数理解析研究所講究録
1004 巻 1997 年 90-108

Approximation Algorithms for MAX SAT: Semidefinite Programming and Network Flows Approach

Takao Asano (浅野孝夫) Kuniaki Hori (堀邦彰)
Department of Information and System Engineering, Chuo University
asano@ise.chuo-u.ac.jp

Takao Ono (小野孝男) Tomio Hirata (平田富夫)
School of Engineering, Nagoya University
{takao,hirata}@hirata.nuee.nagoya-u.ac.jp

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 present an approximation algorithm for MAX SAT which is a refinement of Yannakakis's algorithm. This algorithm leads to a better approximation algorithm with performance guarantee 0.767 if it is combined with the previous algorithms for MAX SAT.

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 2SAT, the restricted version of MAX SAT where each clause has at most 2 literals, is well known to be NP-hard even if the weights of the clauses are identical, and thus MAX SAT is also NP-hard. Thus, many researchers have proposed approximation algorithms for MAX SAT. Yannakakis [9] and Goemans-Williamson [4] proposed 0.75-approximation algorithms. Later, Goeman-Williamson improved the bound 0.75 to 0.7584 based on semidefinite programming [5]. Recently, Asano-Ono-Hirata also improved the bound and the best approximation algorithm for MAX SAT has the performance guarantee 0.765 [1].

In this paper, we first present a refinement of the 0.75 approximation algorithm of Yannakakis for MAX SAT based on network flows. Then we will show that it leads to a 0.767-approximation algorithm if it is combined with the algorithms based on semidefinite programming approach [1],[2],[5].

2 Preliminaries

An instance of MAX SAT is defined by $(\mathcal{C}, w)$, where $\mathcal{C}$ is a collection of boolean clauses such that each clause $C \in \mathcal{C}$ is a disjunction of literals and has a nonnegative weight $w(C)$ (a *literal* is either a variable x_i or its negation $\bar{x}_i$). We sometimes write an instance $\mathcal{C}$ 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 variables in the weighted clauses of $(\mathcal{C}, w)$. We assume that no variable appears more than once in a clause in $\mathcal{C}$, that is, we do not allow a clause like $x_1 \vee x_1 \vee x_2$. For each variable

$x_i \in X$, we consider $x_i = 1$ ($x_i = 0$, resp.) if x_i is true (false, resp.). Then, $\bar{x}_i = 1 - x_i$ and a clause $C_j \in \mathcal{C}$ can be considered to be a function of $\boldsymbol{x} = (x_1, \ldots, x_n)$ as follows:

$$C_j = C_j(\boldsymbol{x}) = 1 - \prod_{x_i \in X_j^+} (1 - x_i) \prod_{x_i \in X_j^-} x_i \tag{1}$$

where X_j^+ (X_j^-, resp.) denotes the set of variables appearing unnegated (negated, resp.) in C_j. Thus, $C_j = C_j(\boldsymbol{x}) = 0$ or 1 for any *truth assignment* $\boldsymbol{x} \in \{0,1\}^n$ (i.e., an assignment of 0 or 1 to each $x_i \in X$), and C_j is *satisfied* (*not satisfied*, resp.) if $C_j(\boldsymbol{x}) = 1$ ($C_j(\boldsymbol{x}) = 0$, resp.). The *value* of a truth assignment $\boldsymbol{x}$ is defined to be

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

That is, the value of $\boldsymbol{x}$ is the sum of the weights of the clauses in $\mathcal{C}$ satisfied by $\boldsymbol{x}$. Thus, MAX SAT is to find a truth assignment of maximum value.

Let A be an algorithm for MAX SAT and let $F_{\mathcal{C}}(x^A(\mathcal{C}))$ be the value of a truth assignment $x^A(\mathcal{C})$ produced by A for an instance $\mathcal{C}$. If $F_{\mathcal{C}}(x^A(\mathcal{C}))$ is at least α times the value $F_{\mathcal{C}}(x^*(\mathcal{C}))$ of an optimal truth assignment $x^*(\mathcal{C})$ 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.*

The 0.75-approximation algorithm of Yannakakis is based on the probabilistic method proposed by Johnson [6]. Let $\boldsymbol{x}^p$ be a *random* truth assignment with $0 \le x_i^p = p_i \le 1$, that is, $\boldsymbol{x}^p$ is obtained by setting independently each variable $x_i \in X$ to be true with probability p_i. Then the probability of a clause $C_j \in \mathcal{C}$ satisfied by the assignment $\boldsymbol{x}^p$ is

$$C_j(\boldsymbol{x}^p) = 1 - \prod_{x_i \in X_j^+} (1 - p_i) \prod_{x_i \in X_j^-} p_i \,. \tag{3}$$

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

$$F_{\mathcal{C}}(\boldsymbol{x}^p) = \sum_{C_j \in \mathcal{C}} w(C_j) C_j(\boldsymbol{x}^p). \tag{4}$$

The probabilistic method assures that there is a truth assignment $\boldsymbol{x}^q \in \{0,1\}^n$ such that its value is at least $F_{\mathcal{C}}(\boldsymbol{x}^p)$. Such a truth assignment $\boldsymbol{x}^q$ can be obtained by the method of conditional probability [4],[9].

Yannakakis introduced *equivalent* instances for MAX SAT [9]: 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. In this paper, we call $(\mathcal{C}, w)$,$(\mathcal{C}', w')$ are *strongly equivalent* if, for every random truth assignment, $(\mathcal{C}, w)$ and $(\mathcal{C}', w')$ have the same expected value. Note that, 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). Our notion of equivalence will be required when we try to obtain an improved bound 0.767. The following lemma plays a central role throughout this paper.

Lemma 1 *Let all clauses below have the same weight.*

1. $\mathcal{A} = \{\bar{x}_i \vee x_{i+1} | i = 1, \ldots, k\}$ *and* $\mathcal{A}' = \{x_i \vee \bar{x}_{i+1} | i = 1, \ldots, k\}$ *are strongly equivalent (we consider* $k + 1 = 1$*).*

2. $\mathcal{B} = \{x_1\} \cup \{\bar{x}_i \vee x_{i+1} | i = 1, ..., \ell\}$ *and* $\mathcal{B}' = \{x_i \vee \bar{x}_{i+1} | i = 1, ..., \ell\} \cup \{x_{\ell+1}\}$ *are strongly equivalent.*

Proof. We can assume weights are all equal to 1. For a random truth assignment x^p with $x_i^p = p_i$, let $F_{\mathcal{D}}(x^p) \equiv \sum_{C \in \mathcal{D}} C(x^p)$ be the expected value of x^p for $\mathcal{D}$ ($\mathcal{D} = \mathcal{A}, \mathcal{A}', \mathcal{B}, \mathcal{B}'$). Then, by $k + 1 = 1$, we have

$$F_{\mathcal{A}}(x^p) = \sum_{i=1}^{k}(1 - p_i(1 - p_{i+1})) = k - \sum_{i=1}^{k} p_i + \sum_{i=1}^{k} p_i p_{i+1},$$

$$F_{\mathcal{A}'}(x^p) = \sum_{i=1}^{k}(1 - p_{i+1}(1 - p_i)) = k - \sum_{i=1}^{k} p_i + \sum_{i=1}^{k} p_i p_{i+1},$$

$$F_{\mathcal{B}}(x^p) = p_1 + \sum_{i=1}^{\ell}(1 - p_i(1 - p_{i+1})) = \ell - \sum_{i=2}^{\ell} p_i + \sum_{i=1}^{\ell} p_i p_{i+1},$$

$$F_{\mathcal{B}'}(x^p) = p_{\ell+1} + \sum_{i=1}^{\ell}(1 - p_{i+1}(1 - p_i)) = \ell - \sum_{i=2}^{\ell} p_i + \sum_{i=1}^{\ell} p_i p_{i+1}.$$

Thus, $F_{\mathcal{A}}(x^p) = F_{\mathcal{A}'}(x^p)$ and $F_{\mathcal{B}}(x^p) = F_{\mathcal{B}'}(x^p)$ for any random truth assignment x^p and we have the lemma. □

3 A Refinement of 0.75-Approximation Algorithm of Yannakakis

The 0.75-approximation algorithm of Yannakakis [9] is based on the probabilistic method and divides the variables $X = \{x_1, \ldots, x_n\}$ of a given instance $(\mathcal{C}, w)$ into three groups P', $(P - P') \cup Q$ and Z based on maximum network flows. 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 $x_i \in (P - P') \cup Q$ and $p_i = 1/2$ if $x_i \in Z$. The expected value $F_{\mathcal{C}}(x^p)$ of this random truth assignment $x^p = (p_1, p_2, ..., p_n)$ is shown to satisfy

$$F_{\mathcal{C}}(x^p) \geq \frac{3}{4}W_1^* + \frac{3}{4}W_2^* + \frac{3}{4}W_3^* + \frac{49}{64}W_4^* + \sum_{k \geq 5}(1 - (\frac{3}{4})^k)W_k^* \geq \frac{3}{4}F_{\mathcal{C}}(x^*), \tag{5}$$

where $\mathcal{C}_k$ is the set of clauses in $\mathcal{C}$ with k literals and $W_k^* = \sum_{C \in \mathcal{C}_k} w(C)C(x^*)$ for an optimal truth assignment x^* (and thus, $F_{\mathcal{C}}(x^*) = \sum_{k \geq 1} W_k^*$). The probabilistic method assures that a truth assignment $x^Y \in \{0, 1\}^n$ with value

$$F_{\mathcal{C}}(x^Y) \geq F_{\mathcal{C}}(x^p) \geq 0.75 F_{\mathcal{C}}(x^*)$$

can be obtained in polynomial time. Thus, Yannakakis's algorithm is a 0.75-approximation algorithm. In this section, we will refine Yannakakis's algorithm and find a random truth assignment $x^p = (p_1, p_2, ..., p_n)$ with value

$$F_{\mathcal{C}}(x^p) \geq \frac{3}{4}W_1^* + \frac{3}{4}W_2^* + \frac{31}{40}W_3^* + \frac{101}{128}W_4^* + \frac{1037}{1280}W_5^* + \sum_{k \geq 6}(1 - (\frac{3}{4})^k)W_k^*. \tag{6}$$

To divide the variables X of a given instance $(\mathcal{C}, w)$ into three groups P', $(P - P') \cup Q$ and Z, Yannakakis transformed $(\mathcal{C}, w)$ into an equivalent instance $(\mathcal{C}', w')$ of the weighted clauses with some nice property by using network flows. Our algorithm is also based on network flows and consists of five steps three of which are almost similar to Steps 1-3 of Yannakakis. Let $\mathcal{C}_{1,2} \equiv \mathcal{C}_1 \cup \mathcal{C}_2$ (the set of clauses in $\mathcal{C}$ with one or two literals). As Yannakakis did, we first construct a network $N(\mathcal{C})$ which represents the weighted clauses in $(\mathcal{C}_{1,2}, w)$ as follows. The set of nodes of $N(\mathcal{C})$ consists of the set of literals in $\mathcal{C}$ and two new nodes s and t which represent true (T) and false (F) respectively. The (directed) arcs of $N(\mathcal{C})$ are corresponding to the clauses in $\mathcal{C}_{1,2}$. Each clause $C = x \vee y \in \mathcal{C}_2$ corresponds to two arcs $(\bar{x}, y)$ and $(\bar{y}, x)$ with capacity $cap(\bar{x}, y) = cap(\bar{y}, x) = w(C)/2$ ($\bar{\bar{x}} = x$). Similarly, each clause $C = x \in \mathcal{C}_1$ corresponds to two arcs (s, x) and $(\bar{x}, t)$ with capacity $cap(s, x) = cap(\bar{x}, t) = w(C)/2$. Thus, we can regard a clause $C = x \in \mathcal{C}_1$ as $x \vee F$ when considering a network as above. Note that $N(\mathcal{C})$ is a naturally defined network since $x \vee y = \bar{x} \to y = \bar{y} \to x$.

Two arcs $(\bar{x}, y)$ and $(\bar{y}, x)$ are called *corresponding arcs.* If each corresponding two arcs in a network are of the same capacity, then the network is called *symmetric.* By the above correspondence of a clause and two corresponding arcs, a symmetric network N exactly corresponds to a set $\mathcal{C}(N)$ of weighted clauses with one or two literals. In the case of $N = N(\mathcal{C})$ defined above, we have $\mathcal{C}(N(\mathcal{C})) = (\mathcal{C}_{1,2}, w)$. Thus, we can always construct the set $\mathcal{C}(N)$ of weighted clauses with one or two literals from a symmetric network N and we sometimes use the term "the set of weighted clauses of a symmetric network".

Then we consider a symmetric flow f of maximum value $v(f)$ in $N_0 \equiv N(\mathcal{C})$ from source node s to sink node t (flow f is called *symmetric* if $f(\bar{x}, y) = f(\bar{y}, x)$ for each corresponding arcs $(\bar{x}, y), (\bar{y}, x)$). Let M_0 be the network obtained from the residual network $N_0(f)$ of N_0 with respect to f by deleting all arcs into s and all arcs from t. Then M_0 is clearly symmetric since N_0 is a symmetric network and f is a symmetric flow.

Let $(\mathcal{C}'_{1,2}, w')$ be the set of weighted clauses of the symmetric network M_0 $((\mathcal{C}'_{1,2}, w') = \mathcal{C}(M_0))$ and let $(\mathcal{C}', w')$ be the set of weighted clauses obtained from $(\mathcal{C}, w)$ by replacing $(\mathcal{C}_{1,2}, w)$ with $(\mathcal{C}'_{1,2}, w')$. Then, for each truth assignment $\boldsymbol{x}$,

$$F_{\mathcal{C}}(\boldsymbol{x}) = \sum_{C \in \mathcal{C}} w(C)C(\boldsymbol{x}) = \sum_{C' \in \mathcal{C}'} w'(C')C'(\boldsymbol{x}) + v(f) = F_{\mathcal{C}'}(\boldsymbol{x}) + v(f). \tag{7}$$

Note that (7) holds even if $\boldsymbol{x}$ is a random truth assignment. This can be obtained by Lemma1 using an observation similar to the one in [9]. Note also that, for $\mathcal{A}, \mathcal{A}', \mathcal{B}, \mathcal{B}'$ in Lemma 1, $\mathcal{A}$ corresponds to a cycle and $\mathcal{A}'$ corresponds to the reverse cycle. Similarly, $\mathcal{B}$ corresponds to a path from x_1 to $x_{\ell+1}$ (plus (s, x_1)) and $\mathcal{B}'$ corresponds to the reverse path from $x_{\ell+1}$ to x_1 (plus $(s, x_{\ell+1})$).

Since f is a maximum flow, there is no path from s to t in M_0. Let R be the set of nodes that are reachable from s in M_0 and let $\bar{Y} = \{\bar{y} | y \in Y\}$ for $Y \subseteq X$. Then, there is no arc from a node in R to a node not in R and the set of nodes that can reach t is $\bar{R}$ (in a symmetric network, $x_1, x_2, ..., x_{k-1}, x_k$ is a path if and only if $\bar{x}_k, \bar{x}_{k-1}, ..., \bar{x}_2, \bar{x}_1$ is a path) and R does not contain any complementary literals, since M_0 is a symmetric network and f is a maximum flow ($x, \bar{x} \in R$ implies that there is a path from s to t since M_0 is symmetric and there are paths from s to x (by $x \in R$) and x to t (by $\bar{x} \in R$), which contradicts the maximality of f). This implies that every clause of form $\bar{x} \vee y$ with $x \in R$ satisfies $y \in R$. Thus, we can set all literals of R to be true consistently and, for each truth assignment $\boldsymbol{x}$ in which all literals of R are true, every clause in $\mathcal{C}'_{1,2}$ that contains a literal in $R \cup \bar{R}$ is satisfied. ¿From now on we assume that all literals in R are unnegated ($R \subseteq X$ and thus all literals in

$\bar{R}$ are negated).

By the argument above we can summarize Step 0 of our algorithm as follows.

Step 0. Find R and $(\mathcal{C}', w')$ from $(\mathcal{C}, w)$ using the network N_0, a symmetric flow f of N_0 of maximum value and the network M_0 defined above.

Note that, by (7), if we have an α-approximation algorithm for $(\mathcal{C}', w')$, then it will also be an α-approximation algorithm for $(\mathcal{C}, w)$. Thus, for simplicity, we can assume from now on $(\mathcal{C}', w') = (\mathcal{C}, w)$ (and thus, $f = 0$ and $M_0 = N_0$) and have the following assumption.

Assumption. $\mathcal{C}$ and $N_0 = N(\mathcal{C})$ satisfy:

(a) $R \subseteq X$ and $x \in R$ for each $C = x \in \mathcal{C}$ (there are arcs $(s, x), (\bar{x}, t)$).

(b) $y \in R$ for each $C = \bar{x} \vee y \in \mathcal{C}$ with $x \in R$ (there is no arc going outside from a node in R).

To obtain a 0.75-approximation algorithm, Yannakakis tried to set each variable in R to be true with probability 3/4 and each variable in $X - R$ to be true with probability 1/2. Then the probability of a clause in $\mathcal{C}_{1,2}$ being satisfied is at least 3/4. Similarly, the probability of a clause in $\mathcal{C}$ with five or more literals being satisfied is at least 3/4. Clauses satisfied with probability less than 3/4 have 3 or 4 literals and are of form $\bar{x} \vee \bar{y} \vee \bar{z}$ with $x, y, z \in R$ or of form $\bar{x} \vee \bar{y} \vee \bar{z} \vee \bar{u}$ with $x, y, z, u \in R$ or of form $\bar{x} \vee \bar{y} \vee a$ with $x, y \in R$ and $a \in (X \cup \bar{X}) - (R \cup \bar{R})$. Let $\mathcal{A}_k$ be the set of clauses C of form $C = \bar{x}_1 \vee \bar{x}_2 \vee \cdots \vee \bar{x}_k$ with $x_1, x_2, \ldots, x_k \in R$ $(k = 3, 4, 5)$.

To split off such clauses in $\mathcal{A}_3 \cup \mathcal{A}_4 \cup \mathcal{A}_5$, we consider the network N_1 obtained from $M_0 = N_0$ as follows (we split off clauses in $\mathcal{A}_5$ too for later use, although Yannakakis split off the clauses in $\mathcal{A}_3 \cup \mathcal{A}_4$ and did not split off the clauses in $\mathcal{A}_5$). Let M_0^- be the network obtained from M_0 by deleting all arcs from $\bar{R}$ to R, all arcs from $\bar{R}$ to $(X - R) \cup (\bar{X} - \bar{R})$ and all arcs from $(X - R) \cup (\bar{X} - \bar{R})$ to R. Let $(\mathcal{C}_{1,2}^-, w) = \mathcal{C}(M_0^-)$ (the set of weighted clauses of the symmetric network M_0^-). N_1 is the network obtained from M_0^- as follows. For each clause $C = \bar{x}_1 \vee \bar{x}_2 \vee \cdots \vee \bar{x}_k \in \mathcal{A}_k$ with $x_1, x_2, \ldots, x_k \in R$ $(k = 3, 4, 5)$, we add two nodes $C, \bar{C}$ and $2k + 2$ arcs $(x_1, C), (x_2, C), \ldots, (x_k, C)$, $(\bar{C}, \bar{x}_1), (\bar{C}, \bar{x}_2), \ldots, (\bar{C}, \bar{x}_k)$, $(s, \bar{C}), (C, t)$. Furthermore, we set, for $k = 3, 4$, all arcs of forms (x_i, C) and $(\bar{C}, \bar{x}_i)$ to have capacity $w(C)/(2k)$ and arcs $(s, \bar{C}), (C, t)$ to have capacity $w(C)/2$. If $k = 5$, we set all arcs of forms (x_i, C) and $(\bar{C}, \bar{x}_i)$ to have capacity $w(C)/12$ and arcs $(s, \bar{C}), (C, t)$ to have capacity $5w(C)/12$.

Then, we find a symmetric flow g of maximum value from s to t in N_1 such that $g(x_1, C) = g(x_2, C) = \cdots = g(x_k, C)$ for each clause $C = \bar{x}_1 \vee \bar{x}_2 \vee \cdots \vee \bar{x}_k \in \mathcal{A}_k$ with $x_1, x_2, \ldots, x_k \in R$ $(k = 3, 4, 5)$. Such a flow g can be obtained in a polynomial time by [8]. Let M_1 be the network obtained from the residual network $N_1(g)$ of N_1 with respect to g by deleting all arcs into s, all arcs from t and all nodes $C, \bar{C}$ (and incident arcs) with $C \in \mathcal{A}_3 \cup \mathcal{A}_4 \cup \mathcal{A}_5$.

Now we can split off clauses in $\mathcal{A}_3 \cup \mathcal{A}_4 \cup \mathcal{A}_5$. For each $C = \bar{x}_1 \vee \bar{x}_2 \vee \cdots \vee \bar{x}_k \in \mathcal{A}_k$ with $x_1, x_2, \ldots, x_k \in R$ $(k = 3, 4, 5)$, let $\mathcal{G}^k(C) = \{x_1, x_2, \ldots, x_k, C\}$. The weights of the clauses in $\mathcal{G}^k(C)$ are defined as follows: Let $g(C) = g(x_1, C)$. Then, $w_1(x_1) = w_1(x_2) = \cdots = w_1(x_k) = 2g(C)$ and if $k = 3, 4$ then $w_1(C) = 2kg(C)$ else (i.e., $k = 5$) $w_1(C) = 12g(C)$. Let $\mathcal{G}^3 = \cup_{C \in \mathcal{A}_3} \mathcal{G}^3(C)$, $\mathcal{G}^4 = \cup_{C \in \mathcal{A}_4} \mathcal{G}^4(C)$ and $\mathcal{G}^5 = \cup_{C \in \mathcal{A}_5} \mathcal{G}^5(C)$.

Let $(\mathcal{D}_{1,2}^-, w_1) = \mathcal{C}(M_1)$ (i.e., $(\mathcal{D}_{1,2}^-, w_1)$ is the set of weighted clauses of the symmetric network M_1) and let $(\mathcal{D}, w_1)$ be the set of clauses with weight function w_1 obtained from

$(\mathcal{C}, w)$ by replacing $(\mathcal{C}^-_{1,2}, w)$ with $(\mathcal{D}^-_{1,2}, w_1)$ and by replacing the weight $w(C)$ of each clause $C \in \mathcal{A}_3 \cup \mathcal{A}_4 \cup \mathcal{A}_5$ with

$$w_1(C) = \begin{cases} w(C) - 6g(C) & (C \in \mathcal{A}_3) \\ w(C) - 8g(C) & (C \in \mathcal{A}_4) \\ w(C) - 12g(C) & (C \in \mathcal{A}_5) \end{cases}$$

(note that $w_1(C) \geq 0$ and we assume clauses with weight 0 are not included in $\mathcal{D}$).

Then $(\mathcal{C}, w)$ and $(\mathcal{C}^1 \equiv \mathcal{D} \cup \mathcal{G}^3 \cup \mathcal{G}^4 \cup \mathcal{G}^5, w_1)$ are shown to be strongly equivalent based on Lemma 1 (note that a clause $C \in \mathcal{C}_k$ with $k = 3, 4, 5$ may be split off and appear in two groups of $\mathcal{C}^1$, for example, in $\mathcal{D}$ and $\mathcal{G}^3$, but the total weight of C is not changed). Let R_1 be the set of nodes reachable from s in M_1. Clearly, $R_1 \subseteq R$ ($\bar{R}_1 \subseteq \bar{R}$). Furthermore, there are no clauses in $\mathcal{D}$ with k ($k = 3, 4, 5$) literals all contained in $\bar{R}_1$ by the maximality of g.

By the argument above, we can summarize Step 1 of our algorithm and have a lemma as follows.

Step 1. Find R_1 and $(\mathcal{C}^1, w_1)$ $(\mathcal{C}^1 = \mathcal{D} \cup \mathcal{G}^3 \cup \mathcal{G}^4 \cup \mathcal{G}^5)$ using the network N_1, a symmetric flow g of N_1 of maximum value and the network M_1 defined above.

Lemma 2 *$(\mathcal{C}, w)$ and $(\mathcal{C}^1, w_1)$ are strongly equivalent. Furthermore, the following statements hold.*

(a) $x \in R_1$ for each $C = x \in \mathcal{D}$.

(b) $y \in R_1$ for each $C = \bar{x} \vee y \in \mathcal{D}$ with $x \in R_1$.

(c) there is no clause in $\mathcal{D}$ with 3,4 or 5 literals all contained in $\bar{R}_1$.

(d) $R_1 \subseteq R$.

Next we will split off clauses $C \in \mathcal{D}$ such that $C = \bar{x} \vee \bar{y} \vee a$ with $x, y \in R_1$ and $a \in Z_1 \cup \bar{Z}_1$ ($Z_1 \equiv X - R_1$). Let $\mathcal{B}_3$ be the set of such clauses in $\mathcal{D}$. Let M_1^+ be the network obtained from the network $N(\mathcal{D})$ representing the set of weighted clauses in $\mathcal{D}$ with one or two literals by deleting all arcs from $\bar{X} \cup Z_1$ to R_1 and all arcs from $\bar{R}_1$ to $Z_1 \cup \bar{Z}_1$. Let $(\mathcal{D}'_{1,2}, w_1) = \mathcal{C}(M_1^+)$ (the set of weighted clauses of the symmetric network M_1^+). Let N_2 be the network obtained from M_1^+ as follows. For each clause $C = \bar{x} \vee \bar{y} \vee a \in \mathcal{B}_3$, we add two nodes $C, \bar{C}$ and 8 arcs $(x, C), (y, C), (C, a), (C, t), (\bar{C}, \bar{x}), (\bar{C}, \bar{y}), (\bar{a}, \bar{C}), (s, \bar{C})$ all with capacity $w_1(C)/4$. Then, we find a symmetric flow h of maximum value such that $h(x, C) = h(y, C) = h(C, a) = h(C, t)$ for each clause $C = \bar{x} \vee \bar{y} \vee a \in \mathcal{B}_3$. Let M_2 be the network obtained from the residual network $N_2(h)$ with respect to h by deleting all arcs into s, all arcs from t and all nodes $C, \bar{C}$ (and incident arcs) with $C = \bar{x} \vee \bar{y} \vee a \in \mathcal{B}_3$.

Now we can split off clauses $C \in \mathcal{B}_3$. For each $C = \bar{x} \vee \bar{y} \vee a \in \mathcal{B}_3$, using $h(C) \equiv h(x, C)$, let $\mathcal{H}(C) = \{x, y, \bar{a}, C, x_0, \bar{x}_0\}$ with weights $w_2(x) = w_2(y) = w_2(\bar{a}) = 2h(C)$, $w_2(C) = 4h(C)$ and $w_2(x_0) = w_2(\bar{x}_0) = -h(C)$ (x_0 is any variable in X and the negative weights are accepted in this case). Let $\mathcal{H} = \cup_{C \in \mathcal{B}_3} \mathcal{H}(C)$. Let $(\mathcal{E}'_{1,2}, w_2) = \mathcal{C}(M_2)$ (the set of weighted clauses of the symmetric network M_2) and let $(\mathcal{E}, w_2)$ be the set of weighted clauses obtained from $(\mathcal{D}, w_1)$ by replacing $(\mathcal{D}'_{1,2}, w_1)$ with $(\mathcal{E}'_{1,2}, w_2)$ and by replacing the weight $w_1(C)$ of each clause $C \in \mathcal{B}_3$ with $w_2(C) = w_1(C) - 4h(C) \geq 0$ (we assume clauses with weight 0 are not included in $\mathcal{E}$).

Then, by the same argument as before, $(\mathcal{D}, w_1)$ and $(\mathcal{E} \cup \mathcal{H}, w_2)$ are shown to be strongly equivalent based on Lemma 1. Let R_2 be the set of nodes reachable from s in M_2. Clearly, $R_2 \subseteq R_1$ $(\bar{R}_2 \subseteq \bar{R}_1)$. A node $a \in Z_1 \cup \bar{Z}_1 \cup (R_1 - R_2)$ is called *uncovered* if there is a clause $C = \bar{x} \vee \bar{y} \vee a \in \mathcal{E}$ such that $x, y \in R_2$ $(w_2(C) > 0)$. Let Q'_2 be the set of nodes in $Z_1 \cup \bar{Z}_1 \cup (R_1 - R_2)$ that are reachable from an uncovered node by a path in M_2. Let R' be the set of nodes $a \in R_1 - R_2$ such that there is a clause $C = \bar{x} \vee a \in \mathcal{E}$ with $x \in Q'_2 - (R_1 - R_2)$ (note that such arcs from $Q'_2 - (R_1 - R_2)$ to $(R_1 - R_2)$ are deleted in M_1^+) and let R'_2 be the set of nodes in $(R_1 - R_2)$ that are reachable from a node in R' by a path in M_2. Let $Q_2 = R'_2 \cup Q'_2$. Then, by the symmetry and maximality of h, Q'_2 and Q_2 contain no complementary literals and we can assume all literals in Q_2 are unnegated. Note that some variable in $R - R_1$ will be in Q_2 and we have to correct the previous assumption that $R \subseteq X$. It suffices to assume that $R_1 \subseteq X$ (not $R \subseteq X$) in the argument below.

By the argument above we can summarize Step 2 of our algorithm and have a lemma as follows.

Step 2. Find R_2, Q_2 and $(\mathcal{E} \cup \mathcal{H}, w_2)$ from $(\mathcal{D}, w_1)$ using the network M_1^+, N_2, a symmetric flow h of N_2 of maximum value and the network M_2 defined above.

Lemma 3 *Let $\mathcal{C}^2 = \mathcal{E} \cup \mathcal{H} \cup \mathcal{G}^3 \cup \mathcal{G}^4 \cup \mathcal{G}^5$ and let the weight function w_2 be generalized to be the same as w_1 for $\mathcal{G}^3 \cup \mathcal{G}^4 \cup \mathcal{G}^5$. Then $(\mathcal{C}, w)$ and $(\mathcal{C}^2, w_2)$ are strongly equivalent. Furthermore, the following statements hold.*

(a) $x \in R_2$ for each $C = x \in \mathcal{E}$.

(b) $y \in R_2$ for each $C = \bar{x} \vee y \in \mathcal{E}$ with $x \in R_2$.

(c) $y \in Q_2 \cup R_2$ for each $C = \bar{x} \vee y \in \mathcal{E}$ with $x \in Q_2$.

(d) there is no clause in $\mathcal{E}$ with 3,4 or 5 literals all contained in $\bar{R}_2$.

(e) $a \in Q_2 \cup R_2$ for each $C = \bar{x} \vee \bar{y} \vee a \in \mathcal{E}$ with $x, y \in R_2$.

(f) $R_2 \subseteq R_1$ and $Q_2 \subseteq X - R_2$.

Now we would like to set each variable in R_2 to be true with probability 3/4, each variable in Q_2 to be true with probability 3/5 and each variable in $Z_2 \equiv X - (Q_2 \cup R_2)$ to be true with probability 1/2. Then, each clause in $\mathcal{E}$ except for a clause C of form $C = \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3$ with $x_1 \in R_2$ and $x_2, x_3 \in Q_2$ or of form $C = \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4$ with $x_1, x_2, x_3 \in R_2$ and $x_4 \in Q_2$ is satisfied with probability at least 3/4.

Thus, we will try to split off such clauses. Let $\mathcal{A}'_3$ be the set of clauses $C \in \mathcal{E}$ of form $C = \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3$ with $x_1 \in R_2$ and $x_2, x_3 \in Q_2$. Similarly, let $\mathcal{A}'_4$ be the set of clauses $C \in \mathcal{E}$ of form $C' = \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4$ with $x_1, x_2, x_3 \in R_2$ and $x_4 \in Q_2$. Let $\mathcal{B}'_3$ be the set of clauses $C \in \mathcal{E}$ of form $C = \bar{x}_1 \vee \bar{x}_2 \vee a$ with $x_1, x_2 \in R_2$ and $a \in Q_2$.

Let M_2^+ be the network obtained from $N(\mathcal{E})$ by deleting all arcs from $\bar{X} \cup Q_2 \cup Z_2$ to R_2, all arcs from $\bar{X} \cup Z_2$ to Q_2 and their symmetric arcs. Let $(\mathcal{E}''_{1,2}, w_2) = \mathcal{C}(M_2^+)$ (the set of weighted clauses of the symmetric network M_2^+) and let N_3 be the network obtained from M_2^+ as follows. For each clause $C = \bar{x}_1 \vee \bar{x}_2 \vee a \in \mathcal{B}'_3$ with $x_1, x_2 \in R_2$ and $a \in Q_2$, we add two nodes $C, \bar{C}$ and 8 arcs $(x_1, C), (x_2, C), (C, a), (C, t)$, $(\bar{C}, \bar{x}_1), (\bar{C}, \bar{x}_2), (\bar{a}, \bar{C}), (s, \bar{C})$ all with capacity $w_2(C)/4$. For each clause $C = \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \in \mathcal{A}'_3$ with $x_1 \in R_2$ and $x_2, x_3 \in Q_2$, we add two nodes $C, \bar{C}$, 6 arcs $(x_1, C), (x_2, C), (x_3, C)$, $(\bar{C}, \bar{x}_1), (\bar{C}, \bar{x}_2), (\bar{C}, \bar{x}_3)$ all with capacity $w_2(C)/6$ and two arcs $(s, \bar{C}), (C, t)$ each with capacity $w_2(C)/2$. For each clause $C = \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4 \in \mathcal{A}'_4$ with $x_1, x_2, x_3 \in R_2$ and $x_4 \in Q_2$, we add two nodes $C, \bar{C}$, 8 arcs

(x_1, C), (x_2, C), (x_3, C), (x_4, C), $(\bar{C}, \bar{x}_1)$, $(\bar{C}, \bar{x}_2)$, $(\bar{C}, \bar{x}_3)$, $(\bar{C}, \bar{x}_4)$ all with capacity $w_2(C)/8$ and two arcs $(s, \bar{C}), (C, t)$ each with capacity $w_2(C)/2$. Then, we find a symmetric flow h' of maximum value such that $h'(x_1, C) = h'(x_2, C) = h'(C, a) = h'(C, t)$ for each clause $C = \bar{x}_1 \vee \bar{x}_2 \vee a \in \mathcal{B}'_3$ with $x_1, x_2 \in R_2$ and $a \in Q_2$, $h'(x_1, C) = h'(x_2, C) = h'(x_3, C) = h'(C, t)/3$ for each clause $C = \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \in \mathcal{A}'_3$ with $x_1 \in R_2$ and $x_2, x_3 \in Q_2$ and that $h'(x_1, C) = h'(x_2, C) = h'(x_3, C) = h'(x_4, C) = h'(C, t)/4$ for each clause $C = \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4 \in \mathcal{A}'_4$ with $x_1, x_2, x_3 \in R_2$ and $x_4 \in Q_2$. Let M_3 be the network obtained from the residual network $N_3(h')$ with respect to h' by deleting all arcs into s, all arcs from t and all nodes $C, \bar{C}$ (and incident arcs) in $\mathcal{B}'_3 \cup \mathcal{A}'_3 \cup \mathcal{A}'_4$.

Now we can split off clauses $C \in \mathcal{B}'_3 \cup \mathcal{A}'_3 \cup \mathcal{A}'_4$. For each $C = \bar{x}_1 \vee \bar{x}_2 \vee a \in \mathcal{B}'_3$ with $x_1, x_2 \in R_2$ and $a \in Q_2$, let $\mathcal{H}'(C) = \{x_1, x_2, \bar{a}, C, x_0, \bar{x}_0\}$ with weights $w_3(x_1) = w_3(x_2) = w_3(\bar{a}) = 2h'(C)$, $w_3(C) = 4h'(C)$ and $w_3(x_0) = w_3(\bar{x}_0) = -2h'(C)$ using $h'(C) \equiv h'(x_1, C)$ (x_0 is any variable in X). Let $\mathcal{H}' = \cup_{C \in \mathcal{B}'_3} \mathcal{H}'(C)$. For each clause $C \in \mathcal{E}$ of form $C = \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \in \mathcal{A}'_3$ with $x_1 \in R_2$ and $x_2, x_3 \in Q_2$, let $\mathcal{G}'_3(C) = \{x_1, x_2, x_3, C\}$ with weights $w_3(x_1) = w_3(x_2) = w_3(x_3) = 2h'(C)$ and $w_3(C) = 6h'(C)$ using $h'(C) \equiv h'(x_1, C)$. For each clause $C \in \mathcal{E}$ of form $C = \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4 \in \mathcal{A}'_4$ with $x_1, x_2, x_3 \in R_2$ and $x_4 \in Q_2$, let $\mathcal{G}^4(C') = \{x_1, x_2, x_3, x_4, C'\}$ with weights $w_3(x_1) = w_3(x_2) = w_3(x_3) = w_3(x_4) = 2h'(C')$ and $w_3(C) = 8h'(C')$ using $h'(C') \equiv h'(x_1, C')$. Let $\mathcal{G}'^3 = \cup_{C \in \mathcal{A}'_3} \mathcal{G}'^3(C)$ and $\mathcal{G}'^4 = \cup_{C \in \mathcal{A}'_4} \mathcal{G}'^4(C)$.

Let $(\mathcal{F}'_{1,2}, w_3) = \mathcal{C}(M_3)$ (the set of weighted clauses of the symmetric network M_3) and let $(\mathcal{F}, w_3)$ be the set of weighted clauses obtained from $(\mathcal{E}, w_2)$ by replacing $(\mathcal{E}''_{1,2}, w_2)$ with $(\mathcal{F}'_{1,2}, w_3)$ and by replacing the weight $w_2(C)$ of each clause $C \in \mathcal{B}'_3 \cup \mathcal{A}'_3 \cup \mathcal{A}'_4$ with $w_3(C) = w_2(C) - 3h'(C)$ ($C \in \mathcal{A}'_3$) or $w_3(C) = w_2(C) - 4h'(C)$ ($C \in \mathcal{B}'_3 \cup \mathcal{A}'_4$) ($w_3(C) \geq 0$ and we assume clauses with weight 0 are not included in $\mathcal{F}$).

Then, by the same argument as before, we have $(\mathcal{C}, w)$ and $(\mathcal{C}^3, w_3)$ ($\mathcal{C}^3 \equiv \mathcal{F} \cup \mathcal{G}^3 \cup \mathcal{G}^4 \cup \mathcal{G}^5 \cup \mathcal{H} \cup \mathcal{G}'^3 \cup \mathcal{G}'^4 \cup \mathcal{H}'$, $w_3 = w_1$ for $\mathcal{G}^3 \cup \mathcal{G}^4 \cup \mathcal{G}^5$ and $w_3 = w_2$ for $\mathcal{H}$) are strongly equivalent based on Lemma 1. Let R_3 be the set of nodes reachable from s in M_3. Clearly, $R_3 \subseteq R_2$ ($\bar{R}_3 \subseteq \bar{R}_2$). We call a node $a \in Q_2$ an *entrance* if there is a clause $C = \bar{x}_1 \vee \bar{x}_2 \vee a \in \mathcal{F}$ such that $x_1, x_2 \in R_3$ ($w_2(C) > 0$). Let Q_3 be the set of nodes reachable from entrances in M_3. Clearly, $Q_3 \subseteq Q_2$ ($\bar{Q}_3 \subseteq \bar{Q}_2$).

By the argument above, we can summarize Step 3 of our algorithm and a lemma as follows.

Step 3. Find R_3, Q_3 and $(\mathcal{F} \cup \mathcal{G}'^3 \cup \mathcal{G}'^4 \cup \mathcal{H}', w_3)$ from $(\mathcal{E}, w_2)$ using the network M_2^+, N_3, a symmetric flow h' of N_3 of maximum value and the network M_3 defined above.

Lemma 4 *$(\mathcal{C}, w)$ and $(\mathcal{C}^3, w_3)$ ($\mathcal{C}^3 \equiv \mathcal{F} \cup \mathcal{G}^3 \cup \mathcal{G}^4 \cup \mathcal{G}^5 \cup \mathcal{H} \cup \mathcal{G}'^3 \cup \mathcal{G}'^4 \cup \mathcal{H}'$, $w_3 = w_1$ for $\mathcal{G}^3 \cup \mathcal{G}^4 \cup \mathcal{G}^5$ and $w_3 = w_2$ for $\mathcal{H}$) are strongly equivalent. Furthermore, the following statements hold.*

(a) $x \in R_3$ for each $C = x \in \mathcal{F}$.

(b) $y \in R_3$ for each $C = \bar{x} \vee y \in \mathcal{F}$ with $x \in R_3$.

(c) $y \in R_2$ for each $C = \bar{x} \vee y \in \mathcal{F}$ with $x \in R_2 - R_3$.

(d) $y \in Q_3 \cup R_2$ for each $C = \bar{x} \vee y \in \mathcal{F}$ with $x \in Q_3$.

(e) there is no clause in $\mathcal{F}$ with 3, 4 or 5 literals all contained in $\bar{R}_2$.

(f) $a \in Q_3 \cup R_2$ for each $C = \bar{x} \vee \bar{y} \vee a \in \mathcal{F}$ with $x, y \in R_3$.

(g) there is no clause $C \in \mathcal{F}$ of form $C = \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3$ with $x_1 \in R_3$ and $x_2, x_3 \in Q_3 \cup (R_2 - R_3)$ or of form $C = \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4$ with $x_1, x_2, x_3 \in R_3$ and $x_4 \in Q_3 \cup (R_2 - R_3)$.

(h) $R_3 \subseteq R_2$ *and* $Q_3 \subseteq Q_2$.

(i) $\sum_{C \in \mathcal{C}_2} w(C) = \sum_{C \in \mathcal{F}_2} w(C)$ $(\mathcal{F}_2 = \mathcal{C}_2^3)$.

(j) For each clause $C \in C_k$ *with* $k \geq 3$, $w(C) = \sum w_3(C)$ *where summation is taken over for all* $\mathcal{I} = \mathcal{F}, \mathcal{G}^3, \mathcal{G}^4, \mathcal{G}^5, \mathcal{H}, \mathcal{G}'^3, \mathcal{G}'^4, \mathcal{H}'$ *with* $\mathcal{I} \ni C$.

Now we are ready to set the probabilities of variables to be true.

Step 4. Obtain a random truth assignment $\boldsymbol{x}^p$ by setting independently each variable x_i to be true with probability p_i as follows:

$$p_i = \begin{cases} 3/4 & (x_i \in R_3) \\ 3/5 & (x_i \in Q_3 \cup (R_2 - R_3)) \\ 1/2 & (x_i \in Z_3 = X - (R_2 \cup Q_3)). \end{cases}$$

Then find a truth assignment $\boldsymbol{x}^A \in \{0,1\}^n$ with value $F_{\mathcal{C}}(\boldsymbol{x}^A) \geq F_{\mathcal{C}}(\boldsymbol{x}^p)$ by the probabilistic method.

4 Analysis

In this section we consider the expected value $F_{\mathcal{C}}(\boldsymbol{x}^p)$ of the random truth assignment $\boldsymbol{x}^p$ obtained by Step 4. Let $\boldsymbol{x}^*$ be an optimal truth assignment for $(\mathcal{C}, w)$. Then, the random truth assignment $\boldsymbol{x}^p$ satisfies (6), which will be shown below.

Let $\boldsymbol{x}^r$ be any random truth assignment and let $W_k^r(\mathcal{I})$ be the expected value of $\boldsymbol{x}^r$ for weighted clauses in $(\mathcal{I}, w_3)$ with k literals $(\mathcal{I} = \mathcal{F}, \mathcal{G}^3, \mathcal{G}^4, \mathcal{G}^5, \mathcal{H}, \mathcal{G}'^3, \mathcal{G}'^4, \mathcal{H}')$. Similarly, let $W_k^r = W_k^r(\mathcal{C})$ be the expected value of $\boldsymbol{x}^r$ for weighted clauses in $(\mathcal{C}, w)$ with k literals. Thus, $W_k^*(\mathcal{I})$ is the value of the optimal truth assignment $\boldsymbol{x}^*$ for weighted clauses in $(\mathcal{I}, w_3)$ with k literals and $W_k^* = W_k^*(\mathcal{C})$ is the value of $\boldsymbol{x}^*$ for weighted clauses in $(\mathcal{C}, w)$ with k literals. Then we have the following lemmas by Lemma 4 and $(\mathcal{C}, w)$ and $(\mathcal{C}^3, w_3)$ are strongly equivalent.

Lemma 5 *For any random truth assignment* $\boldsymbol{x}^r$*, the following statements hold.*

(a) $W_k^r = W_k^r(\mathcal{C}^3)$ $(W_k^r(\mathcal{C}^3) = \sum_{\mathcal{I} \in \{\mathcal{F}, \mathcal{G}^3, \mathcal{G}^4, \mathcal{G}^5, \mathcal{H}, \mathcal{G}'^3, \mathcal{G}'^4, \mathcal{H}'\}} W_k^r(\mathcal{I}))$ *for all* $k \geq 3$. *More specifically,*

$$\begin{aligned} W_3^r &= W_3^r(\mathcal{F}) + W_3^r(\mathcal{G}^3) + W_3^r(\mathcal{H}) + W_3^r(\mathcal{G}'^3) + W_3^r(\mathcal{H}'), \\ W_4^r &= W_4^r(\mathcal{F}) + W_4^r(\mathcal{G}^4) + W_4^r(\mathcal{G}'^4), \\ W_5^r &= W_5^r(\mathcal{F}) + W_5^r(\mathcal{G}^5) \text{ and} \\ W_k^r &= W_k^r(\mathcal{F}) \text{ for all } k \geq 6. \end{aligned}$$

(b) $W_2^r(\mathcal{C}^3) = W_2^r(\mathcal{F})$ *and* $W_1^r(\mathcal{C}^3) = \sum_{\mathcal{I} \in \{\mathcal{F}, \mathcal{G}^3, \mathcal{G}^4, \mathcal{G}^5, \mathcal{H}, \mathcal{G}'^3, \mathcal{G}'^4, \mathcal{H}'\}} W_1^r(\mathcal{I})$, *i.e.,*

$$W_1^r(\mathcal{C}^3) = W_1^r(\mathcal{F}) + W_1^r(\mathcal{G}^3) + W_1^r(\mathcal{G}^4) + W_1^r(\mathcal{G}^5) + W_1^r(\mathcal{H}) + W_1^r(\mathcal{G}'^3) + W_1^r(\mathcal{G}'^4) + W_1^r(\mathcal{H}').$$

Furthermore, $W_{1,2}^r = W_{1,2}^r(\mathcal{C}^3)$ *where* $W_{1,2}^r \equiv W_1^r + W_2^r$ *and* $W_{1,2}^r(\mathcal{C}^3) \equiv W_1^r(\mathcal{C}^3) + W_2^r(\mathcal{C}^3)$.

Lemma 6 *For the random truth assignment* $\boldsymbol{x}^p$ *obtained in Section 4 and an optimal truth assignment* $\boldsymbol{x}^*$*, if*

$$F_{\mathcal{C}^3}(\boldsymbol{x}^p) \geq \frac{3}{4} W_1^*(\mathcal{C}^3) + \frac{3}{4} W_2^*(\mathcal{C}^3) + \frac{31}{40} W_3^*(\mathcal{C}^3) + \frac{101}{128} W_4^*(\mathcal{C}^3) + \frac{1037}{1280} W_5^*(\mathcal{C}^3) + \sum_{k \geq 6} (1 - (\frac{3}{4})^k) W_k^*(\mathcal{C}^3) \tag{8}$$

then $F_{\mathcal{C}}(\boldsymbol{x}^p)$ *satisfies (6).*

Proof. By Lemma 6, we have $W_1^* + W_2^* = W_1^*(\mathcal{C}^3) + W_2^*(\mathcal{C}^3)$ and $W_k^* = W_k^*(\mathcal{C}^3)$ for all $k \geq 3$ and (8) implies

$$F_{\mathcal{C}^3}(\boldsymbol{x}^p) = F_{\mathcal{C}}(\boldsymbol{x}^p) \geq \frac{3}{4}(W_1^* + W_2^*) + \frac{31}{40}W_3^* + \frac{101}{128}W_4^* + \frac{1037}{1280}W_5^* + \sum_{k\geq 6}(1-(\frac{3}{4})^k)W_k^*$$

by Lemma 5. □

By Lemma 6, we have only to show that $F_{\mathcal{C}^3}(\boldsymbol{x}^p)$ satisfies (8). Furthermore, it suffices to show that each group $\mathcal{I}$ satisfies (8) for $\mathcal{I} = \mathcal{F}, \mathcal{G}^3, \mathcal{G}^4, \mathcal{G}^5, \mathcal{H}, \mathcal{G}'^3, \mathcal{G}'^4, \mathcal{H}'$, since $F_{\mathcal{C}^3}(\boldsymbol{x}^p) = F_{\mathcal{F}}(\boldsymbol{x}^p) + F_{\mathcal{G}^3}(\boldsymbol{x}^p) + F_{\mathcal{G}^4}(\boldsymbol{x}^p) + F_{\mathcal{G}^5}(\boldsymbol{x}^p) + F_{\mathcal{H}}(\boldsymbol{x}^p) + F_{\mathcal{G}'^3}(\boldsymbol{x}^p) + F_{\mathcal{G}'^4}(\boldsymbol{x}^p) + F_{\mathcal{H}'}(\boldsymbol{x}^p)$. Similarly, if each $\mathcal{I}(C)$ with $C \in \mathcal{J}$ satisfies (8) then $\mathcal{I}$ satisfies (8), since $F_{\mathcal{I}}(\boldsymbol{x}^p) = \sum_{C\in\mathcal{J}} F_{\mathcal{I}}(C)(\boldsymbol{x}^p)$ for each pair $(\mathcal{I}, \mathcal{J}) = (\mathcal{G}^k, \mathcal{A}_k), (\mathcal{H}, \mathcal{B}_3), (\mathcal{G}'^{k'}, \mathcal{A}'_{k'}), (\mathcal{H}', \mathcal{B}'_3)$ ($k = 3,4,5$ and $k' = 3,4$). Thus, for simplicity, we assume the following (in fact, we can always assume so without loss of generality in our argument below):

$\mathcal{G}^3 = \{x_1, x_2, x_3, \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3\}$ with $x_1, x_2, x_3 \in R$ of weight K_{G_3} and $\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3$ of weight $3K_{G_3}$,

$\mathcal{G}^4 = \{y_1, y_2, y_3, y_4, \bar{y}_1 \vee \bar{y}_2 \vee \bar{y}_3 \vee \bar{y}_4\}$ with $y_i \in R$ of weight K_{G_4} ($i = 1,2,3,4$) and $\bar{y}_1 \vee \bar{y}_2 \vee \bar{y}_3 \vee \bar{y}_4$ of weight $4K_{G_4}$,

$\mathcal{G}^5 = \{z_1, z_2, z_3, z_4, z_5, \bar{z}_1 \vee \bar{z}_2 \vee \bar{z}_3 \vee \bar{z}_4 \vee \bar{z}_5\}$ with $z_i \in R$ of weight K_{G_5} ($i = 1,2,3,4,5$) and $\bar{z}_1 \vee \bar{z}_2 \vee \bar{z}_3 \vee \bar{z}_4 \vee \bar{z}_5$ of weight $6K_{G_5}$,

$\mathcal{H} = \{x_{h_1}, x_{h_2}, \bar{x}_{h_3}, \bar{x}_{h_1} \vee \bar{x}_{h_2} \vee x_{h_3}, x_0, \bar{x}_0\}$ with $x_{h_1}, x_{h_2} \in R_1$, $x_{h_3} \in Z_1 \cup \bar{Z}_1$ ($Z_1 = X - R_1$) of weight $2K_H$, $\bar{x}_{h_1} \vee \bar{x}_{h_2} \vee x_{h_3}$ of weight $4K_H$ and $x_0, \bar{x}_0$ of weight $-K_H$ (x_0 is any variable in X),

$\mathcal{G}'^3 = \{x'_1, x'_2, x'_3, \bar{x}'_1 \vee \bar{x}'_2 \vee \bar{x}'_3\}$ with $x'_1 \in R_2$, $x'_2, x'_3 \in Q_2$ of weight $K_{G'_3}$ and $\bar{x}'_1 \vee \bar{x}'_2 \vee \bar{x}'_3$ of weight $3K_{G'_3}$,

$\mathcal{G}'^4 = \{y'_1, y'_2, y'_3, y'_4, \bar{y}'_1 \vee \bar{y}'_2 \vee \bar{y}'_3 \vee \bar{y}'_4\}$ with $y'_1, y'_2, y'_3 \in R_2$, $y'_4 \in Q_2$ of weight $K_{G'_4}$, $\bar{y}'_1 \vee \bar{y}'_2 \vee \bar{y}'_3 \vee \bar{y}'_4$ of weight $4K_{G'_4}$,

$\mathcal{H}' = \{x'_{h_1}, x'_{h_2}, \bar{x}'_{h_3}, \bar{x}'_{h_1} \vee \bar{x}'_{h_2} \vee x'_{h_3}, x_0, \bar{x}_0\}$ with $x'_{h_1}, x'_{h_2} \in R_2$, $x'_{h_3} \in Q_2$ of weight $2K_{H'}$, $\bar{x}'_{h_1} \vee \bar{x}'_{h_2} \vee x'_{h_3}$ of weight $4K_{H'}$ and $x_0, \bar{x}_0$ of weight $-2K_{H'}$.

For each set $\mathcal{F}_k$ of the clauses in $\mathcal{F}$ with k literals ($k = 1,2,...$),

$\sum_{C\in\mathcal{F}_1} w_3(C) = \sum_{C\in\mathcal{C}_1} w(C) - 3K_{G_3} - 4K_{G_4} - 5K_{G_5} - 4K_H - 3K_{G'_3} - 4H_{G'_4} - 4K_{H'}$
$\sum_{C\in\mathcal{F}_2} w_3(C) = \sum_{C\in\mathcal{C}_2} w(C)$
$\sum_{C\in\mathcal{F}_3} w_3(C) = \sum_{C\in\mathcal{C}_3} w(C) - 3K_{G_3} - 4K_H - 3K_{G'_3} - 3K_{H'}$
$\sum_{C\in\mathcal{F}_4} w_3(C) = \sum_{C\in\mathcal{C}_4} w(C) - 4K_{G_4} - 4K_{G'_4}$
$\sum_{C\in\mathcal{F}_5} w_3(C) = \sum_{C\in\mathcal{C}_5} w(C) - 6K_{G_5}$
$\mathcal{F}_k = \mathcal{C}_k$ for all $k \geq 6$ (weight of a clause in this class is not changed).

Thus, it is easily shown that

$$F_{\mathcal{G}^3}(\boldsymbol{x}^*) \leq 5K_{G_3},\ F_{\mathcal{G}^4}(\boldsymbol{x}^*) \leq 7K_{G_4},\ F_{\mathcal{G}^5}(\boldsymbol{x}^*) \leq 10K_{G_5},\ F_{\mathcal{H}}(\boldsymbol{x}^*) \leq 7K_H,$$
$$F_{\mathcal{G}'^3}(\boldsymbol{x}^*) \leq 5K_{G'_3},\ F_{\mathcal{G}'^4}(\boldsymbol{x}^*) \leq 7K_{G'_4} \text{ and } F_{\mathcal{H}'}(\boldsymbol{x}^*) \leq 5K_{H'}.$$

Now we will find a lower bound on the expected value of $F_{\mathcal{I}}(\boldsymbol{x}^p)$ for each $(\mathcal{I}, w_3)$. We first consider the expected value $F_{\mathcal{G}^3}(\boldsymbol{x}^p)$ of $\boldsymbol{x}^p$ for $(\mathcal{G}^3 = \{x_1, x_2, x_3, \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3\}, w_3)$. Let $p = \sqrt[3]{p_1p_2p_3}$ and $f(\mathcal{G}^3) = 3K_{G_3}(p + (1-p^3))$. Then

$$F_{\mathcal{G}^3}(\boldsymbol{x}^p) = K_{G_3}(p_1 + p_2 + p_3 + 3(1 - p_1p_2p_3)) \geq f(\mathcal{G}^3)$$

by the arithmetic/geometric mean inequality. Here, $x_i \notin \bar{R}_2$ $(i = 1,2,3)$, since $x_i \in R$ and $x_i \notin \bar{R}_2 \subseteq \bar{R}$ $(i = 1,2,3)$. Thus, $p_i \neq \frac{1}{4}$ and $\frac{2}{5} \leq p_i \leq \frac{3}{4}$. This implies $p \in [\frac{2}{5}, \frac{3}{4}]$ and, in this interval, $f(\mathcal{G}^3)$ takes a minimum value at $p = \frac{3}{4}$. Thus,

$$f(\mathcal{G}^3) \geq 3K_{G_3}(\frac{3}{4} + 1 - (\frac{3}{4})^3) = \frac{255}{64}K_{G_3} = 3.984375K_{G_3}.$$

On the other hand, $F_{\mathcal{G}^3}(\boldsymbol{x}^*) = W_1^*(\mathcal{G}^3) + W_3^*(\mathcal{G}^3)$, $W_1^*(\mathcal{G}^3) = K_{G_3}(x_1^* + x_2^* + x_3^*)$ and $W_3^*(\mathcal{G}^3) = 3K_{G_3}(1 - x_1^*x_2^*x_3^*)$. Note that

$$1 - \prod_{i=1}^{k} x_i^* \leq \min\{1, k - \sum_{i=1}^{k} x_i^*\} \tag{9}$$

for $x_i^* = 0,1$ (this holds even for $0 \leq x_i^* \leq 1$). Thus,

$$\begin{aligned} \frac{3}{4}W_1^*(\mathcal{G}^3) + \frac{31}{40}W_3^*(\mathcal{G}^3) &\leq K_{G_3}(\frac{3}{4}(x_1^* + x_2^* + x_3^*) + \frac{31}{40}(3)\min\{1, 3 - (x_1^* + x_2^* + x_3^*)\}) \\ &\leq K_{G_3}(\frac{3}{4}(2) + \frac{31}{40}(3)) = 3.825K_{G_3} \end{aligned}$$

and we have

$$F_{\mathcal{G}^3}(\boldsymbol{x}^p) \geq f(\mathcal{G}^3) \geq 3.984375K_{G_3} \geq 3.825K_{G_3} \geq \frac{3}{4}W_1^*(\mathcal{G}^3) + \frac{31}{40}W_3^*(\mathcal{G}^3). \tag{10}$$

Similarly, the expected value $F_{\mathcal{G}^4}(\boldsymbol{x}^p)$ of $\boldsymbol{x}^p$ for $(\mathcal{G}^4 = \{x_1, x_2, x_3, x_4, \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4\}, w_3)$ is expressed as follows (for simplicity, we assume $y_i = x_i$).

$$F_{\mathcal{G}^4}(\boldsymbol{x}^p) = K_{G_4}(p_1 + p_2 + p_3 + p_4 + 4(1 - p_1p_2p_3p_4)) \geq f(\mathcal{G}^4)$$

where $p \equiv \sqrt[4]{p_1p_2p_3p_4}$ and $f(\mathcal{G}^4) \equiv 4K_{G_4}(p + (1-p^4))$. For the same reason as above, we have $p \in [\frac{2}{5}, \frac{3}{4}]$ and $f(\mathcal{G}^4)$ takes a minimum value at $p = \frac{2}{5}$. Thus,

$$f(\mathcal{G}^4) \geq 4K_{G_4}(\frac{2}{5} + 1 - (\frac{2}{5})^4) = \frac{3436}{625}K_{G_4} = 5.4976K_{G_4}.$$

On the other hand, $F_{\mathcal{G}^4}(\boldsymbol{x}^*) = W_1^*(\mathcal{G}^4) + W_4^*(\mathcal{G}^4)$, $W_1^*(\mathcal{G}^4) = K_{G_4}(x_1^* + x_2^* + x_3^* + x_4^*)$, $W_4^*(\mathcal{G}^4) = 4K_{G_4}(1 - x_1^*x_2^*x_3^*x_4^*)$ and $1 - x_1^*x_2^*x_3^*x_4^* \leq \min\{1, 4 - (x_1^* + x_2^* + x_3^* + x_4^*)\}$ by (9). Thus,

$$\begin{aligned} &\frac{3}{4}W_1^*(\mathcal{G}^4) + \frac{101}{128}W_4^*(\mathcal{G}^4) \\ \leq\ & K_{G_4}(\frac{3}{4}(x_1^* + x_2^* + x_3^* + x_4^*) + \frac{101}{128}(4)\min\{1, 4 - (x_1^* + x_2^* + x_3^* + x_4^*)\}) \\ \leq\ & K_{G_4}(\frac{3}{4}(3) + \frac{101}{128}(4)) = 5.40625K_{G_4} \end{aligned}$$

and we have

$$F_{\mathcal{G}^4}(\boldsymbol{x}^p) \geq f(\mathcal{G}^4) \geq 5.4976K_{G_4} \geq 5.40625K_{G_4} \geq \frac{3}{4}W_1^*(\mathcal{G}^4) + \frac{101}{128}W_4^*(\mathcal{G}^4). \qquad (11)$$

The expected value $F_{\mathcal{G}^5}(\boldsymbol{x}^p)$ of $\boldsymbol{x}^p$ for $(\mathcal{G}^5 = \{x_1, x_2, x_3, x_4, x_5, \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4 \vee \bar{x}_5\}, w_3)$ is expressed as follows (for simplicity, we assume $z_i = x_i$).

$$F_{\mathcal{G}^5}(\boldsymbol{x}^p) = K_{G_5}(p_1 + p_2 + p_3 + p_4 + p_5 + 6(1 - p_1p_2p_3p_4p_5)) \geq f(\mathcal{G}^5)$$

where $p \equiv \sqrt[5]{p_1p_2p_3p_4p_5}$ and $f(\mathcal{G}^5) \equiv K_{G_5}(5p + 6(1-p^5))$. For the same reason as above, we have $p \in [\frac{2}{5}, \frac{3}{4}]$ and $f(\mathcal{G}^5)$ takes a minimum value at $p = \frac{2}{5}$. Thus,

$$f(\mathcal{G}^5) \geq K_{G_5}(5(\frac{2}{5}) + 6(1 - (\frac{2}{5})^5)) \geq 7.93856K_{G_5}.$$

On the other hand, $F_{\mathcal{G}^5}(\boldsymbol{x}^*) = W_1^*(\mathcal{G}^5) + W_5^*(\mathcal{G}^5)$, $W_1^*(\mathcal{G}^5) = K_{G_5}(x_1^* + x_2^* + x_3^* + x_4^* + x_5^*)$, $W_5^*(\mathcal{G}^5) = 6K_{G_5}(1 - x_1^*x_2^*x_3^*x_4^*x_5^*)$ and $1 - x_1^*x_2^*x_3^*x_4^*x_5^* \leq \min\{1, 5 - (x_1^* + x_2^* + x_3^* + x_4^* + x_5^*)\}$ by (9). Thus,

$$\begin{aligned} & \frac{3}{4}W_1^*(\mathcal{G}^5) + \frac{1037}{1280}W_5^*(\mathcal{G}^5) \\ \leq\ & K_{G_5}(\frac{3}{4}(x_1^* + x_2^* + x_3^* + x_4^* + x_5^*) + \frac{1037}{1280}(6)\min\{1, 5 - (x_1^* + x_2^* + x_3^* + x_4^* + x_5^*)\}) \\ \leq\ & K_{G_5}(\frac{3}{4}(4) + \frac{1037}{1280}(6)) = 7.8609375K_{G_5} \end{aligned}$$

and we have

$$F_{\mathcal{G}^5}(\boldsymbol{x}^p) \geq f(\mathcal{G}^5) \geq 7.93856K_{G_5} \geq 7.8609375K_{G_5} \geq \frac{3}{4}W_1^*(\mathcal{G}^5) + \frac{1037}{1280}W_5^*(\mathcal{G}^5). \qquad (12)$$

The expected value $F_{\mathcal{H}}(\boldsymbol{x}^p)$ of $\boldsymbol{x}^p$ for $(\mathcal{H} = \{x_1, x_2, x_3, \bar{x}_1 \vee \bar{x}_2 \vee x_3\}, w_3)$ is expressed as follows (for simplicity, we assume $x_{h_i} = x_i$).

$$F_{\mathcal{H}}(\boldsymbol{x}^p) = K_H(2(p_1 + p_2 + 1 - p_3) - 1 + 4(1 - p_1p_2(1 - p_3))) \geq f(\mathcal{H})$$

where $p \equiv \sqrt{p_1p_2}$ and $f(\mathcal{H}) \equiv K_H(4p + 2(1-p_3) - 1 + 4(1 - p^2(1-p_3)))$. Here, $x_1, x_2 \in R_1$, $x_3 \in Z_1 \cup \bar{Z}_1$ and thus, $p_1, p_2, p \in [\frac{1}{2}, \frac{3}{4}]$ and $p_3 \in [\frac{2}{5}, \frac{3}{5}]$ and $f(\mathcal{H})$ takes a minimum value at $p = \frac{1}{2}$ and $p_3 = \frac{3}{5}$. Thus,

$$f(\mathcal{H}) \geq K_H(4(\frac{1}{2}) + 2(\frac{2}{5}) - 1 + 4(1 - \frac{1}{4}\frac{2}{5})) = 5.4K_H.$$

On the other hand, $F_{\mathcal{H}}(\boldsymbol{x}^*) = W_1^*(\mathcal{H}) + W_3^*(\mathcal{H})$, $W_1^*(\mathcal{H}) = K_H(2(x_1^* + x_2^* + 1 - x_3^*) - 1)$, $W_3^*(\mathcal{H}) = 4K_H(1 - x_1^*x_2^*(1 - x_3^*))$ and $1 - x_1^*x_2^*(1 - x_3^*) \leq \min\{1, 3 - (x_1^* + x_2^* + (1 - x_3^*))\}$ by (9). Thus,

$$\begin{aligned} & \frac{3}{4}W_1^*(\mathcal{H}) + \frac{31}{40}W_3^*(\mathcal{H}) \\ \leq\ & K_H(\frac{3}{4}(2(x_1^* + x_2^* + 1 - x_3^*) - 1) + \frac{31}{40}(4)\min\{1, 3 - (x_1^* + x_2^* + 1 - x_3^*)\}) \\ \leq\ & K_H(\frac{3}{4}(4 - 1) + \frac{31}{40}(4)) = 5.35K_H \end{aligned}$$

and we have

$$F_{\mathcal{H}}(\boldsymbol{x}^p) \geq f(\mathcal{H}) \geq 5.4K_H \geq 5.35K_H \geq \frac{3}{4}W_1^*(\mathcal{H}) + \frac{31}{40}W_3^*(\mathcal{H}). \tag{13}$$

The expected value $F_{\mathcal{G}'^3}(\boldsymbol{x}^p)$ of $\boldsymbol{x}^p$ for $(\mathcal{G}'^3 = \{x_1, x_2, x_3, \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3\}, w_3)$ is expressed as follows (for simplicity, we assume $x_i' = x_i$).

$$F_{\mathcal{G}'^3}(\boldsymbol{x}^p) = K_{G_3'}(p_1 + p_2 + p_3 + 3(1 - p_1p_2p_3)) \geq f(\mathcal{G}'^3)$$

where $p \equiv \sqrt{p_2p_3}$ and $f(\mathcal{G}'^3) \equiv K_{G_3'}(p_1 + 2p + 3(1 - p_1p^2))$. Since $x_1 \in R_2$, $x_2, x_3 \in Q_2$, we have $p_1 \in [\frac{3}{5}, \frac{3}{4}]$ and $p, p_2, p_3 \in [\frac{1}{2}, \frac{3}{5}]$ and $f(\mathcal{G}'^3)$ takes a minimum value at $p_1 = \frac{3}{4}$ and $p = \frac{3}{5}$. Thus,

$$f(\mathcal{G}'^3) \geq K_{G_3'}(\frac{3}{4} + 2(\frac{3}{5}) + 3(1 - \frac{3}{4}(\frac{2}{5})^2)) = 4.14K_{G_3'}.$$

On the other hand, for the same reason as for $\mathcal{G}_3$, we have $\frac{3}{4}W_1^*(\mathcal{G}'^3) + \frac{31}{40}W_3^*(\mathcal{G}'^3) \leq K_{G_3'}(\frac{3}{4}(2) + \frac{31}{40}(3)) = 3.825K_{G_3'}$ and

$$F_{\mathcal{G}'^3}(\boldsymbol{x}^p) \geq f(\mathcal{G}'^3) \geq 4.14K_{G_3'} \geq 3.825K_{G_3'} \geq \frac{3}{4}W_1^*(\mathcal{G}'^3) + \frac{31}{40}W_3^*(\mathcal{G}'^3). \tag{14}$$

The expected value $F_{\mathcal{G}'^4}(\boldsymbol{x}^p)$ of $\boldsymbol{x}^p$ for $(\mathcal{G}'^4 = \{x_1, x_2, x_3, x_4, \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4\}, w_3)$ is expressed as follows (for simplicity, we assume $y_i' = x_i$).

$$F_{\mathcal{G}'^4}(\boldsymbol{x}^p) = K_{G_4'}(p_1 + p_2 + p_3 + p_4 + 4(1 - p_1p_2p_3p_4)) \geq f(\mathcal{G}'^4)$$

where $p \equiv \sqrt[3]{p_1p_2p_3}$ and $f(\mathcal{G}'^4) \equiv K_{G_4'}(3p + p_4 + 4(1 - p^3p_4))$. Since $x_1, x_2, x_3 \in R_2$, $x_4 \in Q_2$, we have $p_1, p_2, p_3, p \in [\frac{3}{5}, \frac{3}{4}]$ and $p_4 \in [\frac{1}{2}, \frac{3}{5}]$ and $f(\mathcal{G}'^4)$ takes a minimum value at $p = \frac{3}{4}$ and $p_4 = \frac{3}{5}$. Thus, $f(\mathcal{G}'^4) \geq K_{G_4'}(3(\frac{3}{4}) + \frac{3}{5} + 4(1 - (\frac{3}{4})^3\frac{3}{5})) = 5.8375K_{G_4'} \geq 5.40625K_{G_4'}$. On the other hand, for the same reason as for $\mathcal{G}_4$, we have $\frac{3}{4}W_1^*(\mathcal{G}'^4) + \frac{101}{128}W_4^*(\mathcal{G}'^4) \leq K_{G_4'}(\frac{3}{4}(3) + \frac{101}{128}(4)) = 5.40625K_{G_4'}$ and

$$F_{\mathcal{G}'^4}(\boldsymbol{x}^p) \geq f(\mathcal{G}'^4) \geq 5.8375K_{G_4'} \geq 5.40625K_{G_4'} \geq \frac{3}{4}W_1^*(\mathcal{G}'^4) + \frac{101}{128}W_4^*(\mathcal{G}'^4). \tag{15}$$

The expected value $F_{\mathcal{H}'}(\boldsymbol{x}^p)$ of $\boldsymbol{x}^p$ for $(\mathcal{H}' = \{x_1, x_2, x_3, \bar{x}_1 \vee \bar{x}_2 \vee x_3\}, w_3)$ is expressed as follows (for simplicity, we assume $x_{h_i'} = x_i$).

$$F_{\mathcal{H}'}(\boldsymbol{x}^p) = K_{H'}(2(p_1 + p_2 + 1 - p_3) - 2 + 4(1 - p_1p_2(1 - p_3))) \geq f(\mathcal{H}')$$

where $p \equiv \sqrt{p_1p_2}$ and $f(\mathcal{H}') \equiv K_{H'}(4p + 2(1 - p_3) - 2 + 4(1 - p^2(1 - p_3)))$. Since $x_1, x_2 \in R_1$, $x_3 \in Z_1 \cup \bar{Z}_1$, we have $p_1, p_2, p \in [\frac{3}{5}, \frac{3}{4}]$ and $p_3 \in [\frac{1}{2}, \frac{3}{5}]$ and $f(\mathcal{H})$ takes a minimum value at $p = \frac{3}{5}$ and $p_3 = \frac{3}{5}$. Thus,

$$f(\mathcal{H}') \geq K_{H'}(4(\frac{3}{5}) + 2(\frac{1}{2}) - 2 + 4(1 - \frac{9}{25}\frac{2}{5})) = 4.624K_{H'}.$$

On the other hand, for the same reason as for $\mathcal{H}$, we have $\frac{3}{4}W_1^*(\mathcal{H}') + \frac{31}{40}W_3^*(\mathcal{H}') \leq K_H(\frac{3}{4}(4 - 2) + \frac{31}{40}(4)) = 4.6K_H'$ and

$$F_{\mathcal{H}'}(\boldsymbol{x}^p) \geq f(\mathcal{H}') \geq 4.624K_H' \geq 4.6K_H' \geq \frac{3}{4}W_1^*(\mathcal{H}') + \frac{31}{40}W_3^*(\mathcal{H}'). \tag{16}$$

Let $W_k(\mathcal{F}) = \sum_{C\in\mathcal{F}_k} w_3(C)$. Then $W_k(\mathcal{F}) \geq W_k^*(\mathcal{F}) = \sum_{C\in\mathcal{F}_k} w_3(C)C(x^*)$. Furthermore, by Lemma 4, the expected value $F_{\mathcal{F}_k}(\boldsymbol{x}^p)$ of $\boldsymbol{x}^p$ for $(\mathcal{F}_k, w_3)$ satisfies

$$F_{\mathcal{F}_k}(\boldsymbol{x}^p) \geq \delta_k W_k(\mathcal{F}) \geq \delta_k W_k^*(\mathcal{F}), \tag{17}$$

where

$$\delta_1 = \delta_2 = \tfrac{3}{4},\ \delta_3 = \tfrac{31}{40},\ \delta_4 = \tfrac{101}{128},\ \delta_5 = \tfrac{1037}{1280} \text{ and } \delta_k = 1 - (\tfrac{3}{4})^k \quad (k \geq 6).$$

Thus, we have shown that each group $\mathcal{I}$ satisfies (8) for $\mathcal{I} = \mathcal{F}, \mathcal{G}^3, \mathcal{G}^4, \mathcal{G}^5, \mathcal{H}, \mathcal{G}'^3, \mathcal{G}'^4, \mathcal{H}'$ by (10) through (17) and that, by Lemma 6, $F_{\mathcal{C}^3}(\boldsymbol{x}^p)$ of $\boldsymbol{x}^p$ satisfies (6), i.e.,

$$F_{\mathcal{C}}(\boldsymbol{x}^p) = F_{\mathcal{C}}^3(\boldsymbol{x}^p) \geq \frac{3}{4}W_1^* + \frac{3}{4}W_2^* + \frac{31}{40}W_3^* + \frac{101}{128}W_4^* + \frac{1037}{1280}W_5^* + \sum_{k\geq 6}(1-(\frac{3}{4})^k)W_k^*.$$

5 0.767-Approximation Algorithm

In this section we give an 0.767-approximation algorithm which is obtained by combining the modified Yannakakis's algorithm presented in Section 3 with the algorithm proposed in [1]. In their algorithm in [1], they have considered the following relaxation of MAX SAT for $(\mathcal{C}, w)$ which is based on the linear programming relaxation and the semidefintie programming method [3],[4].

$$
\begin{aligned}
(S):\quad & \text{Maximize} && \sum_{C_j\in\mathcal{C}} w(C_j)z_j && (18)\\
& \text{subject to:} && \sum_{x_i\in X_j^+}\frac{1+y_{0i}}{2} + \sum_{x_i\in X_j^-}\frac{1-y_{0i}}{2} \geq z_j \quad \forall C_j \in \mathcal{C} && (19)\\
& && \frac{2^{k+1}}{4k}c_j^{(1)}(Y) \geq z_j \qquad \forall C_j \in \mathcal{C}_k,\ \forall k \geq 1 && (20)\\
& && y_{ii} = 1 \qquad 0 \leq \forall i \leq n &&\\
& && 0 \leq z_j \leq 1 \qquad \forall C_j \in \mathcal{C} &&\\
& && Y = (y_{i_1 i_2}) \text{ is a symmetric, positive semidefinite matrix.} &&
\end{aligned}
$$

We briefly review the notation in the above problem (S). Variables $\boldsymbol{y} = (y_0, y_1, \cdots, y_n)$ corresponding to

$$y_0 y_i \equiv 2x_i - 1 \text{ with } |y_0| = |y_i| = 1 \tag{21}$$

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 $\boldsymbol{y} = (y_0, y_1, \cdots, y_n)$ as follows by (1):

$$C_j = C_j(\boldsymbol{y}) = 1 - \prod_{x_i\in X_j^+}\frac{1-y_0y_i}{2}\prod_{x_i\in X_j^-}\frac{1+y_0y_i}{2}. \tag{22}$$

Let $c_j^{(1)}(\boldsymbol{y})$ be the sum of the terms in $C_j(\boldsymbol{y})$ of forms $1 \pm y_0y_i$ and $1 \pm y_{i_1}y_{i_2}$, i.e., for $C_j \in \mathcal{C}_k$,

$$
\begin{aligned}
c_j^{(1)}(\boldsymbol{y}) &= \frac{1}{2^k}\sum_{x_i\in X_j^+}(1+y_0y_i) + \frac{1}{2^k}\sum_{x_i\in X_j^-}(1-y_0y_i) + \frac{1}{2^k}\sum_{x_{i_1},x_{i_2}\in X_j^+}(1-y_{i_1}y_{i_2})\\
&\quad + \frac{1}{2^k}\sum_{x_{i_1},x_{i_2}\in X_j^-}(1-y_{i_1}y_{i_2}) + \frac{1}{2^k}\sum_{x_{i_1}\in X_j^+,x_{i_2}\in X_j^-}(1+y_{i_1}y_{i_2}).
\end{aligned}\tag{23}
$$

Using an $(n+1)$-dimensional vector $\boldsymbol{v}_i$ with norm $\|\boldsymbol{v}_i\| = 1$ corresponding to y_i with $|y_i| = 1$, we replace $y_{i_1}y_{i_2}$ with an inner vector product $\boldsymbol{v}_{i_1} \cdot \boldsymbol{v}_{i_2}$ and set $y_{i_1i_2} = \boldsymbol{v}_{i_1} \cdot \boldsymbol{v}_{i_2}$. Then, the matrix $Y = (y_{i_1i_2})$ is symmetric and positive semidefinite since $Y = \boldsymbol{v}^T\boldsymbol{v}$ for $\boldsymbol{v} = (\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_n)$ and $c_j^{(1)}$ is a function of Y.

The first constraints (19) imply 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 $\boldsymbol{x}$. The second constraints are introduced in [1] and serve as a kind of approximation of original MAX SAT constraints. Of course, they hold for any truth assignment $\boldsymbol{x}$. The second constraint (20) is the same as the first one for a clause C_j with one literal ($z_j \leq C_j(Y)$). The other constraints also hold for any truth assignment and thus, (S) can be considered to a relaxation of MAX SAT. In this paper we use the following relaxation of MAX SAT.

$$
\begin{array}{lll}
(T): & \text{Maximize} & \displaystyle\sum_{C_j \in \mathcal{C}_{1,2}} w(C_j)C_j(Y) + \sum_{k\geq 3}\sum_{C_j\in\mathcal{C}_k} w(C_j)z_j \\
& \text{subject to:} & \displaystyle\frac{2^{k+1}}{4k}c_j^{(1)}(Y) \geq z_j \qquad \forall C_j \in \mathcal{C}_k \text{ with } k \geq 3 \qquad (24) \\
& & y_{i_1i_2} + y_{i_2i_3} + y_{i_3i_1} \geq -1, \quad -y_{i_1i_2} + y_{i_2i_3} - y_{i_3i_1} \geq -1, \\
& & -y_{i_1i_2} - y_{i_2i_3} + y_{i_3i_1} \geq -1, \quad y_{i_1i_2} - y_{i_2i_3} - y_{i_3i_1} \geq -1 \\
& & \qquad\qquad 0 \leq \forall i_1 < \forall i_2 < \forall i_3 \leq n \qquad (25) \\
& & y_{ii} = 1 \qquad \forall 0 \leq i \leq n \\
& & 0 \leq z_j \leq 1 \qquad \forall C_j \in \mathcal{C}_k \text{with } k \geq 3 \\
& & Y = (y_{i_1i_2}) \text{ is a symmetric, positive semidefinite matrix.} \qquad (26)
\end{array}
$$

We first show that (T) is a relaxation of MAX SAT. Let $\boldsymbol{x}^q = (x_i^q) \in \{0,1\}^n$ be any truth assignment for $(\mathcal{C}, w)$. Define $Y^q = (y_{i_1i_2})$ to be $y_{0i}^q = 2x_i^q - 1$ and $y_{i_1i_2}^q = y_{0i_1}^q y_{0i_2}^q$ for $0 \leq i_1 \leq i_2 \leq n$. Then $y_{0i}^q \in \{-1,1\}$, $y_{i_1i_2}^q \in \{-1,1\}$ and $y_{ii}^q = 1$. Furthermore, (25) can be shown to be satisfied. For example, $y_{0i_1}^q + y_{0i_2}^q + y_{i_1i_2}^q = 2x_{i_1}^q - 1 + 2x_{i_2}^q - 1 + (2x_{i_1}^q - 1)(2x_{i_2}^q - 1) = (2x_{i_1}^q - 1 + 1)(2x_{i_2}^q - 1 + 1) - 1 = (2x_{i_1}^q)(2x_{i_2}^q) - 1 \geq -1$. 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., all $y_{0i_1}^q$, $y_{0i_2}^q$, $y_{0i_3}^q$ are equal to -1), $y_{i_1i_2}^q + y_{i_2i_3}^q + y_{i_3i_1}^q = 3 \geq -1$. Other cases are similarly shown.

Define $z_j = 1$ if C_j is satisfied by $\boldsymbol{x}$ and $z_j = 0$ otherwise. If C_j is satisfied by $\boldsymbol{x}^q$, 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 [1], $\frac{2^{k+1}}{4k}c_j^{(1)}(Y^q) \geq 1$. Otherwise, all literals in C_j are false and $(1 + y_{0i}^q)/2 = x_i = 0$ and $(1 - y_{0i'}^q)/2 = \bar{x}_{i'}^q = 0$ and $c_j^{(1)}(Y^q) = 0$. Thus, (24) holds. Since $Y^q = (1, y_{01}^q, y_{02}^q, \ldots, y_{0n}^q)^T(1, y_{01}^q, y_{02}^q, \ldots, y_{0n}^q)$, Y^q is a symmetric and positive semidefinite matrix. Thus, (T) was shown to be a relaxation of MAX SAT.

We next show that a solution $(Y, \boldsymbol{z})$ to (T) leads to a solution to (S), that is, $(Y, \boldsymbol{z})$ with appropriately setted z_j for $C_j \in \mathcal{C}_{1,2}$ satisfies (19) and (20). Note that $c_j^{(1)}(Y) = C_j(Y)$ for any $C_j \in \mathcal{C}_{1,2}$ and

$$C_j(Y) = \begin{cases} (1+y_{0i})/2 & (C_j = x_i \in \mathcal{C}_1) \\ (1-y_{0i})/2 & (C_j = \bar{x}_i \in \mathcal{C}_1) \\ (1+y_{0i_1}+1+y_{0i_2}+1-y_{i_1i_2})/4 & (C_j = x_{i_1} \vee x_{i_2} \in \mathcal{C}_2) \\ (1-y_{0i_1}+1+y_{0i_2}+1+y_{i_1i_2})/4 & (C_j = \bar{x}_{i_1} \vee x_{i_2} \in \mathcal{C}_2) \\ (1-y_{0i_1}+1-y_{0i_2}+1-y_{i_1i_2})/4 & (C_j = \bar{x}_{i_1} \vee \bar{x}_{i_2} \in \mathcal{C}_2). \end{cases} \tag{27}$$

Thus, we set $z_j = C_j(Y)$ for each $C_j \in \mathcal{C}_{1,2}$. Then, clearly (19) and (20) are satisfied for $C_j \in \mathcal{C}_1$ (in fact, (19) and (20) are the same for $C_j \in \mathcal{C}_1$). Similarly, (20) is satisfied for $C_j \in \mathcal{C}_2$. Note that, for a clause C_j with two literals, (19) is redundant since if $C_j = x_{i_1} \vee x_{i_2}$ then

$$\frac{1}{2}(1+y_{0i_1}+1+y_{0i_2}) - \frac{1}{4}(1+y_{0i_1}+1+y_{0i_2}+1-y_{i_1i_2}) = \frac{1}{4}(1+y_{0i_1}+y_{0i_2}+y_{i_1i_2}) \geq 0$$

by (25) (by symmetry we can argue the other cases similarly). Furthermore, for a cluase C_j with one or two literals, $z_j \leq 1$ is automatically satisfied since $C_j(Y) \leq 1$ by (25) and (27), $y_{ii} = 1$ 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 (S) and (Y, z) and (Y, z_S) have the same value. Thus, (Y, z) is an optimal solution to (T) if and only if (Y, z_S) is an optimal solution to (S).

Let $(Y^{\#}, z^{\#})$ be an optimal solution to (T) and let $W_k^{\#}(\mathcal{C})$ be the value of $(Y^{\#}, z^{\#})$ for the weighted clauses in $(\mathcal{C}, w)$ with k literals. Thus, $W_1^{\#}(\mathcal{C}) = \sum_{C \in \mathcal{C}_1} w(C)C(Y^{\#})$, $W_2^{\#}(\mathcal{C}) = \sum_{C \in \mathcal{C}_2} w(C)C(Y^{\#})$ and $W_k^{\#}(\mathcal{C}) = \sum_{C_j \in \mathcal{C}_k} w(C)z_j^{\#}$ for $k \geq 3$. Now we would like to have the following lemma.

Lemma 7 *For the random truth assignment x^p obtained in Section 4 and an optimal solution $(Y^{\#}, z^{\#})$ to (S), the following inequality holds.*

$$F_{\mathcal{C}}(x^p) \geq \frac{3}{4}W_1^{\#} + \frac{3}{4}W_2^{\#} + \frac{31}{40}W_3^{\#} + \frac{101}{128}W_4^{\#} + \frac{1037}{1280}W_5^{\#} + \sum_{k \geq 6}(1-(\frac{3}{4})^k)W_k^{\#}. \tag{28}$$

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

$$\begin{array}{lll} (P): & \text{Maximize} & \sum_{C_j \in \mathcal{C}_{1,2}} w(C_j)C_j(Y) \\ & \text{subject to:} & y_{i_1i_2} + y_{i_2i_3} + y_{i_1i_3} \geq -1, \quad -y_{i_1i_2} + y_{i_2i_3} - y_{i_1i_3} \geq -1, \\ & & -y_{i_1i_2} - y_{i_2i_3} + y_{i_1i_3} \geq -1, \quad y_{i_1i_2} - y_{i_2i_3} - y_{i_1i_3} \geq -1 \\ & & \qquad\qquad 0 \leq \forall i_1 < \forall i_2 < \forall i_3 \leq n \\ & & y_{ii} = 1 \qquad\qquad 0 \leq \forall i \leq n \\ & & Y = (y_{i_1i_2}) \text{ is a symmetric, positive semidefinite matrix.} \end{array}$$

As noted before, for any truth assignment $x = (x_1, x_2, \ldots, x_n)$ for $\mathcal{C}$, $Y = (y_{i_1i_2})$ with $y_{i_1i_2} = y_{i_1}y_{i_2}$, $y_iy_0 = 2x_i - 1$ and $|y_i| = 1$ satisfies the constraints of (P). Furthermore, if $C_j \in \mathcal{C}_{1,2}$ is satisfied by x then $C_j(Y) = 1$. Thus, (P) can be considered to be a relaxation of MAX 2SAT. An optimal solution Y to (P) has the value $F_{\mathcal{C}_{1,2}}(Y) = \sum_{C_j \in \mathcal{C}_{1,2}} w(C_j)C_j(Y)$ at least the value $F_{\mathcal{C}_{1,2}}(\boldsymbol{x}^*) = \sum_{C_j \in \mathcal{C}_{1,2}} w(C_j)C_j(\boldsymbol{x}^*)$ of an optimal truth assignment $\boldsymbol{x}^*$ for $(\mathcal{C}_{1,2}, w)$. Let $\mathcal{C}'_{1,2}$ be a set of weighted clauses obtained from $\mathcal{C}_{1,2}$ by using strongly equivalent tranformations in Lemma 1. Then the MAX 2SAT formulation (P') for $\mathcal{C}'_{1,2}$ is expressed as follows.

$$
\begin{array}{lll}
(P'): & \text{Maximize} & \displaystyle\sum_{C'_j \in \mathcal{C}'_{1,2}} w'(C'_j)C'_j(Y) \\
& \text{subject to:} & y_{i_1i_2} + y_{i_2i_3} + y_{i_1i_3} \geq -1, \quad -y_{i_1i_2} + y_{i_2i_3} - y_{i_1i_3} \geq -1, \\
& & -y_{i_1i_2} - y_{i_2i_3} + y_{i_1i_3} \geq -1, \quad y_{i_1i_2} - y_{i_2i_3} - y_{i_1i_3} \geq -1 \\
& & \qquad\qquad 0 \leq \forall i_1 < \forall i_2 < \forall i_3 \leq n \\
& & y_{ii} = 1 \qquad\qquad 0 \leq \forall i \leq n \\
& & Y = (y_{i_1i_2}) \text{ is a symmetric, positive semidefinite matrix.}
\end{array}
$$

Then we have the following lemma.

Lemma 8 *Two problems (P) and (P') have the same feasible solutions and 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 $\mathcal{C}'_{1,2} = \mathcal{A}' = \{x_i \vee \bar{x}_{i+1} | i = 1, ..., k\}$ (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 $\mathcal{C}'_{1,2} = \mathcal{B}' = \{x_i \vee \bar{x}_{i+1} | i = 1, ..., \ell\} \cup \{x_{\ell+1}\}$ 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 $\mathcal{C}'_{1,2} = \mathcal{A}' = \{x_i \vee \bar{x}_{i+1} | i = 1, ..., k\}$ 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 $\mathcal{C}'_{1,2} = \mathcal{B}'$.

□

Since the transformations described in Section 3 use only the strongly equivalent transformations in Lemma 1, we have the following equivalent MAX SAT formulation (Q) for $(\mathcal{C}^3, w_3)$ by Lemma 8.

$$
\begin{array}{lll}
(Q): & \text{Maximize} & \displaystyle\sum_{C_j \in \mathcal{C}^3_{1,2}} w_3(C_j)C_j(Y) + \sum_{k \geq 3} \sum_{C_j \in \mathcal{C}^3_k} w_3(C_j)z_j \\
& \text{subject to:} & \dfrac{2^{k+1}}{4k} c_j^{(1)}(Y) \geq z_j \qquad \forall C_j \in \mathcal{C}^3_k \text{ with } k \geq 3 \\
& & y_{i_1i_2} + y_{i_2i_3} + y_{i_1i_3} \geq -1, \quad -y_{i_1i_2} + y_{i_2i_3} - y_{i_1i_3} \geq -1, \\
& & -y_{i_1i_2} - y_{i_2i_3} + y_{i_1i_3} \geq -1, \quad y_{i_1i_2} - y_{i_2i_3} - y_{i_1i_3} \geq -1 \\
& & \qquad\qquad 0 \leq \forall i_1 < \forall i_2 < \forall i_3 \leq n \\
& & y_{ii} = 1 \qquad\qquad \forall 0 \leq i \leq n \\
& & 0 \leq z_j \leq 1 \qquad\qquad \forall C_j \in \mathcal{C}^3 \\
& & Y = (y_{i_1i_2}) \text{ is a symmetric, positive semidefinite matrix.}
\end{array}
\tag{29}
$$

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 $(\mathcal{C}^3, w_3)$ (C may be contained in two or more groups in $(\mathcal{C}^3, w_3)$). Thus, the constraints of (T) and (Q) are the same and they have the same optimal solution by Lemma 8, since $(\mathcal{C}_{1,2}, w)$ and $(\mathcal{C}^3_{1,2}, w_3)$ are strongly equivalent.

Since (Q) is a semidefinite programming problem as in [3], 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 (T)). An optimal solution $\boldsymbol{v}^{\#} = (\boldsymbol{v}_0^{\#}, \boldsymbol{v}_1^{\#}, \ldots, \boldsymbol{v}_n^{\#})$ can be obtained by Cholesky decomposition of $Y^{\#} = (y^{\#}_{i_1 i_2})$. Thus,

$$W^{\#}_{1,2}(\mathcal{C}^3) = \sum_{C \in \mathcal{C}^3_{1,2}} w_3(C) C(Y^{\#}) = \sum_{C \in \mathcal{C}_{1,2}} w(C) C(Y^{\#})$$

and

$$W^{\#}_k(\mathcal{C}^3) = \sum_{C_j \in \mathcal{C}^3_k} w_3(C_j) z^{\#}_j.$$

Since $C(Y^{\#}) \leq 1$ for $C \in \mathcal{C}^3_{1,2}$ and $z^{\#}_j \leq 1$ for $C_j \in \mathcal{C}_k$ with $k \geq 3$, $W^{\#}_k(\mathcal{C}_k) \leq W_k = \sum_{C_j \in \mathcal{C}_k} w(C_j)$. By an argument similar to one in Section 4, we have lemma 8 using $\boldsymbol{x}^{\#} = (x^{\#}_i)$ with $x^{\#}_i = \frac{1}{2}(1 + y^{\#}_{0i})$ instead of $\boldsymbol{x}^*$. Note that $z^{\#}_j \leq \sum_{x_i \in X_j^+} x^{\#}_i + \sum_{x_i \in X_j^-} (1 - x^{\#}_i)$ for each $C_j \in \mathcal{C}_k$ with $k \geq 3$ and $z_j \leq \min\{1, \sum_{x_i \in X_j^+} x^{\#}_i + \sum_{x_i \in X_j^-} (1 - x^{\#}_i)\}$.

To achieve the bound 0.767, we consider Algorithm B consisting of the following four algorithms:

(1) set each variable x_i true independently with probability $\frac{1}{2}$;

(2) set x_i true independently with probability $p_i = \frac{1+y^*_{0i}}{2}$ using the optimal solution (Y^*, z^*) to (S);

(3) take a random $(n+1)$-dimensional unit vector $\boldsymbol{r}$ and set x_i true if and only if $\text{sgn}(\bar{\boldsymbol{v}}^*_i \cdot \boldsymbol{r}) = \text{sgn}(\bar{\boldsymbol{v}}^*_0 \cdot \boldsymbol{r})$ using the optimal solution (Y^*, z^*) to (S) and (R') ($\bar{\boldsymbol{v}}^* = (\bar{\boldsymbol{v}}^*_0, \bar{\boldsymbol{v}}^*_1, \ldots, \bar{\boldsymbol{v}}^*_n)$ is obtained by Cholesky decomposition of $\bar{Y}^* = (\bar{y}^*_{i_1 i_2})$ and $y^*_{i_1 i_2} = \boldsymbol{v}^*_{i_1} \cdot \boldsymbol{v}^*_{i_2}$).

(4) set each variable x_i in R_3, $Q_3 \cup (R_2 - R_3)$ or $Z_3 \equiv X - (R_2 \cup Q_3)$ true independently with probability $\frac{3}{4}$, $\frac{3}{5}$ or $\frac{1}{2}$, respectively based on the refinement algorithm in Section 3.

Suppose we use algorithm (i) with probability p_i, where $p_1 + p_2 + p_3 + p_4 = 1$. If we set $p_1 = p_2 = p = 0.269184528$, $p_3 = 0.133774497$ and $p_4 = 1 - 2p - p_3 = 0.327856447$, then

$$W^B \geq \sum_{k \geq 1} (2\beta_k p + \alpha \alpha_k p_3 + \delta_k p_4) W^*_k$$

$(2\beta_k = 1 - \frac{1}{2^k} + 1 - (1 - \frac{1}{k})^k)$. Thus, we obtain Algorithm B is a 0.767198-approximation algorithm, which can be verified by checking

$$2\beta_k p + \alpha \alpha_k p_3 + \delta_k p_4 \geq 0.767198$$

for $k \leq 8$ and noticing that $2\beta_k p + \alpha \alpha_k p_3 + \delta_k p_4$ decreases as k increases, and that, for $k = \infty$, $\beta_k = 1 - \frac{1}{2e}$, $\alpha_k = 0$ and $\delta_k = 1$ and $2\beta_k p + \alpha \alpha_k p_3 + \delta_k p_4 = 0.269184528(2 - \frac{1}{e}) + 1 \geq 0.767198$.

Thus, if we choose the best solution among the solutions obtained by Algorithms (1) – (4) then its value is at least 0.767198 times the value of an optimal solution, and we have the following theorem.

Theorem 1 *A 0.767198-approximation algorithm can be obtained based on the refinement of Yannakakis's algorithm in Section 3.*

6 Concluding Remarks

We have presented a refinement of Yannakakis's algorithm and a 0.767198-approximation algorithm. We believe this approach can be used to further improve the performance guarantee for MAX SAT. For example, if the refinement of Yannakakis's algorithm in this paper is combined with the 0.931-approximation algorithm for MAX 2SAT proposed recently by Feige and Goemans [2], it will lead to a 0.768844-approximation algorithm.

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 and The Telecommunications Advancement Foundation.

参考文献

[1] T. Asano, T. Ono and T. Hirata, Approximation algorithms for the maximum satisfiability problem, In *Proc. 5th SWAT*, 1996, pp.100–111.

[2] 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*, 1995, pp.182–189.

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

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

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

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

[7] S. Mahajan and H. Ramesh, Derandomizing semidefinite programming based approximation algorithms, In *Proc. 36th FOCS*, 1995, pp.162–169.

[8] E. Tardos, A strongly polynomial algorithm for solving combinatoral linear program, *Operations Research*, 11 (1986), pp.250-256.

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