f_2^a(\boldsymbol{y}^*)) = 1 - \prod_{i=1}^{k} (1 - f_2^a(y_i^*)) \ge \delta_k^a z_j^*.$$

Let

$$r(z_j) = 1 - \left(a - \frac{(2a-1)z_j}{k}\right)^k, \quad \text{and} \quad R(z_j) = \delta_k^a z_j = \left(1 - \left(a - \frac{2a-1}{k}\right)^k\right) z_j.$$

Then, since $r(z_j)$ is concave in z_j with $0 \le z_j \le 1$ and $r(0) = 1 - a^k \ge 0 = R(0)$ and $r(1) = 1 - (a - \frac{2a-1}{k})^k = R(1)$, we have

$$r(z_j) = 1 - \left(a - \frac{(2a-1)z_j}{k}\right)^k \ge \left(1 - \left(a - \frac{2a-1}{k}\right)^k\right) z_j = R(z_j).$$

Thus,

$$\begin{aligned}
C_j(f_2^a(\boldsymbol{y}^*)) &= 1 - \prod_{i=1}^{k} (1 - f_2^a(y_i^*)) = 1 - \prod_{i=1}^{k} (a - (2a-1)y_i^*) \\
&\ge 1 - \left(a - \frac{(2a-1)\sum_{i=1}^{k} y_i^*}{k}\right)^k \\
&\ge 1 - \left(a - \frac{(2a-1)z_j^*}{k}\right)^k \\
&\ge \left(1 - \left(a - \frac{2a-1}{k}\right)^k\right) z_j^* = \delta_k^a z_j^*
\end{aligned}$$

by the arithmetic/geometric mean inequality and $y_1^* + y_2^* + \cdots + y_k^* \geq z_j^*$. $\square$

To make a proof of Theorem 2.1 substantially shorter, we use the following lemma.

Lemma 2.4 1. For a fixed a with $\frac{1}{2} \leq a \leq 1$, let $\frac{1}{2a} \leq y \leq 1$. Then for $k \geq 2$,

$$(1-ay)\left(1-\frac{1-y}{k-1}\right)^{k-1} \leq \frac{1}{2}\left(1-\frac{1-\frac{1}{2a}}{k-1}\right)^{k-1}. \tag{12}$$

2. Let ℓ be a positive integer and let b and z satisfy $0 \leq b \leq z \leq 1$. Let μ_b be a positive function depending only on b and decreasing with b and let μ be a positive constant such that $\mu_b \leq \mu$ and $(b+1)\mu \leq 1$. Then

$$1-\mu_b\left(1-\frac{z-b}{\ell}\right)^{\ell} \geq \left(1-\mu_b\left(1-\frac{1-b}{\ell}\right)^{\ell}\right)z \geq \left(1-\mu\left(1-\frac{1-b}{\ell}\right)^{\ell}\right)z. \tag{13}$$

Proof: To prove Part 1, we show that

$$g(y) \equiv (1-ay)\left(1-\frac{1-y}{k-1}\right)^{k-1}$$

is decreasing with $y \geq \frac{1}{2a}$. Then we have Eq.(12), since $g(y) \leq g(\frac{1}{2a}) = \frac{1}{2}(1-\frac{1-\frac{1}{2a}}{k-1})^{k-1}$. To see that $g(y)$ is decreasing for $y \geq \frac{1}{2a}$, we observe that the derivative $g'(y)$ of $g(y)$ is

$$\begin{aligned} g'(y) &= -a\left(1-\frac{1-y}{k-1}\right)^{k-1} + (1-ay)\left(1-\frac{1-y}{k-1}\right)^{k-2} \\ &= \left(1-\frac{1-y}{k-1}\right)^{k-2}\left(-a\left(1-\frac{1-y}{k-1}\right)+1-ay\right) \\ &= \left(1-\frac{1-y}{k-1}\right)^{k-2}\left(1-\frac{a(k-2)+aky}{k-1}\right) \leq 0, \end{aligned}$$

since $a(k-2)+aky \geq a(k-2)+\frac{k}{2} \geq \frac{k-2}{2}+\frac{k}{2} = k-1$ by the conditions on y, a, and k.

In Eq.(13) of Part 2, the second inequality is clear and we consider only the first inequality. For

$$h(z) \equiv 1-\mu_b\left(1-\frac{z-b}{\ell}\right)^{\ell} \quad \text{and} \quad H(z) \equiv \left(1-\mu_b\left(1-\frac{1-b}{\ell}\right)^{\ell}\right)z,$$

$h(z)$ is concave in z and $h(1) = 1-\mu_b(1-\frac{1-b}{\ell})^{\ell} = H(1)$. Thus, if we show that $h(b) \geq H(b)$ given the conditions that $\mu_b \leq \mu$ and $(b+1)\mu \leq 1$, we will have $h(z) \geq H(z)$ for $b \leq z \leq 1$ which implies Eq.(13). Set

$$G(b) = 1-b-\mu\left(1-\left(1-\frac{1-b}{\ell}\right)^{\ell}b\right).$$

Then

$$\begin{aligned} h(b)-H(b) &= (1-\mu_b) - \left(1-\mu_b\left(1-\frac{1-b}{\ell}\right)^{\ell}\right)b \\ &= 1-b-\mu_b\left(1-\left(1-\frac{1-b}{\ell}\right)^{\ell}b\right) \\ &\geq 1-b-\mu\left(1-\left(1-\frac{1-b}{\ell}\right)^{\ell}b\right) = G(b). \end{aligned}$$

This implies that, if $G(b) \geq 0$ then we have $h(b) - H(b) \geq 0$ and Eq.(13). Thus, we will show $G(b) \geq 0$. The derivative of $G(b)$ with respect to b is

$$\begin{aligned} G'(b) &= -1 + \mu\left(1 - \frac{1-b}{\ell}\right)^{\ell} + \mu b\left(1 - \frac{1-b}{\ell}\right)^{\ell-1} \\ &= -1 + \mu\left(1 - \frac{1-b}{\ell}\right)^{\ell-1}\left(b + \left(1 - \frac{1-b}{\ell}\right)\right) \\ &\leq -1 + \mu(b+1) \leq 0, \end{aligned}$$

by the stated conditions. Thus, $G(b)$ is decreasing with b, and, by $G(1) = 0$, we have $G(b) \geq 0$. □

Proof of Theorem 2.1: Noting that clause $C_j = x_1 \vee x_2 \vee \cdots \vee x_k$ corresponds to the constraint

$$y_1^* + y_2^* + \cdots + y_k^* \geq z_j^* \tag{14}$$

in the LP relaxation (GW) of MAX SAT, we will show that

$$C_j(f_1^a(\boldsymbol{y}^*)) = 1 - \prod_{i=1}^{k}(1 - f_1^a(y_i^*)) \geq \gamma_k^a z_j^*.$$

If $k = 1$, then

$$C_j(f_1^a(\boldsymbol{y}^*)) = f_1^a(y_1^*) \geq a y_1^* \geq a z_j^* = \gamma_1^a z_j^*$$

by Eq.(14). For $k \geq 2$, we assume $y_1^* \leq y_2^* \leq \cdots \leq y_k^*$ by symmetry and consider three cases as follows. Case 1: $y_k^* \leq 1 - \frac{1}{2a}$; Case 2: $y_{k-1}^* \leq 1 - \frac{1}{2a} < y_k^*$; Case 3: $1 - \frac{1}{2a} < y_{k-1}^* \leq y_k^*$.

Case 1: $y_k^* \leq 1 - \frac{1}{2a}$. Since all $y_i^* \leq 1 - \frac{1}{2a}$ $(i = 1, 2, ..., k)$, we have $f_1^a(y_i^*) = a y_i^* + 1 - a$ and $1 - f_1^a(y_i^*) = a(1 - y_i^*)$. Thus, we have

$$\begin{aligned} C_j(f_1^a(\boldsymbol{y}^*)) &= 1 - \prod_{i=1}^{k}(1 - f_1^a(y_i^*)) = 1 - a^k \prod_{i=1}^{k}(1 - y_i^*) \\ &\geq 1 - a^k\left(1 - \frac{\sum_{i=1}^{k} y_i^*}{k}\right)^k \\ &\geq 1 - a^k\left(1 - \frac{z_j^*}{k}\right)^k \\ &\geq \left(1 - a^k\left(1 - \frac{1}{k}\right)^k\right) z_j^* \\ &= \gamma_{k,2}^a z_j^* \geq \gamma_k^a z_j^*, \end{aligned}$$

where the first inequality follows by the arithmetic/geometric mean inequality, the second by Eq.(14), the third by Eq.(13) in Lemma 2.4 with $\mu_b = a^k = \mu$ and $b = 0$ satisfying $(b+1)\mu \leq 1$.

Case 2: $y_{k-1}^* \leq 1 - \frac{1}{2a} < y_k^*$. Since $1 - f_1^a(y_i^*) = a(1 - y_i^*)$ $(i = 1, 2, ..., k-1)$, we have

$$\begin{aligned} C_j(f_1^a(\boldsymbol{y}^*)) &= 1 - \prod_{i=1}^{k}(1 - f_1^a(y_i^*)) = 1 - (1 - f_1^a(y_k^*))\, a^{k-1} \prod_{i=1}^{k-1}(1 - y_i^*) \\ &\geq 1 - (1 - f_1^a(y_k^*))\, a^{k-1}\left(1 - \frac{\sum_{i=1}^{k-1} y_i^*}{k-1}\right)^{k-1} \\ &\geq 1 - (1 - f_1^a(y_k^*))\, a^{k-1}\left(1 - \frac{z_j^* - y_k^*}{k-1}\right)^{k-1} \end{aligned}$$

by the arithmetic/geometric mean inequality and Eq.(14). We now assume[3] that $y_k \le z_j^*$. Let

$$\mu = \frac{1}{2}a^{k-1}, \quad \text{and} \quad \mu_b = (1 - f_1^a(b))a^{k-1} \text{ using } b = y_k^*.$$

Then

$$\mu_b = \begin{cases} \frac{1}{2}a^{k-1} & \text{if } 1 - \frac{1}{2a} < y_k^* \le \frac{1}{2a}, \\ (1 - ay_k^*)a^{k-1} & \text{if } \frac{1}{2a} < y_k^* \le 1. \end{cases}$$

Thus, in either case, μ_b is a positive function decreasing with $b = y_k^*$ and the inequalities

$$\mu \ge \mu_b \quad \text{and} \quad (b+1)\mu \le 1$$

hold, since if $y_k^* \le \frac{1}{2a}$ then $\mu_b = \mu$ and if $y_k^* > \frac{1}{2a}$ then $\mu_b = (1 - ay_k^*)a^{k-1} \le \frac{1}{2}a^{k-1} = \mu$. Furthermore, $(b+1)\mu \le 2 \cdot \frac{1}{2}a^{k-1} = a^{k-1} \le 1$, since $b = y_k^* \le 1$ and $\frac{1}{2} \le a \le 1$. Thus, we have

$$\begin{aligned} C_j(f_1^a(\boldsymbol{y}^*)) &\ge 1 - (1 - f_1^a(y_k^*))\, a^{k-1} \left(1 - \frac{z_j^* - y_k^*}{k-1}\right)^{k-1} \\ &\ge \left(1 - (1 - f_1^a(y_k^*))\, a^{k-1} \left(1 - \frac{1 - y_k^*}{k-1}\right)^{k-1}\right) z_j^* \\ &\ge \left(1 - \frac{1}{2}a^{k-1} \left(1 - \frac{1 - \frac{1}{2a}}{k-1}\right)^{k-1}\right) z_j^* \\ &= \gamma_{k,1}^a z_j^* \ge \gamma_k^a z_j^*, \end{aligned}$$

where the second inequality follows by Eq.(13) in Lemma 2.4 and the third inequality follows from the fact that if $y_k^* \le \frac{1}{2a}$ then $\left(1 - \frac{1-y_k^*}{k-1}\right)^{k-1}$ is increasing with y_k^* or from Eq.(12) in Lemma 2.4 if $y_k^* > \frac{1}{2a}$.

Case 3: $1 - \frac{1}{2a} < y_{k-1}^* \le y_k^*$. Let ℓ be the number satisfying $y_\ell^* \le 1 - \frac{1}{2a} < y_{\ell+1}^*$. Then, $\ell \le k-2$, and, since $1 - f_1^a(y_{\ell+i}^*) \le \frac{1}{2} \le a$ $(i = 1, 2, ..., k-\ell)$ and $1 - f_1^a(y_i^*) = a(1 - y_i^*) \le a$ $(i = 1, 2, ..., \ell)$, we have

$$\begin{aligned} C_j(f_1^a(\boldsymbol{y}^*)) &= 1 - \prod_{i=1}^{k}(1 - f_1^a(y_i^*)) = 1 - \prod_{i=\ell+1}^{k}(1 - f_1^a(y_i^*)) \prod_{i=1}^{\ell}(a(1 - y_i^*)) \\ &\ge 1 - \left(\frac{1}{2}\right)^{k-\ell} a^\ell \ge 1 - \left(\frac{1}{2}\right)^2 a^{k-2}. \end{aligned}$$

Furthermore, since $\left(1 - \frac{1-\frac{1}{2a}}{k-1}\right)^{k-1}$ is increasing with $k \ge 2$ and

$$\frac{1}{4} = \frac{1}{2} \cdot a \cdot \frac{1}{2a} \le \frac{1}{2} \cdot a \left(1 - \frac{1 - \frac{1}{2a}}{k-1}\right)^{k-1},$$

we have

$$C_j(f_1^a(\boldsymbol{y}^*)) \ge 1 - \frac{1}{2}a^{k-1}\left(1 - \frac{1 - \frac{1}{2a}}{k-1}\right)^{k-1} = \gamma_{k,1}^a \ge \gamma_k^a \ge \gamma_k^a z_j^*$$

[3] In an optimal solution to the LP (GW), $z_j^* = \min\{1, \sum_{i=1}^k y_i^*\}$ and $y_i^* \le 1$, so that $y_k^* \le z_j^*$. In later sections, we will be able to make this assumption by adding it as an explicit constraint to the linear program or semidefinite program.

by $z_j^* \leq 1$. $\square$

We will further show (in Theorem 2.6) that we can state precisely when $\gamma_k^a = \gamma_{k,1}^a$. Now, we will give a proof of Theorem 2.3 by using the following lemma which makes its proof substantially shorter.

Lemma 2.5 Let $k \geq 2$ be an integer and let x, y be two numbers independent of k. Let

$$d(x,k,y) = \ln \frac{(1-\frac{y}{k})^k}{\left(1-\frac{1-x}{k-1}\right)^{k-1}}. \tag{15}$$

Then the following statements hold.

1. Let x, y satisfy $0 < 2(1-x) \leq y \leq 1$. Then $\frac{\partial d(x,k,y)}{\partial k} > 0$ for $k > 2$, where $\frac{\partial d(x,k,y)}{\partial k}$ is the derivative of $d(x,k,y)$ with respect to k. Furthermore $\frac{\partial d(x,k,y)}{\partial k} \geq 0$ for $k = 2$ and $d(x,k,y)$ is increasing with $k \geq 2$.
2. Let x, y satisfy $0 < y \leq 1 - x \leq 1$. Then $\frac{\partial d(x,k,y)}{\partial k} < 0$ for $k \geq 2$ and $d(x,k,y)$ is decreasing with $k \geq 2$.

Proof: Since $d(x,k,y) = k\ln(k-y) - k\ln k - (k-1)\ln(k-2+x) + (k-1)\ln(k-1)$, we have

$$\begin{aligned}\frac{\partial d(x,k,y)}{\partial k} &= \ln(k-y) + \frac{k}{k-y} - \ln k - 1 - \ln(k-2+x) - \frac{k-1}{k-2+x} + \ln(k-1) + 1 \\ &= -\ln\frac{k(k-2+x)}{(k-y)(k-1)} + \frac{k(k-2+x)-(k-y)(k-1)}{(k-y)(k-2+x)}. \end{aligned} \tag{16}$$

To prove Part 1, we assume that $0 < 2(1-x) \leq y \leq 1$ and $k \geq 2$. Then

$$k(k-2+x) - (k-y)(k-1) = (k-1)y - k(1-x) \geq (k-2)(1-x) \geq 0 \tag{17}$$

and we have

$$\ln\frac{k(k-2+x)}{(k-y)(k-1)} = \ln\left(1 + \frac{(k-1)y-k(1-x)}{(k-y)(k-1)}\right) \leq \frac{(k-1)y-k(1-x)}{(k-y)(k-1)}$$

since $\ln(1+z) \leq z$ for any $z \geq 0$. Thus, by Eq.(16) and Eq.(17),

$$\begin{aligned}\frac{\partial d(x,k,y)}{\partial k} &\geq -\frac{(k-1)y-k(1-x)}{(k-y)(k-1)} + \frac{(k-1)y-k(1-x)}{(k-y)(k-2+x)} \\ &= \frac{(k-1)y-k(1-x)}{(k-y)(k-2+x)(k-1)}(1-x) > 0\end{aligned}$$

if $k > 2$, and $\frac{\partial d(x,k,y)}{\partial k} \geq 0$ if $k = 2$. Thus, $d(x,k,y)$ is increasing with $k \geq 2$ for all x, y with $0 < 2(1-x) \leq y \leq 1$.

To prove Part 2, we assume that $0 < y \leq 1-x \leq 1$ and $k \geq 2$. Then

$$k(k-2+x) - (k-y)(k-1) = (k-1)y - k(1-x) \leq -(1-x) < 0 \tag{18}$$

and we have

$$\ln\frac{(k-y)(k-1)}{k(k-2+x)} = \ln\left(1 + \frac{k(1-x)-(k-1)y}{k(k-2+x)}\right) \leq \frac{k(1-x)-(k-1)y}{k(k-2+x)}$$

since $\ln(1+z) \le z$ for any $z \ge 0$. Thus, by Eq.(16) and Eq.(18),

$$\frac{\partial d(x,k,y)}{\partial k} \le \frac{k(1-x)-(k-1)y}{k(k-2+x)} - \frac{k(1-x)-(k-1)y}{(k-y)(k-2+x)} = -\frac{k(1-x)-(k-1)y}{k(k-y)(k-2+x)}y < 0$$

for $k \ge 2$. Thus, $d(x,k,y)$ is decreasing with $k \ge 2$ for all x, y with $0 < y \le 1-x \le 1$.

□

Proof of Theorem 2.3: Since

$$\gamma_{k,1}^a - \delta_k^a = a^k \left(\left(1 - \frac{2(1-\frac{1}{2a})}{k}\right)^k - \frac{1}{2a}\left(1 - \frac{1-\frac{1}{2a}}{k-1}\right)^{k-1} \right)$$

for $k \ge 2$, we consider

$$B(x,k) \equiv \frac{\left(1-\frac{2(1-x)}{k}\right)^k}{x\left(1-\frac{1-x}{k-1}\right)^{k-1}} = \frac{\left(1-\frac{2(1-\frac{1}{2a})}{k}\right)^k}{\frac{1}{2a}\left(1-\frac{1-\frac{1}{2a}}{k-1}\right)^{k-1}}, \quad b(x,k) \equiv \ln xB(x,k) = \ln \frac{\left(1-\frac{2(1-x)}{k}\right)^k}{\left(1-\frac{1-x}{k-1}\right)^{k-1}},$$

using $\frac{1}{2} \le x \equiv \frac{1}{2a} < 1$. Then by Eq.(15) in Lemma 2.5, we have $b(x,k) = d(x,k,y)$ with $y = 2(1-x)$ and thus, by Part 1 of Lemma 2.5,

$$\frac{\partial b(x,k)}{\partial k} > 0 \text{ for } k > 2, \quad \text{and} \quad \frac{\partial b(x,k)}{\partial k} \ge 0 \text{ for } k \ge 2,$$

for all $\frac{1}{2} \le x < 1$. Thus, $b(x,k)$ and $B(x,k)$ are increasing with $k \ge 2$. Since $B(x,2) = 1$, we have $B(x,k) > 1$ for all $k \ge 3$ and for all $\frac{1}{2} < x < 1$ (i.e., $1 > a > \frac{1}{2}$). This completes the proof of $\gamma_{k,1}^a > \delta_k^a$ for all $k \ge 3$ and $\frac{1}{2} < a < 1$ ($1 > x > \frac{1}{2}$). It is easy to see that

$$\gamma_{k,2}^a = 1 - a^k\left(1-\frac{1}{k}\right)^k > \delta_k^a = 1 - a^k\left(1 - \frac{2-\frac{1}{a}}{k}\right)^k$$

for all $k \ge 2$ and $\frac{1}{2} < a < 1$, since $2(1-\frac{1}{2a}) < 1$ implies

$$1 - \left(1-\frac{1}{k}\right)^k > 1 - \left(1 - \frac{2(1-\frac{1}{2a})}{k}\right)^k.$$

Thus, $\gamma_k^a = \min\{\gamma_{k,1}^a, \gamma_{k,2}^a\} > \delta_k^a$ for all $k \ge 3$ and $\frac{1}{2} < a < 1$.

For $k = 1, 2$, it is clear that $\gamma_1^a = \delta_1^a = a$ and $\gamma_2^a = \delta_2^a = \frac{3}{4}$.

□

In the following theorem, we give conditions on when $\gamma_{k,1}^a$ dominates $\gamma_{k,2}^a$, which will be used to obtain improved approximation algorithms in the next section.

Theorem 2.6 Let $\alpha = \sqrt{2} - \frac{1}{2} \approx .914213$ and let β be the number satisfying $2\beta = e^{\frac{1}{2\beta}}$ ($\beta \approx .881611$). Then, the following statements hold.

1. For $\frac{3}{4} \le a \le \alpha$ and $k \ge 3$, $\gamma_{k,1}^a$ $\gamma_{k,2}^a$ and $\gamma_k^a = \min\{\gamma_{k,1}^a, \gamma_{k,2}^a\}$ are all decreasing with a and increasing with k. Furthermore, $\gamma_k^a \ge \frac{3}{4}$ for $\frac{3}{4} \le a \le \alpha$ and $k \ge 1$.

2. For $\frac{3}{4} \le a \le \beta$, $\gamma^a_{k,1} \le \gamma^a_{k,2}$ for all $k \ge 2$ and thus, $\gamma^a_k = \gamma^a_{k,1}$.

3. For $\beta \le a \le \alpha$, there is a unique integer k_a such that

$$\gamma^a_{k,1} \le \gamma^a_{k,2} \quad \text{for all } 2 \le k \le k_a \quad \text{and} \quad \gamma^a_{k,1} > \gamma^a_{k,2} \quad \text{for all } k > k_a.$$

   In particular, $k_\alpha = 7$ for $\alpha = \sqrt{2} - \frac{1}{2}$. Thus,

$$\gamma^\alpha_{k,1} < \gamma^\alpha_{k,2} \quad \text{for all } 2 \le k \le 7 \quad \text{and} \quad \gamma^\alpha_{k,1} > \gamma^\alpha_{k,2} \quad \text{for all } k \ge 8.$$

Proof: We will first prove Part 1. Since

$$\gamma^a_{k,1} = 1 - \frac{1}{2}a^{k-1}\left(1 - \frac{1-\frac{1}{2a}}{k-1}\right)^{k-1} = 1 - \frac{1}{2}\left(\frac{2(k-2)a+1}{2(k-1)}\right)^{k-1}, \quad \gamma^a_{k,2} = 1 - a^k\left(1-\frac{1}{k}\right)^k,$$

both $\gamma^a_{k,1}$ and $\gamma^a_{k,2}$ are decreasing with a. Similarly, for $\frac{3}{4} \le a \le \alpha$,

$$1 - \gamma^a_{k,1} = \frac{1}{2}a^{k-1}\left(1 - \frac{1-\frac{1}{2a}}{k-1}\right)^{k-1} \qquad \text{and} \qquad 1 - \gamma^a_{k,2} = a^k\left(1 - \frac{1}{k}\right)^k$$

are decreasing ($\gamma^a_{k,1}$ and $\gamma^a_{k,2}$ are increasing) with $k \ge 3$, which can be shown as follows. Consider

$$C_1(x,k) \equiv \frac{1-\gamma^a_{k+1,1}}{1-\gamma^a_{k,1}} = \frac{a\left(1 - \frac{1-\frac{1}{2a}}{k}\right)^k}{\left(1 - \frac{1-\frac{1}{2a}}{k-1}\right)^{k-1}} = \frac{\left(1 - \frac{1-x}{k}\right)^k}{2x\left(1 - \frac{1-x}{k-1}\right)^{k-1}}$$

using $\frac{1}{2} < x \equiv \frac{1}{2a} < 1$. Let

$$c_1(x,k) = \ln 2xC_1(x,k) = \ln \frac{\left(1 - \frac{1-x}{k}\right)^k}{\left(1 - \frac{1-x}{k-1}\right)^{k-1}}.$$

Then by using $y = 1 - x$ satisfying $0 < y \le 1 - x \le 1$ in Part 2 of Lemma 2.5, we have $c_1(x,k)$ is decreasing with $k \ge 2$ and so is $C_1(x,k)$. Thus, for all $\frac{3}{4} \le a = \frac{1}{2x} \le \alpha = \sqrt{2} - \frac{1}{2}$ ($\frac{2}{3} \ge x \ge \frac{1}{2\alpha} = \frac{2\sqrt{2}+1}{7} \approx .546918$), we have

$$C_1(x,k) < C_1(x,2) = \frac{(1+x)^2}{8x^2} = \frac{(2a+1)^2}{8} \le \frac{(2\alpha+1)^2}{8} = 1$$

and

$$\gamma^a_{k+1,1} > \gamma^a_{k,1} \quad (1 - \gamma^a_{k+1,1} < 1 - \gamma^a_{k,1})$$

for $k \ge 3$.

Similarly, for

$$C_2(k) \equiv \frac{1-\gamma^a_{k+1,2}}{1-\gamma^a_{k,2}} = \frac{a\left(1-\frac{1}{k+1}\right)^{k+1}}{\left(1-\frac{1}{k}\right)^k} \quad \text{and} \quad c_2(k) \equiv \ln\frac{C_2(k)}{a} = \ln\frac{\left(1-\frac{1}{k+1}\right)^{k+1}}{\left(1-\frac{1}{k}\right)^k}$$

with $x = 0$ and $y = 1$ satisfying $0 < y \le 1 - x \le 1$ in Part 2 of Lemma 2.5, we have $c_2(k)$ is decreasing with $k \ge 1$ and so is $C_2(k)$. Thus, for all $\frac{3}{4} \le a \le \alpha = \sqrt{2} - \frac{1}{2}$, we have

$$C_2(k) \le C_2(3) = \frac{3^7 a}{2^{11}} \le \frac{3^7 \alpha}{2^{11}} \approx 0.976262 < 1$$

and

$$\gamma^a_{k+1,2} > \gamma^a_{k,2} \quad (1 - \gamma^a_{k+1,2} < 1 - \gamma^a_{k,2})$$

for $k \ge 3$.

By the argument above, we have $\gamma^a_k = \min\{\gamma^a_{k,1}, \gamma^a_{k,2}\}$ is decreasing with a for all $\frac{3}{4} \le a \le \alpha$ and increasing with $k \ge 3$. Thus, $\gamma^a_k \ge \gamma^a_3 \ge \frac{3}{4}$ for all $\frac{3}{4} \le a \le \alpha$ and $k \ge 3$ since γ^a_k is increasing with $k \ge 3$ and $\gamma^a_3 = \gamma^a_{3,1} = 1 - \frac{(2a+1)^2}{32} \ge \frac{3}{4}$ by $\gamma^a_{3,1} = 1 - \frac{(2a+1)^2}{32}$ and $\gamma^a_{3,2} = 1 - \frac{8a^3}{27}$. It is clear that $\gamma^a_1 = a \ge \frac{3}{4}$ and $\gamma^a_2 = \frac{3}{4}$ by $\gamma^a_{2,1} = \frac{3}{4}$ and $\gamma^a_{2,2} = 1 - \frac{a^2}{4}$. This completes the proof of Part 1.

Next we prove Part 2. Since

$$\gamma^a_{k,1} - \gamma^a_{k,2} = a^k \left(1 - \frac{1}{k}\right)^k - \frac{1}{2} a^{k-1} \left(1 - \frac{1 - \frac{1}{2a}}{k-1}\right)^{k-1} \tag{19}$$

we consider

$$D(x, k) \equiv \frac{(1 - \frac{1}{k})^k}{x\left(1 - \frac{1-x}{k-1}\right)^{k-1}} = \frac{(1 - \frac{1}{k})^k}{\frac{1}{2a}\left(1 - \frac{1-\frac{1}{2a}}{k-1}\right)^{k-1}} \tag{20}$$

using $\frac{1}{2} \le x \equiv \frac{1}{2a} < 1$. Let

$$d(x, k) = \ln D(x, k) = \ln \frac{(1 - \frac{1}{k})^k}{x\left(1 - \frac{1-x}{k-1}\right)^{k-1}}. \tag{21}$$

Then $d(x, k) = d(x, k, 1) - \ln x$, where $d(x, k, y)$ is the function defined by Eq.(15) and $y = 1$ satisfies the condition $0 < 2(1-x) \le y \le 1$ in Part 1 of Lemma 2.5. Thus, $d(x, k) = d(x, k, 1) - \ln x$ is increasing with $k \ge 2$ for all $\frac{1}{2} \le x < 1$. Furthermore, it is clear that $d(x, k)$ is decreasing with x, since

$$\frac{\partial d(x, k)}{\partial x} = -\frac{1}{x} - \frac{k-1}{k-2+x} < 0 \quad \text{for all } \tfrac{1}{2} \le x \le 1 \text{ and } k \ge 2.$$

From the above argument, for a fixed $\frac{1}{2} \le x < 1$, if

$$d(x, \infty) = \lim_{k\to\infty} \ln \frac{(1 - \frac{1}{k})^k}{x\left(1 - \frac{1-x}{k-1}\right)^{k-1}} = \ln \frac{e^{-1}}{xe^{-1+x}} = \ln \frac{1}{xe^x} \le 0,$$

then $d(x, k) \le 0$ for all $k \ge 2$, since $d(x, k)$ is increasing with $k \ge 2$. Let x_β be a number satisfying

$$x_\beta e^{x_\beta} = 1. \tag{22}$$

Then x_β is unique ($x_\beta = \frac{1}{2\beta} \approx .567143$) and $d(x_\beta, \infty) = 0$, and thus, $d(x, k) < 0$ for all $k \ge 2$ and $x > x_\beta$ (i.e., $xe^x > 1$), since $d(x, k)$ is increasing with k and decreasing with x. Similarly, $d(x_\beta, k) < 0$ for all finite $k \ge 2$. Thus, by Eqs.(20) and (21),

$$D(x, k) = D(\frac{1}{2a}, k) < 1 \quad \text{for all (finite) } k \ge 2 \text{ and } \frac{1}{2} < a = \frac{1}{2x} \le \frac{1}{2x_\beta} = \beta \approx .881611.$$

By Eq.(19), this implies that

$$\gamma_{k,1}^{a} - \gamma_{k,2}^{a} < 0 \quad \text{for all (finite) } k \geq 2 \text{ and } \frac{1}{2} < a \leq \beta \approx .881611$$

and Part 2 is proved.

The proof of Part 3 is almost similar. For $\frac{1}{2} \leq x < x_\beta$, there is an integer k_x such that

$$d(x,k) \leq 0 \quad \text{for all } 2 \leq k \leq k_x \quad \text{and} \quad d(x,k) > 0 \quad \text{for all } k > k_x,$$

since $d(x,k)$ is increasing with k and

$$d(x,2) = \ln \frac{(1-\frac{1}{2})^2}{x\left(1 - \frac{1-x}{2-1}\right)^{2-1}} = \ln \frac{1}{4x^2} \leq 0 \quad \text{and} \quad d(x,\infty) = \ln \frac{1}{xe^x} > 0.$$

Thus, for $a = \frac{1}{2x} > \frac{1}{2x_\beta} = \beta \approx .881611$, there is a unique integer k_a such that

$$D(x,k) = D(\frac{1}{2a}, k) \leq 1 \quad \text{for all } 2 \leq k \leq k_a \quad \text{and} \quad D(x,k) = D(\frac{1}{2a}, k) > 1 \quad \text{for all } k > k_a.$$

By Eq.(19), this implies that, for $a = \frac{1}{2x} > \beta$ there is a unique integer k_a such that

$$\gamma_{k,1}^{a} - \gamma_{k,2}^{a} \leq 0 \quad \text{for all } 2 \leq k \leq k_a \quad \text{and} \quad \gamma_{k,1}^{a} - \gamma_{k,2}^{a} > 0 \quad \text{for all } k > k_a.$$

For $\alpha = \sqrt{2} - \frac{1}{2} \approx .914213 > \beta = .881611$, it is easy to see $k_\alpha = 7$, that is

$$\gamma_{7,1}^{\alpha} < \gamma_{7,2}^{\alpha} \quad \text{and} \quad \gamma_{8,1}^{\alpha} > \gamma_{8,2}^{\alpha}.$$

Thus,

$$\gamma_{k,1}^{\alpha} < \gamma_{k,2}^{\alpha} \quad \text{for all } 2 \leq k \leq 7 \quad \text{and} \quad \gamma_{k,1}^{\alpha} > \gamma_{k,2}^{\alpha} \quad \text{for all } k \geq 8.$$

□

As a corollary of Theorems 2.1 and 2.6, we obtain a new family of 3/4-approximation algorithms. Furthermore, by using $a = \frac{3}{4}$ and $a = \alpha = \sqrt{2} - \frac{1}{2} \approx .914213$, we have the following results which will be used to obtain improved approximation algorithms in the next section.

Corollary 2.7 Let $\alpha = \sqrt{2} - \frac{1}{2} \approx .914213$. Then, for any a with $\frac{3}{4} \leq a \leq \alpha$, the expected value $F(f_1^a(\boldsymbol{y}^*))$ of the random truth assignment $\boldsymbol{x}^p = f_1^a(\boldsymbol{y}^*)$ satisfies $F(f_1^a(\boldsymbol{y}^*)) \geq \frac{3}{4}\sum_{k\geq 1} W_k^*$. In particular, the following statements hold for $a = \frac{3}{4}$ and $a = \alpha$.

1. The expected value $F(f_1^{\frac{3}{4}}(\boldsymbol{y}^*))$ of the random truth assignment $\boldsymbol{x}^p = f_1^{\frac{3}{4}}(\boldsymbol{y}^*)$ satisfies

$$\begin{aligned} F(f_1^{\frac{3}{4}}(\boldsymbol{y}^*)) &\geq \sum_{k\geq 1} \gamma_k^{\frac{3}{4}} W_k^* \\ &= .75W_1^* + .75W_2^* + .804W_3^* + .851W_4^* + .888W_5^* + .915W_6^* \\ &\quad +.936W_7^* + .952W_8^* + .964W_9^* + .973W_{10}^* + \sum_{k\geq 11} \gamma_{k,1}^{\frac{3}{4}} W_k^*. \end{aligned}$$

2. The expected value $F(f_1^{\alpha}(\boldsymbol{y}^*))$ of the random truth assignment $\boldsymbol{x}^p = f_1^{\alpha}(\boldsymbol{y}^*)$ satisfies

$$\begin{aligned} F(f_1^{\alpha}(\boldsymbol{y}^*)) &\geq \sum_{k\geq 1} \gamma_k^{\alpha} W_k^* \\ &= \alpha W_1^* + .75W_2^* + .75W_3^* + .766W_4^* + .784W_5^* + .801W_6^* \\ &\quad +.817W_7^* + .832W_8^* + .845W_9^* + .857W_{10}^* + \sum_{k\geq 11} \gamma_{k,2}^{\alpha} W_k^*. \end{aligned}$$

3 The Improved Approximation Algorithms

In this section, we give our improved appproximation algorithms for MAX SAT based on a hybrid approach. We use a semidefinite programming relaxation of MAX SAT which is a combination of ones given by Goemans and Williamson [7], Feige and Goemans [4], Karloff and Zwick [11], and Zwick [16]. We give some notation needed to describe the formulation, and the formulation itself, in Section 3.1. We then give our algorithm in Section 3.2. We will use a combination of the MAX SAT algorithm in the previous section which uses the function $f_1^{3/4}$, the Feige-Goemans MAX 2SAT algorithm, the Karloff-Zwick MAX 3SAT algorithm to obtain our .7846-approximation algorithm for MAX SAT. Finally, if we use the MAX SAT algorithm of the previous section f_1^a for $a = \sqrt{2} - \frac{1}{2}$ and combine it with Zwick's algorithm, we obtain a performance guarantee of .8331, assuming the correctness of Zwick's conjecture. The analysis of these algorithms is presented in Section 3.3.

3.1 The MAX SAT relaxation

We begin by defining some notation we will need to describe our semidefinite programming relaxation of the MAX SAT problem. For a clause $C_j = x_{j_1} \vee x_{j_2} \vee \cdots \vee x_{j_{k_j}}$ with $k_j \geq 3$, let P_j be the set of all possible clauses C with two literals in C_j (e.g., if $C_j = x \vee y \vee z$ then $P_j = \{x \vee y, x \vee z, y \vee z\}$). Similarly, for $k_j \geq 4$, let Q_j be the set of all possible clauses C with three literals in C_j (e.g., if $C_j = x \vee y \vee z \vee u$ then $Q_j = \{x \vee y \vee z, x \vee y \vee u, x \vee z \vee u, y \vee z \vee u\}$). For simplicity, we use $P_j = C_j$ if C_j is a clause with one or two literals and $Q_j = C_j$ if C_j is a clause of three literals.

To give the semidefinite programming relaxation, we follow [7] by introducing variables $\boldsymbol{y} = (y_0, y_1, \ldots, y_{2n})$ corresponding to

$$y_0 y_i \equiv 2x_i - 1 \text{ with } |y_0| = |y_i| = 1 \text{ and } y_{n+i} = -y_i.$$

Thus, x_i becomes $\frac{1+y_0y_i}{2}$ and a clause $C_j = x_{j_1} \vee x_{j_2} \in \mathcal{C}_2$ can be considered to be a function on $\boldsymbol{y} = (y_0, y_1, \ldots, y_{2n})$ as follows:

$$\begin{aligned} C_j = C_j(\boldsymbol{y}) &= 1 - \frac{1 - y_0 y_{j_1}}{2} \frac{1 - y_0 y_{j_2}}{2} \\ &= \frac{3 + y_0 y_{j_1} + y_0 y_{j_2} - y_{j_1} y_{j_2}}{4}. \end{aligned}$$

We also introduce variables $\boldsymbol{z}$ for clauses corresponding to $z_j = C_j(\boldsymbol{y})$. We relax this formulation to a "vector programming" problem using $(2n+1)$-dimensional vectors $\boldsymbol{v}_i$ with norm $\|\boldsymbol{v}_i\| = 1$ (i.e., $\boldsymbol{v}_i \in S^{2n}$) and $\boldsymbol{v}_{n+i} = -\boldsymbol{v}_i$ corresponding to y_i with $|y_i| = 1$ and $y_{n+i} = -y_i$. We replace each $y_{i_1} y_{i_2}$ with an inner vector product $\boldsymbol{v}_{i_1} \boldsymbol{v}_{i_2}$ and set $y_{i_1 i_2} = \boldsymbol{v}_{i_1} \boldsymbol{v}_{i_2}$.

We need to use constraints on the products $\boldsymbol{v}_i \boldsymbol{v}_j$ and the variables z_j as given in [7, 4, 11, 16]. While we can give these explicitly for the cases of [7, 4, 11], the number of constraints in [16] becomes too large to give explicitly (although still polynomially sized). Thus we will write the latter constraints as $Canon(\boldsymbol{v}, \boldsymbol{z}) \geq \boldsymbol{b}$. For the other constraints, let $C = x_{i_1} \vee \cdots \vee x_{i_k}$ be a

$$
\begin{array}{lll}
\max & \displaystyle\sum_{C_j \in \mathcal{C}} w(C_j) z_j & \\
\text{s.t.} & \displaystyle\sum_{i=1}^{k_j} \frac{1+\boldsymbol{v}_0\boldsymbol{v}_{j_i}}{2} \geq z_j \quad \forall C_j = x_{j_1} \vee x_{j_2} \vee \cdots \vee x_{j_{k_j}} \in \mathcal{C} & (23) \\
& \displaystyle\frac{1+\boldsymbol{v}_0\boldsymbol{v}_{j_i}}{2} \leq z_j \quad \forall 1 \leq i \leq k_j, \forall C_j = x_{j_1} \vee x_{j_2} \vee \cdots \vee x_{j_{k_j}} \in \mathcal{C} & (24) \\
& \displaystyle\frac{1}{k-1} \sum_{C \in P_j} relax(C) \geq z_j \quad \forall C_j \in \mathcal{C}_k, \quad \forall k \geq 2 & (25) \\
(AW) & \displaystyle\frac{1}{\binom{k-1}{2}} \sum_{C \in Q_j} relax(C) \geq z_j \quad \forall C_j \in \mathcal{C}_k, \quad \forall k \geq 3 & (26) \\
& fg(i_1, i_2, i_3) \geq -1 \quad \forall i_1, i_2, i_3 \text{ with } 0 \leq i_1 < i_2 < i_3 \leq n & (27) \\
& Canon(\boldsymbol{v}, \boldsymbol{z}) \geq \boldsymbol{b} & (28) \\
& \boldsymbol{v}_i \in S^n \quad 0 \leq \forall i \leq 2n & (29) \\
& \boldsymbol{v}_{n+i} = -\boldsymbol{v}_i \quad 1 \leq \forall i \leq n & (30) \\
& 0 \leq z_j \leq 1 \quad \forall C_j \in \mathcal{C}. & (31)
\end{array}
$$

Figure 2: The semidefinite relaxation (AW).

clause with $k \leq 3$ literals. Define

$$
relax(C) = \begin{cases}
\frac{1+\boldsymbol{v}_0\boldsymbol{v}_{i_1}}{2} & \text{if } k=1 \\
\frac{4-(-\boldsymbol{v}_0+\boldsymbol{v}_{i_1})(-\boldsymbol{v}_0+\boldsymbol{v}_{i_2})}{4} & \text{if } k=2 \\
\min\{\frac{4-(-\boldsymbol{v}_0+\boldsymbol{v}_{i_1})(\boldsymbol{v}_{i_2}+\boldsymbol{v}_{i_3})}{4}, & \\
\quad \frac{4-(-\boldsymbol{v}_0+\boldsymbol{v}_{i_2})(\boldsymbol{v}_{i_3}+\boldsymbol{v}_{i_1})}{4}, & \\
\quad \frac{4-(-\boldsymbol{v}_0+\boldsymbol{v}_{i_3})(\boldsymbol{v}_{i_1}+\boldsymbol{v}_{i_2})}{4}\} & \text{if } k=3
\end{cases}
$$

as Karloff and Zwick did [11]. They show that for a clause C_j, $relax(C_j) \geq z_j$ is a valid constraint. Finally, define

$$
fg(i_1, i_2, i_3) = \min \left\{ \begin{array}{l}
\boldsymbol{v}_{i_1}\boldsymbol{v}_{i_2} + \boldsymbol{v}_{i_1}\boldsymbol{v}_{i_3} + \boldsymbol{v}_{i_2}\boldsymbol{v}_{i_3}, \\
-\boldsymbol{v}_{i_1}\boldsymbol{v}_{i_2} + \boldsymbol{v}_{i_1}\boldsymbol{v}_{i_3} - \boldsymbol{v}_{i_2}\boldsymbol{v}_{i_3}, \\
-\boldsymbol{v}_{i_1}\boldsymbol{v}_{i_2} - \boldsymbol{v}_{i_1}\boldsymbol{v}_{i_3} + \boldsymbol{v}_{i_2}\boldsymbol{v}_{i_3}, \\
\boldsymbol{v}_{i_1}\boldsymbol{v}_{i_2} - \boldsymbol{v}_{i_1}\boldsymbol{v}_{i_3} - \boldsymbol{v}_{i_2}\boldsymbol{v}_{i_3}.
\end{array} \right\}
$$

Feige and Goemans [4] show that $fg(i_1, i_2, i_3) \geq -1$ is a valid constraint for all $0 \leq i_1 < i_2 < i_3 \leq n$.

Now we are ready to formulate MAX SAT as a vector programming problem. The formulation (AW) is shown in Figure 2. It can be considered to be a semidefinite programming problem in the standard way (see [7], for example). Let $V = (\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{2n})$ and $Y = V^T V$, so that $y_{ij} = \boldsymbol{v}_i\boldsymbol{v}_j$. Then, the matrix $Y = (y_{i_1 i_2})$ is symmetric and positive semidefinite and $y_{ii} = 1$ $(i = 0, 1, ..., 2n)$. Note that $y_{n+i_1, n+i_2} = y_{i_1 i_2}$ and $y_{n+i_1, i_2} = y_{i_1, n+i_2} = -y_{i_1 i_2}$ $(1 \leq i_1, i_2 \leq n)$. From now on we will not distinguish between semidefinite programming and vector programming.

We now argue that (AW) is a valid relaxation of the MAX SAT problem. The first set of constraints (23) corresponds to the set of LP relaxation constraints and implies that, if $C_j = 1$

(i.e., $z_j = 1$) then one of the literals in C_j is true. Thus, they hold for any truth assignment $\boldsymbol{x}$ if we identify $\boldsymbol{x}$ with $\boldsymbol{y}$ such that $y_0 = (1, 0, ..., 0)$, and $y_i = (1, 0, ..., 0)$ if $x_i = 1$ and $y_i = (-1, 0, ..., 0)$ if $x_i = 0$. Similarly, the set of constraints (24) implies that $z_j = 1$ if one of the literals in C_j is true, and thus they hold for any truth assignment. The next two sets of constraints (25) and (26) represent a relaxation of each MAX SAT clause by sets of MAX 2SAT and MAX 3SAT clauses, respectively. If any literal in $C_j \in \mathcal{C}_k$ is set true, then at least $k-1$ length 2 clauses in P_j are true and at least $\binom{k-1}{2}$ length 3 clauses in Q_j are true. Thus, the constraints hold for any truth assignment $\boldsymbol{x}$. The other constraints also hold for any truth assignment as was shown in the papers in which they were defined. Thus, (AW) can be considered to be a relaxation of MAX SAT, which is a combination of an LP relaxation, and MAX 2SAT, MAX 3SAT, and MAX SAT relaxations based on semidefinite programming.

3.2 The algorithm

Our algorithm for MAX SAT is to find an optimal solution $(V^*, \boldsymbol{z}^*)$ to the SDP (AW), and use it to produce four different solutions, as described below. We then output the best solution of the four. Let $V^* = (\boldsymbol{v}_0^*, \boldsymbol{v}_1^*, \ldots, \boldsymbol{v}_{2n}^*)$. We run the following four polynomial-time algorithms.

Algorithm (1) Use a random truth assignment $\boldsymbol{x}^p = (x_i^p)$ with $x_i^p = f_1^a(\frac{1+\boldsymbol{v}_0^*\boldsymbol{v}_i^*}{2})$ corresponding to LP relaxation of Goemans-Williamson [6], where f_1^a is the function defined in Eq.(4) for a specified value of a.

Algorithm (2) Use a random truth assignment corresponding to MAX 2SAT algorithm given by Feige and Goemans [4] as follows. Let $U^* = (\boldsymbol{u}_0^*, \boldsymbol{u}_1^*, \ldots, \boldsymbol{u}_{2n}^*)$ be obtained from $V^* = (\boldsymbol{v}_0^*, \boldsymbol{v}_1^*, \ldots, \boldsymbol{v}_{2n}^*)$ by slightly rotating $V^* = (\boldsymbol{v}_0^*, \boldsymbol{v}_1^*, \ldots, \boldsymbol{v}_{2n}^*)$. More specifically, let $\boldsymbol{u}_0^* = \boldsymbol{v}_0^*$ and let $\boldsymbol{u}_i^*$ be the vector, coplanar with $\boldsymbol{v}_0^*$ and $\boldsymbol{v}_i^*$, on the same side of $\boldsymbol{v}_0^*$ as $\boldsymbol{v}_i^*$ is, and which forms an angle with $\boldsymbol{v}_0^*$ equal to $f(\theta_i)$ for some function f, where θ_i is the angle between $\boldsymbol{v}_0^*$ and $\boldsymbol{v}_i^*$. Feige and Goemans use the function

$$f(\theta) = \theta + .806765\Big[\frac{\pi}{2}(1 - \cos\theta) - \theta\Big].$$

Note that $f(\pi - \theta) = \pi - f(\theta)$ and thus, $\boldsymbol{u}_{n+i}^* = -\boldsymbol{u}_i^*$ for each i, $1 \le i \le n$. Then, take a random $(2n+1)$-dimensional unit vector $\boldsymbol{r}$ uniformly from the unit sphere and set x_i to be true if and only if sgn$(\boldsymbol{u}_i^* \cdot \boldsymbol{r})$=sgn$(\boldsymbol{u}_0^* \cdot \boldsymbol{r})$.

Algorithm (3) Use a random truth assignment obtained by using $f(\theta) = \theta$ in Algorithm (2). This random truth assignment was originally proposed by Goemans and Williamson [5] and it corresponds to MAX 3SAT algorithm given by Karloff and Zwick [11].

Algorithm (4) Use a random truth assignment corresponding to the MAX SAT algorithm given by Zwick [16]. Let $V = (\boldsymbol{v}_0^*, \boldsymbol{v}_1^*, \ldots, \boldsymbol{v}_{2n}^*)$ and $Y = V^T V$. Then set $Y' = (\cos^2\beta)Y + (\sin^2\beta)I$ for $\beta = .4555$, where I is the $(2n+1)\times(2n+1)$ identity matrix. Let $Y' = U^T U$, for $U = (\boldsymbol{u}_0^*, \boldsymbol{u}_1^*, \ldots, \boldsymbol{u}_{2n}^*)$. Obtain the random truth assignment from the $\boldsymbol{u}_i^*$ as in Algorithm (3).

3.3 Analysis

We now turn to the proof of our main theorem.

Theorem 3.1 Let $F(V^*, z^*)$ denote the expected value of the algorithm above, with $a = 3/4$ for Algorithm (1). Then

$$F(V^*, z^*) \geq .7846 \sum_{k \geq 1} W_k^*, \tag{32}$$

where $W_k^* = \sum_{C_j \in \mathcal{C}_k} w(C_j) z_j^*$. Let $F'(V^*, z^*)$ denote the expected value of a modified version of the algorithm in which $a = \alpha = \sqrt{2} - 1/2$ in Algorithm (1). Then if Zwick's algorithm performs as conjectured

$$F'(V^*, z^*) \geq .8331 \sum_{k \geq 1} W_k^*. \tag{33}$$

This implies that we have a .7846-approximation algorithm for MAX SAT. Furthermore, if Zwick's algorithm behaves as conjectured, we obtain a .8331-approximation algorithm for MAX SAT.

Proof of Theorem 3.1: The expected value of the solution returned by the algorithm above is at least as good as the expected value of an algorithm that uses Algorithm (i) with probability p_i, where $p_1 + p_2 + p_3 + p_4 = 1$. From the arguments in Section 2, the probability that a clause $C_j \in \mathcal{C}_k$ is satisfied by Algorithm (1) is at least $\gamma_k^a z_j$, where γ_k^a is defined in Eq. (6). Similarly, from the arguments in [5, 7, 4], the probability that a clause $C_j \in \mathcal{C}_k$ is satisfied by Algorithm (2) is at least

$$\begin{aligned} .93109 \cdot \frac{1}{\binom{k}{2}} \sum_{C \in P_j} relax(C) &\geq .93109 \cdot \frac{2}{k} \frac{1}{k-1} \sum_{C \in P_j} relax(C) \\ &\geq .93109 \cdot \frac{2}{k} z_j^* \quad \text{for } k \geq 2, \end{aligned}$$

and at least $.97653 z_j^*$ for $k = 1$. By an analysis obtained by Karloff and Zwick [11] and an argument similar to one in [5, 7], we have that the probability that a clause $C_j \in \mathcal{C}_k$ is satisfied by Algorithm (3) is at least

$$\begin{aligned} \frac{7}{8} \cdot \frac{1}{\binom{k}{3}} \sum_{C \in Q_j} relax(C) &\geq \frac{7}{8} \cdot \frac{3}{k} \frac{1}{\binom{k-1}{2}} \sum_{C \in Q_j} relax(C) \\ &\geq \frac{3}{k} \frac{7}{8} z_j^* \qquad \text{for } k \geq 3, \end{aligned}$$

and at least $.87856 z_j^*$ for $k = 1, 2$.

Zwick [16, 17] conjectures that the worst-case ratio of the value of a certain semidefinite programming bound to the performance of a certain algorithm is attained at a certain point; if the conjecture is true, he obtains an analytic expression for the probability that a clause $C_j \in \mathcal{C}_k$ is satisfied by Algorithm (4). This analytic expression is quite complicated, and we do not repeat it here. Instead, we use the fact that the probability that a clause $C_j \in \mathcal{C}_k$ is satisfied by Algorithm (4) is at least $\alpha_k z_j^*$, where $\alpha_k \geq .7977$; the values of α_k for $k = 1, \ldots, 39$ were obtained from Zwick [17], who computed the value of the analytic expression at these points. We list these values in Appendix A. Importantly, $\alpha_1 = \alpha_{35} = .7977$ and α_k is increasing for $k > 35$ [17].

We now analyze what happens if we use Algorithm (i) with probability p_i. The expected value of this algorithm is at least $(\gamma_1^a p_1 + .97653 p_2 + .87856 p_3 + .7977 p_4) w_j z_j^*$ on clauses $C_j \in \mathcal{C}_1$. Similarly, the value of the algorithm is at least $(\gamma_2^a p_1 + .93109 p_2 + .87856 p_3 + .87995 p_4) w_j z_j^*$ on clauses $C_j \in \mathcal{C}_2$, $(\gamma_3^a p_1 + .93109 \frac{2}{3} p_2 + \frac{7}{8} p_3 + \frac{7}{8} p_4) w_j z_j^*$ on clauses $C_j \in \mathcal{C}_3$, and $(\gamma_k^a p_1 + .93109 \frac{2}{k} p_2 + \frac{7}{8} \cdot \frac{3}{k} p_3 + \alpha_k p_4) w_j z_j^*$ on clauses $C_j \in \mathcal{C}_k$ for $k \geq 4$. If we can show that this quantity is at least $\beta w(C_j)z_j^*$ for all clauses C_j, then we obtain an approximation algorithm with performance guarantee β since then the expected value of the solution is at least $\beta \sum_{C_j \in \mathcal{C}} w(C_j)z_j^* \geq \beta W^* \geq \beta \hat{W}$.

Suppose first that we do not assume Zwick's conjectuire, and we set $a = 3/4$ (that is, we use the function $f_1^{3/4}$ in Algorithm (1)). If we set $p_4 = 0$, $p_1 = .7846081$, $p_2 = .1316834$, and $p_3 = 1 - (p_1 + p_2) = .0837085$, then we have

$$\begin{aligned} \frac{3}{4}p_1 + 0.97653p_2 + 0.87856p_3 &\geq \frac{3}{4}p_1 + 0.93109p_2 + 0.87856p_3 \\ &\geq p_1 \geq 0.7846 \quad \text{for } k = 1, 2, \end{aligned}$$

and for $k \geq 3$,

$$\gamma_k^{3/4} p_1 + \frac{2 \times .93109}{k} p_2 + \frac{3}{k}\frac{7}{8} p_3 \geq p_1 \geq .7846.$$

This can be proved by the following observation. Since, for $k \geq 3$,

$$k(1 - \gamma_k^{3/4}) = k \left(\frac{1}{2} \left(\frac{3}{4} \right)^{k-1} \left(1 - \frac{1}{3(k-1)} \right)^{k-1} \right)$$

takes the maximum value for $k = 4$, we have

$$\frac{(2 \times .93109)p_2 + \frac{21}{8}p_3}{k(1 - \gamma_k^{3/4})} \geq \frac{(2 \times .93109)p_2 + \frac{21}{8}p_3}{4(1 - \gamma_4^{3/4})} \geq p_1.$$

This implies the inequality (32), and thus the algorithm has performance guarantee .7846.

We now prove inequality (33). Suppose we use all four algorithms, and we set $a = \sqrt{2} - \frac{1}{2}$. We attempt to determine the best weighting of the algorithms by using a linear program, in which the probabilities p_i and the performance guarantee β are variables, and there is a constraint for every clause size from 1 to 39 (e.g., for clause size 1, the constraint is $.91421p_1 + .97653p_2 + .87856p_3 + .79778p_4 \geq \beta$). We attempt to maximize β subject to these constraints, and obtain the solution $p_1 = .303636$, $p_4 = .696364$, $\beta = .8331$, $p_2 = p_3 = 0$. To verify the answer, one need only check that $p_1\gamma_k^a + p_4\alpha_k \geq .8331$ for k ranging from 1 to 35, since both α_k and γ_k^a are increasing for $k > 35$. $\square$

Since the algorithms in [9, 16] are parametrized, we expect that small improvements to our bounds can be made by altering some of these parameters.

Acknowledgements

The first author was supported in part by Grant in Aid for Scientific Research of the Ministry of Education, Science, Sports and Culture of Japan and The Institute of Science and Engineering, Chuo University. We thank Howard Karloff and Uri Zwick for useful comments, and Uri Zwick for calculating the bounds α_k of his algorithm for us.

References

[1] T. Asano, Approximation algorithms for MAX SAT: Yannakakis vs. Goemans-Williamson, In *Proc. 5th Israel Symposium on Theory of Computing and Systems*, pp. 24–37, 1997.

[2] T. Asano, T. Ono and T. Hirata, Approximation algorithms for the maximum satisfiability problem, *Nordic Journal of Computing*, 3, pp. 388–404, 1996.

[3] J. Chen, D. Friesen, and H. Zheng. Tight bound on Johnson's algorithm for MAX SAT. *Journal of Comput. and Sys. Sci.* 58, pp. 622–640, 1999.

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

[5] Michel X. Goemans and David P. Williamson, .878-approximation algorithms for MAX CUT and MAX 2SAT, In *Proc. 26th ACM Symposium on the Theory of Computating*, pp. 422–431, 1994.

[6] Michel X. Goemans and David P. Williamson, New 3/4-approximation algorithms for the maximum satisfiability problem, *SIAM Journal of Disc. Math.* 7, pp. 656–666, 1994.

[7] Michel X. Goemans and David P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, *Journal of the ACM* 42, pp. 1115–1145, 1995.

[8] Johan Håstad, Some optimal inapproximability results. In *Proc. 28th ACM Symposium on the Theory of Computing*, pp. 1–10, 1997.

[9] Erin Halperin and Uri Zwick, Approximation algorithms for MAX 4-SAT and rounding procedures for semidefinite programs. In Proc. 7th MPS Conference on Integer Programming and Combinatorial Optimization, LNCS 1610, pp. 202–217, 1999.

[10] David S. Johnson, Approximation algorithms for combinatorial problems, *Journal of Comput. and Sys. Sci.* 9, pp. 256–278, 1974.

[11] Howard Karloff and Uri Zwick, A 7/8-Approximation algorithm for MAX 3SAT?, In *Proc. 38th IEEE Symposium on the Foundations of Computer Science*, pp. 406–415, 1997.

[12] R. Kohli and R. Krishnamurti, Average performance of heuristics for satisfiability, *SIAM Journal of Disc. Math.* 2, pp. 508–523, 1989.

[13] K. Lieberherr and E. Specker, Complexity of partial satisfaction, *Journal of the ACM* 28, pp. 411–421, 1981.

[14] Luca Trevisan, Gregory B. Sorkin, Madhu Sudan and David P. Williamson, Gadgets, approximation, and linear programming, *SIAM Journal of Computing* 29, pp. 2074–2097, published electronically April 18, 2000.

[15] Mihalis Yannakakis, On the approximation of maximum satisfiability, *J. Algorithms* 17, pp. 475–502, 1994.

[16] Uri Zwick, Outward rotations: a tool for rounding solutions of semidefinite programming relaxations, with applications to MAX CUT and other problems. In *Proc. 31st ACM Symposium on the Theory of Computing*, pp. 679–687, 1999.

[17] Uri Zwick, Personal communication, 1999.

Appendix

A Performance of Zwick's Algorithm

Here we list the performance of Zwick's algorithm for various clause lengths [17].

k	α_k
1	.79777594
2	.87995339
3	.87500007
4	.86480993
5	.85523413
6	.84695541
7	.83991613
8	.83393899
9	.82884858
10	.82449552
11	.82075749
12	.81753539
13	.81474892
14	.81233263
15	.81023288
16	.80840537
17	.80681324
18	.80542565
19	.80421661
20	.80316408
21	.80224927
22	.80145609
23	.80077067
24	.80018100
25	.79967666
26	.79924856
27	.79888872
28	.79859015
29	.79834667
30	.79815282
31	.79800376
32	.79789519
33	.79782326
34	.79778453
35	.79777594
36	.79779472
37	.79783838
38	.79790468
39	.79799160