A Simplified NP-Complete Maximum Satisfiability Problem

Venkatesh Raman
The Institute of Mathematical Sciences
C. I. T. Campus
Madras 600 113. India

B. Ravikumar
Department of Computer Science
University of Rhode Island
Kingston RI 02881. USA

July 11, 1996

Abstract

It is shown that the Max2Sat problem is NP-Complete even if every variable appears in at most three clauses, and that even the general MaxSat problem is solvable in polynomial time if every variable appears at most twice.

1 Introduction

It is known that the CNF Boolean Satisfiability Problem (SAT) is NP-Complete even if every clause contains at most three literals, even with every variable appearing at most three times (either purely or in its complemented form)[1]. When every clause contains exactly three literals, it has long been known that the problem remains NP-Complete if every variable appears at most five times[1]. Recently Tovey[5] showed that the problem remains NP-Complete in this case even if every variable appears at most four times (We first give a much simpler reduction to prove this result). Tovey also showed as a nice application of Hall's theorem that the problem becomes trivial (i.e. the instance is always satisfiable) when every variable appears at most three times. It follows easily that all these results carry over to the MaxSat problem when each clause contains either exactly three or at most three variables.

Here we address the analagous question for the Max2Sat problem where each clause contains at most two variables: what is the minimum for the maximum number of appearances for any variable required for the Max2Sat problem to remain NP-Complete? The original reduction from 3SAT to Max2Sat has some variable appearing 12 times[1], [2] if we start from a 3SAT instance in which each variable appears at most 3 times. Jaumard and Simeone[3], while discussing the complexity of the Maximum Satisfiability Problem for Horn Formulas give a reduction from Vertex Cover to the Max2Sat problem. If we start from a cubic graph, every variable appears at most four times in the resulting Max2Sat instance. This proves that the Max2Sat problem remains NP-Complete even if every variable appears at most four times. We reproduce this reduction in the next section for completeness. Then, by a reduction from Max2Sat instances in which each variable appears at most four times, we show that the Max2Sat problem remains NP-Complete even if every variable appears at most three times. We can also prove that Max2Sat remains NP-Complete even if every clause contains exactly two literals and every variable appears at most three times.

Tovey[5] has shown that when no variable appears in more than two clauses SAT can be solved in linear time. We show that his algorithm can be modified to work for MaxSat in $O(n\sqrt{(n+m)})$ time where n is the number of variables and m is the number of clauses when no variable appears in more than two clauses.

2 Max2Sat when every variable appears at most thrice

First we observe the following from the reduction of Jaumard and Simeone.

Lemma 2.1: *Given a Boolean Formula in CNF form in which every clause has at most two literals and every variable appears in at most four clauses, and an integer k, it is NP-Complete to decide whether there is a truth assignment to the variables that satisfies at least k of the clauses.*

Proof: It is obvious to see that the problem is in NP. The following is the reduction of B. Jaumard and B. Simeone from Vertex Cover to the Max2Sat problem. We will assume that the graph in the instance of the Vertex Cover problem is cubic, i.e. every vertex has degree 3. Vertex Cover problem remains NP-Complete for cubic graphs[2].

Given a cubic graph $G(V, E)$ with n vertices and m edges, introduce for each $i = 1$ to n a variable x_i for the vertex i and define the formula

$$\Phi = (\wedge_{i=1}^{n} x_i) \wedge (\wedge_{\{i,j\} \in E}(\bar{x_i} \vee \bar{x_j}))$$

Φ has $n + m$ clauses with each clause having at most two literals and every variable appearing in at most four clauses.

Now, if G has a vertex cover of size at most k, then setting those variables corresponding to the vertices in the vertex cover 0, all but at most k clauses of Φ can be satisfied. Conversely if there is an assignment that satisfies at least $n + m - k$ clauses of Φ, we can assume without loss of generality that the unsatisfied clauses are the clauses with single literals. For, if there is an edge $\{i, j\} \in E$ in which both x_i and x_j are 1, arbitrarily setting one of x_i and x_j to 0 satisfies that clause (and perhaps more clauses) and unsatisfies only one clause, thus keeping the number of clauses satisfied at least $n + m - k$. Now it is easy to see that the set of vertices corresponding to the variables assigned 0, form a vertex cover of size at most k in G. ▮

Now we prove the main result of the paper.

Theorem 1: *Given a Boolean Formula in CNF form in which every clause has at most two literals and every variable appears in at most three clauses, and an integer k, it is NP-Complete to decide whether there is a truth assignment to the variables that satisfies at least k of the clauses.*

Proof: We reduce from the Max2Sat problem in which every clause has at most two literals and every variable appears in at most four clauses. Let F be a formula in which every clause has at most two variables and every variable appears in at most four clauses. Let U be the set of variables in F and let $d(x)$, for $x \in U$ be the number of clauses in which the variable x appears in F. Let b the number of variables x for which $d(x) = 4$.

For every variable x for which $d(x) = 4$, do the following. First, create a separate variable for each of its occurrences: x_1, x_2, x_3, x_4 and replace the i-th occurrence of the variable with x_i. We further introduce the variables $y_1, y_2, ...y_8$ and the following clauses:

$$
\begin{gathered}
(x_1 \vee y_1)(x_2 \vee y_2)(x_3 \vee y_3)(x_4 \vee y_4) \\
(\bar{x_1} \vee \bar{y_7})(\bar{x_2} \vee \bar{y_8})(\bar{x_3} \vee \bar{y_5})(\bar{x_4} \vee \bar{y_6}) \\
(\bar{y_1} \vee y_5)(\bar{y_2} \vee y_6)(\bar{y_3} \vee y_7)(\bar{y_4} \vee y_8) \\
(\bar{y_1} \vee y_2)(\bar{y_3} \vee y_4)(y_5 \vee \bar{y_6})(y_7 \vee \bar{y_8})
\end{gathered}
$$

Let X be the above set of 16 clauses and let G be the resulting Max2Sat formula. Now clearly every variable in the formula G appears in at most three clauses. We claim that F has an assignment that satisfies at least k of the clauses if and only there is an assignment to the variables of G that satisfies at least $k + 16b$ clauses of G.

If F has an assignment that satisfies at least k clauses, then by assigning the truth value of x to x_1, x_2, x_3 and x_4 for every x appearing 4 times, we can satisfy the k original clauses and the new $16b$ clauses as follows. If all the x_i's are 0, then set all the y_i variables to 1; if all the x_i's are 1, then set al the y_i variables to 0.

Conversely if there is an assignment to G that satisfies at least $k + 16b$ clauses, then we first show that there is an equivalent assignment that satisfies at least $k + 16b$ clauses of G that satisfies all the new $16b$ clauses also. We then show that the only assignments that satisfy all the $16b$ new clauses are the ones in which the truth value of all the x_i values corresponding to the variable x in F are the same (either all 0's or all 1s) and from that we can recover a truth assignment for F that satisfies at least k of the original clauses. The proof will be complete from the following observations.

1. All the 16 clauses of X are satisfiable if $x_1 = x_2 = x_3 = x_4 = 1$ or $x_1 = x_2 = x_3 = x_4 = 0$.

2. If exactly one of the x_is is 1 or if exactly one of the x_is is 0, then at most 15 clauses of X are satisfiable.

3. If exactly two of the x_is are 1, then at most 14 clauses of X are satisfiable.

To obtain an equivalent assignment that satisfies all the $16b$ clauses and still $k+16b$ clauses in total, for each set of 16 new clauses, in case 2 above, we flip the only variable which is 1 or 0 by which we will satisfy one extra (new) clause and we may falsify at most one original clause (the clause containing that variable).

In case 3 above, by flipping the two x_i's from 1 to 0, we will satisfy two extra (new) clauses and falsify at most two original clauses (the two original clauses containing these two variables). ▮

Observe that the new clauses introduced in the above reduction are only two literal clauses. Hence the following corollary follows.

Corollary 2.0: *Given a Boolean Formula in CNF form in which every clause has exactly two literals and every variable appears in at most three clauses, and an integer k, it is NP-Complete to decide whether there is a truth assignment to the variables that satisfies at least k of the clauses.*

Proof: Start with a Max2Sat instance F in which each clause has at most two literals and every variable appears in at most three clauses. For every variable x appearing in a unit clause, introduce a new variable y and replace the unit clause (x) by the two 2 literal clauses $(x \vee y) \wedge (x \vee \bar{y})$ to obtain a formula G in which every clause has exactly two literals. If u out of the total m clauses of F are unit clauses in F, then G has $m + u$ clauses. Furthermore there is an assignment that satisfies at least k clauses of F if and only if there is an assignment that satisfies at least $k + u$ clauses of G.

Now the variables (x) that appeared in unit clauses in F may be appearing in four clauses. Hence use the reduction used in the proof of Theorem 1 to obtain a formula in which there are exactly two literals per clause and every variable appears in at most three clauses. ▮

3 MaxSat when every variable appears at most twice

Tovey[5] has shown that when every variable appears in at most two clauses, SAT can be solved in linear time. His observation is that if every variable appears at most twice and if every clause contains at least two variables, then the formula is always satisfiable. So then it suffices to eliminate the unit clauses, and Tovey deals with the unit clauses in the obvious way. However we have the extra complication that some unit clauses may necessarily have to be falsified to obtain the solution to MaxSat instance.

We observe that his bipartite graph construction can be used to solve the

MaxSat problem also. Given a boolean formula F in which every variable appears in at most two clauses, we assume without loss of generality that each variable, in fact, appears in exactly in two clauses: one purely and another complemented (other variables can be eliminated by suitably assigning values to them). Now construct a bipartite graph $G(V, C, E)$ from the given formula F with the set V containing n vertices corresponding to the n variables, and C containing m vertices corresponding to the m clauses. A vertex $v \in V$ is joined to a vertex $c \in C$ if the variable v is present in the clause C in the formula F. The biparitite graph has $n + m$ vertices and $2n$ edges and every vertex $v \in V$ has degree two. It can be seen that any truth assignment to the variables of F correspond to a matching in G with the size of the matching corresponding to the number of clauses satisfied in F. A matching corresponds to a truth assignment to variables of Fin the following way: if a vertex $c \in C$ is matched with a vertex $v \in V$ then we satisfy the clause C by setting the truth value of v to be true or false respectively depending on whether v appears purely or in complemented form in C. Conversely given any truth assignment to variables of F, we match each satisfied clause with one of the variables that makes it true (The same variable cannot satisfy more than one clause) yielding a matching in G. So now find a maximum matching in G which yields an assignment that satisfies the maximum number of clauses. The maximum matching in the bipartite graph can be found in $O(n\sqrt{(m+n)})$ time using the $O(m\sqrt{(n)})$ algorithm[4] for maximum matching. Thus we have the following theorem.

Theorem 2: *Given a boolean formula in conjunctive normal form with n variables and m clauses in which every variable appears in at most two clauses, an assignment that satisfies the maximum number of clauses can be found in $O(n\sqrt{(m+n)})$ time.*

References

[1]

[2]

[3]

[4]

[5]