Best possible approximation algorithm for MAX SAT with cardinality constraint

M.I. Sviridenko *

We consider the MAX SAT problem with the additional constraint that at most P variables have a true value. We obtain a $(1-e^{-1})$-approximation algorithm for this problem. Feige [6] has proved that for MAX SAT with cardinality constraint with clauses without negations this is the best possible performance guarantee unless $P = NP$.

1 Introduction

An instance of the Maximum Satisfiability Problem (MAX SAT) is defined by a collection C of Boolean clauses, where each clause is a disjunction of literals drawn from a set of variables $\{x_1, \ldots, x_n\}$. A literal is either a variable x or its negation $\bar{x}$. In addition, for each clause $C_j \in C$, there is an associated nonnegative weight w_j. An optimal solution to a MAX SAT instance is an assignment of truth values to the variables $x_1, \ldots, x_n$ that maximizes the sum of the weights of the satisfied clauses (i.e. clauses with at least one true literal). In this work we consider cardinality constrained MAX SAT (CC-MAX SAT). An instance of this problem is a pair (C, P) where C is a collection of clauses and P is an integer parameter. A feasible solution is a truth assignment that gives value true to at most P variables.

MAX SAT is one of the central problems in theoretical computer science and is well studied, both from a practical viewpoint [9] and a theoretical one. The best known approximation algorithm for MAX SAT has a performance guarantee slightly better than 0.77 [3]. In [10] it is shown that the MAX E3SAT, the version of MAX SAT problem in which each clause is of length exactly three, cannot be approximated in polynomial time to within a ratio greater than 7/8, unless $P = NP$. For general MAX 3SAT there exists an approximation algorithm with performance guarantee 7/8 [11]. The best known positive and negative results for MAX 2SAT are 0,931 [7] and 21/22 [10], respectively. We can see that there is a gap between positive and negative results for MAX SAT.

Khanna and Motwani [12] define a class MPSAT of optimization problems and present an approximation scheme for all problems in this class. Since planar CC-MAX SAT belongs to MPSAT the existence of an approximation scheme for this planar problem

*Supported by grants 97-01-00890 and 99-01-00510 from the Russian Foundation for Basic Research.

follows. On the other hand, Feige [6] has proved that the existence of an approximation algorithm with performance guarantee better than $1-e^{-1}$ for CC-MAX SAT with clauses without negations implies $P = NP$.

In this work we present an approximation algorithm for CC-MAX SAT with performance guarantee $1 - e^{-1}$. We use the method of randomized rounding of an optimal solution to a linear relaxation. Notice that for satisfiability problems without cardinality constraint the best known algorithms (sometimes, the best possible) are obtained by using semidefinite programming relaxations (compare [4] and [7, 5, 11]) but for CC-MAX SAT the best possible approximation is obtained here via a linear programming relaxation.

2 Linear relaxation and approximation algorithm

Consider the following mixed integer program

$$\max \sum_{C_j \in C} w_j z_j, \tag{1}$$

subject to

$$\sum_{i \in I_j^+} y_i + \sum_{i \in I_j^-} (1 - y_i) \geq z_j \quad \text{for all } C_j \in C, \tag{2}$$

$$\sum_{i=1}^{n} y_i \leq P, \tag{3}$$

$$0 \leq z_j \leq 1 \quad \text{for all } C_j \in C, \tag{4}$$

$$y_i \in \{0, 1\} \quad i = 1, \ldots, n, \tag{5}$$

where I_j^+ (respectively I_j^-) denotes the set of variables appearing unnegated (respectively negated) in C_j. By associating $y_i = 1$ with x_i set true, $y_i = 0$ with x_i false, $z_j = 1$ with clause C_j satisfied, and $z_j = 0$ with clause C_j not satisfied, the mixed integer program (1)-(5) is a formulation of CC-MAX SAT. A similar integer program was first used by Goemans and Williamson [4] for designing an approximation algorithm for MAX SAT.

Let M be an integer constant with $1 \leq M \leq P$. We define M in the next section. Consider the problem (1)-(5) with the additional constraint $\sum_{i=1}^{n} y_i \leq M$. We can find an optimal solution (y', z') of this problem in polynomial time by complete enumeration. Now, consider the problem (1)-(5) with the additional constraint $\sum_{i=1}^{n} y_i \geq M$ and let (y'', z'') be an α-approximation solution of this problem. Clearly, the better of these two solutions is an α-approximation solution of CC-MAX SAT. Consequently, without loss of generality we may consider the problem (1)-(5) with constraint $\sum_{i=1}^{n} y_i \geq M$.

For $t = M, \ldots, P$ consider now the linear programs LP_t formed by replacing $y_i \in \{0, 1\}$ constraints with the constraints $0 \leq y_i \leq 1$ and by replacing (3) with the constraint

$$\sum_{i=1}^{n} y_i = t. \tag{6}$$

Let F_t^* be the value of an optimal solution of LP_t. Let k denote an index such that $F_k^* = \max_{M \le t \le P} F_t^*$. Since any optimal solution of the problem (1)-(5) with constraint $\sum_{i=1}^n y_i \ge M$ is a feasible solution of LP_t for some t, we obtain that F_k^* is an upper bound of the optimal value of this problem. We now present a randomized approximation algorithm for CC-MAX SAT.

1. Solve the linear programs LP_t for all $t = M, \ldots, P$. Let (y^*, z^*) be an optimal solution of LP_k where k is an index such that $F_k^* = \max_{M \le t \le P} F_t^*$.
2. The second part of the algorithm consists of k independent steps. At each step the algorithm chooses an index i from the set $\{1, \ldots, n\}$ at random with probability $P_i = \frac{y_i^*}{k}$. Let S denote the set of the chosen indices. Notice that $P \ge k \ge |S|$. We set $x_i = 1$ if $i \in S$ and $x_i = 0$, otherwise, and $z_j = \min\{1, \sum_{i \in I_j^+} x_i + \sum_{i \in I_j^-} (1 - x_i)\}$.

In the first part of our algorithm we can solve linear programs LP_t for all $t = M, \ldots, P$ by using any known polynomial algorithm for linear programming. The second part is a derandomization of the randomized part of the algorithm. We will show in Section 4 that derandomization can be done in polynomial time. In the next section we evaluate the expected value of the rounded solution.

3 Analysis of the algorithm

3.1 Preliminaries

In this subsection we state some technical lemmas.

Lemma 1 *The probability of realization of at least one among the events $A_1, \ldots, A_n$ is given by*

$$Pr(A_1 \cup \ldots \cup A_n) = \sum_{1 \le i \le n} Pr(A_i) - \sum_{1 \le i_1 < i_2 \le n} Pr(A_{i_1} \cap A_{i_2}) + \ldots$$
$$+ (-1)^{t-1} \sum_{1 \le i_1 < \ldots < i_t \le n} Pr(A_{i_1} \cap \ldots \cap A_{i_t}) + \ldots$$

Proof. See [8], V.1, Chapter IV, (1.5). $\square$

Lemma 2 *The probability of realization of at least one among the events $B, A_1, \ldots, A_n$ is given by*

$$Pr(B \cup A_1 \cup \ldots \cup A_n) = Pr(B) + \sum_{1 \le i \le n} Pr(\bar{B} \cap A_i) - \sum_{1 \le i_1 < i_2 \le n} Pr(\bar{B} \cap A_{i_1} \cap A_{i_2}) + \ldots$$
$$+ (-1)^{t-1} \sum_{1 \le i_1 < \ldots < i_t \le n} Pr(\bar{B} \cap A_{i_1} \cap \ldots \cap A_{i_t}) + \ldots$$

Proof. The claim follows from Lemma 1 and the facts

$$Pr(B \cup A_1 \cup \ldots \cup A_n) = Pr(B) + Pr(\bar{B} \cap (A_1 \cup \ldots \cup A_n)) =$$

$$Pr(B) + Pr((\bar{B} \cap A_1) \cup \ldots \cup (\bar{B} \cap A_n)).\square$$

Lemma 3 *The inequalities*

$$\begin{aligned} 1 - e^{-y} &\leq e^{-1+y}, \\ 1 - e^{-4/k}e^{-y} &\leq e^{-1+y} - g(k), \\ 1 - e^{-4/k}e^{-y} - e^{-4/k}e^{-x} + e^{-x-y} &\leq e^{-2+x+y} \end{aligned}$$

hold for all $y, x \in [0,1]$, $k \geq M$ where M is a sufficiently large constant independent of x and y, and $\lim_{k\to+\infty} g(k) = 0$.

Proof. Let $t = e^y$, then the first inequality is equivalent to $t^2 - et + e \geq 0$. Since $e^2 - 4e < 0$ we obtain the desired statement. Using the same argument we can prove the second inequality for sufficiently large k. We now prove the third inequality

$$1 - e^{-4/k}e^{-y} - e^{-4/k}e^{-x} + e^{-x-y} = (1 - e^{-4/k}e^{-x})(1 - e^{-4/k}e^{-y}) + g_1(k)$$

where $g_1(k) = e^{-x}e^{-y}(1 - e^{-8/k})$. Let $g(k) = e \cdot g_1(k)$, so $g(k) \geq 0$ and $\lim_{k\to+\infty} g(k) = 0$. We continue using the second inequality

$$\leq (e^{-1+x} - g(k))e^{-1+y} + g_1(k) = e^{-2+x+y} - g_1(k)e^y + g_1(k) \leq e^{-2+x+y}.\square$$

In the proof of the following statements we will use the inequalities:

$$e^{-a} \geq (1 - a/k)^k \geq e^{-a-a^2/k} \tag{7}$$

for all $k \geq 2a \geq 0$. We can simply derive (7) from the well-known inequalities: $e^{-1} \geq (1 - 1/x)^x \geq e^{-1-1/x}$ for all $x \geq 2$.

Lemma 4 *The inequality*

$$g(x,y,z) = \left(1 - \frac{x}{k}\right)^k + \left(1 - \frac{y}{k}\right)^k + \left(1 - \frac{z}{k}\right)^k - \left(1 - \frac{x+y}{k}\right)^k - \left(1 - \frac{y+z}{k}\right)^k -$$

$$- \left(1 - \frac{z+x}{k}\right)^k + \left(1 - \frac{x+y+z}{k}\right)^k > 1 - e^{-1}.$$

holds for all $x, y, z \in [0,1]$ and $k \geq M$ where M is a sufficiently large constant independent of x, y, z.

Proof. Notice that the following inequalities hold for all $x \in [0,1]$

$$\begin{aligned} e^{-x} - (1 - x/k)^k &\leq \text{ by (8)} \\ e^{-x}(1 - e^{-x^2/k}) &\leq 1 - e^{-1/k}. \end{aligned}$$

Using similar arguments we have

$$\lim_{k \to +\infty} \left\{ e^{-x} + e^{-y} + e^{-z} + e^{-x-y-z} - \left(1 - \frac{x}{k}\right)^k - \right.$$

$$\left. - \left(1 - \frac{y}{k}\right)^k - \left(1 - \frac{z}{k}\right)^k - \left(1 - \frac{x+y+z}{k}\right)^k \right\} = 0$$

and therefore for large k we obtain

$$g(x,y,z) \geq e^{-x} + e^{-y} + e^{-z} - e^{-x-y} - e^{-x-z} - e^{-y-z} + e^{-x-y-z} - o(1) =$$

$$= 1 - (1 - e^{-x})(1 - e^{-y})(1 - e^{-z}) - o(1) \geq 1 - (1 - e^{-1})^3 - o(1) > 0.74 > 1 - e^{-1}. \square$$

3.2 Evaluation of expectation

Let S denote the set of indices produced by the randomized algorithm of Section 2, let $f(S)$ be the value of the solution defined by the set S and let $E(f(S))$ be the expectation of $f(S)$. We now prove our main result.

Theorem 1

$$F_k^* \geq E(f(S)) \geq (1 - e^{-1}) F_k^*$$

Proof. Using linearity of expectation we obtain

$$E(f(S)) = \sum_{C_j \in C} w_j Pr(z_j = 1).$$

Fix a clause C_j and let $X^+ = \sum_{i \in I_j^+} y_i^*$. We now consider four cases.

Case 1 Assume that $I_j^- = \emptyset$. Since the steps of the algorithm are independent and $X^+ \geq z_j^*$ we have

$$Pr(z_j = 1) = Pr(S \cap I_j^+ \neq \emptyset) = 1 - \left(1 - \frac{X^+}{k}\right)^k \geq$$

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

The last inequality follows from the concavity of the function $g'(z) = 1 - (1 - z/k)^k$ and the facts $g'(0) = 0$ and $g'(1) = 1 - (1 - 1/k)^k$.

Case 2 Assume that $|I_j^-| = 1$. Let $I_j^- = \{i\}$ and $a = y_i^*$. If $X^+ > 1$, then using the argument of the previous case we obtain $Pr(z_j = 1) \geq Pr(S \cap I_j^+ \neq \emptyset) \geq 1 - e^{-1}$. Assume that $X^+ \leq 1$, then

$$\begin{aligned}
Pr(z_j = 1) = Pr(S \cap I_j^+ \neq \emptyset \text{ or } i \notin S) &= \quad \text{by Lemma 2} \\
Pr(S \cap I_j^+ \neq \emptyset) + Pr(S \cap I_j^+ = \emptyset \text{ and } i \notin S) &= \quad \text{by the independence of the steps} \\
&\quad\quad \text{of the randomized algorithm} \\
1 - \left(1 - \frac{X^+}{k}\right)^k + \left(1 - \frac{X^+ + a}{k}\right)^k &\geq \quad \text{by inequalities (7)} \\
1 - e^{-X^+} + e^{-X^+ - a - (X^+ + a)^2/k} &\geq \quad \text{by inequality } X^+ + a \leq 2 \\
1 - e^{-X^+}(1 - e^{-4/k} e^{-a}) &\geq \quad \text{by Lemma 3} \\
1 - e^{-X^+} e^{-1+a} &\geq \quad \text{by inequality } X^+ + (1 - a) \geq z_j^* \\
1 - e^{-z_j^*} &\geq \quad \text{by concavity} \\
(1 - e^{-1}) z_j^*. &
\end{aligned}$$

Case 3 Assume that $|I_j^-| = 2$. Let $I_j^- = \{i_1, i_2\}, a = y_{i_1}^*$ and $b = y_{i_2}^*$. Without loss of generality assume that $X^+ \leq 1$, then

$$\begin{aligned}
Pr(z_j = 1) = Pr(S \cap I_j^+ \neq \emptyset \text{ or } i_1 \notin S \text{ or } i_2 \notin S) &= \quad \text{by Lemma 2} \\
Pr(S \cap I_j^+ \neq \emptyset) + Pr(S \cap I_j^+ = \emptyset \text{ and } i_1 \notin S) + & \\
Pr(S \cap I_j^+ = \emptyset \text{ and } i_2 \notin S) - & \\
Pr(S \cap I_j^+ = \emptyset \text{ and } i_1 \notin S \text{ and } i_2 \notin S) &= \quad \text{by the independence of the steps} \\
&\quad\quad \text{of the randomized algorithm} \\
1 - \left(1 - \frac{X^+}{k}\right)^k + \left(1 - \frac{X^+ + a}{k}\right)^k + & \\
\left(1 - \frac{X^+ + b}{k}\right)^k - \left(1 - \frac{X^+ + a + b}{k}\right)^k &\geq \quad \text{by inequalities (7)} \\
1 - e^{-X^+} + e^{-4/k} e^{-X^+ - a} + e^{-4/k} e^{-X^+ - b} - e^{-X^+ - a - b} &\geq \quad \text{by Lemma 3} \\
1 - e^{-X^+} e^{-2+a+b} &\geq \quad \text{by inequality} \\
&\quad\quad X^+ + (1 - a) + (1 - b) \geq z_j^* \\
1 - e^{-z_j^*} &\geq \quad \text{by concavity} \\
(1 - e^{-1}) z_j^*. &
\end{aligned}$$

Case 4 Assume that $|I_j^-| \geq 3$. Let i_1, i_2, i_3 be arbitrary indices from the set I_j^-. Then

$$\begin{aligned}
Pr(z_j = 1) \geq Pr(i_1 \notin S \text{ or } i_2 \notin S \text{ or } i_3 \notin S) &= \quad \text{by Lemma 1} \\
Pr(i_1 \notin S) + Pr(i_2 \notin S) + Pr(i_3 \notin S) - &
\end{aligned}$$

$$
\begin{aligned}
& Pr(i_1 \notin S \text{ and } i_2 \notin S) - \\
& Pr(i_2 \notin S \text{ and } i_3 \notin S) - \\
& Pr(i_1 \notin S \text{ and } i_3 \notin S) + \\
Pr(i_1 \notin S \text{ and } i_2 \notin S \text{ and } i_3 \notin S) \quad &= \quad \text{by the independence of the steps} \\
& \qquad \text{of the randomized algorithm} \\
\left(1-\frac{y_{i_1}^*}{k}\right)^k + \left(1-\frac{y_{i_2}^*}{k}\right)^k + \left(1-\frac{y_{i_3}^*}{k}\right)^k & - \\
\left(1-\frac{y_{i_1}^*+y_{i_2}^*}{k}\right)^k - \left(1-\frac{y_{i_2}^*+y_{i_3}^*}{k}\right)^k & - \\
\left(1-\frac{y_{i_3}^*+y_{i_1}^*}{k}\right)^k + \left(1-\frac{y_{i_1}^*+y_{i_2}^*+y_{i_3}^*}{k}\right)^k \quad &> \quad \text{by Lemma 4} \\
1-e^{-1}. \square &
\end{aligned}
$$

4 Derandomization

In this section we apply the method of conditional expectations [2] to find an approximate truth assignment in polynomial time. The straightforward use of this method doesn't give a polynomial-time algorithm since if an instance of CC-MAX SAT contains a clause with a non-constant number of negations we cannot directly calculate (by using Lemma 2) the conditional expectations in polynomial time.

Let IP_1 be an instance of CC-MAX SAT given by a set of clauses $C = \{C_j : j = 1, \ldots, m\}$ and a set of variables $\{x_1, \ldots, x_n\}$. Let F_k^* be the value of an optimal solution of the relaxation LP_k for IP_1. We define an instance IP_2 of CC-MAX SAT by replacing each clause C_j in which $|I_j^-| \geq 3$ with the clause $C_j' = \bar{x}_{i_1} \vee \bar{x}_{i_2} \vee \bar{x}_{i_3}$ where $i_1, i_2, i_3 \in I_j^-$.

We apply our randomized approximation algorithm using probabilities defined by the optimal solution of LP_k. Let S be a solution obtained by randomized rounding, $f_1(S)$ the value of S for the problem IP_1 and $f_2(S)$ the value of S for the problem IP_2. Then using the fact that $Pr(z_j = 1) > 1 - e^{-1}$ for all clauses C_j with $|I_j^-| \geq 3$, we have $E(f_2(S)) \geq (1 - e^{-1})F_k^*$. We can derandomize this algorithm using the following procedure

Description of the derandomization

The derandomized algorithm consists of k steps indexed by $s = 1, \ldots, k$. In step s we choose an index i_s which maximizes the conditional expectation, i.e.

$$E(f_2(S)|i_1 \in S, \ldots, i_{s-1} \in S, i_s \in S) = \max_{j \in \{1,\ldots,n\}} E(f_2(S)|i_1 \in S, \ldots, i_{s-1} \in S, j \in S).$$

Since

$$\max_{j\in\{1,\ldots,n\}} E(f_2(S)|i_1 \in S,\ldots,i_{s-1}\in S, j\in S) \geq E(f_2(S)|i_1\in S,\ldots,i_{s-1}\in S)$$

at the end of the derandomization we obtain a solution $\tilde{S}$ such that $f_2(\tilde{S}) \geq E(f_2(S)) \geq (1-e^{-1})F_k^*$. Since $F_k^* \geq f_1(\tilde{S}) \geq f_2(\tilde{S})$ this solution is $(1-e^{-1})$-approximation solution for IP_1. We can calculate the conditional expectations in polynomial time using their linearity, Lemma 2 and the fact that $|I_j^-| \leq 3$ in the instance IP_2.

5 Discussion

In this paper we have presented a polynomial time approximation algorithm for CC-MAX SAT with a worst case performance guarantee of $1-e^{-1}$. Feige [6] proved that this is a best possible performance guarantee unless P=NP. Ageev and Sviridenko [1] obtained an approximation algorithm with performance guarantee $1-(1-1/k)^k$ for the class of CC-MAX SAT instances with clauses of the following two types: $x_1 \vee \ldots \vee x_t$, $\bar{x}_1 \vee \ldots \vee \bar{x}_t$ and with at most k literals per clause. A major open problem consists in finding a better approximation algorithm for CC-MAX SAT with clauses of bounded length (for example, for CC-MAX 2SAT).

References

[1] A. Ageev and M. Sviridenko, Approximation algorithms for maximum coverage and max cut with given sizes of parts, to appear in IPCO99.

[2] N. Alon and J. H. Spencer, The Probabilistic Method, Wiley, 1992.

[3] T. Asano, Approximation algorithms for MAX SAT: Yannakakis vs. Goemans-Williamson, In Proceedings of the 5th Israel Symposium on Theory and Computing Systems, (1997), pp.182–189.

[4] M. Goemans and D. Williamson, New 3/4-approximation algorithms for the maximum satisfiability problem, SIAM Journal on Discrete Mathematics, 7, (1994), pp.656–666.

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

[6] U. Feige, A threshold of $\ln n$ for approximating set cover, Journal of the ACM **45** (1998), pp.634–652.

[7] U. Feige and M. Goemans, Approximating the value of two-prover proof systems, with applications to MAX 2-SAT and MAX-DICUT, In Proceedings of the 3rd Israel Symposium on Theory and Computing Systems (1995), pp.182–189.

[8] W. Feller, An Introduction to Probability Theory and Its Applications, John Wiley & Sons, New York, 1968.

[9] D. Johnson and M. Trick, Cliques, Colorings and Satisfiability, AMS, DIMACS Series in Discrete Mathematics and Theoretical Computer Science **26** (1996).

[10] J. Håstad, Some optimal inapproximability results, In Proceedings of the 28th Annual ACM Symp. on Theory of Computing (1996), pp.1–10.

[11] H. Karloff and U. Zwick, A 7/8-approximation algorithm for MAX 3SAT?, In Proceedings of the 38th FOCS (1997), pp.406–415.

[12] S. Khanna and R. Motwani, Towards a syntactic characterization of PTAS, In Proceedings of the 28th Annual ACM Symp. on Theory of Computing (1996), pp.329–337.