A New $\frac{3}{4}$-Approximation Algorithm for MAX SAT

*Michel X. Goemans** M.I.T.

David P. Williamson† M.I.T.

Abstract

Recently, Yannakakis [5] presented the first $\frac{3}{4}$-approximation algorithm for the Maximum Satisfiability Problem (MAX SAT). His algorithm makes non-trivial use of solutions to max-flow problems. We present a new $\frac{3}{4}$-approximation algorithm that depends solely on the solution to a linear programming relaxation of MAX SAT. The algorithm uses the probabilistic method/randomized rounding in an unusual way. As a by-product, we obtain a tight worst-case analysis of the corresponding duality gap.

Introduction

An instance of the Maximum Satisfiability Problem (MAX SAT) is defined by a collection $\mathcal{C}$ of boolean clauses, where each clause is a disjunction of literals drawn from the set of variables $\{x_1, x_2, \ldots, x_n\}$. A *literal* is either a variable x or its negation $\overline{x}$. In addition, for each clause $C_j \in \mathcal{C}$, there is an associated nonnegative weight w_j. The optimal solution to a MAX SAT instance is an assignment of truth values to variables $x_1, \ldots, x_n$ that maximizes the sum of the weight of the satisfied clauses (i.e. clauses with at least one true literal). MAX SAT is known to be NP-complete, even when each clause contains at most two variables (sometimes called MAX 2SAT) [2]. Hence there is unlikely to be any polynomial-time algorithm that can solve MAX SAT optimally.

Many people, however, have proposed *α-approximation algorithms* for MAX SAT. An α-approximation algorithm for MAX SAT is a polynomial-time algorithm which for every instance produces a truth assignment with weight at least α times the weight of the optimal solution. Johnson [3] demonstrates a $\frac{1}{2}$-approximation algorithm, which is also a $(1 - \frac{1}{2^k})$-approximation algorithm when each clause contains at least k variables. In particular, if $k \geq 2$ the performance guarantee is at least $\frac{3}{4}$. Lieberherr and Specker [4] give a $\frac{\sqrt{5}-1}{2}$-approximation algorithm ($\frac{\sqrt{5}-1}{2} = 0.618\ldots$) when the clause set does not contain both clauses x_i and $\overline{x_i}$ for any i. Recently Yannakakis improved on both of these results by showing a $\frac{3}{4}$-approximation algorithm [5]. Yannakakis' algorithm transforms a MAX SAT instance into an equivalent instance (in terms of approximability) which does not contain any unit clauses (i.e. clauses with only one

*Address: M.I.T. Room 2-372, 77 Massachusetts Ave., Cambridge, MA, 02139. Email: `goemans@math.mit.edu`. Research supported in part by Air Force contract AFOSR-89-0271 and DARPA contract N00014-89-J-1988.

†Address: Laboratory for Computer Science Room 301, 545 Technology Square, Cambridge, MA 02139. Email: `dpwmson@theory.lcs.mit.edu`. Research partially supported by an NSF Graduate Fellowship and DARPA contract N00014-89-J-1988.

literal). In conjunction with Johnson's algorithm, this leads to the improved performance guarantee. The algorithm uses max flow computations in an elegant way to transform MAX 2SAT instances. However, the transformation becomes more complicated when general clauses are introduced.

The purpose of this note is to present a new $\frac{3}{4}$-approximation algorithm which is conceptually simple for all MAX SAT instances. The algorithm presented here relies solely on applying the probabilistic method to the solution of a single linear program that is a linear programming relaxation of a formulation for the MAX SAT problem. The use of the probabilistic method involves a twist on the standard randomized rounding technique that is interesting in its own right.

The note is structured as follows. In Section 1, Johnson's algorithm is reviewed in terms of the probabilistic method. The new algorithm and its analysis is presented in Section 2. We conclude with a few remarks in Section 3.

1 Johnson's Algorithm and the Probabilistic Method

Suppose we independently set each variable x_i to be true with probability p_i. Then the expected weight of clauses satisfied by this probabilistic assignment is

$$\hat{W} = \sum_{C_j \in \mathcal{C}} w_j (1 - \prod_{i \in I_j^+} (1 - p_i) \prod_{i \in I_j^-} p_i),$$

where I_j^+ (resp. I_j^-) denote the set of variables appearing unnegated (resp. negated) in C_j. The probabilistic method specifies that there must exist an assignment of truth values to variables whose weight is at least this expected value. In fact, the method of conditional probabilities (see [1], p. 223) can be applied to find such an assignment in polynomial time. In the method of conditional probabilities, the value for the ith variable is determined on the ith iteration: given the value of $x_1, \ldots, x_{i-1}$, calculate the expected weight of clauses satisfied by the probabilistic assignment given the current assignment to $x_1, \ldots, x_{i-1}$ and the assignment $x_i = 1$. Then calculate the expected weight given the assignment to $x_1, \ldots, x_{i-1}$ and $x_i = 0$. The variable x_i is assigned the value that maximizes the conditional expectation. Since each conditional expectation can be calculated in polynomial time, the overall algorithm takes polynomial time, and as asserted above, the assignment produced has weight at least $\hat{W}$.

Johnson's algorithm essentially sets $p_i = \frac{1}{2}$ for all i and uses the method of conditional probabilities. It is not hard to see that for this choice of p_i,

$$\hat{W} \geq \sum_{C_j \in \mathcal{C}} (1 - \frac{1}{2}) w_j = \frac{1}{2} \sum_{C_j \in \mathcal{C}} w_j.$$

Since the optimum assignment can have weight at most $\sum_j w_j$, this proves that Johnson's algorithm is a $\frac{1}{2}$-approximation algorithm. Moreover, if all clauses have at least k literals then

$$\hat{W} \geq \left(1 - \frac{1}{2^k}\right) \sum_{C_j \in \mathcal{C}} w_j,$$

implying that Johnson's algorithm is a $(1 - \frac{1}{2^k})$-approximation algorithm for this restricted class of instances.

2 The Algorithm and its Analysis

Consider the following integer program:

$$
(IP)\qquad
\begin{array}{lll}
\text{Max} & \displaystyle\sum_{C_j \in \mathcal{C}} w_j z_j & \\
\text{subject to:} & & \\
& \displaystyle\sum_{i \in I_j^+} y_i + \sum_{i \in I_j^-} (1 - y_i) \geq z_j & \forall C_j \in \mathcal{C} \\
& y_i \in \{0, 1\} & 1 \leq i \leq n \\
& 0 \leq z_j \leq 1 & \forall C_j \in \mathcal{C}.
\end{array}
$$

By corresponding $y_i = 1$ with x_i set true, $y_i = 0$ with x_i set false, $z_j = 1$ with clause C_j satisfied, and $z_j = 0$ with clause C_j not satisfied, the integer program (IP) exactly corresponds to the MAX SAT problem, and its optimal value Z_{IP}^* is equal to the optimal value of the MAX SAT problem. We can now consider the linear programming relaxation of (IP) formed by replacing the $y_i \in \{0, 1\}$ constraints with the constraints $0 \leq y_i \leq 1$. Call this linear program (LP). Obviously the optimal value of (LP) is an upper bound on the optimal value of (IP); that is, $Z_{LP}^* \geq Z_{IP}^*$. Whenever there are no unit clauses, the solution $y_i = 0.5$ for all i and $z_j = 1$ for all j of value $\sum_{C_j \in \mathcal{C}} w_j$ is optimal, independent of the weights w_j. Hence, the relaxation is vacuous in this case. However, when there are unit clauses (the bad case for Johnson's algorithm), we shall show that this relaxation provides some useful information.

We can now introduce the algorithm, which consists of three simple steps. The first step is to solve the linear program (LP). Let (y, z) be an optimal solution. The second step is to calculate the variables p_i according to the formula $p_i = f(y_i)$ for some carefully selected function $f : [0, 1] \rightarrow [0, 1]$. Possible choices for f are discussed below. The third step is to apply the method of conditional probabilities to these p_i to derive an assignment. Clearly, this algorithm takes only polynomial time (using some polynomial-time linear programming algorithm).

The proof of the performance guarantee is similar to the approach described in Section 1 although the expected weight $\hat{W}$ of a random truth assignment is not compared to $\sum_{C_j \in \mathcal{C}} w_j$ but rather to Z_{LP}^*. Notice that if

$$
1 - \prod_{i \in I_j^+} (1 - f(y_i)) \prod_{i \in I_j^-} f(y_i) \geq \frac{3}{4} z_j \tag{1}
$$

for any feasible solution to (LP) and for any clause C_j then

$$
\hat{W} = \sum_{C_j} w_j \left\{ 1 - \prod_{i \in I_j^+} (1 - p_i) \prod_{i \in I_j^-} p_i \right\} \geq \frac{3}{4} \sum_{C_j} w_j z_j = \frac{3}{4} Z_{LP}^* \geq \frac{3}{4} Z_{IP}^*,
$$

implying that the resulting algorithm is a $\frac{3}{4}$-approximation algorithm. Inequality (1) together with the constraints on z_j motivate the following definition.

Definition 1 *A function $f : [0, 1] \rightarrow [0, 1]$ has property $\frac{3}{4}$ if*

$$
1 - \prod_{i=1}^{l} (1 - f(y_i)) \prod_{i=l+1}^{k} f(y_i) \geq \frac{3}{4} \min \left(1, \sum_{i=1}^{l} y_i + \sum_{i=l+1}^{k} (1 - y_i) \right)
$$

for any k, l with $k \geq l$ and any $y_1, \ldots, y_k \in [0, 1]$.

By the previous discussion, any function f with property $\frac{3}{4}$ induces a $\frac{3}{4}$-approximation algorithm.

The standard randomized rounding technique would set $f(y) = y$ for all $y \in [0,1]$. However, this function does not have property $\frac{3}{4}$. Indeed, taking $y_1 = y_2 = \ldots = y_k = \frac{1}{k}$ and letting $k = l$ tend to ∞, one derives that

$$1 - \prod_{i=1}^{l}(1 - f(y_i)) \prod_{i=l+1}^{k} f(y_i) = 1 - \left(1 - \frac{1}{k}\right)^k \to 1 - \frac{1}{e} \not\geq \frac{3}{4}.$$

The following theorems show that not only do there exist functions with property $\frac{3}{4}$ but also there is some flexibility in choosing such a function.

Theorem 1 *Any function f satisfying*

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

for all $y \in [0,1]$ has property $\frac{3}{4}$.

Theorem 2 *The linear function $f_\alpha(y) = \alpha + (1 - 2\alpha)y$, where*

$$2 - \frac{3}{\sqrt[3]{4}} \leq \alpha \leq \frac{1}{4},$$

has property $\frac{3}{4}$. ($2 - \frac{3}{\sqrt[3]{4}} \approx .11$)

Theorem 3 *The function*

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

has property $\frac{3}{4}$.

The possible choices for f following from Theorems 1–3 are depicted in Figure 1. Before proving these theorems, we would like to make a few remarks regarding the functions with property $\frac{3}{4}$ given in these theorems.

- By Theorem 1, there exist functions f with property $\frac{3}{4}$ for which $f(1) = 1$ and $f(0) = 0$. This is the case, for example, for

  $$f(y) = \begin{cases} 4^{y-1} & \text{if } y \geq \frac{1}{2} \\ 1 - 4^{-y} & \text{if } y < \frac{1}{2}. \end{cases}$$

  The property $f(1) = 1$ and $f(0) = 0$ implies that the randomized rounding sets the value of a variable x_i deterministically according to y_i when $y_i \in \{0, 1\}$.

- For $l = k$ and $y_1 = \ldots = y_k = \frac{1}{k}$, notice that any function f with property $\frac{3}{4}$ must satisfy

  $$f\left(\frac{1}{k}\right) \geq 1 - 4^{-1/k}$$

  for all integers $k \geq 1$. This partially explains the choice of the lower bound in Theorem 1. The upper bound can be similarly obtained by considering $l = 0$.

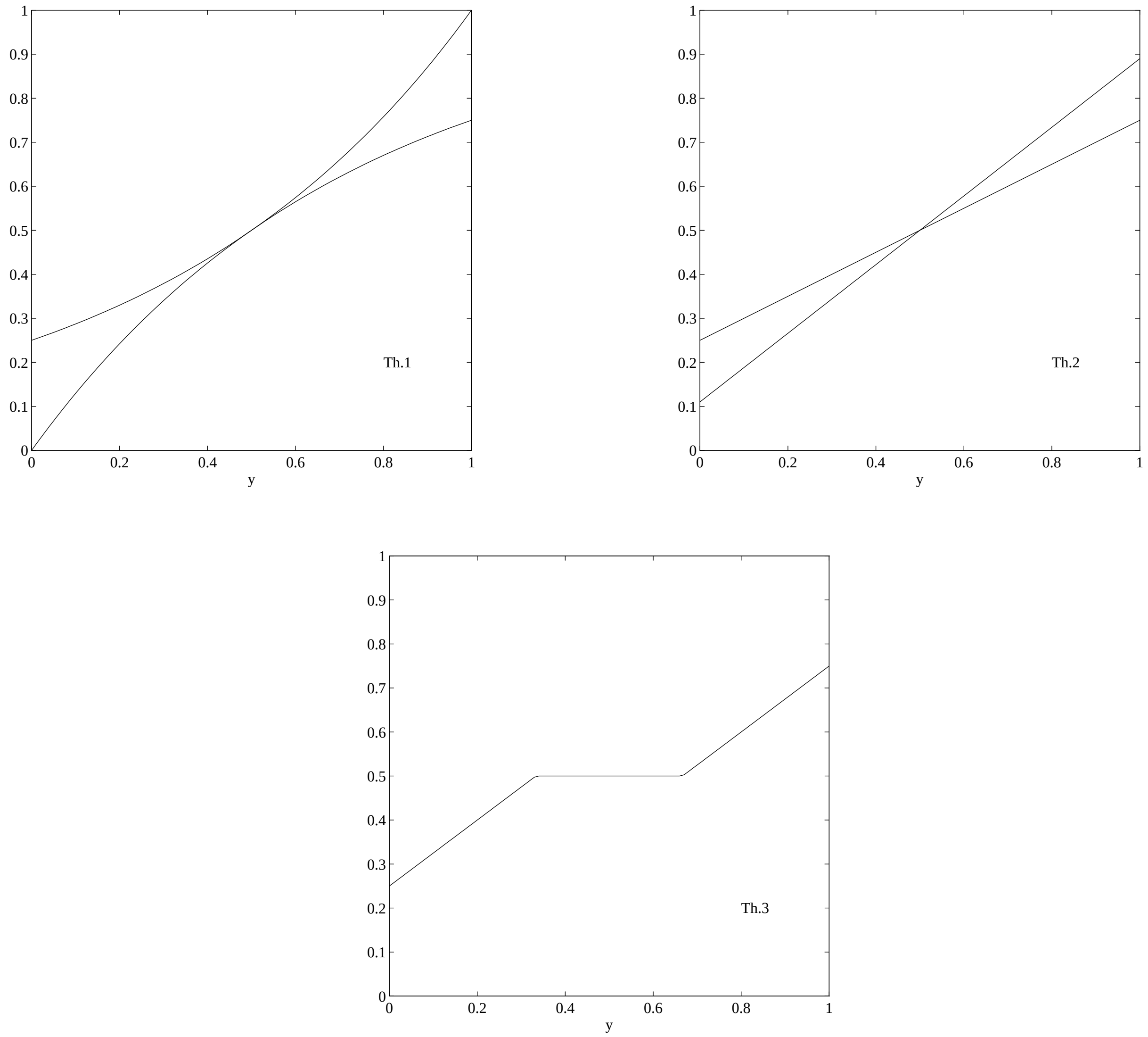

Figure 1: Functions with property $\frac{3}{4}$ from Theorems 1, 2 and 3.

- Although $f(y) = y$ is not a function with property $\frac{3}{4}$, there exist linear functions with property $\frac{3}{4}$ as demonstrated in Theorem 2.

- The function described in Theorem 3 is the "closest" to Johnson's scheme in the sense that, for any $y \in [0,1]$, the function f of Theorem 3 minimizes $|f(y) - \frac{1}{2}|$ over all functions with property $\frac{3}{4}$. Indeed, by considering the case $k = 1$, $l = 0$ or 1, one derives that any function f with property $\frac{3}{4}$ satisfies
$$\frac{3}{4}y \le f(y) \le \frac{3}{4}y + \frac{1}{4}$$
for any $y \in [0,1]$.

To prove Theorems 1–3, we use the following Lemma to restrict our attention to the case $l = k$ (corresponding to a clause with no negated variable).

Lemma 3.1 *Let $g : [0,1] \rightarrow [0,1]$ be a function satisfying*
$$1 - \prod_{i=1}^{k}(1 - g(y_i)) \ge \frac{3}{4}\min\left(1, \sum_{i}^{k} y_i\right) \tag{2}$$
for all k and all $y_1, y_2, \ldots, y_k \in [0,1]$. Consider any function $f : [0,1] \rightarrow [0,1]$ satisfying
$$g(y) \le f(y) \le 1 - g(1-y) \tag{3}$$
for all $y \in [0,1]$. Then f has property $\frac{3}{4}$.

Proof: Consider any k, l with $k \ge l$ and any $y_1, \ldots, y_k \in [0,1]$. Then
$$\begin{aligned} 1 - \prod_{i=1}^{l}(1 - f(y_i)) \prod_{i=l+1}^{k} f(y_i) &\overset{(3)}{\ge} 1 - \prod_{i=1}^{l}(1 - g(y_i)) \prod_{i=l+1}^{k} (1 - g(1 - y_i)) \\ &= 1 - \prod_{i=1}^{k}(1 - g(y_i')) \\ &\overset{(2)}{\ge} \frac{3}{4}\min\left(1, \sum_{i=1}^{k} y_i'\right) \\ &= \frac{3}{4}\min\left(1, \sum_{i=1}^{l} y_i + \sum_{i=l+1}^{k}(1 - y_i)\right), \end{aligned}$$
where $y_i' = y_i$ for $i = 1, \ldots, l$ and $y_i' = 1 - y_i$ for $i = l+1, \ldots, k$. ■

In the proofs of Theorems 1–3, we make use of the fact that over an interval a concave function is minimized at one of its endpoints.

Proof of Theorem 1: By Lemma 3.1, we only need to show that $g(y) = 1 - 4^{-y}$ satisfies (2). We have
$$\begin{aligned} 1 - \prod_{i=1}^{k}(1 - g(y_i)) &= 1 - \prod_{i=1}^{k} 4^{-y_i} \\ &= 1 - 4^{-\sum_{i=1}^{k} y_i} \\ &= 1 - 4^{-Y} = g(Y) \end{aligned}$$

where $Y = \sum_{i=1}^{k} y_i$. Since $g(Y)$ is increasing with Y, in order to prove (2), we only need to show that $g(Y) \geq \frac{3}{4}Y$ for any $Y \in [0,1]$. This follows from the concavity of $g(Y) - \frac{3}{4}Y$ and the facts that $g(0) = 0$ and $g(1) = \frac{3}{4}$. ■

Proof of Theorem 2: Suppose $2 - \frac{3}{\sqrt[3]{4}} \leq \alpha \leq \frac{1}{4}$. Since $f_\alpha(y) = \alpha + (1-2\alpha)y$ satisfies $1 - f_\alpha(1-y) = f_\alpha(y)$, we only need to show that $f_\alpha(y)$ satisfies (2) in order to use Lemma 3.1. We have

$$\begin{aligned} 1 - \prod_{i=1}^{k}(1 - f_\alpha(y_i)) &= 1 - \prod_{i=1}^{k}(1 - \alpha - (1-2\alpha)y_i) \\ &\geq 1 - \left(1 - \alpha - (1-2\alpha)\frac{\sum_{i=1}^{k} y_i}{k}\right)^k \end{aligned}$$

since the geometric mean of k numbers is never greater than the arithmetic mean. Letting $y = (\sum_{i=1}^{k} y_i)/k$, we need to show that

$$1 - (1 - \alpha - (1-2\alpha)y)^k \geq \frac{3}{4}\min(1, ky), \tag{4}$$

for any $y \in [0,1]$. Since the left-hand-side is increasing with y, we can restrict our attention to $y \in [0, \frac{1}{k}]$. Furthermore, since it is concave in y, we can just check (4) for $y = 0$ (for which it is trivially satisfied) and for $y = \frac{1}{k}$. For this latter value, we need to prove that

$$\frac{1}{4} \geq \left(1 - \alpha - (1-2\alpha)\frac{1}{k}\right)^k, \tag{5}$$

for any integer $k \geq 1$. For $k = 1$, (5) reduces to $\alpha \leq \frac{1}{4}$ while (5) always holds for $k = 2$. For $k \geq 3$, (5) is equivalent to

$$\alpha \geq \frac{k - 1 - k4^{-1/k}}{k-2}. \tag{6}$$

One can show that $h(x) = (x - 1 - x4^{-1/x})/(x-2)$ is decreasing in x and, thus, (6) holds provided that $\alpha \geq 2 - \frac{3}{\sqrt[3]{4}}$. ■

Proof of Theorem 3: Again, $1 - f(1-y) = f(y)$ and therefore we only need to show that f satisfies (2) in order to use Lemma 3.1. We begin by making a number of simplifying assumptions about the y_i values. For convenience, we write (2) as $E \geq \frac{3}{4}z$ where

$$E = 1 - \prod_{i=1}^{k}(1 - f(y_i))$$

and

$$z = \min\left(1, \sum_{i=1}^{k} y_i\right).$$

In every case if the simplifying assumption does not hold, then the y_i can be modified in such a way that the assumption does hold and that if $E' \geq \frac{3}{4}z'$ for the new y values, then $E \geq \frac{3}{4}z$ for the old y values.

First, we can assume without loss of generality that none of the $y_1, \ldots, y_k$ are in the interval $[\frac{1}{3}, \frac{2}{3})$. If some y_i is in this interval, then it can be increased to $\frac{2}{3}$ without changing E while $z' \geq z$. Second, all y_i in the interval $[0, \frac{1}{3})$ can be assumed to be equal. If they are not, then replacing all these y_i by their arithmetic mean does not increase E and does not affect z; this follows from the fact that given that some variables sum to a particular amount, their product is maximized when the variables are all equal. Third, we can assume that if there is some y_i in the interval $[0, \frac{1}{3})$, then there does not exist y_j in the interval $(\frac{2}{3}, 1]$. Otherwise increasing y_i by ϵ and decreasing y_j by ϵ decreases E without changing z.

Suppose now that the values $y_1, \ldots, y_k$ obey the properties above. We prove (2) in a series of cases.

- Case 1. $k = 1$. Then, $E = 1 - (1 - f(y_1)) = f(y_1) \geq \frac{3}{4} y_1 \geq \frac{3}{4} z$.
- Case 2. Suppose that $y_1 = \cdots = y_k = y \leq \frac{1}{3}$, and $k \geq 2$. Here $ky \geq z$. Then

$$
\begin{aligned}
E &= 1 - (1 - f(y_1))^k \\
&= 1 - \left(\frac{3}{4} - \frac{3}{4} y\right)^k \\
&\geq 1 - \left(\frac{3}{4} - \frac{3}{4} \cdot \frac{z}{k}\right)^k \\
&\geq 1 - \left(\frac{3}{4}\right)^k \left(1 - \frac{z}{k}\right)^k \\
&\geq 1 - \left(\frac{3}{4}\right)^k e^{-z} && (7) \\
&\geq 1 - \frac{9}{16} e^{-z} && (8) \\
&= (1 - \frac{9}{16} e^{-z} - \frac{3}{4} z) + \frac{3}{4} z \\
&\geq \min\left(\frac{7}{16}, \frac{1}{4} - \frac{9}{16e}\right) + \frac{3}{4} z && (9) \\
&\geq \frac{3}{4} z.
\end{aligned}
$$

  Equation (7) follows since $(1 - \frac{z}{k})^k \leq e^{-z}$, and (8) follows since $k \geq 2$. Equation (9) follows from the concavity of $1 - \frac{9}{16} e^{-z} - \frac{3}{4} z$ over the interval [0,1].

- Case 3. Suppose $k \geq 2$ and all but one $y_i \leq \frac{1}{3}$; that is, suppose $y_1 = \cdots = y_{k-1} = y \leq \frac{1}{3}$, and $y_k = \frac{2}{3}$. Here $z \leq (k-1)y + \frac{2}{3}$. Then

$$
\begin{aligned}
E &= 1 - \frac{1}{2}(1 - f(y_1))^{k-1} \\
&= 1 - \frac{1}{2}\left(\frac{3}{4} - \frac{3}{4} y\right)^{k-1} \\
&\geq 1 - \frac{1}{2}\left(\frac{3}{4}\right)^{k-1} \left(1 - \frac{z - \frac{2}{3}}{k-1}\right)^{k-1}.
\end{aligned}
$$

  When $k = 2$, the inequality above yields $E \geq \frac{3}{8} + \frac{3}{8} z$, so that $E \geq \frac{3}{4} z$ on the interval [0,1]. When $k \geq 3$, we can continue the inequality as in Case 2, deriving

$$
\begin{aligned}
E &\geq 1 - \frac{1}{2}\left(\frac{3}{4}\right)^{k-1} e^{-z+2/3} \\
&\geq 1 - \frac{9}{32} e^{-z+2/3} \\
&= (1 - \frac{9}{32} e^{-z+2/3} - \frac{3}{4} z) + \frac{3}{4} z
\end{aligned}
$$

$$\begin{aligned} &\geq \min\left(1 - \frac{9}{32}e^{2/3}, \frac{1}{4} - \frac{9}{32}e^{-1/3}\right) + \frac{3}{4}z \\ &\geq \frac{3}{4}z. \end{aligned}$$

- Case 4. Suppose that at least two of the $y_i \geq \frac{2}{3}$. Then it follows straightforwardly that $E \geq \frac{3}{4} \geq \frac{3}{4}z$.

In every case, $E \geq \frac{3}{4}z$, so the proof is complete. ■

3 Concluding Remarks

The existence of functions with property $\frac{3}{4}$ proves a worst-case bound on the duality gap associated with (LP), namely that

$$Z^*_{LP} \leq \frac{4}{3}Z^*_{IP}.$$

Moreover, this worst-case analysis is tight, as can be seen from the MAX 2SAT instance

$$\begin{cases} x_1 \vee x_2 \\ x_1 \vee \overline{x_2} \\ \overline{x_1} \vee x_2 \\ \overline{x_1} \vee \overline{x_2} \end{cases}$$

with unit weights. As we observed previously, the LP solution $y_i = \frac{1}{2}$ for all i and $z_j = 1$ for all j is optimal for any instance without unit clauses, and has value $\sum_j w_j$. In this case, $Z^*_{LP} = 4$, while $Z^*_{IP} = 3$.

The performance guarantee of $\frac{3}{4}$ for the algorithm is also tight. For any instance without unit clauses, our algorithm reduces to Johnson's algorithm, since $y_i = \frac{1}{2}$ for all i is an optimal solution to (LP) and all the functions given above have $f(\frac{1}{2}) = \frac{1}{2}$. Johnson's algorithm is a $\frac{3}{4}$-approximation algorithm on this class of instances, and he gives instances of this type that are tight for his algorithm [3]. Furthermore, *any* function f with property $\frac{3}{4}$ must satisfy $f(\frac{1}{2}) = \frac{1}{2}$. This follows from the definition of property $\frac{3}{4}$ for the values $k = 2$, $l = 0$ or 2, and $y_1 = y_2 = \frac{1}{2}$. Therefore, without changing our analysis or strengthening the linear programming relaxation, one cannot expect to beat the performance guarantee of $\frac{3}{4}$.

Thus it is an interesting open question as to whether the linear programming relaxation can be strengthened so that a better performance guarantee is possible using these techniques. It would also be interesting to obtain a complete characterization of the functions with property $\frac{3}{4}$. Finally, we would like to know if the technique used here of randomized rounding with a function other than the identity function can be applied to other problems with natural linear programming relaxations.

Acknowledgements

We would like to thank Rick Vohra for suggesting the search for simple functions with property $\frac{3}{4}$.

References

[1] N. Alon and J. H. Spencer. *The Probabilistic Method.* John Wiley and Sons, 1992.

[2] M. Garey, D. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1976.

[3] D. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256–278, 1974.

[4] K. Lieberherr and E. Specker. Complexity of partial satisfaction. *Journal of the ACM*, 28(2):411–421, 1981.

[5] M. Yannakakis. On the approximation of maximum satisfiability. In *Proceedings of the 3rd ACM-SIAM Symposium on Discrete Algorithms*, pages 1–9, 1992.