On a variant of Monotone NAE-3SAT and the Triangle-Free Cut problem.

Peiyush Jain, Microsoft Corporation.

March 30, 2010

Abstract

In this paper we define a restricted version of Monotone NAE-3SAT and show that it remains NP-Complete even under that restriction. We expect this result would be useful in proving NP-Completeness results for problems on \( k \)-colourable graphs \((k \geq 5)\). We also prove the NP-Completeness of the Triangle-Free Cut problem.

1 Introduction

SAT was first proven to be NP-Complete in a seminal paper by Cook \[1\] which led to the current theory around NP-Completeness. Many variants of SAT have been shown to be NP-Complete and the Not-All-Equal 3SAT \[2\] (or NAE-3SAT in short) has been found to be particularly useful.

SAT variants have been especially useful in proving NP-Completeness of various graph related problems for restricted classes of graphs. For instance, the planar version of 3SAT, Planar-3SAT is known to be NP-Complete \[4\] and has been useful in proving NP-Completeness of planar graph related problems. Unfortunately, the planar version of NAE-3SAT is in P \[5\].

We consider a version of NAE-3SAT which might prove useful in proving NP-Completeness results of \( k \)-colourable graphs, for \( k \geq 5 \). For instance, from the result of this paper, it immediately follows that MAX-CUT is NP-Complete for 5-colourable graphs. Of course, this is weaker than the currently known result that MAXCUT is NP-Complete for 3-colourable graphs \[7\], but we expect this result might still prove to be useful for other problems.

We also show that the triangle-free cut problem is NP-Complete. This is the vertex colouring variant of the Monochromatic Triangle Problem, which is known to be NP-Complete \[3\].
2 Preliminaries

In this section we define the problems about which we are going to prove NP-Completeness results.

2.1 \( k \)-colourable Monotone NAE-3SAT

Given any monotone CNF expression \( \phi \) with exactly three distinct (un-negated) literals per clause, we associate a simple undirected graph \( G(\phi) \) with \( \phi \) as follows:

Each variable of \( \phi \) is a vertex of the graph, and two vertices have an edge between them iff they appear in some clause together. A variant is to have an additional vertex for each clause and connect them to the variables which appear in them. The result of this paper holds true for both the variants.

We can now define our problem.

Definition. The \( k \)-colourable Monotone NAE-3SAT decision problem: Given a monotone CNF expression \( \phi \) with exactly three distinct variables in each clause, such that the corresponding graph \( G(\phi) \) defined earlier is \( k \)-colourable, is the expression \( \phi \) not-all-equal satisfiable?

It is easy to see that 4-colourable monotone NAE-3SAT is in P. We show that 5-colourable Monotone NAE-3SAT is NP-Complete, by going via a reduction that also proves the NP-Completeness of the Triangle-Free cut problem (defined below).

2.2 Triangle Free Cut

Suppose we are given a graph and we want to know if there is a colouring of the vertices with two colours so that there is no monochromatic triangle. This problem can be stated in terms of cuts and triangle-free graphs.

Definition. Triangle-Free Cut: Given a simple, undirected graph \( G(V, E) \) with vertex set \( V \) and edge set \( E \), a cut of \( G \) is a partition of \( V \) into two non-empty sets \( V_1 \) and \( V - V_1 \). We say that the cut is triangle-free if the induced sub-graphs of \( G \) on \( V_1 \) and \( V - V_1 \) both do not contains any triangles.

Thus we have:

Definition. The Triangle-Free Cut decision problem: Given a simple, undirected graph \( G(V, E) \), is there a triangle-free cut of \( G \)?

It is easy to see that this problem is in NP.

For the class of 4-colourable graphs, the problem of determining the existence of a triangle-free cut is trivial: Just form a cut by putting vertices...
of two colours in one set and the remaining coloured vertices in the other.

We show that Triangle-Free Cut problem is NP-Complete for the class of 5-colourable and bounded degree (\(\leq 8\)) graphs.

3 NP-Completeness of Triangle-Free Cut

We start with the version of Monotone-NAE-3SAT in which each clause has exactly three distinct variables [6]. Given such a Monotone-NAE-3SAT expression, we construct a graph which has a triangle free cut if and only if the expression is NAE-satisfiable. We show that the constructed graph has the property that it is 5-colourable and has maximum degree \(\leq 8\).

i.e. we prove the following theorem:

**Theorem 1.** The Triangle-Free Cut decision problem is NP-Complete for the class of 5-colourable and maximum degree \(\leq 8\) graphs.

**Proof.** Suppose we are given a monotone expression with variables \(x_1, x_2, ..., x_n\) and clauses \(c_1, ..., c_m\). We first transform this expression as follows:

If a variable \(x = x_1\) (say) appears \(k > 1\) times, we replace it with \(k\) variables \(y_j, 1 \leq j \leq k\) and add \(k - 1\) clauses to ensure that the \(y_j\) variables get the same truth values.

wlog, assume that \(x = x_1\) appears in clauses \(c_1, c_2, ..., c_k\).

Say \(c_i = (x \lor a_i \lor b_i), 1 \leq i \leq k\)

We replace these clauses with the \(k\) new clauses:

\[ c''_i = (y_i \lor a_i \lor b_i), 1 \leq i \leq k \]

and add \(k - 1\) new clauses

\[ c''_i = (y_i \lor \neg y_{i+1}), 1 \leq i \leq k - 1 \]

We do this for each repeated variable. This transformation is a polynomial time transform and the resulting expression is NAE-satisfiable if and only if the original expression is NAE-satisfiable.

The resulting expression has the following properties:

1. Each clause is of the form \((x \lor y \lor z)\) or \((x \lor \neg y)\).
2. Each variable appears no more than three times.
3. Any two variables appear together in no more than one clause.
4. A variable appears exactly once in a clause with three literals.
5. If a variable appears exactly once or twice, it appears un-negated.
6. If a variable appears thrice, the negated literal appears exactly once.

We now construct a graph \(G(V, E)\) from the expression we just created.

We associate a vertex with each variable. For any clause with exactly three variables (note that the literals here are unnegated), we connect
the corresponding vertices, forming a triangle. Observe that no two such triangles have a common vertex or edge.

For clauses of the form \((x \lor \neg y)\), we 'connect' them using the glued-tetrahedra gadget in figure 1.

**Figure 1: Tetrahedra Gadget**

In the gadget, notice that \(x\) and \(y\) are not adjacent, of degree 3 and that it is 5-colourable even if we assign arbitrary colours to \(x\) and \(y\). Also notice that any triangle-free cut of this gadget necessarily has \(x\) and \(y\) in the same set and that there actually exists a triangle-free cut of this gadget.

Now the graph so formed is 5-colourable: We pick a three literal clause and colour the corresponding vertices of the triangle arbitrarily with three different colours. Given any clause of the form \((x \lor \neg y)\), we already have a colouring of \(x\) and \(y\) and the vertices in the gadget connecting them can be coloured so as to maintain 5-colourability.

Also notice that the maximum degree of any vertex in this graph is 8, as it can be part of atmost two gadgets and one triangle.

Now consider any triangle in the graph. This is either a triangle formed due to a three variable clause, or is a triangle completely within a gadget instance.

We now claim that this graph has a triangle-free cut if and only if the expression is NAE-Satisfiable.
3.1 NAE-Satisfiability implies existence of triangle-free cut

Assume the expression was NAE-Satisfiable. Now put the variables which are set to true in one set and the variables which are set to false in another. We now show that this can be extended to give a triangle-free cut of the graph.

For any triangle which is formed because of a clause, since not all variables are true (or false), at least one edge of the triangle is part of the cut.

Any clause of the form \((x \lor \neg y)\) is NAE-Satisfiable if both \(x\) and \(y\) get assigned true, or both get assigned false. Now this implies that the corresponding gadget vertices can be assigned to the vertex sets so that all the triangles in the gadget are ‘cut’ by the cut.

Thus, if the expression is NAE-Satisfiable, then the graph has a triangle-free cut.

3.2 Existence of triangle-free cut implies NAE-Satisfiability

Now assume that the graph has a triangle-free cut. Pick any one vertex set of the cut and set the variables in those to true. Set the remaining variables to false. Now this assignment makes the expression NAE-satisfiable.

Each triangle corresponding to a three variable clause has at least one true and one false vertex, thus making the clause NAE-Satisfiable. For any clause of the form \((x \lor \neg y)\), \(x\) and \(y\) are necessarily in the same set, thus making the clause NAE-Satisfiable.

This proves the NP-Completeness of the triangle-free cut problem on 5-colourable, bounded degree \((\leq 8)\) graphs. \(\square\)

4 NP-Completeness of \(k\)-colourable Monotone-NAE-3SAT

We prove the following:

**Theorem 2.** \(k\)-colourable Monotone-NAE-3SAT is NP-Complete for \(k \geq 5\). Moreover, it remains NP-Complete if each variable is restricted to appear no more than 7 times.

**Proof.** It is easy to see that the above reduction to triangle-free cut carries over to the 5-colourable Monotone NAE-3SAT in a natural manner (by making the vertices variables, and triangles the clauses) such that the graph has a triangle free cut if and only if the corresponding 5-colourable Monotone NAE-3SAT expression is satisfiable. Moreover, each vertex of the graph appears in at most 7 triangles (2 gadgets and 1 clause triangle.
or 5 triangles for a gadget vertex) and so each variable in the expression appears no more than 7 times.

Thus 5-colourable Monotone NAE-3SAT is NP-Complete and remains so even under the restriction that each variable appears no more than 7 times.

5 Conclusion

We have shown that the decision versions of $k$-colourable Monotone NAE-3SAT and Triangle-Free Cut problem for $k$-colourable graphs are NP-Complete for $k \geq 5$. The next steps would be to prove the NP-Completeness (or give polynomial time algorithms) of the corresponding search problems when $k = 3$ or $k = 4$.

References

[1] Cook, S. A. 1971. "The complexity of theorem-proving procedures". In Proceedings of the Third Annual ACM Symposium on theory of Computing (Shaker Heights, Ohio, United States, May 03 - 05, 1971). STOC ’71. ACM, New York, NY, 151-158.

[2] T. J. Schaeffer. 1978. "The complexity of satisfiability problems", Proc. 10th STOC, (1978), pp. 216–226.

[3] M.R. Garey, D.S. Johnson. 1979. Computers and Intractability, A Guide to the Theory of NP-Completeness, W.H. Freeman and Co., New York, 1979.

[4] D. Lichtenstein. 1981. "Planar formulae and their uses", SIAM J. on Computing 11 (1981).

[5] Moret, B. M. 1988. "Planar NAE3SAT is in P". SIGACT News 19, 2 (Jun. 1988), 51-54.

[6] Moret, B. M. 1998. "The Theory of Computation", Addison-Wesley-Longman, 1998. Chapter 7, Problem 7.1, Part 2.

[7] H. L. Bodlaender and K. Jansen. 2000. "On the complexity of the maximum cut problem". Nordic Journal of Computing, 7(1):14-31, 2000.