Hybrid Approaches to MAX SAT

Takao Asano* Kuniaki Hori* Takao Ono** Tomio Hirata**
* Department of Information and System Engineering
Chuo University (asano@ise.chuo-u.ac.jp)
** School of Engineering, Nagoya University
({takao,hirata}@hirata.nuee.nagoya-u.ac.jp)

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 several hybrid approaches to MAX SAT proposed so far and give a new hybrid approach combining the algorithms of Goemans-Williamson and Yannakakis. We discuss the relations among these hybrid approaches and show that our new approach leads to a unified analysis of the performance guarantee.

MAX SATに対するハイブリッド法

浅野 孝夫* 堀 邦彰* 小野 孝男** 平田 富夫**
* 中央大学 理工学部 情報工学科 ** 名古屋大学 工学部

MAX SAT(最大充足化問題)とは, 節の集合と各節の重みが与えられたとき, 充足する節の重みの総和を最大にするような真偽割り当てを求める問題である. この論文では、これまでに提案されたMAX SATに対するハイブリッド法およびGoemans-WillaimsonとYannakakisを組み合わせた新しいハイブリッド法を考え、それらの間の関係について議論する。さらに、新しいハイブリッド法により統一的な性能解析が可能であることを示す。

1 Introduction

We consider MAX SAT (the maximum satisfiability problem): given a set of clauses with weights, find a truth assignment that maximizes the sum of the weights of the satisfied clauses. MAX SAT is NP-hard and many researchers have proposed approximation algorithms for MAX SAT. See [1] for algorithms proposed so far and the notations in this paper. The hybrid approach of Goemans-Williamson combining semidefinite programming and linear programming relaxations is quite natural [7]. On the other hand, the hybrid approach in [2] did not seem so natural and used a rather ad hoc technique for analysis. This is because Yannakakis' algorithm [10] changes a given instance to another instance and it seems difficult to analyze the performance quarantee of the hybrid approach of combining Yannakakis' algorithm with the hybrid approach of Goemans-Williamson.

In this paper, we consider several hybrid approaches to MAX SAT proposed so far and give a new hybrid approach combining the algorithms of Goemans-Williamson and Yannakakis. We discuss the relations among these hybrid approaches and show that our new approach leads to a unified analysis of the performance guarantee. The new approach leads to a better approximation algorithm with performance guarantee 0.770, if we use a refinement of Yannakakis' algorithm proposed in [1].

2 Formulations of Hybrid Approaches

In this section we will give formal formulations of hybrid approaches to MAX SAT proposed so far. Goemans-Williamson gave the following formulation of MAX SAT:

$$
\begin{array}{lll}
(IP): & \text{Maximize} & \sum_{C_j \in \mathcal{C}} w(C_j) z_j \\
& \text{subject to:} & \sum_{x_i \in X_j^+} y_i + \sum_{x_i \in X_j^-} (1 - y_i) \geq z_j \quad \forall C_j \in \mathcal{C} \\
& & y_i \in \{0,1\} \qquad \forall x_i \in X \\
& & z_j \in \{0,1\} \qquad \forall C_j \in \mathcal{C}.
\end{array}
$$

In this formulation, variables $y = (y_i)$ correspond to variables $X = \{x_1, x_2, ..., x_n\}$ and $z = (z_j)$ 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 and only if the clause is satisfied. The objective function represents the total weight of the satisfied clauses. Thus, the above formulation exactly corresponds to MAX SAT.

Then, Goemans-Williamson considered the following linear programming relaxation of MAX SAT where the variables y and z can take any value between 0 and 1.

$$
\begin{aligned}
(GW1): \quad & \text{Maximize} \quad \sum_{C_j \in \mathcal{C}} w(C_j) z_j \\
& \text{subject to:} \quad \sum_{x_i \in X_j^+} y_i + \sum_{x_i \in X_j^-} (1 - y_i) \geq z_j \qquad \forall C_j \in \mathcal{C} && (1) \\
& \qquad 0 \leq y_i \leq 1 \qquad \forall x_i \in X && (2) \\
& \qquad 0 \leq z_j \leq 1 \qquad \forall C_j \in \mathcal{C}. && (3)
\end{aligned}
$$

Using an optimal solution $(y^{\#}, z^{\#})$ to this program, they set each variable x_i to be true with probability $y_i^{\#}$. Then a clause C_j with k literals is satisfied with the pobability $1 - \prod_{x_i \in X_j^+}(1 - y_i^{\#}) \prod_{x_i \in X_j^-} y_i^{\#} \geq (1 - (1 - \frac{1}{k})^k) z_j^{\#}$, since

$$
\begin{aligned}
& 1 - \prod_{x_i \in X_j^+} (1 - y_i^{\#}) \prod_{x_i \in X_j^-} y_i^{\#} \\
\geq \; & 1 - \left(\frac{\sum_{x_i \in X_j^+}(1 - y_i^{\#}) + \sum_{x_i \in X_j^-}(1 - (1 - y_i^{\#}))}{k}\right)^k \\
\geq \; & 1 - (1 - \frac{z_j^{\#}}{k})^k \geq (1 - (1 - \frac{1}{k})^k) z_j^{\#} \qquad (4)
\end{aligned}
$$

by (1) and the arithmetic/geometric mean inequality. Thus, the expected value $F_{\mathcal{C}}(x^{\#})$ of this random truth assignement $x^{\#} = y^{\#} = (y_i^{\#})$ is at least

$$
W_1^{\#} + 0.75 W_2^{\#} + (1 - \frac{8}{27}) W_3^{\#} + \sum_{k \geq 4} (1 - (1 - \frac{1}{k})^k) W_k^{\#} \geq (1 - \frac{1}{e}) W^{\#}, \qquad (5)
$$

where $W_k^{\#} = \sum_{C_j \in \mathcal{C}_k} w(C_j) z_j^{\#}$ and $W^{\#} = \sum_{C_j \in \mathcal{C}} w(C_j) z_j^{\#} = \sum_{k \geq 1} W_k^{\#}$.

Note that this bound is good for short clauses and the expected value of the random truth assignment obtained by Johnson is good for long clauses. Thus, if we choose the better one between these two random truth assignments then the expected value is at least the bound $\sum_{k \geq 1} \beta_k W_k^*$ with $2\beta_k = 2 - \frac{1}{2^k} - (1 - \frac{1}{k})^k$.

On the other hand, Goemans-Williamson obtained a 0.878-approximation algorithm for MAX 2SAT based on semidefinite programming, and made a breakthrough for MAX 2SAT algorithms [6],[7]. They also applied this to the above linear programming relaxation, to obtain a better performance for MAX SAT. Thus, they used another hybrid approach combining a linear programming relaxation with their MAX 2SAT algorithm based on semidefinite programming and obtained a 0.758-approximation algorithm. This is a kind of breakthrough for MAX SAT, since it is believed to be difficult to overcome the bound 0.75. Let X_j denote the set of variables in C_j (i.e., $X_j = X_j^+ \cup X_j^-$) and let

$$
sgn_j(x_i) = \begin{cases} 1 & (x_i \in X_j^+) \\ -1 & (x_i \in X_j^-). \end{cases} \qquad (6)
$$

Then the following is the formal formulation of their new hybrid approach.

$$
\begin{aligned}
(GW2): \quad & \text{Maximize} \quad \sum_{C_j \in \mathcal{C}} w(C_j) z_j \\
& \text{subject to:} \quad \sum_{x_i \in X_j} \frac{1 + sgn_j(x_i) y_{0i}}{2} \geq z_j \qquad \forall C_j \in \mathcal{C} && (7)
\end{aligned}
$$

$$\frac{k}{2}C_j^{(2)}(Y) \geq z_j \qquad \forall C_j \in \mathcal{C}_k,\ \forall k \geq 2 \tag{8}$$

$$y_{ii} = 1 \qquad 0 \leq \forall i \leq n \tag{9}$$

$$0 \leq z_j \leq 1 \qquad \forall C_j \in \mathcal{C} \tag{10}$$

$$Y = (y_{i_1 i_2}) \text{ is a symmetric positive semidefinite matrix.} \tag{11}$$

Since all hybrid approaches desribed in this paper are closely related to this formulation, we briefly review the notations. Variables $y = (y_0, y_1, \ldots, y_n)$ corresponding to

$$y_0 y_i \equiv 2x_i - 1 \text{ with } |y_0| = |y_i| = 1 \tag{12}$$

are first introduced for semidefinite programming. Thus, x_i ($\bar{x}_i$, resp.) becomes $\frac{1+y_0y_i}{2}$ ($\frac{1-y_0y_i}{2}$, resp.) and a clause $C_j \in \mathcal{C}$ can be considered to be a function of $y = (y_0, y_1, \ldots, y_n)$ as follows by (??):

$$C_j = C_j(y) = 1 - \prod_{x_i \in X_j} \frac{1 - sgn_j(x_i) y_0 y_i}{2}. \tag{13}$$

Using an $(n+1)$-dimensional vector v_i with norm $\|v_i\| = 1$ corresponding to y_i with $|y_i| = 1$, they replace $y_{i_1} y_{i_2}$ with an inner vector product $v_{i_1} \cdot v_{i_2}$ and set $y_{i_1 i_2} = v_{i_1} \cdot v_{i_2}$. Thus, $\frac{1+y_{0i}}{2}$ corresponds to x_i and the matrix $Y = (y_{i_1 i_2})$ is symmetric and positive semidefinite since $Y = v^T v$ for $v = (v_0, v_1, \ldots, v_n)$.

The first set of constraints (7) exactly corresponds to (1), the set of constraints of the linear programming relaxation of MAX SAT of Goemans-Williamson explained before. It implies that, if $C_j = 1$ (i.e., $z_j = 1$) then one of the literals in C_j is true and thus, it holds for any truth assignment x.

The second set of constraints (8) corresponds to a MAX 2SAT relaxation. Each clause $C_j \in \mathcal{C}_k$ with $k \geq 2$ is approximately represented by $C_j^{(2)}$, the set of weighted clauses with two literals in C_j. The weight of C_j is evenly shared and the weight of each clause in $C_j^{(2)}$ becomes $\frac{2w(C_j)}{k(k-1)}$. For example, if $C_j = a_1 \vee a_2 \vee a_3$ then $C_j^{(2)} = \{a_1 \vee a_2, a_1 \vee a_3, a_2 \vee a_3\}$ and each clause in $C_j^{(2)}$ has the weight $w(C_j)/3$ (thus total weight of clauses in $C_j^{(2)}$ is $w(C_j)$). Clearly, $C_j^{(2)} = C_j$ for a clause C_j with two literals. Thus, by (13),

$$C_j^{(2)}(y) = \frac{1}{2k(k-1)} \sum_{x_{i_1}, x_{i_2} \in X_j, x_{i_1} \neq x_{i_2}} (3 + sgn_j(x_{i_1}) y_0 y_{i_1} + sgn_j(x_{i_2}) y_0 y_{i_2} - sgn_j(x_{i_1}) sgn_j(x_{i_2}) y_0^2 y_{i_1} y_{i_2})$$

and $C_j^{(2)}$ is a function of Y as follows.

$$C_j^{(2)}(Y) = \frac{1}{2k(k-1)} \sum_{x_{i_1}, x_{i_2} \in X_j, x_{i_1} \neq x_{i_2}} (3 + sgn_j(x_{i_1}) y_{0i_1} + sgn_j(x_{i_2}) y_{0i_2} - sgn_j(x_{i_1}) sgn_j(x_{i_2}) y_{i_1 i_2}). \tag{14}$$

Note that, if a clause $C_j \in \mathcal{C}_k$ is satisfied then one of the literals in C_j is true and at least $(k-1)$ clauses in $C_j^{(2)}$ are satisfied. Thus, the second set of constraints holds for all truth assignments and the formulation above is a combination of a linear programming relaxation and a MAX 2 SAT relaxation based on semidefinite programming.

The solution to this program is used in two ways. One is as a random truth assignment correponding to the linear programming relaxation and the other is as a random truth assignment correponding to the MAX2SAT relaxation. Choosing the best random truth assignment among the random truth assignments including above two assignments and the random truth assignment obtained by Johnson leads to a 0.7584-approximation algorithm.

After Goemans-Williamson's hybrid apporaches, various hybrid approaches are proposed to obtain better bounds. Feige-Goemans [4] used a hybrid approach which adds the following constraints to $(GW2)$.

$$(FG): \quad \text{Maximize} \quad \sum_{C_j \in \mathcal{C}} w(C_j) z_j$$

$$\text{subject to:}\quad (7), (8), (9), (10), (11) \text{ and} \tag{15}$$

$$y_{i_1i_2} + y_{i_2i_3} + y_{i_3i_1} \geq -1, \quad -y_{i_1i_2} + y_{i_2i_3} - y_{i_3i_1} \geq -1,$$
$$-y_{i_1i_2} - y_{i_2i_3} + y_{i_3i_1} \geq -1, \quad y_{i_1i_2} - y_{i_2i_3} - y_{i_3i_1} \geq -1$$
$$\forall i_1, i_2, i_3 \text{ with } 0 \leq i_1 < i_2 < i_3 \leq n. \tag{16}$$

Asano-Ono-Hirata [2] have proposed another relaxation of MAX SAT which is a hybrid approach combining the linear programming relaxation and the semidefintie programming method as follows.

$$(A96): \quad \text{Maximize} \quad \sum_{C_j \in \mathcal{C}} w(C_j) z_j$$

$$\text{subject to:} \quad (7), (9), (10), (11) \text{ and} \tag{17}$$

$$\frac{2^{k-1}}{k} c_j^{(1)}(Y) \geq z_j \qquad \forall C_j \in \mathcal{C}_k, \forall k \geq 2. \tag{18}$$

We briefly review the notation $c_j^{(1)}$. $c_j^{(1)}(y)$ is the sum of the terms in $C_j(y)$ of forms $1 + sgn_j(x_i)y_0y_i$ and $1 - sgn_j(x_{i_1})sgn_j(x_{i_2})y_{i_1}y_{i_2}$, i.e., for $C_j \in \mathcal{C}_k$,

$$c_j^{(1)}(y) = \frac{1}{2^k} \sum_{x_i \in X_j} (1 + sgn_j(x_i)y_0y_i) + \frac{1}{2^k} \sum_{x_{i_1}, x_{i_2} \in X_j, x_{i_1} \neq x_{i_2}} (1 - sgn_j(x_{i_1})sgn_j(x_{i_2})y_{i_1}y_{i_2})$$

and

$$c_j^{(1)}(Y) = \frac{1}{2^k} \sum_{x_i \in X_j} (1 + sgn_j(x_i)y_{0i}) + \frac{1}{2^k} \sum_{x_{i_1}, x_{i_2} \in X_j, x_{i_1} \neq x_{i_2}} (1 - sgn_j(x_{i_1})sgn_j(x_{i_2})y_{i_1i_2}) \tag{19}$$

The constraints (18) are introduced in [2] and serve as a kind of approximation of original MAX SAT constraints. Of course, they hold for any truth assignment x and thus, $(A96)$ can be considered to a relaxation of MAX SAT. In this paper we also consider the following relaxation of MAX SAT obtained by adding the constraints introduced by Feige-Goemans [4].

$$(A97): \quad \text{Maximize} \quad \sum_{C_j \in \mathcal{C}_{1,2}} w(C_j)C_j(Y) + \sum_{k \geq 3} \sum_{C_j \in \mathcal{C}_k} w(C_j) z_j$$

$$\text{subject to:} \quad (9), (11), (16) \text{ and} \tag{20}$$

$$\frac{2^{k-1}}{k} c_j^{(1)}(Y) \geq z_j \qquad \forall C_j \in \mathcal{C}_k \text{ with } k \geq 3 \tag{21}$$

$$0 \leq z_j \leq 1 \qquad \forall C_j \in \mathcal{C}_k \text{with } k \geq 3, \tag{22}$$

where $\mathcal{C}_{1,2} = \mathcal{C}_1 \cup \mathcal{C}_2$.

For completeness, we first show that $(A97)$ is a relaxation of MAX SAT, although it was suggested by Feige-Goemans [4]. Let $x^q = (x_i^q) \in \{0, 1\}^n$ be any truth assignment for $(\mathcal{C}, w)$. Define $Y^q = (y_{i_1i_2}^q)$ to be

$$y_{00}^q = 1,\ y_{0i}^q = y_{i0}^q = 2x_i^q - 1\ (1 \leq i \leq n) \text{ and } y_{i_1i_2}^q = y_{i_2i_1}^q = y_{0i_1}^q y_{0i_2}^q\ (1 \leq i_1 \leq i_2 \leq n).$$

Then

$$y_{0i}^q \in \{-1, 1\},\ y_{i_1i_2}^q \in \{-1, 1\} \text{ and } y_{ii}^q = 1.$$

Thus, (9) is satisfied. (16) can be shown to be satisfied as follows. For example,

$$y_{0i_1}^q + y_{0i_2}^q + y_{i_1i_2}^q = 2x_{i_1}^q - 1 + 2x_{i_2}^q - 1 + (2x_{i_1}^q - 1)(2x_{i_2}^q - 1) = (2x_{i_1}^q)(2x_{i_2}^q) - 1 \geq -1.$$

Similarly,

$$y_{i_1i_2}^q + y_{i_2i_3}^q + y_{i_3i_1}^q = y_{0i_1}^q y_{0i_2}^q + y_{0i_2}^q y_{0i_3}^q + y_{0i_3}^q y_{0i_1}^q = (y_{0i_1}^q + y_{0i_2}^q)(y_{0i_1}^q + y_{0i_3}^q) - (y_{0i_1}^q)^2.$$

Thus, by symmetry, if (at least) one of $y_{0i_1}^q, y_{0i_2}^q, y_{0i_3}^q$ is equal to 1 then $y_{i_1i_2}^q + y_{i_2i_3}^q + y_{i_3i_1}^q \geq -1$ is obtained as above. Otherwise (i.e., $y_{0i_1}^q = y_{0i_2}^q = y_{0i_3}^q = -1$), $y_{i_1i_2}^q + y_{i_2i_3}^q + y_{i_3i_1}^q = 3 \geq -1$. Other cases are similarly shown. Thus, (16) is satisfied. For $C_j \in \mathcal{C}_k$ with $k \geq 3$, define

$$z_j^q = 1 \text{ if } C_j \text{ is satisfied by } x^q \text{ and } z_j^q = 0 \text{ otherwise.}$$

Then (22) is satisfied. If C_j is satisfied by x^q, then some literal in C_j, $x_i \in X_j^+$ or $\bar{x}_{i'}$ with $x_{i'} \in X_j^-$ is true and $(1+y_{0i}^q)/2 = x_i^q = 1$ or $(1-y_{0i'}^q)/2 = \bar{x}_{i'}^q = 1$ and $c_j^{(1)}(Y^q) \neq 0$. Thus, by Lemma 1 in [2], $\frac{2^{k-1}}{k}c_j^{(1)}(Y^q) \geq 1 = z_j^q$ for $k \geq 3$. Otherwise, all literals in C_j are false and $(1+y_{0i}^q)/2 = x_i^q = 0$ and $(1-y_{0i'}^q)/2 = \bar{x}_{i'}^q = 0$ and $c_j^{(1)}(Y^q) = 0 = z_j^q$ for $k \geq 3$. Thus, (21) holds. Similarly, since $c_j^{(1)}(Y) = C_j(Y)$ for any $C_j \in \mathcal{C}_{1,2}$ and

$$C_j(Y) = \begin{cases} (1+y_{0i})/2 & (C_j = x_i \in \mathcal{C}_1) \\ (1-y_{0i})/2 & (C_j = \bar{x}_i \in \mathcal{C}_1) \\ (3+y_{0i_1}+y_{0i_2}-y_{i_1i_2})/4 & (C_j = x_{i_1} \vee x_{i_2} \in \mathcal{C}_2) \\ (3-y_{0i_1}+y_{0i_2}+y_{i_1i_2})/4 & (C_j = \bar{x}_{i_1} \vee x_{i_2} \in \mathcal{C}_2) \\ (3-y_{0i_1}-y_{0i_2}-y_{i_1i_2})/4 & (C_j = \bar{x}_{i_1} \vee \bar{x}_{i_2} \in \mathcal{C}_2), \end{cases} \tag{23}$$

$\sum_{C_j \in \mathcal{C}_{1,2}} w(C_j)C_j(Y^q) + \sum_{k\geq 3}\sum_{C_j \in \mathcal{C}_k} w(C_j)z_j^q$ is the total weight of the clauses satisfied by x^q. Furthemore, Y^q is a symmetric positive semidefinite matrix and (11) is satisfied, since $Y^q = (1, y_{01}^q, y_{02}^q, ..., y_{0n}^q)^T(1, y_{01}^q, y_{02}^q, ..., y_{0n}^q)$. Thus, $(A97)$ was shown to be a relaxation of MAX SAT.

We next show that a solution (Y, z) to $(A97)$ leads to a solution to $(A96)$, that is, (Y, z) with appropriately setted z_j for $C_j \in \mathcal{C}_{1,2}$ satisfies (7), (10) and (18). We set

$$z_j = C_j(Y) \text{ for each } C_j \in \mathcal{C}_{1,2}.$$

Then, clearly (7) is satisfied for $C_j \in \mathcal{C}_1$ and (18) is satisfied for $C_j \in \mathcal{C}_2$. Note that, for a clause $C_j \in \mathcal{C}_k$ with $k \geq 2$, (7) is redundant since if $C_j = x_1 \vee x_2 \vee \cdots \vee x_k$ then

$$\begin{aligned} &\frac{1}{2}\sum_{i=1}^{k}(1+y_{0i}) - \frac{2^{k-1}}{k}c_j^{(1)}(Y) \\ &= \frac{1}{2}\sum_{i=1}^{k}(1+y_{0i}) - \frac{1}{2k}(\sum_{i=1}^{k}(1+y_{0i}) + \sum_{1\leq i_1<i_2\leq k}(1-y_{i_1i_2})) \\ &= \frac{1}{2k}\sum_{1\leq i_1<i_2\leq k}(1+y_{0i_1}+1+y_{0i_2}-1+y_{i_1i_2}) \geq 0 \end{aligned} \tag{24}$$

by (16) (by symmetry we can argue the other cases similarly). Furthermore, for a clause $C_j \in \mathcal{C}_{1,2}$, $z_j \leq 1$ is automatically satisfied since $C_j(Y) \leq 1$ by (16) and (23), $y_{ii} = 1$ and Y is a symmetric positive semidefinite matrix. Thus, (Y, z) with $z_j = C_j(Y)$ for $C_j \in \mathcal{C}_{1,2}$, say (Y, z_S), is a solution to $(A96)$ and both (Y, z) and (Y, z_S) have the same value. Thus, for an optimal solution (Y, z) to $(A97)$, (Y, z_S) is a solution to $(A96)$ (but not necessarily an optimal solution). Thus, $(A97)$ can be considered to be a better relaxation of MAX SAT than $(A96)$.

Similarly, a solution (Y, z) to $(A97)$ leads to a solution to (FG), that is, (Y, z) with $z_j = C_j(Y)$ for $C_j \in \mathcal{C}_{1,2}$ satisfies (7), (8) and (10). Here, note that, for a clause $C_j \in \mathcal{C}_k$ with $k \geq 3$, (8) is redundant since if $C_j = x_1 \vee x_2 \vee \cdots \vee x_k$ then

$$\begin{aligned} &\frac{k}{2}c_j^{(2)}(Y) - \frac{2^{k-1}}{k}c_j^{(1)}(Y) \\ &= \frac{1}{4(k-1)}(\sum_{i=1}^{k}(k-1)(1+y_{0i}) + \sum_{1\leq i_1<i_2\leq k}(1-y_{i_1i_2})) - \frac{1}{2k}(\sum_{i=1}^{k}(1+y_{0i}) + \sum_{1\leq i_1<i_2\leq k}(1-y_{i_1i_2})) \\ &= \frac{k-2}{4k(k-1)}\sum_{1\leq i_1<i_2\leq k}(1+y_{0i_1}+1+y_{0i_2}-1+y_{i_1i_2}) \geq 0 \end{aligned} \tag{25}$$

by (16) (by symmetry we can argue the other cases similarly).

The arguments above also show that, for a clause $C_j \in \mathcal{C}_k$, (8) with (16) implies (7), since if $C_j = x_1 \vee x_2 \vee \cdots \vee x_k$ then

$$\frac{1}{2}\sum_{i=1}^{k}(1+y_{0i}) - \frac{k}{2}c_j^{(2)}(Y)$$

$$= \frac{1}{2}\sum_{i=1}^{k}(1+y_{0i}) - \frac{1}{4(k-1)}(\sum_{i=1}^{k}(k-1)(1+y_{0i}) + \sum_{1\le i_1<i_2\le k}(1-y_{i_1i_2}))$$

$$= \frac{1}{4(k-1)}\sum_{1\le i_1<i_2\le k}(1+y_{0i_1}+1+y_{0i_2}-1+y_{i_1i_2}) \ge 0 \qquad (26)$$

(by symmetry we can argue the other cases similarly).

As a summary we have the following theorem (the other cases not discussed above can be simlarly treated).

Theorem 1 *Let $S(GW1)$, $S(GW2)$, $S(FG)$, $S(A96)$, $S(A97)$ be the set of solutions to the programs $(GW1)$, $(GW2)$, (FG), $(A96)$, $(A97)$ respectively. Then the following relations hold.*

$S(GW1) \subset S(GW2) \subset S(FG) \subset S(A97)$ and $S(A96) \subset S(A97)$.

These inclusions are strict and $(A96)$ gives the smallest upper bound among the programs proposed so far.

3 Strongly Equivalent Transformations

We have discussed the relations of the solutions to relaxations of MAX SAT proposed so far and shown that (A97) is the best dual heuristic [8] in a sense that it gives the smallest upper bound among them. In this section, we will consider how we can combine Yannakakis' algorithm with the hybrid approach $A(97)$ proposed above. Roughly speaking, we might combine Yannakakis' algorithm (including its refinements) with any hybrid approach in the previous section, we will only consider $(A97)$, since it gives a more chance to get a better performance guarantee. First, we will briefly review essential parts of Yannakakis' algorithm and its existing refinements below.

The 0.75-approximation algorithm of Yannakakis divides the variables $X = \{x_1, \ldots, x_n\}$ of a given instance $(\mathcal{C}, w)$ into three groups P, P' and P'' based on maximum network flows (some variables will be negated appropriately). Then it sets independently each variable $x_i \in X$ to be true with probability p_i such that $p_i = 3/4$ if $x_i \in P$, $p_i = 5/9$ if $x_i \in P'$ and $p_i = 1/2$ if $x_i \in P''$. The expected value $F_{\mathcal{C}}(x^p)$ of this random truth assignment $x^p = (p_1, p_2, ..., p_n)$ is at least the bound in (??).

To divide the variables X of a given instance $(\mathcal{C}, w)$ into three groups P, P' and P'', Yannakakis transformed $(\mathcal{C}, w)$ into an equivalent instance $(\mathcal{C}', w')$ of the weighted clauses with some nice property by using network flows. Note that two sets $(\mathcal{C}, w)$, $(\mathcal{C}', w')$ of weighted clauses over the same set of variables are called *equivalent* if, for every truth assignment, $(\mathcal{C}, w)$ and $(\mathcal{C}', w')$ have the same value. Similarly, based on [3], $(\mathcal{C}, w)$,$(\mathcal{C}', w')$ are called *strongly equivalent*, if, for every *random* truth assignment, $(\mathcal{C}, w)$ and $(\mathcal{C}', w')$ have the same *expected* value. Clearly, if $(\mathcal{C}, w)$,$(\mathcal{C}', w')$ are strongly equivalent then they are also equivalent since a truth assignment is always a random truth assignment (the converse is not true). The following lemma [3] plays a central role throughout this paper.

Lemma 1 *Let all clauses below have the same weight. Then $\mathcal{A} = \{\bar{x}_i \vee x_{i+1} | i = 1, ..., k\}$ and $\mathcal{A}' = \{x_i \vee \bar{x}_{i+1} | i = 1, ..., k\}$ are strongly equivalent (we consider $k+1 = 1$). Furthermore, $\mathcal{B} = \{x_1\} \cup \{\bar{x}_i \vee x_{i+1} | i = 1, ..., \ell\}$ and $\mathcal{B}' = \{x_i \vee \bar{x}_{i+1} | i = 1, ..., \ell\} \cup \{x_{\ell+1}\}$ are strongly equivalent.*

For a given instance $(\mathcal{C}, w)$, Yannakakis' algorithm and all existing refinements consist of several steps, say $L+1$ steps, and divide the variables X into groups using flows in networks defined based on Lemma 1. In each step except for Step $L+1$, the algorithm outputs a set of weighted clauses which is strongly equivalent to a set of weighted clauses given as an input of that step. The output of Step i ($i = 1, 2, ..., L$) consists of groups of weighted clauses and all but one group are set aside (we call such groups being split off). The remaining group becomes an input of Step $i+1$. Thus, we have the groups of weighted clauses split off in Steps 1 to L and the remaining group $(\mathcal{D}^{(L)}, w_L)$ after Step L. Let $(\mathcal{C}^L, w_L)$ be the set of all split groups together with $(\mathcal{D}^{(L)}, w_L)$. Then $(\mathcal{C}^L, w_L)$ is strongly equivalent to a given instance $(\mathcal{C}, w)$. The crucial point is that a clause $C \in \mathcal{C}_k$ with $k \ge 3$ may be split off and appear in several groups of the finally obtained instance $(\mathcal{C}^L, w_L)$ but the total weight of C is the same as in the given input instance $(\mathcal{C}, w)$. Only the total weight of a clause with one or two literals may change.

After Step L, we obtain a partition of X into several sets $\{X_k \mid k = 1, 2, ...\}$ and in Step $L+1$, we obtain a random truth assignment $x^p = (p_1, p_2, ..., p_n)$ by appropriately setting each variable x_i to be true with probability p_i such that $p_i = p_{i'}$ if x_i and $x_{i'}$ are in the same set X_k in the partition of X. Then, all groups of weighted clauses split off in Steps 1 to L and the remaining group $(\mathcal{D}^{(L)}, w_L)$ of weighted clauses after Step L are shown to have the expected values at least the bound $\sum_k \gamma_k W_k^*$. Furthermore, we have the following lemma since $(\mathcal{C}, w)$ and $(\mathcal{C}^L, w_L)$ are strongly equivalent (see also [1]).

Lemma 2 *Let x^r be any random truth assignment and let $W_k^r(\mathcal{C}^L)$ be the expected value of x^r for the weighted clauses in $(\mathcal{C}^L, w_L)$ with k literals. Similarly, let $W_k^r(\mathcal{C})$ be the expected value of x^r for the weighted clauses in $(\mathcal{C}, w)$ with k literals. Let $W_k^*(\mathcal{C})$ ($W_k^*(\mathcal{C}^L)$, resp.) be the value of an optimal truth assignment x^* for the weighted clauses in $(\mathcal{C}, w)$ (($\mathcal{C}^L, w_L$), resp.) with k literals. Then the following statements hold.*
(a) $W_k^r(\mathcal{C}) = W_k^r(\mathcal{C}^L)$ and $W_k^*(\mathcal{C}) = W_k^*(\mathcal{C}^L)$ for all $k \geq 3$.
(b) $W_1^r(\mathcal{C}) + W_2^r(\mathcal{C}) = W_1^r(\mathcal{C}^L) + W_2^r(\mathcal{C}^L)$ and $W_1^*(\mathcal{C}) + W_2^*(\mathcal{C}) = W_1^*(\mathcal{C}^L) + W_2^*(\mathcal{C}^L)$.

Now we would like to find a relation between a solution to $(A97)$ and a solution to the following MAX SAT formulation (Q) which is equal to $(A97)$ for the instance produced by a refinement of Yannakakis.

$$(Q): \quad \text{Maximize} \sum_{C_j \in \mathcal{C}_{1,2}^L} w_L(C_j)C_j(Y) + \sum_{k\geq 3}\sum_{C_j \in \mathcal{C}_k^L} w_L(C_j)z_j$$
$$\text{subject to the constraints (9), (11), (16) and}$$
$$\tfrac{2^{k-1}}{k} c_j^{(1)}(Y) \geq z_j \qquad \forall C_j \in \mathcal{C}_k^L \text{ with } k \geq 3,$$
$$0 \leq z_j \leq 1 \qquad \forall C_j \in \mathcal{C}_k^L \text{with } k \geq 3.$$

As noted before, each clause C of $(\mathcal{C}, w)$ with three or more literals has the weight equal to the sum of the weights of C in $(\mathcal{C}^L, w_L)$ in the strongly equivalent tranformations in Section 2 in Lemma 1, (i.e., $(\mathcal{C}_k, w) = (\mathcal{C}_k^L, w_L)$ for $k \geq 3$ even if C may be contained in two or more groups in $(\mathcal{C}^L, w_L)$). Thus, the constraints of $(A97)$ and (Q) are the same and we can have the following lemma.

Lemma 3 *Both $(A97)$ and (Q) have the same solutions and the same optimal solutions.*

Before proving the above lemma, we consider the following relaxation of MAX 2SAT formulation (P):

$$(P): \text{Maximize} \sum_{C_j \in \mathcal{C}_{1,2}} w(C_j)C_j(Y) \text{ subject to the constraints (9), (11), (16).}$$

By the same argument as for MAX SAT, (P) can be considered to be a relaxation of MAX 2SAT. Since $\mathcal{C}_{1,2}^L$ is a set of weighted clauses obtained from $\mathcal{C}_{1,2}$ by the strongly equivalent tranformations in Lemma 1, the following MAX 2SAT formulation (P') exactly corresponds to (P) for the instance $\mathcal{C}_{1,2}^L$.

$$(P'): \text{Maximize} \sum_{C'_j \in \mathcal{C}_{1,2}^L} w_L(C'_j)C'_j(Y) \text{ subject to the constraints (9), (11), (16).}$$

Then we have the following lemma.

Lemma 4 *Two problems (P) and (P') have the same feasible solutions and values. Thus,*

$$\sum_{C_j \in \mathcal{C}_{1,2}} w(C_j)C_j(Y) = \sum_{C'_j \in \mathcal{C}_{1,2}^L} w_L(C'_j)C'_j(Y)$$

for any feasible solution Y and (P) and (P') have the same optimal solutions.

Proof. Clearly (P) and (P') have the same feasible solutions since constraints are the same. It suffices to show that both have the same optimal value for the case $\mathcal{C}_{1,2} = \mathcal{A} = \{\bar{x}_i \vee x_{i+1} | i = 1, ..., k\}$ and $\mathcal{C}_{1,2}^L = \mathcal{A}' = \{x_i \vee \bar{x}_{i+1} | i = 1, ..., k\}$ (we consider $k + 1 = 1$) and the case $\mathcal{C}_{1,2} = \mathcal{B} = \{x_1\} \cup \{\bar{x}_i \vee x_{i+1} | i = 1, ..., \ell\}$ and $\mathcal{C}_{1,2}^L = \mathcal{B}' = \{x_i \vee \bar{x}_{i+1} | i = 1, ..., \ell\} \cup \{x_{\ell+1}\}$ in Lemma 1, since $\mathcal{C}_{1,2}^L$ is obtained from $\mathcal{C}_{1,2}$ by the strongly equivalent tranformations in Lemma 1. We can assume weights are all equal to 1. Let $\mathcal{C}_{1,2} = \mathcal{A} = \{\bar{x}_i \vee x_{i+1} | i = 1, ..., k\}$ and $\mathcal{C}_{1,2}^L = \mathcal{A}' = \{x_i \vee \bar{x}_{i+1} | i = 1, ..., k\}$ and $C_j = \bar{x}_j \vee x_{j+1}$ and $C'_j = \bar{x}_{j+1} \vee x_j$. Then

$$\sum_{j=1}^{k} C_j(Y) = \sum_{j=1}^{k} C'_j(Y)$$

since $\sum_{j=1}^k C_j(Y) = \sum_{j=1}^k \frac{1}{4}(1 - y_{0j} + 1 + y_{0j+1} + 1 + y_{jj+1}) = \sum_{j=1}^k \frac{1}{4}(3 + y_{jj+1})$ and $\sum_{j=1}^k C'_j(Y) = \sum_{j=1}^k \frac{1}{4}(1 + y_{0j} + 1 - y_{0j+1} + 1 + y_{jj+1}) = \sum_{j=1}^k \frac{1}{4}(3 + y_{jj+1})$.

Analogous argument can be done for the case $\mathcal{C}_{1,2} = \mathcal{B}$ and $\mathcal{C}_{1,2}^L = \mathcal{B}'$. □

The proof of Lemma 3 can be obtained immediately by Lemma 4, since each clause C of $(\mathcal{C}, w)$ with three or more literals has the weight equal to the sum of the weights of C in $(\mathcal{C}^L, w_L)$ in the strongly equivalent tranformations in Section 2.

Since (Q) is a semidefinite programming problem as in [7], we can find an approximate optimal solution $(Y^\#, z^\#)$ within a small constant error ϵ in polynomial time. For convenience, we call it an optimal solution to (Q) (and $(A97)$). An optimal solution $v^\# = (v_0^\#, v_1^\#, \ldots, v_n^\#)$ can be obtained by Cholesky decomposition of $Y^\# = (y^\#_{i_1 i_2})$. Thus,

$$W^\#_{1,2}(\mathcal{C}^L) = \sum_{C \in \mathcal{C}^L_{1,2}} w_L(C) C(Y^\#) = \sum_{C \in \mathcal{C}_{1,2}} w(C) C(Y^\#) = W^\#_{1,2}(\mathcal{C})$$

and

$$W^\#_k(\mathcal{C}^L) = \sum_{C_j \in \mathcal{C}^L_k} w_L(C_j) z^\#_j = W^\#_k(\mathcal{C}) \text{ for } k \geq 3.$$

Since the existing refinements of Yannakakis' algorithm use only strongly equivalent transformations which keep the total weight of a clause with three or more literals, we can use the same analysis for the optimal solution (Q) (and $(A97)$) as that for an optimal truth assignment x^* of $(\mathcal{C}, w)$. We have only to use $x^\# = (x^\#_i)$ with $x^\#_i = \frac{1}{2}(1 + y^\#_{0i})$ instead of x^*. There, the following inequality plays an essential role: $z^\#_j \leq \min\{1, \sum_{x_i \in X^+_j} x^\#_i + \sum_{x_i \in X^-_j} (1 - x^\#_i)\}$ for each $C_j \in \mathcal{C}_k$ with $k \geq 3$.

Lemma 3 and the above argument have shown the following theorem.

Theorem 2 *In a hybrid approach of combining Goemans-Williamson's algorithm with a refinement of Yannakakis' algorithm based on the strongly equivalent transformations which keep the total weight of any clause with three or more literals, we can consider them independently. Furthermore, in an analysis, we can use the same argumemt for the optimal solution (Q) (and $(A97)$) as that for an optimal truth assignment x^* of $(\mathcal{C}, w)$.*

4 Concluding Remarks

We have presented a theoretical framewok of hybrid approaches combining the algorithms of Goemans-Williamson and Yannakakis. Since the framework presented is so general and can be applied to any refinements of Yannakakis' algorithm using strongly equivalent transformations in Lemma 1, we believe further improvements of the performance guarantee for MAX SAT can be achieved by this framework. Furthermore the framework is easily modified to match with the 0.931-approximation algorithm for MAX 2SAT by Feige and Goemans [4].

Acknowledgments.

The first author was supported in part by Grant in Aid for Scientific Research of the Ministry of Education, Science and Culture of Japan, The Institute of Science and Engineering of Chuo University, and The Telecommunications Advancement Foundation.

References

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

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

[3] T. Asano, K. Hori, T. Ono and T. Hirata, A refinement of Yannakakis's algorithm for MAX SAT. *Information Processing Society of Japan, SIGAL-TR-54-11*, 1996.

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

[5] M. X. Goemans and D. P. Williamson, .878-approximation algorithms for MAX CUT and MAX 2SAT. *Proc. 26th STOC*, 1994, pp.422–431.

[6] M. X. Goemans and D. P. Williamson, New 3/4-approximation algorithms for the maximum satisfiability problem. *SIAM J. Disc. Math.*, 7 (1994), pp.656–666.

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

[8] M. Grötschel and L. Lovász, Combinatorial Optimization. *in Handbook of Combinatorics (R. Graham, M. Grötschel and L. Lovász, eds.)*, Elsevier Science B.V., 1995, pp.1541–1597.

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

[10] M. Yannakakis, On the approximation of maximum satisfiability. *Proc. 3rd SODA*, 1992, pp.1–9 (and also *J. Algorithms*, 17 (1994), pp.475-502).