Advanced Algorithms 2024 12/10 2023

Sample Solutions 03

Lecturer: Johannes Lengler *Teaching Assistant: Patryk Morawski*

1 MAX-SAT

1. Consider an arbitrary boolean assignment to the variables $x_1, x_2, \ldots, x_n$. Furthermore, let $i \in [m]$. For this exercise we use the following definitions:

   - L_{ij}: A random variable defined as: $L_{ij} = \begin{cases} 1 & \text{if the } j\text{th literal in clause } i \text{ is satisfied} \\ 0 & \text{otherwise} \end{cases}$
   - C_i : A random variable defined as: $C_i = \begin{cases} 1 & \text{if the clause } c_i \text{ is satisfied} \\ 0 & \text{otherwise} \end{cases}$
   - C: The weighted sum of all C_i's: $C = \sum_{k=1}^{m} w_k \cdot C_k$
   - D_i : A random variable defined as $D_i = 1 - C_i = \begin{cases} 1 & \text{if the clause } c_i \text{ is not satisfied} \\ 0 & \text{otherwise} \end{cases}$
   - D : The weighted sum of all D_i's: $D = \sum_{k=1}^{m} w_k \cdot D_k$
   - $\mathcal{A}_{1/2}$: An algorithm which assigns each boolean variable x the value $True$ with probability 1/2 and then outputs the sum of the weights of all satisfied clauses
   - OPT: An algorithm which outputs the highest possible sum of weights of satisfied clauses possible.

   If we set all boolean variables $x_1, x_2, \ldots, x_n$ to $True$ with probability 1/2, then all literals in each clause will be $True$ with probability 1/2. We have for all $i \in [m]$:

   $$Pr[L_{ij} = 1] \;=\; Pr[L_{ij} = 0] \;=\; \frac{1}{2}.$$

   Assume that clause i has $k \geq 1$ literals. As each clause is the OR of its literals, it is only unsatisfied if all literals evaluate to $False$. Thus, the probability of clause i not being satisfied under a random assignment is:

   $$Pr[D_i = 1] \;=\; \prod_{j=1}^{k} Pr[L_{ij} = 0] \;=\; \frac{1}{2^k} \;\leq\; \frac{1}{2}.$$

   From this we can conclude that the probability of clause i being satisfied is:

   $$Pr[C_i = 1] \;=\; 1 - Pr[D_i = 1] \;=\; 1 - \frac{1}{2^k} \;\geq\; \frac{1}{2}.$$

   We are now able to compute a lower bound on the expected value of the output of algorithm $\mathcal{A}_{1/2}$. Observe that the output of algorithm $\mathcal{A}_{1/2}$ can be expressed by the random variable $C = \sum_{k=1}^{m} w_k \cdot C_k$. The expected value of C is:

   $$\mathbb{E}[C] \;=\; \mathbb{E}\left[\sum_{k=1}^{m} w_k \cdot C_k\right] \;\overset{(1)}{=}\; \sum_{k=1}^{m} w_k \cdot \mathbb{E}[C_k] \;\overset{(2)}{=}\; \sum_{k=1}^{m} w_k \cdot Pr[C_k = 1] \;\overset{(3)}{\geq}\; \frac{1}{2}\sum_{k=1}^{m} w_k,$$

where in (1) we used linearity of expectation, in (2) we used the fact that C_k is an indicator variable and in (3) we used the approximation $Pr[C_k = 1] \geq \frac{1}{2}$.

The maximum possible value which OPT could attain is when all clauses are satisfied. For an arbitrary problem instance I we therefore have:

$$OPT(I) \;\leq\; \sum_{k=1}^{m} w_k.$$

This leads to the conclusion that in expectation, algorithm $\mathcal{A}_{1/2}$ is a 1/2 approximation of OPT:

$$\mathbb{E}[\mathcal{A}_{1/2}(I)] \;=\; \mathbb{E}[C] \;\geq\; \frac{1}{2}\sum_{k=1}^{m} w_k \;\geq\; \frac{1}{2}OPT(I).$$

We now try to improve algorithm $\mathcal{A}_{1/2}$ such that it returns a 0.49 approximation of OPT with probability at least 99%. For this we first look at the expected sum of weights of the unsatisfied clauses of algorithm $\mathcal{A}_{1/2}$. This sum of weights can be expressed by the random variable $D = \sum_{k=1}^{m} w_k \cdot D_k$. Similar to C we can calculate an upper bound of the expected value of D:

$$\mathbb{E}[D] \;=\; \mathbb{E}\left[\sum_{k=1}^{m} w_k \cdot D_k\right] \;\overset{(1)}{=}\; \sum_{k=1}^{m} w_k \cdot \mathbb{E}[D_k] \;\overset{(2)}{=}\; \sum_{k=1}^{m} w_k \cdot Pr[D_k = 1] \;\overset{(3)}{\leq}\; \frac{1}{2}\sum_{k=1}^{m} w_k,$$

where in (1) we used linearity of expectation, in (2) we used the fact that D_k is an indicator variable and in (3) we used the approximation $Pr[D_k = 1] \leq \frac{1}{2}$.

We now look at the probability of D being larger than $0.51 \cdot \sum_{k=1}^{m} w_k$ (in which case $\mathcal{A}_{1/2}$ did not produce a 0.49 approximation). For this we can use Markov's inequality:

$$Pr[D \geq 0.51 \cdot \sum_{k=1}^{m} w_k] \;\leq\; \frac{\mathbb{E}[D]}{0.51 \cdot \sum_{k=1}^{m} w_k} \;\leq\; \frac{\sum_{k=1}^{m} w_k}{2 \cdot 0.51 \cdot \sum_{k=1}^{m} w_k} \;=\; \frac{1}{1.02} \;\leq\; 0.99.$$

To decrease this probability we amplify it by repeating algorithm $\mathcal{A}_{1/2}$ a times and returning the result which yielded the highest sum of weights. For this new algorithm $\mathcal{A}_{1/2}^{a}$ to fail (i.e. not returning a value which is a 0.49 approximation), all a repetitions must return a value which is smaller than a 0.49 approximation. The probability of this event happening is (we denote the ith run of algorithm $\mathcal{A}_{1/2}$ as $D^{(i)}$):

$$Pr[\mathcal{A}_{1/2}^{a}(I) \leq 0.49 \cdot OPT(I)] \;=\; \prod_{i=1}^{a} Pr\left[D^{(i)} \geq 0.51 \sum_{k=1}^{m} w_k\right] \;\leq\; 0.99^{a}.$$

Setting a to 459 will reduce this probability to at most $0.99^{459} \approx 0.0099 < 0.01$ which is lower than 1%. Therefore, by repeating algorithm $\mathcal{A}_{1/2}$ 459 times and outputting the best result of all the runs we get a new algorithm which outputs a 0.49 approximation of OPT with probability at least 99%.

2. Let ILP be the integer linear program which is defined in the exact same way as the linear program in the problem description with the exception that:

$$\forall j \in \{1, 2, ..., m\} : z_j \in \{0, 1\}$$
$$\forall i \in \{1, 2, ..., n\} : y_i \in \{0, 1\}$$

Algorithm 1 $\mathcal{A}_{nr}$ which uses randomized rounding to solve the MAX-SAT problem

function $\mathcal{A}_{nr}$(Input: I = formula in CNF form)
  Transform I to its corresponding linear program form I_{LP}
  $(y^*, z^*) \leftarrow$ Solve linear program I_{LP}
  for $i \in [n]$ **do**
    $x_i \leftarrow \mathbf{1}$ with probability y_i^* (otherwise $\mathbf{0}$)
  end for
  Transform the problem back to a CNF instance I_{ILP}
  return the sum of the weights of all clauses in I_{ILP} which are satisfied
end function

You can see clearly that the linear program in the problem description is a relaxation of ILP (as we only changed the domain of all variables from integers to real numbers).

Let's first explain how ILP relates to our problem of finding a boolean variable assignment which maximizes the total weight of all satisfied clauses. Consider an arbitrary assignment of truth values to the boolean variables $x_1, x_2, ..., x_n$. We can then interpret the variables of ILP as follows:

- y_i: represents the truth value of the boolean variable x_i: $y_i = \begin{cases} 1 & \text{if } x_i = True \\ 0 & \text{otherwise} \end{cases}$
- z_i: represents the truth value of clause c_i: $z_i = \begin{cases} 1 & \text{if clause } c_i \text{ is satisfied} \\ 0 & \text{otherwise} \end{cases}$

ILP has one constraint per clause. This constraint basically restricts the integer linear program to only set z_i to 1 (i.e. marking the clause as satisfied) if at least one of its literals is satisfied (described by the sum of all variables y_j which are contained in clause i.). As an example, let's assume that clause i of our input CNF is:

$$(x_1 \vee x_2 \vee \neg x_3 \vee \neg x_4 \vee x_5).$$

The corresponding constraint in the integer linear program would then be:

$$(y_1 + y_2 + y_5) + ((1 - y_3) + (1 - y_4)) \ \geq z_i,$$

or if you define $S_i^+ = \{y_1, y_2, y_5\}$ and $S_i^- = \{y_3, y_4\}$:

$$\sum_{j \in S_i^+} y_j + \sum_{j \in S_i^-} y_j \geq z_i.$$

Finally, the maximization constraint of ILP is:

$$\text{maximize} \sum_{j=1}^{m} w_j \cdot z_j,$$

which can be interpreted as the goal of maximizing the total sum of all satisfied clauses.

Let us now look at algorithm $\mathcal{A}_{nr}$ which is described in algorithm 1. We are interested in the probability that a clause i with $k \geq 1$ literals $(l_1 \vee l_2 \vee \cdots \vee l_k)$ will be satisfied by algorithm $\mathcal{A}_{nr}$. Let C_i denote the random variable which is 1 if clause i is satisfied by algorithm $\mathcal{A}_{nr}$ and 0 otherwise. Let's first look at the probability that $C_i = 0$ (i.e. that clause i will not be satisfied by algorithm $\mathcal{A}_{nr}$). The literals of clause i can be divided into two groups. The group of literals S_i^+ which consist of a simple boolean variable x

and the literals S_i^- which consist of a negated boolean variable $\neg x$. In order that clause i is not satisfied, all boolean variables of literals in S_i^+ have to be assigned $False$ and all boolean variables of literals in S_i^- have to be assigned $True$. For a literal l let y_l be the variable which corresponds to the boolean variable x of l. Because algorithm $\mathcal{A}_{nr}$ assigns 1 to the boolean variable x_j with probability y_j^* we can express the probability of clause i being unsatisfied as:

$$Pr[C_i = 0] \;=\; \left(\prod_{l \in S_i^+} (1 - y_l^*)\right) \cdot \left(\prod_{l \in S_i^-} y_l^*\right).$$

We will now make use of the AM-GM inequality (Arithmetic Mean - Geometric Mean) which states that for any set of non-negative real numbers $a_1, a_2, \ldots, a_n$ it holds that:

$$\frac{a_1 + a_2 + \cdots + a_n}{n} \;\geq (a_1 \cdot a_2 \cdot \cdots \cdot a_n)^{\frac{1}{n}}$$

As the statement above is basically just a multiplication of k non-negative real numbers we can apply this inequality in the following way:

$$\begin{aligned}
Pr[C_i = 0] \;&=\; \left(\prod_{l \in S_i^+} (1 - y_l^*)\right) \cdot \left(\prod_{l \in S_i^-} y_l^*\right) \;=\; \left(\left(\left(\prod_{l \in S_i^+} (1 - y_l^*)\right) \cdot \left(\prod_{l \in S_i^-} y_l^*\right)\right)^{\frac{1}{k}}\right)^k \\
&\leq\; \left(\frac{1}{k}\left(\underbrace{\sum_{l \in S_i^+} (1 - y_l^*) \;+\; \sum_{l \in S_i^-} y_l^*}_{TEMP}\right)\right)^k.
\end{aligned}$$

We will now transform the expression denoted by $TEMP$ in the previous equation. First note that $|S_i^+| + |S_i^-| = k$ because, by assumption, clause i consists of k literals. Secondly, notice that, by definition of the linear program, we have $y_j^* \leq 1$ for all $j \in [n]$. Thus:

$$\begin{aligned}
k - TEMP &= k - \left(\sum_{l \in S_i^+} (1 - y_l^*) \;+\; \sum_{l \in S_i^-} y_l^*\right) \\
&= \left(|S_i^+| - \sum_{l \in S_i^+} (1 - y_l^*)\right) + \left(|S_i^-| - \sum_{l \in S_i^-} y_l^*\right) = \sum_{l \in S_i^+} y_l^* \;+\; \sum_{l \in S_i^-} (1 - y_l^*).
\end{aligned}$$

Returning to our previous computation of $Pr[C_i = 0]$ we can transform this expression now as follows:

$$\begin{aligned}
Pr[C_i = 0] \;&\leq\; \left(\frac{1}{k} \cdot TEMP\right)^k \;=\; \left(1 - 1 + \frac{1}{k} \cdot TEMP\right)^k \\
&= \left(1 - \frac{1}{k} \cdot k - \frac{1}{k} \cdot (-TEMP)\right)^k \;=\; \left(1 - \frac{1}{k} \cdot (k - TEMP)\right)^k \\
&\leq\; \left(1 - \frac{1}{k} \cdot \left(\sum_{l \in S_i^+} y_l^* \;+\; \sum_{l \in S_i^-} (1 - y_l^*)\right)\right)^k.
\end{aligned}$$

Note, that the inner most expression is one of the constraints of our linear program from the problem statement. As (y^*, z^*) is the optimal solution of this linear program we know that:

$$\sum_{l \in S_i^+} y_l^* \ + \ \sum_{l \in S_i^-} (1 - y_l^*) \ \geq \ z_i^*.$$

The expression above therefore further simplifies to:

$$Pr[C_i = 0] \ \leq \ \left(1 - \frac{1}{k} \cdot \left(\sum_{l \in S_i^+} y_l^* \ + \ \sum_{l \in S_i^-} (1 - y_l^*)\right)\right)^k \ \leq \ \left(1 - \frac{z_i^*}{k}\right)^k.$$

From this it follows that:

$$Pr[C_i = 1] \ = \ 1 - Pr[C_i = 0] \ \geq \ 1 - \left(1 - \frac{z_i^*}{k}\right)^k. \tag{1}$$

Remember that our goal is to prove that:

$$Pr[C_i = 1] \ \geq \ \left(1 - \left(1 - \frac{1}{k}\right)^k\right) \cdot z_i^*.$$

We are going to prove this by proceeding as follows:

a) Express the probability in (1) as a function of z_i^*.
b) Prove that this function is concave on the interval $[0, 1]$.
c) Use a property of concave functions to prove our inequality.

The proof works as follows:

a) The goal is to convert the expression in (1) as a function. For $k \geq 1$, define the function $F_k(z) \ = \ 1 - \left(1 - \frac{z}{k}\right)^k$. Notice that $F_k(0) = 0$.
b) To prove that F_k is concave on the interval $[0, 1]$ for $k \geq 1$ we use the following lemma:

   Lemma 1. *A differentiable function f is concave on an interval $[a, b]$ if and only if its derivative function f' is monotonically decreasing on that interval.*

   For $k \geq 1$ the derivative of F_k is:

   $$\frac{dF_k}{dz}\left(1 - \left(1 - \frac{z}{k}\right)^k\right) \ = \ -k \cdot \left(1 - \frac{z}{k}\right)^{k-1} \cdot \frac{dF_k}{dz}\left(1 - \frac{z}{k}\right) \ = \ \left(1 - \frac{z}{k}\right)^{k-1}.$$

   To prove that this derivative is monotonically decreasing on the interval $[0, 1]$ for all $k \geq 1$ consider the two integers $0 \leq a < b \leq 1$:

   $$a < b \ \Rightarrow \ 1 - \frac{b}{k} < 1 - \frac{a}{k} \ \Rightarrow \ \left(1 - \frac{b}{k}\right)^{k-1} \leq \left(1 - \frac{a}{k}\right)^{k-1} \ \Rightarrow \ \frac{dF_k}{dz}(b) \leq \frac{dF_k}{dz}(a).$$

   which concludes the proof.
c) To finally prove our inequality, we will make use of the following lemma for concave functions:

   Definition 2. *A real-valued function f on an interval is said to be concave if, for any x and y in the interval and for any $\alpha \in [0, 1]$:*

   $$f((1 - \alpha)x + \alpha y) \ \geq (1 - \alpha) \cdot f(x) + \alpha \cdot f(y).$$

As our function F_k is concave on the interval $[0,1]$ for all $k \geq 1$ it must therefore hold that:

$$F_k(z_i^* \cdot 1 + \underbrace{(1 - z_i^*) \cdot 0}_{=0}) \;\geq\; z_i^* \cdot F_k(1) + (1 - z_i^*) \cdot \underbrace{F_k(0)}_{=0}.$$

Applying the definition of function F_k and combining it with the probability $Pr[C_i = 1]$ we finally get:

$$Pr[C_i = 1] \;\geq\; 1 - \left(1 - \frac{z_i^*}{k}\right)^k \;\geq \left(1 - \left(1 - \frac{1}{k}\right)^k\right) \cdot z_i^*.$$

Having computed this upper bound for the probability of a random clause i being satisfied under algorithm $\mathcal{A}_{nr}$, we can now estimate how well of an approximation algorithm $\mathcal{A}_{nr}$ provides in expectation. For this let $obj(x) = \sum_{i=1}^m w_i \cdot x_i$ be the objective function of the (integer) linear program. Furthermore, let (y^*, z^*) be the optimal solution of the linear program and $OPT = (y^{OPT}, z^{OPT})$ the optimal solution of the integer linear program. Notice that $obj(z^{OPT}) \;\leq\; obj(z^*)$ because every feasible solution of ILP is also a feasible solution to the corresponding linear program. Furthermore, notice that the output of algorithm $\mathcal{A}_{nr}$ can be described by the random variable $C = \sum_{k=1}^m w_k \cdot C_k$. The expected value of C can be estimated in the following way:

$$\begin{aligned}
\mathbb{E}[C] \;&=\; \mathbb{E}\left[\sum_{k=1}^m w_k \cdot C_k\right] \;\overset{(1)}{=}\; \sum_{k=1}^m w_k \cdot \mathbb{E}[C_k] \;\overset{(2)}{=}\; \sum_{k=1}^m w_k \cdot Pr[C_k = 1] \\
&\overset{(3)}{\geq} \left(1 - \left(1 - \frac{1}{k}\right)^k\right) \cdot \sum_{k=1}^m w_k \cdot z_k^* \;\overset{(4)}{\geq}\; \left(1 - \frac{1}{e}\right) \cdot \sum_{k=1}^m w_k \cdot z_k^{OPT}.
\end{aligned}$$

where in (1) we used linearity of expectation, in (2) we used the fact that C_k is an indicator variable, in (3) we used the approximation of $Pr[C_k = 1]$ which we proved previously and in (4) we used the fact that $\left(1 - \frac{1}{k}\right)^k \;\leq\; \frac{1}{e}$ for all $k \geq 1$ and that $obj(z^{OPT}) \;\leq\; obj(z^*)$. We have therefore shown that algorithm $\mathcal{A}_{nr}$ yields in expectation an $\left(1 - \frac{1}{e}\right)$ approximation of the optimal result.

3. Let us now try to solve MAX-SAT by combining the algorithms from part 1 and 2. Let's call the algorithm from part 1 `LARGE-SAT` and the algorithm from part 2 `SMALL-SAT`. The idea of the combined algorithm is very simple and the following one: toss a fair coin and, depending on the outcome, choose one the two algorithms of above.

   <u>`MEDIUM-SAT`</u>

   $b \in_R \{0, 1\}$

   if $b = 0$: solve MAX-SAT with `LARGE-SAT`.

   else: solve MAX-SAT with `SMALL-SAT`.

   Theorem 3. *`MEDIUM-SAT` gives a 3/4-approximation for the MAX-SAT problem in expectation.*

   Proof. We know that the following two inequalities must hold by linearity of expectation and what has been established in the previous points.

$$\mathbb{E}\left[\sum_{j\in[m]} w_j z_j \middle| b=0\right] \geq \sum_{j\in[m]} w_j \left(1-2^{-k}\right) \overset{z_j^*\leq 1}{\geq} \sum_{j\in[m]} w_j \left(1-2^{-k}\right) z_j^*$$

$$\mathbb{E}\left[\sum_{j\in[m]} w_j z_j \middle| b=1\right] \geq \sum_{j\in[m]} w_j \left(1-\left(1-\frac{1}{k}\right)^k\right) z_j^*$$

By the law of total expectation, we have that

$$\begin{aligned}
\mathbb{E}\left[\sum_{j\in[m]} w_j z_j\right] &= \Pr[b=0]\cdot \mathbb{E}\left[\sum_{j\in[m]} w_j z_j \middle| b=0\right] + \Pr[b=1]\cdot \mathbb{E}\left[\sum_{j\in[m]} w_j z_j \middle| b=1\right] \\
&\geq \frac{1}{2}\sum_{j\in[m]} w_j z_j^* \left[\left(1-2^{-k}\right)+\left(1-\left(1-\frac{1}{k}\right)^k\right)\right] \geq \frac{1}{2}\sum_{j\in[m]}\frac{3}{2}w_j z_j^* \\
&= \frac{3}{4}\cdot \text{FOPT} \geq \frac{3}{4}\cdot \text{OPT}.
\end{aligned}$$

The second to last inequality comes from the fact that $\left(1-2^{-k}\right)+\left(1-\left(1-\frac{1}{k}\right)^k\right)$ is monotonically non-decreasing in k and evaluates to $\frac{3}{2}$ for $k=1$, $\frac{3}{2}$ for $k=2$ and $\frac{341}{216} > \frac{3}{2}$ for $k=3$, which is already enough. This concludes the proof of the theorem. □

4. Let us consider the following CNF formula

$$F = (x_1 \vee x_2) \wedge (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2).$$

We notice that all 4 possible assignments satisfy exactly 3 of the clauses in F, with the known corresponding LP objective function and LP constraints

$$\begin{aligned}
&y_1 + y_2 \geq z_1 \\
&y_1 + (1-y_2) \geq z_2 \\
&(1-y_1) + y_2 \geq z_3 \\
&(1-y_1) + (1-y_2) \geq z_4 \\
&y_1, y_2, z_1, z_2 \in [0,1]
\end{aligned}$$

This yields optimal solutions

$$\begin{aligned}
&z_1^* = z_2^* = z_3^* = z_4^* = 1 \\
&y_1^* = y_2^* = \frac{1}{2}.
\end{aligned}$$

Thus, F is a CNF formula such that there is a 3/4 gap between the value of the solution of LP described in part 2 and the optimal Boolean assignment to the variables.

5. Similarly to part 2, the general strategy is to bound the probability of each clause j being satisfied as a function of z_j^* and use this to compare the expected value of our solution to the optimum of the LP relaxation.

Recall that we set x_i to 1 with probability $f(y_j)$, where f is an arbitrary function satisfying $f(y) \in [1-4^{-y}, 4^{y-1}]$. The probability of clause j not being satisfied is (omitting the asterisk in y^* and z^* for brevity)

$$\Pr\left[\text{clause } j \text{ is not satisfied}\right] = \prod_{i\in S_j^+}(1-f(y_i)) \cdot \prod_{i\in S_j^-}(f(y_i)).$$

Now we use our assumption on f to bound the probability from above, appropriately substituting the upper or lower bound on f in each of its occurrences:

$$\begin{aligned}\Pr[\text{clause } j \text{ is not satisfied}] &\le \prod_{i\in S_j^+}(1-(1-4^{-y_i}))\cdot\prod_{i\in S_j^-}(4^{y_i-1}) \\ &= \prod_{i\in S_j^+}(4^{-y_i}))\cdot\prod_{i\in S_j^-}(4^{y_i-1}) \\ &= 4^{\alpha_j}\end{aligned}$$

where we define $\alpha_j = \sum_{i\in S_j^+}(-y_i)+\sum_{i\in S_j^-}(y_i-1)$. Now notice that the condition from the LP definition requires that $-\alpha_j \ge z_j$. Therefore we get

$$\begin{aligned}\alpha_j &\le -z_j \\ \Pr[\text{clause } j \text{ is } \textbf{not} \text{ satisfied}] &\le 4^{\alpha_j} \\ &\le 4^{-z_j} \\ \Pr[\text{clause } j \text{ is satisfied}] &\ge 1-4^{-z_j} \\ &\ge \frac{3}{4}z_j\end{aligned}$$

where the last inequality comes from the fact that $h(x) = 1-4^{-x}$ is a concave function, meaning that $h(x) \ge h(0)\cdot x + h(1)\cdot(1-x)$. We now bound the expected value of our solution:

$$\begin{aligned}\mathbb{E}\left[\sum_{j=1}^{m}[\text{clause } j \text{ is satisfied}]\cdot w_j\right] &= \sum_{j=1}^{m}\Pr[\text{clause } j \text{ is satisfied}]\cdot w_j \\ &\ge \frac{3}{4}\sum_{j=1}^{m} z_j w_j \\ &= \frac{3}{4}OPT_{LP} \\ &\ge \frac{3}{4}OPT.\end{aligned}$$

Here OPT_{LP} denotes the value of the optimal LP solution, whereas OPT is the optimal solution to the MAX-SAT problem.