# Planar NAE3SAT is in P

Bernard M.E. Moret

Department of Computer Science
The University of New Mexico
Albuquerque, NM 87131

*Abstract*

*Lichtenstein proposed the use of a "planar" version of 3SAT as a departure point for proofs of* **NP**-*completeness that would at once prove the general graph problem as well as its restriction to planar graphs to be* **NP**-*complete. Dyer and Frieze later proved that Planar 1in3SAT is also* **NP**-*complete. Of the three standard versions of 3SAT, only NAE3SAT remained open. We provide a very simple reduction from NAE3SAT to Simple MaxCut, which respects Lichtenstein's conditions; since it is known that the planar version of (Simple) MaxCut is in* **P**, *so is the planar version of NAE3SAT, thereby settling the question (if in a somewhat unexpected way).*

Among published proofs of **NP**-completeness, one finds more reductions from *3-Satisfiability* (*3SAT* for short) and its main variants, *One-in-three-3SAT (1in3SAT)* and *Not-all-equal 3SAT (NAE3SAT)*, than from any other **NP**-complete problem. In particular, a great many proofs of **NP**-completeness for graph problems are constructed from *3SAT*. Since the transformations used rarely produce planar graphs, such proofs as a rule leave open the question of whether or not the planar restriction of the graph problem is itself **NP**-complete. This question then gets answered laboriously, if in the affirmative, by the construction of a crossing component for the graph.

Remarking that most transformations from *3SAT* are made up of what Garey and Johnson [1979] call local replacements, Lichtenstein [1982] conceived the idea of defining a "planar" version of *3SAT* which would only give rise, through local replacements, to planar instances of the graph problem of interest. Thus only one crossing component had to be designed—for the planar *3SAT* itself—and proofs of **NP**-completeness for planar graph problems would be greatly simplified. Since many **NP**-completeness proofs are also derived from the main two variants of *3SAT* mentioned above, it was natural that interest would turn to planar versions of those. Dyer and Frieze [1986] proved that the planar version of *1in3SAT* is also **NP**-complete, but no one was able to place planar *NAE3SAT* in either camp.

We prove that Planar *NAE3SAT* is in **P** by (what else?) reduction to a known problem in **P**, namely Planar (Simple) *MaxCut*. Our reduction uses only local replacement and carefully expands the single variable nodes of Planar *NAE3SAT* into chains of nodes for *MaxCut* so as to produce only planar instances of *MaxCut*. Orlova and Dorfman [1972] showed that such instances can be solved in polynomial time by a matching process carried on the dual of the graph. Thus Planar *NAE3SAT* is in **P**—a rather surprising result.

An instance of Planar *NAE3SAT* is defined as an instance of *NAE3SAT*, except that the graph produced from it according to the following rules must be planar. The graph has one vertex for each variable of the instance and one vertex for each clause of the instance; all variable vertices are connected in a simple cycle and each clause vertex is connected by an edge to variable vertices corresponding to the literals present in the clause. Note that positive and negative literals are treated exactly alike. (Simple) *MaxCut* is defined as follows: given an undirected graph and given a bound (a positive integer), can the vertices of the graph be partitioned into two subsets such that the number of edges with endpoints in both subsets is no smaller than the given bound?

Our transformation is quite elementary. Given an instance of Planar *NAE3SAT* with $k$ clauses, we transform it in polynomial time into an instance of *MaxCut* with $9k$ vertices and no more than $6k$ edges as follows. For each variable $x$ forming a total of $n_x$ literals in the $k$ clauses, we set up a simple cycle of $2n_x$ vertices (and $2n_x$ edges); alternating vertices represent complemented and uncomplemented literals.[1] For each clause, we set up a triangle, where each vertex of the triangle is connected by an edge to the (complement of the) corresponding "literal vertex." Finally, we set the minimum number of edges to be cut to $11k$ This transformation is clearly feasible in polynomial time.

Now, given a satisfying truth assignment, we put all vertices correspond-

[1] Note that, if $n_x = 1$, this would produce a double edge between two vertices; in this case only, we set up a single edge between the two vertices and lower the required number of cut edges by one. Since this is an elementary adjustment we shall assume in the following that $n_x \geq 2$ rather than qualify every statement to take this special case into account.

ing to true literals on one side of the partition and all others on the other side. Since the truth assignment is valid, each edge between a literal and its complement is cut, thereby contributing a total (for all variables) of $6k$ (i.e., twice the total number of literals) to the cut sum. Since the truth assignment is a solution, each triangle is cut (not all three vertices may be on the same side, as this would correspond to a clause with three false or three true literals), thereby contributing a total of $5k$ to the cut sum. Hence we have a solution to *MaxCut*. Conversely, observe that $11k$ is the maximum attainable cut sum: we cannot do better than cut each clause triangle and each segment between complementary literals. Moreover, the cut sum of $11k$ can only be reached by cutting all triangles and segments. (If all three vertices of a clause triangle are placed on the same side of the partition, at most 4 of the 6 edges associated with the clause can be cut.) Hence a solution to *MaxCut* yields a solution to *NAE3SAT*: cutting the segments ensures a valid truth assignment and cutting each triangle ensures a satisfying truth assignment.

Only one point remains: to verify that the instance produced is indeed planar. No crossing can arise from the replacement of the single clause vertices of Planar *NAE3SAT* the triangles of *MaxCut*, as the summits of the triangles are assigned arbitrarily. The replacement of the single variable vertices of Planar *NAE3SAT* by the simple cycles of *MaxCut* likewise cannot produce non-planar configurations, as the choice of vertex pair for some literal $\tilde{x}$ from among the $n_x$ pairs present in the circle is also arbitrary.[2] The correctness of this transformation completes our proof.

It is interesting to note that rather more transformations from the *3SAT* family of problems to graph problems produce a single edge per variable (with the endpoints intended to represent the two truth value assignments) than produce a single vertex or arbitrarily labelled structure. Thus Lichtenstein's version of Planar *3SAT* is rather limited; a potentially more useful version would constrain instances according to a different graph construction, where each variable is in fact represented by a single edge and where the single vertex clause is connected to the proper (i.e., complemented or not) endpoint of the variable's edge. Simplifying the transformation given above to one where a single edge (rather than a simple cycle) is produced for each variable—thus yielding the very simple transformation given in Moret and Shapiro [1985]—shows (this time predictably) that Planar *NAE3SAT*

[2]Our thanks to Michael Fellows for pointing out that a single pair of vertices per variable could lead to non-planar instances of *MaxCut*.

remains in **P** under this new definition.


Dyer, M.E., and A.M. Frieze [1986], "Planar 3DM is **NP**-complete," J. of Algorithms 7, pp. 174–184.

Garey, M.R., and D.S. Johnson [1979]. *Computers and Intractability: a Guide to the Theory of* **NP**-*Completeness*. W.H. Freeman, San Francisco.

Lichtenstein, D. [1982], "Planar formulae and their uses," SIAM J. Comput. 11, pp. 329–343.

Moret, B.M.E., and H.D. Shapiro [1985], "Using symmetry and rigidity: a simpler approach to **NP**-completeness proofs," University of New Mexico, Dept. of Computer Science Tech. Rep. CS85-8.

Orlova, G.I., and Y.G. Dorfman [1972], "Finding the maximum cut in a graph," Eng. Cybernetics 10, pp. 502–506.