アルゴリズム 70−7
(1999. 11. 8)

MAX SATに対する新しい3/4-近似アルゴリズムとより高性能な近似アルゴリズム

浅野 孝夫
中央大学理工学部情報工学科
〒112-8551 文京区春日 1-13-27
asano@ise.chuo-u.ac.jp

David P. Williamson
IBM Thomas J. Watson Research Center
P.O. Box 218, Yorktown Heights, NY 10598
dpw@watson.ibm.com

MAX SAT(最大充足化問題)は，節の集合$\mathcal{C}$と各節$C \in \mathcal{C}$に対する非負の重み$w(C)$の対$(\mathcal{C}, w)$で入力が規定され，$X = \{x_1, x_2, \ldots, x_n\}$を$\mathcal{C}$の節に現れる変数の集合としたとき，各変数$x_i \in X$に対して，真偽割当をして，満たされる節の重みの総和を最大にする問題である．この論文では，Goemans-Williamsonによって提案されたLP緩和法に基づく近似アルゴリズムを検討し，より一般化した新しい3/4-近似アルゴリズムを与える．さらに，Feige-GoemansのMAX 2SATアルゴリズム，Karloff-ZwickのMAX 3SATアルゴリズムおよびZwickのMAX SATアルゴリズムと適切に組み合わせることにより，MAX SATに対するより高性能な近似アルゴリズムが得られることを示す．より具体的には，MAX 2SATとMAX 3SATアルゴリズムと組み合わせて0.7846-近似アルゴリズムが得られること，およびMAX SATアルゴリズムと組み合わせると現在最高性能の.7977の近似率を上回る.8331-近似アルゴリズムが得られることを示す．

A New Family of 3/4-Approximation Algorithms and Improved Approximation Algorithms for MAX SAT

Takao Asano
Chuo University
Kasuga, Bunkyo-ku, Tokyo 112-8551
asano@ise.chuo-u.ac.jp

David P. Williamson
IBM Thomas J. Watson Research Center
P.O. Box 218, Yorktown Heights, NY 10598
dpw@watson.ibm.com

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 new family of 3/4-approximation algorithms that generalize a previous algorithm of Goemans and Williamson. 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 beats Zwick's conjectured performance guarantee of .7977. The best previous result for MAX SAT without assuming Zwick's conjecture is a .770-approximation algorithm of Asano.

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

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. Shortly after, Goemans and Williamson proposed a .878-approximation algorithm for MAX 2SAT based on semidefinite programming [5]. They also showed that their .878-approximation algorithm, combined with their .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 suggested that it might lead 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. 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 new family of .75-approximation algorithms for MAX SAT which generalize an algorithm of Goemans and Williamson. These algorithms lead to better overall approximation algorithms 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]

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.

2 The MAX SAT Algorithms of Goemans and Williamson

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

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

[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				Yannakakis92 [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].

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.

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

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

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

where e is the base of natural logarithm, $\mathcal{C}_k$ denotes the set of clauses in $\mathcal{C}$ with k literals, $W^* = \sum_{C_j \in \mathcal{C}} w(C_j) z_j^*$ and $W_k^* = \sum_{C_j \in \mathcal{C}_k} w(C_j) z_j^*$ (note that $W^* = \sum_{C_j \in \mathcal{C}} w(C_j) z_j^* \ge \hat{W} = \sum_{C_j \in \mathcal{C}} w(C_j) \hat{z}_j$ 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 Johnson's algorithm, it leads to a $\frac{3}{4}$-approximation algorithm. Note that 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 $\sum_{k \ge 1} \beta_k W_k^*$ where β_k is defined as follows:

$$\beta_k = \frac{1}{2}\left(2 - \frac{1}{2^k} - \left(1 - \frac{1}{k}\right)^k\right). \quad (1)$$

To give an improved approximation algorithm for MAX SAT, we will sharpen the analysis of Goemans and Williamson to provide a more precise statement of the expected weight of satisfied clauses in $\mathcal{C}_k$ for each k. We first give a theorem showing the behavior of the algorithm using a parametrized function f_1^a defined as follows:

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

for $\frac{1}{2} \le a \le 1$.

Theorem 1 *The probability of $C_j = x_{j_1} \vee x_{j_2} \vee \cdots \vee x_{j_{k_j}} \in \mathcal{C}$ 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*

$$C_j(f_1^a(\boldsymbol{y}^*)) = 1 - \prod_{i=1}^{k_j} (1 - f_1^a(y_i^*)) \ge \gamma_{k_j}^a z_j^*, \quad (3)$$

for $\frac{1}{2} \le a \le 1$, where f_1^a is the function defined in Eq. (2) and

$$\gamma_k^a = \begin{cases} a & \text{if } k = 1 \\ \min\{\gamma_k^a(1), \gamma_k^a(2)\} & \text{if } k \ge 2, \end{cases} \quad (4)$$

where

$$\gamma_k^a(1) = 1 - \frac{1}{2}a^{k-1}\left(1 - \frac{1-\frac{1}{2a}}{k-1}\right)^{k-1}, \quad (5)$$

$$\gamma_k^a(2) = 1 - a^k\left(1 - \frac{1}{k}\right)^k. \quad (6)$$

Thus, the expected value $F(f_1^a(y^*))$ of the random truth assignment $x^p = f_1^a(y^*)$ satisfies

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

A corollary of this theorem is that we obtain a new family of 3/4-approximation algorithms.

Corollary 1 *Let $\alpha = \sqrt{2} - \frac{1}{2} \approx .9142$. Then, for $\frac{3}{4} \le a \le \alpha$,*

$$F(f_1^a(y^*)) \ge \frac{3}{4}\sum_k W_k^*.$$

To prove the main theorem, we use the following lemma.

Lemma 1 *For a fixed a with $\frac{1}{2} \le a \le 1$, let $\frac{1}{2a} \le y \le 1$. Then for $k \ge 2$,*

$$(1 - ay)\left(1 - \frac{1-y}{k-1}\right)^{k-1} \le \frac{1}{2}\left(1 - \frac{1-\frac{1}{2a}}{k-1}\right)^{k-1} \quad (7)$$

Furthermore, let ℓ be a positive integer and $0 \le b \le z \le 1$. Then, for a positive constant μ with $(b+1)\mu \le 1$,

$$1 - \mu\left(1 - \frac{z-b}{\ell}\right)^\ell \ge \left(1 - \mu\left(1 - \frac{1-b}{\ell}\right)^\ell\right) z. \quad (8)$$

Proof of Theorem 1: Now we are ready to prove (3). We assume $k_j = k$ and $x_{j_i} = x_i$ for each $i = 1, 2, ..., k$ by symmetry since $f_1^a(x) = 1 - f_1^a(\bar{x})$ and we can set $x := \bar{x}$ if necessary. Thus, we consider clause $C_j = x_1 \vee x_2 \vee \cdots \vee x_k$, which corresponds to the LP constraint

$$y_1^* + y_2^* + \cdots + y_k^* \ge z_j^*, \quad (9)$$

and we will show that

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

If $k = 1$, then $C_j(f_1^a(y^*)) = f_1^a(y_1^*) \ge a y_1^* \ge a z_j^* = \gamma_1^a z_j^*$ by (9). Thus, we assume $k \ge 2$ and $y_1^* \le y_2^* \le \cdots \le y_k^*$ by symmetry and consider three cases as follows. Case 1: $y_k^* \le 1 - \frac{1}{2a}$; Case 2: $y_{k-1}^* \le 1 - \frac{1}{2a} < y_k^*$; Case 3: $1 - \frac{1}{2a} < y_{k-1}^* \le y_k^*$.

Case 1: $y_k^* \le 1 - \frac{1}{2a}$. Since all $y_i^* \le 1 - \frac{1}{2a}$ ($i = 1, 2, ..., k$), 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

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

where the first inequality follows by the arithmetic/geometric mean inequality, the second by (9), the third by (8) with $\mu = a^k$, and $b = 0$.

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

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

by the arithmetic/geometric mean inequality and inequality (9). We now assume[3] that $y_k \le z_j^*$. If $y_k^* \le \frac{1}{2a}$, then $1 - f_1^a(y_k^*) = \frac{1}{2}$ and

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

[3] In an optimal solution to the LP (GW), $z_j^* = \min\{1, \sum_{i=1}^k y_i^*\}$ and $y_k^* \le 1$, 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.

$$\geq \left(1 - \frac{1}{2}a^{k-1}\left(1 - \frac{1-y_k^*}{k-1}\right)^{k-1}\right) z_j^*$$

$$\geq \left(1 - \frac{1}{2}a^{k-1}\left(1 - \frac{1-\frac{1}{2a}}{k-1}\right)^{k-1}\right) z_j^*$$

$$= \gamma_k^a(1) z_j^* \geq \gamma_k^a z_j^*$$

by inequality (8) ($\mu = \frac{1}{2}a^{k-1}$ and $b = y_k^*$) and since $(1 - \frac{1-y_k^*}{k-1})^{k-1}$ is increasing with y_k^*.

If $y_k^* > \frac{1}{2a}$, then $1 - f_1^a(y_k^*) = 1 - ay_k^*$. We show that

$$1 - (1-ay_k^*)\, a^{k-1}\left(1 - \frac{z_j^* - y_k^*}{k-1}\right)^{k-1}$$
$$\geq \left(1 - (1-ay_k^*)\, a^{k-1}\left(1 - \frac{1-y_k^*}{k-1}\right)^{k-1}\right) z_j^*.$$

Let

$$F(z_j^*, y_k^*)$$
$$= 1 - (1-ay_k^*)\, a^{k-1}\left(1 - \frac{z_j^* - y_k^*}{k-1}\right)^{k-1} - \left(1 - (1-ay_k^*)\, a^{k-1}\left(1 - \frac{1-y_k^*}{k-1}\right)^{k-1}\right) z_j^*$$
$$= 1 - z_j^* - (1-ay_k^*)\, a^{k-1}\left(\left(1 - \frac{z_j^* - y_k^*}{k-1}\right)^{k-1} - \left(1 - \frac{1-y_k^*}{k-1}\right)^{k-1} z_j^*\right).$$

To prove $F(z_j^*, y_k^*) \geq 0$, we have only to show that $F(1, y_k^*) \geq 0$ and $F(y_k^*, y_k^*) \geq 0$, since $1 - (1-ay_k^*)\, a^{k-1}\left(1 - \frac{z_j^*-y_k^*}{k-1}\right)^{k-1}$ is a concave function on z_j^* with $y_k^* \leq z_j^* \leq 1$. Clearly $F(1, y_k^*) = 0$ holds and so we show $F(y_k^*, y_k^*) \geq 0$. Let

$$G(y_k^*) = 1 - y_k^* - \frac{1}{2}a^{k-1}\left(1 - \left(1 - \frac{1-y_k^*}{k-1}\right)^{k-1} y_k^*\right)$$

Then we have $F(y_k^*, y_k^*) \geq G(y_k^*)$, since $1 - ay_k^* \leq \frac{1}{2}$ with $\frac{1}{2a} \leq y_k^* \leq 1$. On the other hand, the derivative $G'(y_k^*)$ of $G(y_k^*)$ satisfies

$$G'(y_k^*)$$
$$= -1 + \frac{1}{2}a^{k-1}\left(\left(1 - \frac{1-y_k^*}{k-1}\right)^{k-2} y_k^* + \left(1 - \frac{1-y_k^*}{k-1}\right)^{k-1}\right)$$
$$\leq -1 + \frac{1}{2}a^{k-1}(1+1) = -1 + a^{k-1} \leq 0.$$

Thus, $G(y_k^*)$ is decreasing with y_k^*, and we have $G(y_k^*) \geq 0$ since $G(1) = 0$. Thus, $F(y_k^*, y_k^*) \geq G(y_k^*) \geq 0$ and $F(z_j^*, y_k^*) \geq 0$.

By the above argument and inequality (7),

$$C_j(f_1^a(y^*))$$
$$\geq 1 - (1-ay_k^*)\, a^{k-1}\left(1 - \frac{z_j^* - y_k^*}{k-1}\right)^{k-1}$$
$$\geq \left(1 - (1-ay_k^*)\, a^{k-1}\left(1 - \frac{1-y_k^*}{k-1}\right)^{k-1}\right) z_j^*$$
$$\geq \left(1 - \frac{1}{2}a^{k-1}\left(1 - \frac{1-\frac{1}{2a}}{k-1}\right)^{k-1}\right) z_j^*$$
$$= \gamma_k^a(1) z_j^* \geq \gamma_k^a z_j^*.$$

Case 3: $1 - \frac{1}{2a} < y_{k-1}^* \leq y_k^*$. Proof can be done similarly. □

In the following lemma, we give conditions on when $\gamma_k^a(1)$ dominates $\gamma_k^a(2)$, which will be used to obtain improved approximation algorithms in the next section.

Lemma 2 *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, for $\frac{3}{4} \leq a \leq \beta$,*

$$\gamma_k^a(1) = 1 - \frac{1}{2}a^{k-1}\left(1 - \frac{1-\frac{1}{2a}}{k-1}\right)^{k-1}$$
$$\leq \gamma_k^a(2) = 1 - a^k\left(1 - \frac{1}{k}\right)^k$$

for all $k \geq 2$ and thus, $\gamma_k^a = \gamma_k^a(1)$. Even for $\beta \leq a \leq \alpha$, $\gamma_k^a(1) \leq \gamma_k^a(2)$ for all $k \leq 7$. On the other hand, $\gamma_k^\alpha(1) \geq \gamma_k^\alpha(2)$ for all $k \geq 8$. Thus, for $\frac{3}{4} \leq a \leq \alpha$, $\gamma_k^a = \gamma_k^a(1)$ for $k \leq 7$ and $\gamma_k^a \geq \gamma_k^a(2) \geq \gamma_k^\alpha(2) = 1 - \alpha^k\left(1 - \frac{1}{k}\right)^k$ for $k \geq 8$.

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]. To describe the formulation precisely, we first need some notation. We will then 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.

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 Goemans-Williamson [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 $\frac{1+y_0 y_i}{2}$ 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 $Canon(\boldsymbol{v}, z) \geq \boldsymbol{b}$. For the other constraints, let $C = x_{i_1} \vee \cdots \vee x_{i_k}$ be a 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 $1 \leq i_1 < i_2 < i_3 \leq n$.

Now we are ready to formulate MAX SAT as the following vector programming problem. The formulation (AW) is shown in a separate figure. It can be considered to be the semidefinite programming problem in the standard way. 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 semidefinite programming and vector programming.

Let $V^* = (\boldsymbol{v}_0^*, \boldsymbol{v}_1^*, \ldots, \boldsymbol{v}_{2n}^*)$. Then an optimal solution $(V^*, \boldsymbol{z}^*)$ to this program is used in four ways as follows.

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 the LP relaxation of Goemans-Williamson [6], where f_1^a is the function defined in Eq. (2) 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 $\boldsymbol{v}_0^*$, 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

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

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

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-Williamson [5] and it corresponds to MAX 3SAT algorithm given by Karloff and Zwick [11].

Algorithm (4) Use a random truth assignment corresponding to 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 $\beta = .4555$, and 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 (2).

$$\begin{array}{llr}
\max & \sum_{C_j \in \mathcal{C}} w(C_j) z_j & \\
\text{s.t.} & \sum_{i=1}^{k_j} \frac{1 + v_0 v_{j_i}}{2} \geq z_j \quad \forall C_j = x_{j_1} \vee x_{j_2} \vee \cdots \vee x_{j_{k_j}} \in \mathcal{C} & (10) \\
& \frac{1 + v_0 v_{j_i}}{2} \leq z_j \quad \forall 1 \leq i \leq k_j, \forall C_j = x_{j_1} \vee x_{j_2} \vee \cdots \vee x_{j_{k_j}} \in \mathcal{C} & (11) \\
& \frac{1}{k-1} \sum_{C \in P_j} relax(C) \geq z_j \quad \forall C_j \in \mathcal{C}_k, \quad \forall k \geq 2 & (12) \\
(AW) & \frac{1}{\binom{k-1}{2}} \sum_{C \in Q_j} relax(C) \geq z_j \forall C_j \in \mathcal{C}_k, \quad \forall k \geq 3 & (13) \\
& relax(C) \geq z_j \quad \forall C_j \in \mathcal{C} & (14) \\
& fg(i_1, i_2, i_3) \geq -1 \quad \forall i_1, i_2, i_3 \text{ with } 0 \leq i_1 < i_2 < i_3 \leq n & (15) \\
& Canon(v, z) \geq b & (16) \\
& v_i \in S^n \quad 0 \leq \forall i \leq 2n & (17) \\
& v_{n+i} = -v_i \quad 1 \leq \forall i \leq n & (18) \\
& 0 \leq z_j \leq 1 \quad \forall C_j \in \mathcal{C}. & (19)
\end{array}$$

Suppose we pick the best solution returned by the four algorithms. The behavior of this algorithm is at least as good as the expected behavior 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. (4). 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

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

and at least $.87856 z_j^*$ for $k=1,2$. Zwick [16, 17] conjectures 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 $\alpha_k \geq .7977$; we list his values of α_k for $k=1,\ldots,39$ in the Appendix. Importantly, $\alpha_1 = \alpha_{35} = .7977$ and α_k is increasing for $k > 35$.

We now analyze what happens if we choose the best random truth assignment among those given by the algorithms (1)-(4). The expected behavior 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 behavior 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 $\beta w_j z_j^*$ 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 we consider the case in which we do not use algorithm (4), and we set $a = 3/4$ (that is, we use the function $f_1^{3/4}$ in algorithm (1)). If we set $p_1 = .7846081$, $p_2 = .1316834$, and $p_3 = 1 - (p_1 + p_2) = .0837085$, then we have

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

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 on $k = 4$, 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 above, and thus the algorithm has performance guarantee .7846.

Now 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 $k > 35$.

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.

Acknowledgments

The authors would like to express their gratitude to Professor U. Zwick for giving information about the bounds on how his MAX SAT algorithm performs on clauses of length k. 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.

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, In *Proc. 37th IEEE Symposium on the Foundations of Computer Science*, pp. 617–626, 1996.

[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.