

## THE COMPLEXITY OF PLANAR COUNTING PROBLEMS\*

HARRY B. HUNT III<sup>†</sup>, MADHAV V. MARATHE<sup>‡</sup>, VENKATESH RADHAKRISHNAN<sup>§</sup>,  
AND RICHARD E. STEARNS<sup>†</sup>

**Abstract.** We prove the  $\#P$ -hardness of the counting problems associated with various satisfiability, graph, and combinatorial problems, when restricted to planar instances. These problems include 3SAT, 1-3SAT, 1-EX3SAT, MINIMUM VERTEX COVER, MINIMUM DOMINATING SET, MINIMUM FEEDBACK VERTEX SET, X3C, PARTITION INTO TRIANGLES, AND CLIQUE COVER. We also prove the NP-completeness of the AMBIGUOUS SATISFIABILITY problems [J. B. Saxe, *Two Papers on Graph Embedding Problems*, Tech. Report CMU-CS-80-102, Dept. of Computer Science, Carnegie Mellon Univ., Pittsburgh, PA, 1980] and the  $D^P$ -completeness (with respect to random polynomial reducibility) of the unique satisfiability problems [L. G. Valiant and V. V. Vazirani, *NP is as easy as detecting unique solutions*, in Proc. 17th ACM Symp. on Theory of Computing, 1985, pp. 458–463] associated with several of the above problems, when restricted to planar instances. Previously, very few  $\#P$ -hardness results, no NP-hardness results, and no  $D^P$ -completeness results were known for counting problems, ambiguous satisfiability problems, and unique satisfiability problems, respectively, when restricted to planar instances.

Assuming  $P \neq NP$ , one corollary of the above results is that there are no  $\epsilon$ -approximation algorithms for the problems of maximizing or minimizing a linear objective function subject to a planar system of linear inequality constraints over the integers.

**Key words.** planar, 3SAT, graphs,  $\#P$ -complete,  $D^P$ -complete, NP-complete

**AMS subject classifications.** 68Q15, 68R10.

**PII.** S0097539793304601

**1. Introduction.** A number of papers in the literature, including [1, 5, 16, 20, 21, 25, 26], have considered the complexity of counting problems, proving many such problems to be  $\#P$ -complete. Other papers have studied the complexity of ambiguous and unique satisfiability problems [23, 27], proving such problems to be NP-hard and  $D^P$ -hard,<sup>1</sup> respectively. Still other papers [3, 4, 5, 6, 15] have considered the complexity of decision problems when restricted to planar instances, proving many such problems to be NP-hard. In this paper, we combine these lines of research and prove for the first time that *even when restricted to planar instances*, many counting problems remain  $\#P$ -complete, many ambiguous satisfiability problems remain NP-complete, and many unique satisfiability problems remain  $D^P$ -hard.

Previously, very few  $\#P$ -hardness results, no NP-hardness results, and no  $D^P$ -completeness results were known for counting problems, ambiguous satisfiability problems, and unique satisfiability problems, respectively, when restricted to planar in-

\*Received by the editors August 15, 1993; accepted for publication (in revised form) June 12, 1996; published electronically May 19, 1998.

<http://www.siam.org/journals/sicomp/27-4/30460.html>

<sup>†</sup>Department of Computer Science, State University of New York (SUNY) at Albany, Albany, NY 12222 (hunt@cs.albany.edu, res@cs.albany.edu). This work was supported by NSF grants CCR 89-03319, CCR 94-06611, and CCR 90-06396.

<sup>‡</sup>Los Alamos National Laboratory, P.O. Box 1663, MS B265, Los Alamos, NM 87545 (marathe@lanl.gov). This research was supported by the Department of Energy under contract W-7405-ENG-36. Part of the work was done while the author was at SUNY-Albany and supported by NSF grants CCR 89-03319 and CCR 94-06611.

<sup>§</sup>Mailstop 47LA-2, Hewlett-Packard Company, 19447 Pruneridge Avenue, Cupertino, CA 95014-9913 (rven@cup.hp.com). Part of this work was done while the author was at SUNY-Albany and was supported by NSF grants CCR 89-03319 and CCR 94-06611.

<sup>1</sup>Throughout this paper, all the  $D^P$ -hardness results are with respect to random polynomial time reductions.

stances. Such results are presented for several satisfiability, graph, and combinatorial problems. These results show that planar counting, planar ambiguous satisfiability, and planar unique satisfiability problems are *as hard as* arbitrary such problems, with respect to polynomial time and random polynomial time reducibilities. The results in this paper both extend the results in the literature and also provide additional tools for proving hardness results for planar problems for various complexity classes. These tools include parsimonious and weakly parsimonious crossover boxes, the NP-hardness of various basic planar satisfiability problems, and the NP-hardness of the planar ambiguous satisfiability problems. (Henceforth, we denote the restriction of a problem  $\Pi$  to planar instances by PL- $\Pi$ .) The particular results presented here include the following.

1. The problem 3SAT has a parsimonious planar crossover box. Among other things, this implies that the problem #PL-3SAT is #P-complete, the problem AMBIGUOUS-PL-3SAT is NP-complete, and the problem UNIQUE-PL-3SAT is  $D^P$ -complete.

2. The problem 3SAT is simultaneously polynomial time, planarity-preserving, and parsimoniously reducible to each of the basic CNF satisfiability problems listed in Table 1.1. (Previously, 3SAT was only known to be so reducible to the problem 1-EX3SAT [3].)

3. There exist polynomial time, weakly parsimonious, and planarity-preserving reductions from the problem 1-EX3MONOSAT to several graph problems including MINIMUM VERTEX COVER, MINIMUM DOMINATING SET, and MINIMUM FEEDBACK VERTEX SET.

4. Using the results 1, 2, and 3, variants of known reductions, and new reductions, we show that for all of the problems PL- $\Pi$  in Table 1.1, the problem #PL- $\Pi$  is #P-hard. Similarly, we show that for many of the problems in Table 1.1, the problems AMBIGUOUS PL- $\Pi$  and UNIQUE PL- $\Pi$  are NP-hard and  $D^P$ -hard, respectively.

5. All of the #P-hardness results for planar counting problems in Table 1.1 can easily be shown to hold, even when the input to the problem consists of a formula  $f$  and one of its satisfying assignments, a graph  $G$  and one of its minimum vertex covers, a graph  $G$  and one of its dominating sets, etc.

Thus the #P-hardness of the problems #PL- $\Pi$  in Table 1.1 is not simply a corollary of the NP-hardness of the problem PL- $\Pi$ , since the problems #PL- $\Pi$  are #P-hard even when restricted to sets of instances for which the problems PL- $\Pi$  are trivially polynomial time solvable. Moreover, quoting from [27] we note that “whether the number of solutions of all NP-complete problems are nevertheless polynomial time interreducible (i.e., whether NP-completeness implies #P-completeness) is still open.”

Corollaries of our results and their proofs include the following.

1. The problems #1-VALID 3SAT and #1-VALID PL-3SAT are #P-complete. (It is trivially seen that every instance of the problem 1-VALID 3SAT is satisfiable by the assignment making all variables equal to 1. See Definition 2.10.)

2. The problems AMBIGUOUS 1-VALID 3SAT and AMBIGUOUS 1-VALID PL-3SAT are NP-complete. The problems UNIQUE 1-VALID 3SAT and UNIQUE 1-VALID PL-3SAT are Co-NP-complete.

3. Assuming  $P \neq NP$ , there are no  $\epsilon$ -approximation algorithms for the problems of maximizing or minimizing a linear objective function subject to a planar system of inequalities over the integers.

Table 1.1 gives a summary of our #P-hardness results. The rest of the paper is organized as follows. Section 2 contains definitions and preliminaries. Section 3

discusses the complexity of #PL-3SAT and other basic CNF satisfiability problems. These problems are used to prove the #P-hardness of other problems discussed in the subsequent sections. Section 4 discusses the complexity of various counting problems for planar graphs. Section 5 contains the ambiguous and unique satisfiability results and the result on the nonapproximability of the objective functions of integer linear programs. Finally, section 6 consists of conclusions and open problems.

TABLE 1.1

Summary of NP- and #P-hardness results for planar instances. The third column summarizes the decision complexity of the problems while the fourth column summarizes the complexity of the counting versions. A star (\*) denotes a result obtained in this paper. The numbers in square brackets are the references where the corresponding results are proved.

No.	Problem	Decision problem (NP-complete)	Counting version #P-hard
1	PL-3SAT	[15]	*
2	PL-EX3SAT	*	*
3	PL-1-3SAT	[3]	*
4	PL-1-EX3SAT	[3]	*
5	PL-1-EX3MONOSAT	*	*
6	PL-VERTEX COVER	[6]	*
7	PL-HAMILTONIAN CIRCUIT	[15]	[20]
8	PL-DOMINATING SET	[6]	*
9	PL-FEEDBACK VERTEX SET	[6]	*
10	PL-3-COLORING	[6]	[11]
11	PL-GRAPH HOMOMORPHISM and ONTO HOMOMORPHISM	[5]	[11]
12	PL-SUBGRAPH ISOMORPHISM	[5]	*
13	PL-CLIQUE COVER	[3]	*
14	PL-HITTING SET	[3]	*
15	PL-X3C	[3]	*
16	PL-PARTITION INTO TRIANGLES	[3]	*
17	PL-PARTITION INTO CLAWS	[3]	*

**2. Definitions and preliminaries.** In this section we review the basic definitions and notation used in this paper. Additional definitions can be found in [3, 5, 19, 23].

**DEFINITION 2.1.** A search problem  $\Pi$  consists of a set  $D_\Pi$  of objects called instances, and for each instance  $I \in D_\Pi$ , a set  $S_\Pi[I]$  of objects called solutions for  $I$ . An algorithm is said to solve a search problem  $\Pi$  if, given  $I \in D_\Pi$  as input, the algorithm outputs no if  $S_\Pi[I] = \emptyset$  and outputs an  $s \in S_\Pi[I]$  otherwise.

**DEFINITION 2.2.** The enumeration problem associated with a search problem  $\Pi$  is the problem of determining, given  $I \in D_\Pi$ , the cardinality of  $S_\Pi[I]$ .

**DEFINITION 2.3.** The class #P consists of all enumeration problems associated with search problems  $\Pi$  such that there is a nondeterministic algorithm for solving  $\Pi$  such that, for all  $I \in D_\Pi$ , the number of distinct accepting sequences for  $I$  by the algorithm equals  $|S_\Pi[I]|$  and the length of the longest accepting computation of the algorithm on  $I \in D_\Pi$  is bounded by a polynomial in the length of  $I$ .

**DEFINITION 2.4.** A reduction [5]  $f : D_\Pi \rightarrow D_{\Pi'}$  is parsimonious if and only if  $\forall I \in D_\Pi$  the number of solutions of  $I$  is equal to the number of solutions of  $f(I)$ .

**DEFINITION 2.5.** A reduction  $f$  is weakly parsimonious if and only if  $|S(I)| = g(I)|S(f(I))|$ , where  $|S(I)|$  and  $|S(f(I))|$  denote the number of solutions of  $I$  and  $f(I)$ , respectively, and  $g(I)$  is a polynomial time computable function represented using binary notation.

An enumeration problem is said to be  $\#P$ -hard if each problem in  $\#P$  is polynomial time parsimoniously or weakly parsimoniously Turing reducible to it. If, in addition, the enumeration problem is in  $\#P$ , the problem is said to be  $\#P$ -complete.

DEFINITION 2.6.  $\#3SAT$  is the problem of computing the number of satisfying assignments of a Boolean formula  $F$  in conjunctive normal form with at most three literals per clause.

The following basic results on the complexity of counting problems are used in this paper.

THEOREM 2.7 (see [5, 25]). The problems  $\#3SAT$  and  $\#GRAPH\ 3-COLORING$  are  $\#P$ -complete.

DEFINITION 2.8. The bipartite graph associated with a CNF satisfiability problem is defined as follows. The clauses and variables in a formula are in one to one correspondence with the vertices of the graph. There is an edge between a clause node and a variable node if and only if the variable appears in the clause. A CNF formula is planar if and only if its associated bipartite graph is planar.

DEFINITION 2.9. (1)  $EX3SAT$  is the restriction of the problem  $3SAT$  to formulas in which each clause has exactly three literals.

(2)  $1-3SAT$  is the problem of determining if a CNF formula in which each clause has no more than three literals has a satisfying assignment such that exactly one literal per clause is satisfied.

(3)  $1-EX3SAT$  is the problem of determining if a CNF formula in which each clause has exactly three literals has a satisfying assignment such that exactly one literal per clause is satisfied.

(4)  $1-EX3MONOSAT$  is the restriction of  $1-EX3SAT$  to formulas having no negated literals.

DEFINITION 2.10 (see [24]). A relation  $R(x_1, x_2, \dots, x_n)$  is 1-valid if and only if  $(1, 1, \dots, 1) \in R$ . A CNF formula  $f$  is 1-valid if the formula is satisfied when all the variables in the formula are set to true.

DEFINITION 2.11. Given a Boolean formula  $F$  and an assignment  $\mathbf{v}$  to the variables of  $F$ , the notation  $\mathbf{v}[F]$  denotes the value of  $F$  under  $\mathbf{v}$ .

DEFINITION 2.12. (1)  $EXACT\ COVER\ BY\ 3-SETS\ (X3C)$ : An instance of this problem consists of a set  $X$  with  $3m$  elements and a collection  $C$  of three-element subsets of  $X$ . The question is: does there exist a subcollection  $C'$  of  $C$  such that every element of  $X$  occurs in exactly one set in  $C'$ ?

(2)  $HITTING\ SET$ : An instance of this problem consists of a collection  $C$  of subsets of a finite set  $S$  and a positive integer  $K \leq |C|$ . The question is: is there a subset  $S' \subseteq S$  with  $|S'| \leq K$  such that  $S'$  contains at least one element from each subset in  $C$ ?

As in the case of  $3SAT$ , we can associate a bipartite graph  $G = (S, T, E)$  with an instance of each of the above problems. For example,  $PL-X3C$  is defined as follows. Each element in  $C$  has a corresponding vertex in  $S$ , each element in  $X$  has a corresponding vertex in  $T$ , and a vertex in  $S$  is joined to a vertex in  $T$  if and only if the set corresponding to the vertex in  $S$  contains the element corresponding to the vertex in  $T$ .

DEFINITION 2.13. (1)  $DOMINATING\ SET$ : An instance of this problem consists of an undirected graph  $G = (V, E)$  and an integer  $K \leq |V|$ . The question is: is there a dominating set of size at most  $K$  in  $G$ ; i.e., is there a subset  $V'$  of  $V$ ,  $|V'| \leq K$ , such that for each  $u \in V - V'$  there is a  $v \in V'$  such that  $(u, v) \in E$ ?

(2)  $CLIQUE\ COVER$ : An instance of this problem consists of an undirected graph  $G = (V, E)$  and an integer  $K \leq |V|$ . The question is: is there a clique cover of size

at most  $K$  in  $G$ ; i.e., can the graph be partitioned into at most  $K$  sets of nodes such that each set is a clique?

(3) PARTITION INTO CLAWS: An instance of this problem consists of an undirected graph  $G = (V, E)$ ,  $|E| = m$ . The question is: is there a way to partition the edges of the graph into sets  $E_1, \dots, E_s$ ,  $s = m/3$ , such that each  $E_i$  induces a subgraph isomorphic to  $K_{1,3}$  (i.e., a claw)?

(4) FEEDBACK VERTEX SET: An instance of this problem consists of a directed graph  $G = (V, E)$  and an integer  $K \leq |V|$ . The question is: is there a feedback vertex set of size at most  $K$  in  $G$ ; i.e., does there exist a subset  $V'$  of  $V$  of size at most  $K$  such that  $V'$  contains at least one vertex from every cycle in  $G$ ?

DEFINITION 2.14. Let  $\Pi$  denote a CNF satisfiability problem. Then the associated ambiguous version of  $\Pi$ , denoted by AMBIGUOUS- $\Pi$ , is the problem of determining, given an instance  $I$  of  $\Pi$  and an assignment  $\mathbf{v}$  to the variables of  $I$  such that  $\mathbf{v}[I] = 1$ , if there is an additional assignment of values to the variables of  $I$  satisfying  $I$ . The associated unique version of  $\Pi$ , denoted by UNIQUE- $\Pi$ , is the problem of determining if the given instance  $I$  of  $\Pi$  has exactly one satisfying assignment.

More generally, let  $\Pi$  be any decision problem. Henceforth, when applicable, we denote the restriction of  $\Pi$  to planar instances by PL- $\Pi$ , the counting version of  $\Pi$  by # $\Pi$ , the ambiguous version of  $\Pi$  by AMBIGUOUS- $\Pi$ , and the unique version of the problem by UNIQUE- $\Pi$ . For example, recalling Definition 2.12 and the discussion immediately following it,

(i) PL-X3C is the problem X3C restricted to instances  $(X, C)$  for which the bipartite graph  $G = (S, T, E)$  is planar;

(ii) #X3C is the problem of computing, given  $(X, C)$ , the number of distinct subsets  $C' \subseteq C$  such that each element of  $X$  occurs in exactly one set in  $C'$ ;

(iii) AMBIGUOUS-X3C is the problem of determining, given  $(X, C, C')$ , where  $C' \subseteq C$  and each element of  $X$  occurs in exactly one set in  $C'$ , if there exists another subset  $C'' \subseteq C$  that is an exact cover of the elements in  $X$ .

Finally, henceforth, by *reduction*, we mean a polynomial time many-one reduction.

**3. Basic planar counting problems.** In this section, we prove that the problem PL-3SAT has a parsimonious crossover box. Specifically, we show that the crossover box in [15] is parsimonious. One immediate corollary is that the problem #PL-3SAT is #P-hard. We also prove that the problem 3SAT is planarity-preserving and parsimoniously reducible to each of the basic SAT problems listed in Table 1.1.

DEFINITION 3.1. A crossover box for a satisfiability problem  $\Pi$  is a formula  $f_c$  with four distinguished variables  $a, a_1, b, b_1$ , which can be laid out on the plane with the distinguished variables on the outer face, such that

(1) the old variables  $a$  and  $b$  are opposite to the corresponding new variables  $a_1$  and  $b_1$ ,

(2) each assignment to  $a$  and  $b$  can be extended to a satisfying assignment of  $f_c$ ,

(3) for any satisfying assignment of  $f_c$ ,  $a \equiv a_1$  and  $b \equiv b_1$ .

The crossover box is parsimonious if and only if for each assignment to the old variables, there is exactly one extension of this assignment to the variables in the crossover box such that  $f_c$  is satisfied.

THEOREM 3.2. The problem 3SAT has a parsimonious planar crossover box. Hence, 3SAT is parsimoniously reducible to PL-3SAT and #PL-3SAT is #P-complete.

*Proof.* The crossover box described here is the same as the one given in [15]. Here we prove that the crossover box also preserves the number of solutions. For expository purposes, we describe the crossover box in two steps. The first step is to consider the

following formula:

$$\begin{aligned}
 f_c = & (\overline{a_2} + \overline{b_2} + \alpha) \wedge (a_2 + \overline{\alpha}) \wedge (b_2 + \overline{\alpha}) \bigwedge \\
 & (\overline{a_2} + b_1 + \beta) \wedge (a_2 + \overline{\beta}) \wedge (\overline{b_1} + \overline{\beta}) \bigwedge \\
 & (a_1 + b_1 + \gamma) \wedge (\overline{a_1} + \overline{\gamma}) \wedge (\overline{b_1} + \overline{\gamma}) \bigwedge \\
 & (a_1 + \overline{b_2} + \delta) \wedge (\overline{a_1} + \overline{\delta}) \wedge (b_2 + \overline{\delta}) \bigwedge \\
 & (\alpha + \beta + \gamma + \delta) \bigwedge \\
 & (\overline{\alpha} + \overline{\beta}) \wedge (\overline{\beta} + \overline{\gamma}) \wedge (\overline{\gamma} + \overline{\delta}) \wedge (\overline{\delta} + \overline{\alpha}) \bigwedge \\
 & (a_2 + \overline{a}) \wedge (a + \overline{a_2}) \wedge (b_2 + \overline{b}) \wedge (b + \overline{b_2}).
 \end{aligned}$$

Following [15], we give an intuitive explanation of formula  $f_c$ . Clauses 1, 2, and 3 imply that  $(\alpha \leftrightarrow (a_2 \wedge b_2))$ ; clauses 4, 5, and 6 imply  $(\beta \leftrightarrow (a_2 \wedge \overline{b_1}))$ ; clauses 7, 8, and 9 imply  $(\gamma \leftrightarrow (\overline{a_1} \wedge \overline{b_1}))$ ; clauses 10, 11, and 12 imply  $(\delta \leftrightarrow (\overline{a_1} \wedge b_2))$ . Clause 13 (the four literal clause) implies that at least one of  $\alpha, \beta, \gamma$ , or  $\delta$  is true. Clauses 14, 15, 16, 17 imply that  $(\alpha + \gamma) \rightarrow (\overline{\beta} \wedge \overline{\delta})$  and  $(\beta + \delta) \rightarrow (\overline{\alpha} \wedge \overline{\gamma})$ . Finally, clauses 18 and 19 imply  $(a \leftrightarrow a_2)$  and  $(b \leftrightarrow b_2)$ . It can now be verified that the formula  $f_c$  implies  $(a_1 \leftrightarrow a)$  and  $(b_1 \leftrightarrow b)$ . For example, consider an assignment  $\mathbf{v}$  such that  $\mathbf{v}[a_1] = \mathbf{v}[b_1] = 0$ . Then  $f_c$  implies that  $\mathbf{v}[\beta] = \mathbf{v}[\delta] = \mathbf{v}[\alpha] = \mathbf{v}[a_2] = \mathbf{v}[b_2] = 0$  and  $\mathbf{v}[\gamma] = 1$ . We leave it to the reader to verify the other three cases. Thus, in any satisfying assignment to  $f_c$ , the new variables  $a_1, a_2, b_1, b_2, \alpha, \beta, \gamma$ , and  $\delta$  are *functionally* dependent on  $a$  and  $b$ . In other words, given an assignment to the variable  $a$  and  $b$  there is a unique way to extend this assignment so as to satisfy all the clauses in  $f_c$ . Thus  $f_c$  is a parsimonious crossover box. Even though the formula itself is a parsimonious planar crossover box, it is unsuitable for a reduction to PL-3SAT because it has one clause with four literals, namely,  $(\alpha + \beta + \gamma + \delta)$ . The second step is to obtain the formula  $f'_c$  by replacing this clause with the formula  $(\alpha + \delta + \xi) \wedge (\overline{\xi} + \beta + \gamma)$ . The planarity of the formula  $f'_c$  is demonstrated in Figure 3.1. This step also preserves numbers of satisfying assignments as demonstrated by the following claim.

CLAIM 3.3. (1) *Exactly one of  $\alpha, \beta, \gamma, \delta$  is true in any satisfying assignment to  $f_c$ .* (2)  *$\xi$  is functionally dependent on  $\alpha, \beta, \gamma$ , and  $\delta$ . Thus a satisfying assignment for  $f_c$  can be extended in a unique way to a satisfying assignment to the formula  $f'_c$ .*

*Proof.* We prove the claim for the case when  $\alpha$  is true. The other three cases are similar. As already discussed, clauses 14, 15, 16, 17 imply that  $(\alpha + \gamma) \rightarrow (\overline{\beta} \wedge \overline{\delta})$  and  $(\beta + \delta) \rightarrow (\overline{\alpha} \wedge \overline{\gamma})$ . Consider a satisfying assignment  $\mathbf{v}$  such that  $\mathbf{v}[\alpha] = 1$ . Then the above discussion implies that  $\mathbf{v}[\beta] = \mathbf{v}[\delta] = 0$ . Now, since  $(\beta \leftrightarrow (a_2 \wedge \overline{b_1}))$  and  $\mathbf{v}[\beta] = 0$ , we have  $\mathbf{v}[a_2] = 1$  and  $\mathbf{v}[b_1] = 1$ . Since  $(\gamma \leftrightarrow (\overline{a_1} \wedge \overline{b_1}))$  and  $\mathbf{v}[b_1] = 1$  it implies that  $\mathbf{v}[\gamma] = 0$ . This forces  $\mathbf{v}[\xi] = 0$ .  $\square$

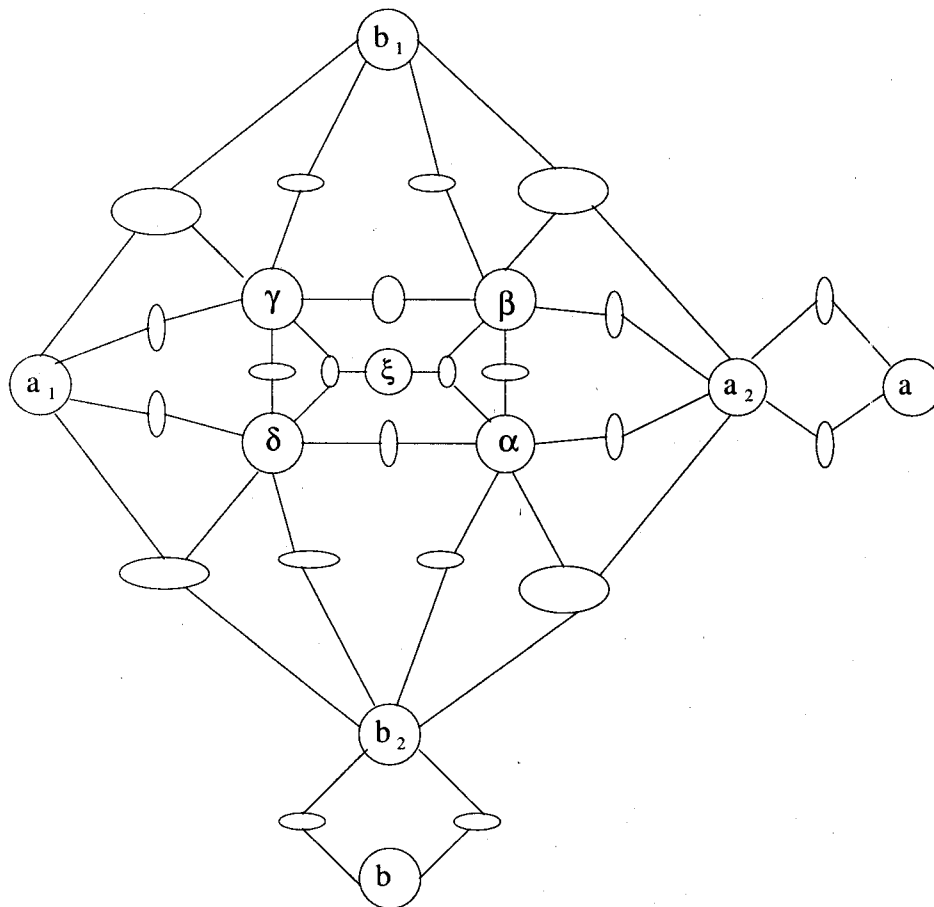


FIG. 3.1. *The parsimonious planar crossover box for 3SAT. The clauses are denoted by ellipses and the variables are denoted by circles.*

Thus, the satisfying assignments to  $f'_c$  satisfy  $f_c$ . It can now be seen that in a satisfying assignment to the variables of  $f'_c$ , the values of  $a_1$ ,  $a_2$ ,  $b_1$ ,  $b_2$ ,  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\delta$ , and  $\xi$  are all functionally dependent on  $a$  and  $b$ . Therefore,  $f'_c$  is a parsimonious crossover box.

We can now describe the reduction given in [15] from 3SAT to PL-3SAT. For any 3CNF formula, lay the formula in the plane, possibly with certain edge pairs crossing over at "crossover points." This layout is a planar graph with vertex set consisting of the variables, clauses, and crossover points. In this layout, we add a new variable node on the edge between two crossover points or between a crossover point and a clause node, as shown in Figure 3.2. The resulting graph is bipartite where the first set of nodes consists of the variable nodes and the second set consists of the clause nodes and the crossover points. Each edge is between a variable and a crossover point, or between a variable and a clause. Also, each crossover point has four distinct variables as neighbors. We now replace each crossover point with the crossover box in Figure 3.1, where  $a_2$ ,  $b_2$ ,  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\delta$ , and  $\xi$  are given distinct names in each replacement. Here  $a$ ,  $b$ ,  $a_1$ , and  $b_1$  are identified with the neighbors of the crossover point in cyclic order in the layout. The new layout is planar and the

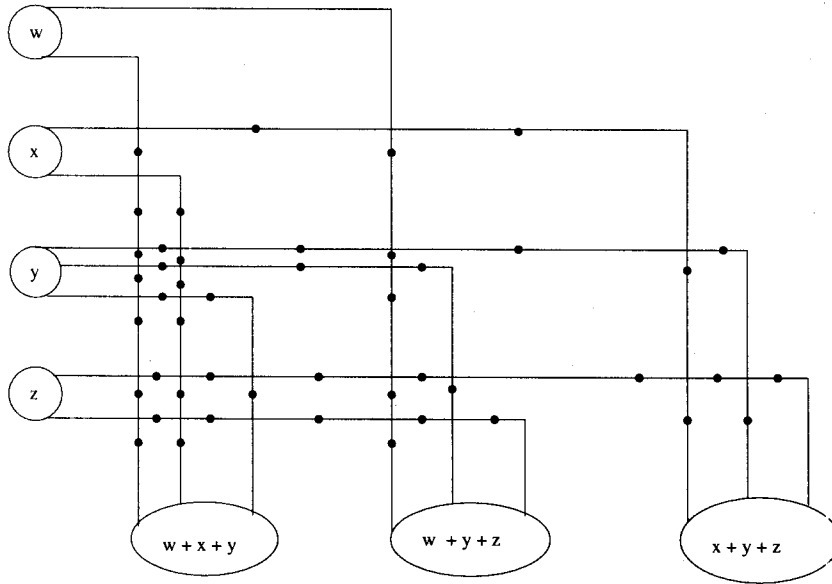


FIG. 3.2. Figure showing how new variables are introduced in the bipartite graph for 3SAT to obtain an instance of PL-3SAT. In the example, the instance of 3SAT is given by  $F = (w - x + y) \wedge (w + y + z) \wedge (x + y + z)$ . The circles denote original variables, ellipses denote the original clauses, and the black dots denote new variables added. Each crossover of edges is replaced by a crossover box shown in Figure 3.1.

new formula has the same number of solutions as the original, since in any satisfying assignment the new variables are functionally dependent on the old.  $\square$

Next, we strengthen Theorem 3.2 by showing that counting the number of satisfying assignments of a planar 3CNF formula is #P-complete, even when the input consists of a planar 3CNF formula  $F$  and an assignment  $\mathbf{v}[F]$  to the variables of  $F$  which satisfies  $F$ .

**THEOREM 3.4.** *Given an instance  $F$  of PL-3SAT and an assignment  $\mathbf{v}$  to the variables of  $F$  such that  $\mathbf{v}[F] = 1$ , the problem of counting the number of satisfying assignments of  $F$  is #P-complete.*

*Proof. Step 1.* Given an arbitrary planar 3CNF formula  $f(x_1, x_2, \dots, x_n)$ , we first construct a new formula  $f_1(x_1, x_2, \dots, x_n, x_{n+1})$ , where  $f_1 = (f \wedge x_{n+1}) \vee [\bar{x}_1 \wedge \bar{x}_2 \wedge \dots \wedge \bar{x}_n \wedge \bar{x}_{n+1}]$ .

Obviously, an assignment  $\mathbf{v}$ , such that  $\mathbf{v}[x_1] = \mathbf{v}[x_2] = \dots = \mathbf{v}[x_n] = \mathbf{v}[x_{n+1}] = 0$ , satisfies  $f_1$ . Hence  $f_1$  is always satisfiable. Also, the number of satisfying assignments of  $f_1$  is one more than the number of satisfying assignments of  $f$ . Therefore, knowing the number of satisfying assignments of  $f_1$  tells us the number of satisfying assignments of  $f$ .

*Step 2.* Convert  $f_1(x_1, x_2, \dots, x_n, x_{n+1})$  into an equivalent 3CNF formula  $f_2(x_1, x_2, \dots, x_n, x_{n+1}, x_{n+2}, \dots, x_{n+p})$ , where  $x_{n+2}, \dots, x_{n+p}$  are new variables. This is done in the standard way as follows. Obtain a parse tree of  $f_1$ . For each non-leaf node in the parse tree introduce new variables  $y_1, y_2, \dots, y_m$ , where  $y_m$  is the variable corresponding to the root of the parse tree. Each node of the parse tree corresponds to an operator applied to one or two inputs. Let the children of a non-leaf node  $i$  be  $j$  and  $k$ .



*Case 1.* If the operator at node  $i$  is an AND operator, construct a new 3CNF formula equivalent to the formula  $y_i \equiv (y_j \wedge y_k)$ .

*Case 2.* If the operator is an OR operator, construct a 3CNF formula equivalent to the formula  $y_i \equiv (y_j + y_k)$ .

*Case 3.* If the operator is a NOT operator, construct a 3CNF formula equivalent to the formula  $y_i \equiv \overline{y_j}$ .

The final 3CNF formula  $f_2$  is a conjunction of all the 3CNF formulas along with  $y_n$ . Now it is easy to see that  $f_2$  is satisfiable if and only if  $f_1$  is satisfiable and the reduction is parsimonious. (The new variables are functionally dependent on the old.)

*Step 3.* Next, lay out  $f_2$  on a plane and replace each crossover in the layout by the crossover box described in Theorem 3.2. Let  $f_3$  be the resulting planar 3CNF formula. By Theorem 3.2, the reduction from  $f_2$  to  $f_3$  is parsimonious. Thus the number of satisfying assignments of  $f_3$  is one more than the number of satisfying assignments of  $f$ . The theorem now follows.  $\square$

Next we extend this result to prove that counting the number of satisfying assignments of a 1-valid planar 3CNF formula is #P-hard.

**THEOREM 3.5.** *The problem #1-VALID PL-3SAT is #P-complete.*

*Proof.* Given an arbitrary PL-3CNF formula  $f(x_1, x_2, \dots, x_n)$  and a satisfying assignment  $\mathbf{v}$  such that  $\mathbf{v}[f] = 1$ , we construct a new formula  $f_1(x_1, x_2, \dots, x_n, y_1, \dots, y_p)$ , where  $y_1, \dots, y_p$  are new variables. The formula  $f_1$  is constructed as follows. Let  $x_{l_1}, x_{l_2}, \dots, x_{l_p}$  be the variables in  $f$  such that  $\mathbf{v}[x_{l_1}] = \mathbf{v}[x_{l_2}] = \dots = \mathbf{v}[x_{l_p}] = 0$ . Then replace  $\overline{x_{l_i}}$  by  $y_i$  and  $x_{l_i}$  by  $\overline{y_i}$ ,  $1 \leq i \leq p$ . Obviously, the formula  $f_1$  is 1-valid and the reduction is parsimonious.  $\square$

Karp and Luby [14] presented randomized fully polynomial time approximation schemes for several #P-complete problems, including #DNF. Since then, substantial research has been done in the area of approximation algorithms for various counting problems. Saluja, Subramaniam, and Thakur [22] give a logical characterization of the counting problems that have a polynomial time random approximation scheme. Our #P-hardness of #PL-3SAT and other problems immediately raise the question of approximating the optimal values of these counting problems. The parsimonious reduction from 3SAT to PL-3SAT implies that, given a deterministic polynomial time algorithm  $A$  to approximately count the number of satisfying assignments of a planar 3CNF formula, we can construct a deterministic polynomial time algorithm  $A'$  with the same performance guarantee to approximately count the number of satisfying assignments of an arbitrary 3CNF formula.

Intuitively Theorem 3.2 and the above observation mean that counting the number of satisfying assignments of a planar 3CNF formula is *as hard as* counting the number of satisfying assignments of an arbitrary 3CNF formula with respect to polynomial time reducibility. We remark that the result holds even for 1-VALID PL-3CNF formulas.

Next, we prove the #P-hardness of other basic satisfiability problems. First we prove two lemmas.

**LEMMA 3.6.** *Let  $F$  be the planar monotone formula  $(c + d + e) \wedge (c + e + f) \wedge (d + e + f)$ . Then there is a unique satisfying assignment  $\mathbf{v}$  to the variables of  $F$  such that each clause has exactly one true literal, namely,  $\mathbf{v}[c] = \mathbf{v}[d] = \mathbf{v}[f] = 0$ ; and  $\mathbf{v}[e] = 1$ .*

*Proof.* The proof is by inspection.  $\square$

**LEMMA 3.7.** *The following EX3CNF formula is planar and has exactly one satisfying assignment, namely, the assignment  $\mathbf{v}$  defined by  $\mathbf{v}[x_i] = 0 (1 \leq i \leq 9)$ :*

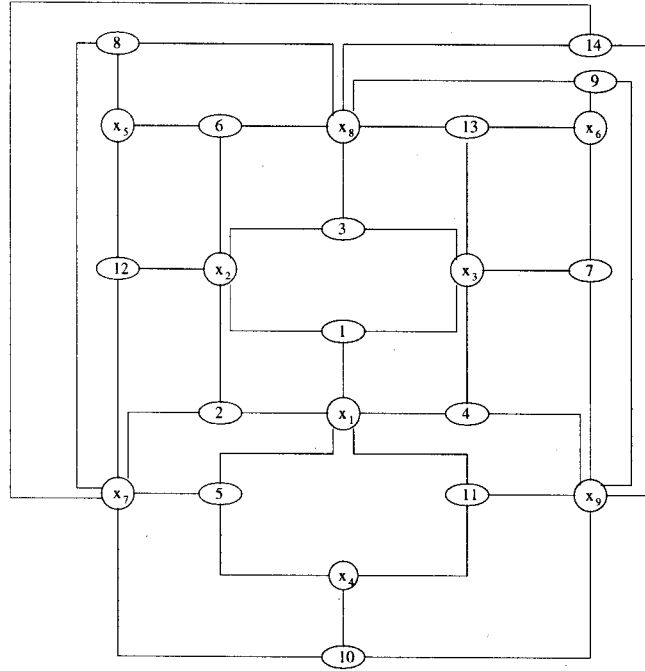


FIG. 3.3. PL-EX3SAT formula as described in Lemma 3.7. Ellipses denote the clauses in  $G$ . The clauses are numbered in the order in which they appear in Lemma 3.7.

$$G(x_1, \dots, x_9) = (\bar{x}_1 + \bar{x}_2 + \bar{x}_3) \wedge (\bar{x}_1 + x_2 + x_7) \wedge (\bar{x}_2 + x_3 + x_8) \wedge (\bar{x}_3 + x_1 + x_9) \wedge (\bar{x}_4 + x_1 + x_7) \wedge (\bar{x}_5 + x_2 + x_8) \wedge (\bar{x}_6 + x_3 + x_9) \wedge (\bar{x}_7 + x_5 + x_8) \wedge (\bar{x}_8 + x_6 + x_9) \wedge (\bar{x}_9 + x_4 + x_7) \wedge (\bar{x}_1 + \bar{x}_4 + \bar{x}_9) \wedge (\bar{x}_2 + \bar{x}_5 + \bar{x}_7) \wedge (\bar{x}_3 + \bar{x}_6 + \bar{x}_8) \wedge (\bar{x}_7 + \bar{x}_8 + \bar{x}_9).$$

*Proof.* Planarity is demonstrated by Figure 3.3. The given assignment  $\mathbf{v}$  satisfies  $G$  since each clause of  $G$  contains a negated literal.

Suppose  $G$  is true. Let  $\mathbf{u}$  be any truth assignment to the variables in  $G$ . Then,  $x_7 \rightarrow x_8$  by clauses 6, 8, and 12;  $x_8 \rightarrow x_9$  by clauses 7, 9, and 13;  $x_9 \rightarrow x_7$  by clauses 5, 10, and 11. Therefore,  $(x_7 + x_8 + x_9) \rightarrow (x_7 \wedge x_8 \wedge x_9)$ . Hence by clause 14,  $\mathbf{u}[x_7] = \mathbf{u}[x_8] = \mathbf{u}[x_9] = 0$ . Given this,  $x_1 \rightarrow x_2$  by clause 2;  $x_2 \rightarrow x_3$  by clause 3; and  $x_3 \rightarrow x_1$  by clause 4. Therefore,  $(x_1 + x_2 + x_3) \rightarrow (x_1 \wedge x_2 \wedge x_3)$ . Hence by clause 1,  $\mathbf{u}[x_1] = \mathbf{u}[x_2] = \mathbf{u}[x_3] = 0$ . But this implies that  $\mathbf{u}[x_4] = \mathbf{u}[x_5] = \mathbf{u}[x_6] = 0$  by clauses 5, 6, and 7. Hence in any satisfying assignment  $\mathbf{u}$  of  $G$ ,  $\mathbf{u}[x_i] = 0$  ( $1 \leq i \leq 9$ ).  $\square$

Next we give planarity-preserving parsimonious reductions from 3SAT to the basic SAT problems listed in Table 1.1. Without loss of generality, we assume that the given instance of the CNF formula does not have any single literal clause.

**THEOREM 3.8.** *There exist planarity-preserving parsimonious reductions from 3SAT to each of the following problems: EX3SAT, 1-3SAT, 1-EX3SAT, 1-EX3MONOSAT and X3C.*

*Proof.* 3SAT  $\rightarrow$  EX3SAT: Let  $f$  be a 3CNF formula with clauses  $c_j$  ( $1 \leq j \leq k$ ). For  $1 \leq j \leq k$ , let  $c'_j$  be  $c_j$ , if  $c_j$  is a three-literal clause. If  $c_j = (l_{j1} + l_{j2})$ , let  $c'_j$  be  $(l_{j1} + l_{j2} + x'_j) \wedge G(x'_1, \dots, x'_9)$ .

$G$  is defined as in Lemma 3.7 and  $x_i^j$  ( $1 \leq i \leq 9$ ) are distinct new variables. Let

$$g = \bigwedge_{j=1}^k c'_j.$$

Then by Lemma 3.7, Figure 3.3, and direct inspection of the definitions of the formulas  $c'_j$ , the reduction mapping  $f$  into  $g$  is seen to be a planarity-preserving and parsimonious reduction of the problem 3SAT to the problem EX3SAT.

3SAT  $\rightarrow$  1-EX3SAT<sup>2</sup>: Let  $f$  be a 3CNF formula with clauses  $c_j$  ( $1 \leq j \leq m$ ).

(1) For each three-literal clause  $c_j = (z_p + z_q + z_r)$  of  $f$ , let  $c'_j = (z_p + u^j + v^j) \wedge (\bar{z}_q + u^j + w^j) \wedge (v^j + w^j + t^j) \wedge (\bar{z}_r + v^j + x^j)$ , where  $u^j, v^j, w^j, t^j$ , and  $x^j$  are distinct new variables local to  $c'_j$ .

(2) For each two-literal clause  $c_j = (z_p + z_q)$  of  $f$ , let  $c'_j = (z_p + u^j + v^j) \wedge (\bar{z}_q + u^j + w^j) \wedge (v^j + w^j + t^j) \wedge (\bar{a}^j + v^j + x^j) \wedge (a^j + d^j + e^j) \wedge (a^j + e^j + f^j) \wedge (d^j + e^j + f^j)$ , where  $u^j, v^j, w^j, t^j, x^j, a^j, d^j$ , and  $e^j$  are all new variables local to  $c'_k$ . Let

$$f' = \bigwedge_{j=1}^m c'_j.$$

To see the planarity of this reduction, see Figure 3.4. We claim that  $f'$  is exactly-one satisfiable *if and only if* the original formula  $f$  is satisfiable. Moreover, the reduction is planarity-preserving and parsimonious. To prove that the reduction is parsimonious, it suffices to show the following two claims.

CLAIM 3.9. *No assignment of truth values to the variables of clause  $c_j$  ( $1 \leq j \leq m$ ) of  $f$  which does not satisfy  $c_j$  can be extended to an assignment of truth values to the variables of the formula  $c'_j$  that exactly-one satisfies  $c'_j$ .*

*Proof.* Let  $c_j = (z_p + z_q + z_r)$  and  $\mathbf{u}$  be an assignment to the variables such that  $\mathbf{u}[z_p] = \mathbf{u}[z_q] = \mathbf{u}[z_r] = 0$ . Let  $\mathbf{v}$  be an exactly-one satisfying assignment to the variables of  $c'_j$  such that  $\mathbf{v}[z_p] = \mathbf{v}[z_q] = \mathbf{v}[z_r] = 0$  (i.e.,  $\mathbf{v}$  is an extension of  $\mathbf{u}$ ). Then, the clauses  $(\bar{z}_q + u^j + w^j)$  and  $(\bar{z}_r + v^j + x^j)$  imply that  $\mathbf{v}[u^j] = \mathbf{v}[w^j] = \mathbf{v}[v^j] = \mathbf{v}[x^j] = 0$ . It follows that  $\mathbf{v}$  does not exactly-one satisfy the clause  $(z_p + u^j + w^j)$  of  $c'_j$ .

Let  $c_j = (z_p + z_q)$  and  $\mathbf{u}$  be an assignment to the variables such that  $\mathbf{u}[z_p] = \mathbf{u}[z_q] = 0$ . Let  $\mathbf{v}$  be an exactly-one satisfying assignment to the variables of  $c'_j$  such that  $\mathbf{v}[z_p] = \mathbf{v}[z_q] = 0$ . Then, the clauses  $(z_p + u^j + v^j)$  and  $(\bar{z}_q + u^j + w^j)$  imply that  $\mathbf{v}[u^j] = \mathbf{v}[w^j] = 0$  and  $\mathbf{v}[v^j] = 1$ . But given this, the clause  $(\bar{a}^j + v^j + x^j)$  implies that  $\mathbf{v}[a^j] = 1$ . Lemma 3.6 now implies that  $\mathbf{v}$  does not exactly-one satisfy  $c'_j$ .  $\square$

CLAIM 3.10. *For each satisfying assignment to the variables of the clause  $c_j$  ( $1 \leq j \leq m$ ) of  $f$ , there is exactly one way the assignment can be extended to the variables of the formula  $c'_j$  so as to exactly-one satisfy  $c'_j$ .*

*Proof.* When  $c_j = (z_p + z_q + z_r)$ , we need to verify that the only exactly-one satisfying assignments  $c'_j$  are the following:

1.  $z_p = 1, z_q = 0, z_r = 0, u^j = 0, v^j = 0, w^j = 0, t^j = 1, x^j = 0$ ;
2.  $z_p = 0, z_q = 1, z_r = 0, u^j = 1, v^j = 0, w^j = 0, t^j = 1, x^j = 0$ ;
3.  $z_p = 0, z_q = 0, z_r = 1, u^j = 0, v^j = 1, w^j = 0, t^j = 0, x^j = 0$ ;

<sup>2</sup>Although they claim to have a parsimonious reduction, the reduction actually given in [3] is *not* parsimonious.

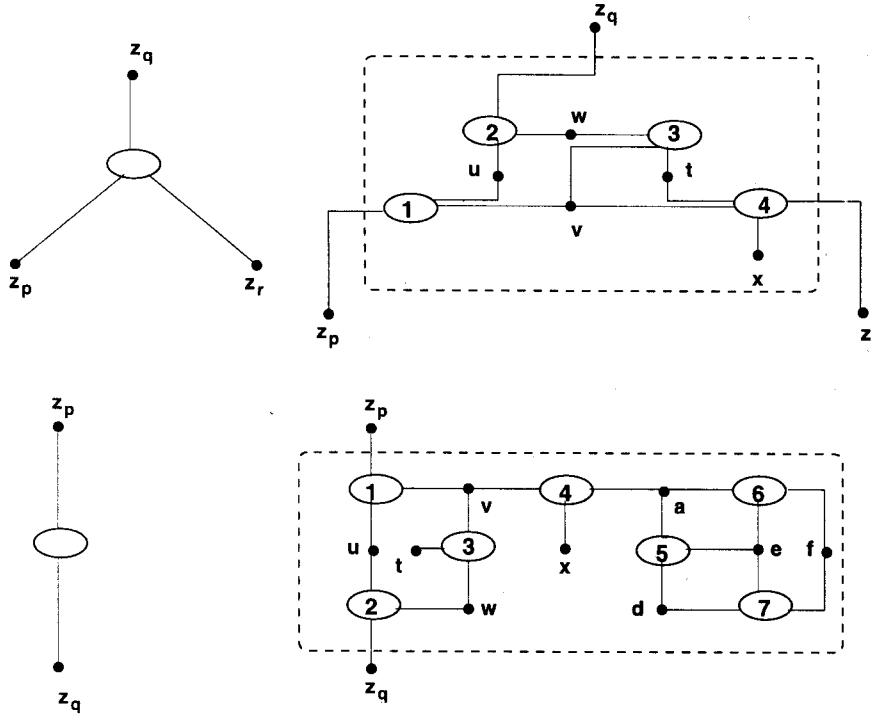


FIG. 3.4. Figure illustrating the reduction from 3SAT to 1-EX3SAT. (a) shows how to transform a three-literal clause. (b) shows how to transform a two-literal clause. The clauses are numbered in the order in which they appear in the reduction outlined in the proof of Theorem 3.8. Note that the reduction is a local replacement-type reduction and hence preserves planarity.

- 4.  $z_p = 1, z_q = 1, z_r = 0, u^j = 0, v^j = 0, w^j = 1, t^j = 0, x^j = 0;$
- 5.  $z_p = 1, z_q = 0, z_r = 1, u^j = 0, v^j = 0, w^j = 0, t^j = 1, x^j = 1;$
- 6.  $z_p = 0, z_q = 1, z_r = 1, u^j = 1, v^j = 0, w^j = 0, t^j = 1, x^j = 1;$

and

- 7.  $z_p = 1, z_q = 1, z_r = 1, u^j = 0, v^j = 0, w^j = 1, t^j = 0, x^j = 1.$

When  $c_j = (z_p + z_q)$ , we need to verify that the only exactly-one assignments of  $c_j^j$  are the following:

- 1.  $z_p = 1, z_q = 0, u^j = 0, v^j = 0, w^j = 0, t^j = 1, x^j = 0, c^j = 0, d^j = 0, e^j = 1, f^j = 0;$
- 2.  $z_p = 0, z_q = 1, u^j = 1, v^j = 0, w^j = 0, t^j = 1, x^j = 0, c^j = 0, d^j = 0, e^j = 1, f^j = 0;$  and
- 3.  $z_p = 1, z_q = 1, u^j = 0, v^j = 0, w^j = 1, t^j = 0, x^j = 0, c^j = 0, d^j = 0, e^j = 1, f^j = 0. \quad \square$

1-EX3SAT  $\rightarrow$  1-EX3MONOSAT: Let  $f$  be an instance of 1-EX3SAT. Let

$$f = \bigwedge_{j=1}^m c_j.$$

For each  $c_j$ , construct  $c_j^j$  as follows. Replace each negated literal of the form  $\overline{z_p}$  appearing in the clause  $c_j$  by a distinct new variable  $y_p^j$  in  $c_j$ , then add the clauses  $(z_p + y_p^j + a_p^j) \wedge (a_p^j + d_p^j + e_p^j) \wedge (a_p^j + f_p^j + e_p^j) \wedge (d_p^j + f_p^j + e_p^j)$ . Note that for each negated

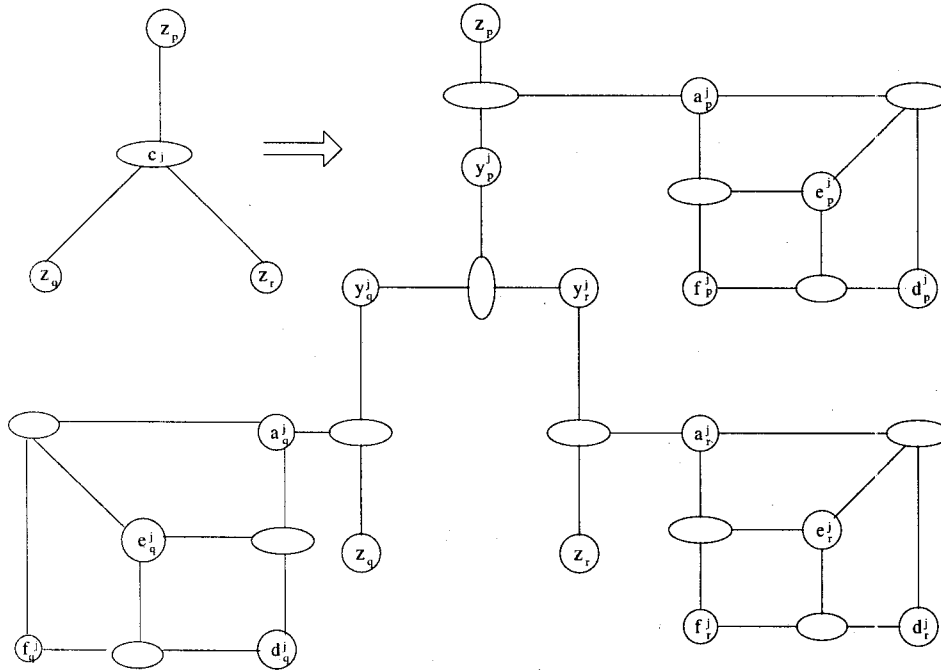


FIG. 3.5. Figure illustrating the reduction from 1-EX3SAT to 1-EX3MONOSAT. The figure illustrates the construction for a three-literal clause  $c_j$  which contains all negated literals

literal, we introduce new copies of the auxiliary variables  $a_p^j, \dots, f_p^j$ . See Figure 3.5 for an example.

Let

$$f' = \bigwedge_{j=1}^m c'_j.$$

Then  $f'$  is an instance of 1-EX3MONOSAT obtained from  $f$ . The result follows from Lemma 3.6 and the fact that for all variables  $x$  and  $y$ ,  $(x + y)$  is exactly-one satisfiable if and only if  $x = \bar{y}$ .

1-EX3MONOSAT  $\rightarrow$  X3C: Although Dyer and Frieze [3] do not observe this, the reduction given in their paper [3] from 1-EX3SAT to X3C is actually parsimonious. The reduction presented here is essentially the same as that given in [3], except that we start from an instance of 1-EX3MONOSAT. Thus in our reduction, we do not have to take care of negated literals. We now describe the reduction. Each variable is represented by a cycle of 3 element sets. If the variable occurs  $r$  times in the 1-EX3SAT instance, then there are  $2r$  sets, with each successive pair of sets sharing an element. This cycle is augmented with  $r$  additional sets and  $2r$  elements by adding a 3-set to one of the external elements in each pair. The 3 elements now corresponding to an appearance of  $v_j$  will be called a connector. The variable  $v_j$  is set to true if and only if all three connector elements are covered by the cycle when  $v_j$  appears in the corresponding clause. Figure 3.6 illustrates the variable component. Next, consider each clause  $c_i$ . Each  $c_i$  is represented by a configuration shown in Figure 3.7. This has 12 elements and 9 sets. Of the 12 elements, 3 are internal and the rest are grouped in groups of 3. Each group of 3 elements is called a terminal of  $c_i$ . Finally, we connect a

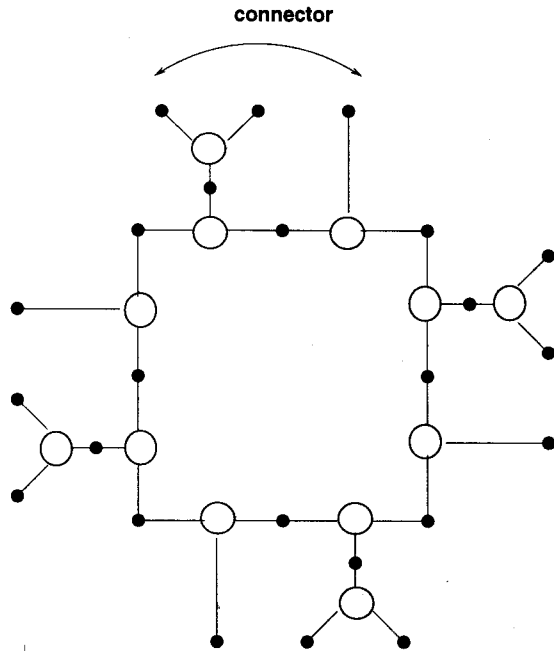


FIG. 3.6. Variable configuration for reduction to X3C. The black dots represent element nodes, while the ellipses denote the triples.

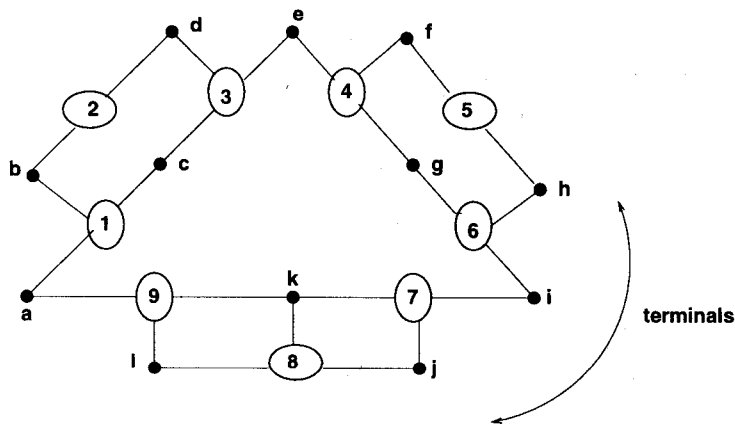


FIG. 3.7. Figure illustrating the clause configuration. The sets  $\{l, a, b\}$ ,  $\{d, c, f\}$ ,  $\{j, i, h\}$  represent the three terminals. The vertices labeled  $c, g, k$  are the internal elements.

clause component to the variable component as follows. For each  $v_j \in c_i$  we identify three distinct connector elements with one of the terminals in  $c_i$ . The construction is depicted in Figure 3.8. Let  $G$  denote the graph obtained as a result of the construction. Planarity of  $G$  follows by the fact that each component replacing a variable and a clause is planar and the components are joined in a planarity-preserving way. We first prove that there is an exact cover of  $c_i$  configuration if and only if exactly one terminal is covered externally, when we restrict the covering such that either none or all 3 of the elements in each terminal are covered externally. But the configuration has the

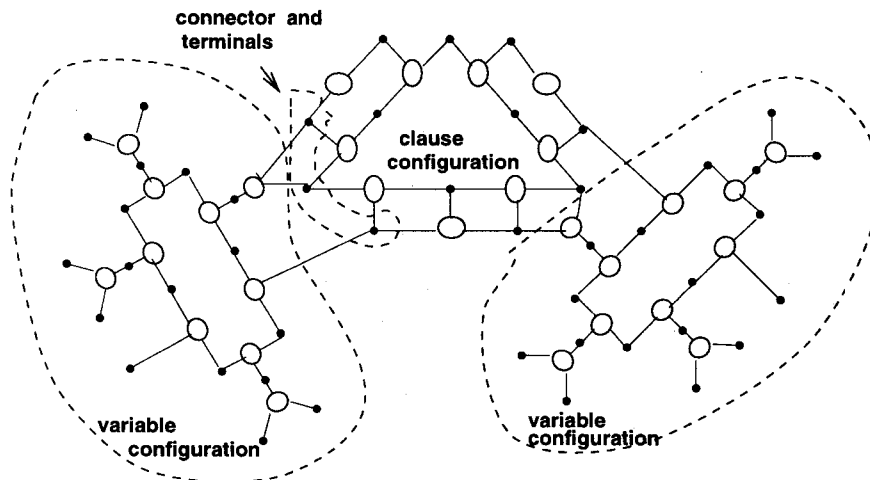


FIG. 3.8. Figure illustrating the way the clause and variable configurations are attached.

property that each of the 3 internal elements appear in 3 of the 9 sets and no 2 appear in the same set. It follows that if this configuration forms part of an exact cover by 3-sets, then exactly 3 of the sets must be used, hence 9 of the 12 elements will be covered internally. Moreover, this can only be done so that exactly 1 of the terminals will be left uncovered. This uncovered terminal is covered by sets in the variable configuration and amounts to setting the literal true. It can then be argued that the exact cover by 3-sets has a solution if and only if the corresponding 1-EX3MONOSAT instance is satisfiable. It is easily verified that the reduction is parsimonious. This is because the clause configuration forces precisely 1 literal to be set to true and the other 2 literals to be false. Moreover, for each satisfying assignment of 1-EX3MONOSAT, there is exactly 1 way the sets can be chosen so as to have an exact cover. Hence, the reduction is parsimonious.  $\square$

**COROLLARY 3.11.** *The problems PL-EX3SAT, PL-1-EX3SAT, and PL-1-EX3MONOSAT are NP-complete. The problems #PL-EX3SAT, #PL-1-3SAT, #PL-1-EX3SAT, #PL-1-EX3MONOSAT, and #PL-X3C are #P-complete.*

#### 4. Planar graph problems.

**4.1. Overview of our proofs.** In this section we give parsimonious/weakly parsimonious and planarity-preserving reductions from PL-1-EX3MONOSAT to various graph problems. The problems considered here are MINIMUM VERTEX COVER, MINIMUM DOMINATING SET, CLIQUE COVER, FEEDBACK VERTEX SET, PARTITION INTO CLAWS, PARTITION IN TRIANGLES, and BIPARTITE DOMINATING SET. Previously, reductions showing that these problems were NP-hard frequently did not preserve the number of solutions. Central to the proofs is the reduction (called RED1) from 1-EX3MONOSAT to 3SAT with the property that every formula is mapped to a formula in which all satisfying assignments satisfy exactly one literal in each three-literal clause. This in turn enables us to obtain (weakly) parsimonious reductions from EX1-3MONOSAT to the problems considered here.

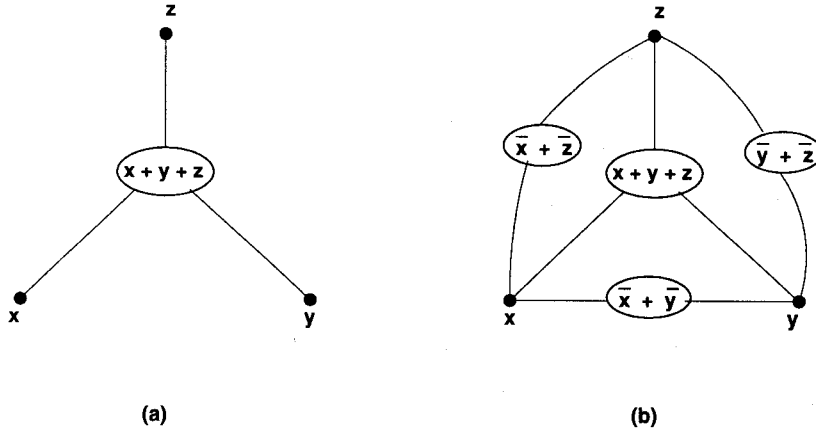


FIG. 4.1. Figure illustrating the reduction RED1 discussed in section 4.2. Observe that the reduction is planarity-preserving.

4.2. **Reduction RED1.** Let RED1 be a mapping from an instance

$$f = \bigwedge_{j=1}^m c_j$$

of 1-EX3MONOSAT to an instance

$$f' = \bigwedge_{j=1}^m c'_j$$

of 3SAT, where for  $c_j = (x + y + z)$ ,  $c'_j = (x + y + z) \wedge (\bar{x} + \bar{y}) \wedge (\bar{x} + \bar{z}) \wedge (\bar{z} + \bar{y})$ .

LEMMA 4.1. *The formula  $f'$  has the following properties.*

- (1) *The satisfying assignments of  $f'$  are exactly the exactly-one satisfying assignments of  $f$ .*
- (2) *In any satisfying assignment of  $f'$ , all but one clause in  $c'_j$  is exactly-one satisfiable.*
- (3) *Each variable in the formula  $f'$  occurs at least twice negated and at least once unnegated.*
- (4) *RED1 is planarity-preserving. (See Figure 4.1.)*
- (5) *RED1 is parsimonious.*

*Proof.* The proof is by inspection. □

We call each of the  $c'_j$  a *clause group*. Observe that each  $c'_j$  has four clauses; one is a three-literal clause and the others are two-literal clauses.

4.3. **Weakly parsimonious reductions and basic graph problems.**

THEOREM 4.2. *There exists a planarity-preserving and weakly parsimonious reduction from 1-EX3MONOSAT to each of the following problems: (1) MINIMUM VERTEX COVER, (2) MINIMUM DOMINATING SET, (3) MINIMUM FEEDBACK VERTEX SET, and (4) SUBGRAPH ISOMORPHISM.*

*Proof.* (1) MINIMUM VERTEX COVER: The reduction is from 1-EX3MONOSAT and is similar to the one given in [5] for proving NP-hardness of MINIMUM VERTEX COVER. Let  $f$  be a 1-EX3MONOSAT formula. Apply RED1 to  $f$  to obtain  $f'$ . Next,



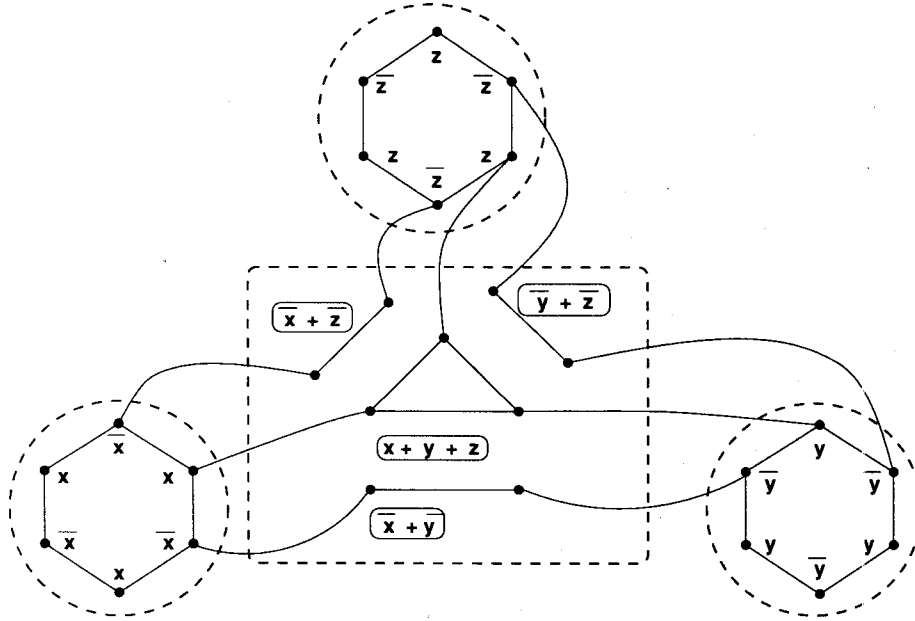


FIG. 4.2. Figure illustrating the reduction from 1-EX3MONOSAT to VERTEX COVER and the transformation of a clause group  $(x + y + z) \wedge (\bar{x} + \bar{y}) \wedge (\bar{x} + \bar{z}) \wedge (\bar{z} + \bar{y})$ . The dotted enclosures depict how to locally transform the clauses as well as the variables so as to preserve planarity of the resulting graph.

starting from  $f'$ , construct an instance  $G(V, E)$  of the vertex cover problem as shown in Figure 4.2 as follows.

1. Consider a clause group  $c'_j = (x + y + z) \wedge (\bar{x} + \bar{y}) \wedge (\bar{x} + \bar{z}) \wedge (\bar{z} + \bar{y})$ . Corresponding to the clause  $(x + y + z)$ , construct a triangle with vertices  $\{x, y, z\}$  and edges  $\{(x, y), (x, z), (y, z)\}$ . Corresponding to a clause of the form  $(\bar{x} + \bar{y})$  add the edge  $\{(x, y)\}$ . Call this the clause graph.

2. For each variable  $x$  that appears  $i$  times we construct a simple cycle with  $2i$  vertices. Let the odd-numbered variables represent the negated occurrences and the even-numbered variables represent the unnegated occurrences. Call this the variable graph.

3. Join the vertices of the clause graph to the vertices of the variable graph as follows. Consider a clause group  $c'_j$ . Corresponding to a clause, join the triangle vertices  $x, y$ , and  $z$  to the corresponding unnegated occurrences of  $x, y$ , and  $z$  in the cycles. Corresponding to a clause of the form  $(\bar{x} + \bar{y})$ , join the two vertices  $x, y$  to the negated occurrence of the variables  $x$  and  $y$ , respectively. Repeat the procedure for each clause group.

Now set  $K = 1/2 \sum C_i + 2m + 3m$ , where  $C_i$  is the length of the cycle of the variable  $i$  and  $m$  is the number of clause groups in  $f'$ . The reduction is illustrated in Figure 4.2.

CLAIM 4.3. (1) The formula  $f'$  is satisfiable if and only if the graph  $G$  has a vertex cover of size  $K$ .

(2) The reduction is planarity-preserving and weakly parsimonious.

Proof. Part 1. Observe that for any vertex cover, one needs to pick at least half of the nodes from each cycle, two of the three nodes from each triangle, and one from the

simple edge for each two-literal clause. (Recall that there are  $3m$  two-literal clauses in  $f'$ .) The sum is exactly  $K$ . Given this observation, the proof is similar to the one given in [5].

*Part 2.* It can be easily verified by observing in Figure 4.2 that the reduction is planarity-preserving. To see that for each distinct satisfying assignment of  $f'$ , there are  $2^m$  distinct vertex covers of size  $K$ , note that, by Lemma 4.1, for each satisfying assignment, all but one clause in each clause group has only one true literal. This forces the choice of vertices from the clause graph for all clauses having only one true literal. For each satisfying assignment and for each clause group there is one clause in the clause group in which both the literals are true. For each such clause any of the two vertices can be included in the vertex cover. Since there are  $m$  clause groups, we have  $m$  such clauses and hence we have  $2^m$  different vertex covers for each satisfying assignment.  $\square$

Note that our reduction shows that counting the number of vertex covers of size  $\leq k$  is #P-hard even if there are no vertex covers of size strictly less than  $k$ . For the next two results, we use such an instance of #-vertex cover for our reductions.

(2) MINIMUM DOMINATING SET: The reduction is from the MINIMUM VERTEX COVER problem. The reduction in [13] from VERTEX COVER to DOMINATING SET can easily be modified to get a parsimonious reduction. Let  $G_1 = (V_1, E_1)$ ,  $V_1 = \{v_1, \dots, v_n\}$  be an instance of the MINIMUM VERTEX COVER problem. We construct an instance  $G_2 = (V_2, E_2)$  of the MINIMUM DOMINATING SET problem as follows. There is one vertex in  $V_2$  corresponding to every vertex in  $V_1$ . For each edge in  $G_1$  we also introduce two additional vertices and join them to the two endpoints of the original edge. Formally,  $V_2 = U_1 \cup U_2$  and  $E_2 = A_1 \cup A_2$ , where

$$U_1 = \{u_i \mid v_i \in V_1\},$$

$$U_2 = \{x_1^{ij}, x_2^{ij} \mid (v_i, v_j) \in E_1\},$$

$$A_1 = \{(u_i, u_j) \mid (v_i, v_j) \in E_1\},$$

$$A_2 = \{(u_i, x_1^{ij}), (x_1^{ij}, u_j), (u_i, x_2^{ij}), (x_2^{ij}, u_j) \mid (v_i, v_j) \in E_1\}.$$

CLAIM 4.4.  $G_1$  has a minimum vertex cover of size  $k$  if and only if  $G_2$  has a minimum dominating set of size  $k$ . Furthermore, the reduction is planarity-preserving and parsimonious.

*Proof.* It is easy to see that the reduction is planarity-preserving. Consider a minimum vertex cover  $VC = \{v_{i_1}, v_{i_2}, \dots, v_{i_k}\}$  of  $G_1$ . Corresponding to  $VC$  we claim that there is exactly one dominating set in the graph  $G_2$ , namely, the vertex set  $DS = \{u_{i_1}, u_{i_2}, \dots, u_{i_k}\}$ . First note that for each edge in the original graph  $G$  we have four new edges and two new vertices in  $G_2$ . Consider a pair of nodes of the form  $x_1^{ij}, x_2^{ij}$  connected to the nodes  $u_i$  and  $u_j$ . It is clear that the only way to dominate both  $x_1^{ij}, x_2^{ij}$  by using only one node is to include one of  $u_i$  or  $u_j$  in the dominating set. We need to consider two cases. First consider the case when exactly one of  $v_i$  or  $v_j$  is in  $VC$ . Then it is clear that there is exactly one dominating set  $DS$  in  $G_2$  corresponding to  $VC$ . When both  $v_i$  and  $v_j$  are in  $VC$ , the minimality of  $VC$  implies that at least one edge incident on  $v_i$  and at least one edge incident on  $v_j$  are covered solely by  $v_i$  and  $v_j$ , respectively. This implies that both  $u_i$  and  $u_j$  have to be in

any feasible dominating set of  $G_2$ . Thus we have exactly one dominating set  $DS$  in  $G_2$  corresponding to the vertex cover  $VC$  in  $G_1$ . Conversely, consider a minimum dominating set  $DS = \{u_{i_1}, u_{i_2}, \dots, u_{i_k}\}$  of size  $k$  in  $G_2$ . Consider an edge  $(v_i, v_j)$  in  $G$ . If  $u_i$  and  $u_j$  are not in  $DS$ , then by construction of  $G_2$ , both  $x_1^{ij}$  and  $x_2^{ij}$  are in  $DS$ . But we could then construct a new dominating set  $DS'$  of  $G_2$ , where

$$DS' = DS - \{x_1^{ij}, x_2^{ij}\} \cup \{u_i\}.$$

Clearly  $|DS'| < |DS|$ , which is a contradiction to the assumption that  $DS$  is a minimum dominating set. Thus,  $DS$  does not contain any vertex from the set  $U_2$ . We now claim that  $VC = \{v_{i_1}, v_{i_2}, \dots, v_{i_k}\}$  is a vertex cover of  $G$ . The claim follows by observing that corresponding to each edge  $(v_i, v_j) \in E_1$ , at least one of the vertices  $u_i, u_j, x_1^{ij}, x_2^{ij}$  are in the set  $DS$ . We have already argued that  $x_1^{ij}, x_2^{ij} \notin DS$ . Thus one of  $u_i, u_j$  is in  $DS$ . The corresponding vertex in  $VC$  is seen to cover the edge  $(v_i, v_j)$ .  $\square$

(3) FEEDBACK VERTEX SET: The reduction is from the MINIMUM VERTEX COVER problem. Starting from an instance  $G_1(V_1, E_1)$  of the MINIMUM VERTEX COVER problem, we construct the graph  $G_2$  that is identical to the one given for the MINIMUM DOMINATING SET problem. By arguments similar to those given in the proof of the MINIMUM DOMINATING SET problem, it is easy to see that the  $G_1$  has a minimum vertex set of size  $K$  if and only if  $G_2$  has a feedback vertex set of size  $K$  and the reduction is parsimonious.

(4) SUBGRAPH ISOMORPHISM: This follows directly by taking the graph  $H$  to be a simple cycle on  $n$  nodes, and the weakly parsimonious reduction from 3SAT to the HAMILTONIAN CIRCUIT problem given in Provan [20].  $\square$

COROLLARY 4.5. *The problems #PL-MINIMUM VERTEX COVER, #PL-MINIMUM DOMINATING SET, #PL-MINIMUM FEEDBACK VERTEX SET, and #PL-SUBGRAPH ISOMORPHISM are #P-complete.*

**4.4. Parsimonious reductions and other graph problems.** In this section we briefly discuss why the reductions studied by [6, 15] and [3, 4] from X3C to various other graph problems are parsimonious and planarity-preserving.

THEOREM 4.6. *There exist planarity-preserving and parsimonious reductions from X3C to each of the problems (1) MINIMUM CLIQUE COVER, (2) PARTITION INTO CLAWS, (3) BIPARTITE DOMINATING SET, (4) PARTITION INTO TRIANGLES, and (5) MINIMUM HITTING SET.*

*Proof.* (1) MINIMUM CLIQUE COVER: The reduction is the same as given in [3, 5]. Given an instance  $I(X, C)$  of X3C such that  $|X| = 3p$  and  $|C| = m$ , we construct an instance  $G$  of the MINIMUM CLIQUE COVER problem such that  $G$  has a clique cover with cliques of size 3 if and only if  $I$  has a solution. The reduction consists of replacing each triple in the instance of  $I$  by a triangle and by replacing an edge from a triple to an element by a set of triangles. The reduction is illustrated in Figure 4.3. Formally, for each element, we have a vertex in  $G$ . Corresponding to each triple  $t_i = \{x_i, y_i, z_i\}$  and the associated edges  $(t_i, x_i), (t_i, y_i), (t_i, z_i)$ , we create the subgraph as shown in Figure 4.3. The graph  $G$  obtained by carrying out the above reduction for each triple has  $3p + 9m$  vertices and  $18m$  edges. The reduction is planarity-preserving, as each component is planar and they are joined in a planarity-preserving way. We claim that  $I$  has a solution if and only if  $G$  has a clique cover of size  $(p + 3m)$ . In particular, as shown in [3, 5], if  $t_1, \dots, t_p$  is the set of triples in an exact cover, then

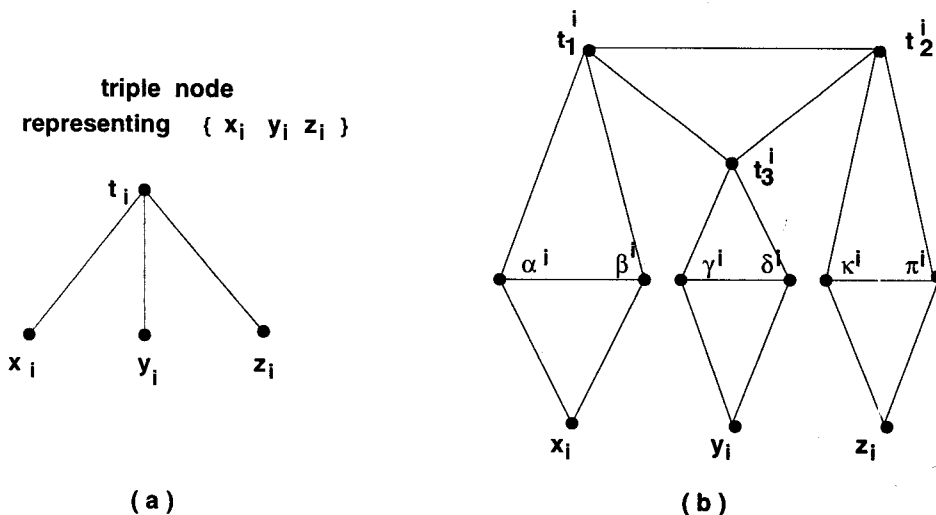


FIG. 4.3. Figure illustrating the reduction from X3C to CLIQUE COVER.

the corresponding clique cover is constructed by taking

$$\{\alpha^i, \beta^i, x_i\}, \{\gamma^i, \delta^i, y_i\}, \{\kappa^i, \pi^i, z_i\}, \{t_1^i, t_2^i, t_3^i\}$$

whenever  $t_i = \{x_i, y_i, z_i\}$  is in the exact cover and by taking the cliques

$$\{\alpha^i, \beta^i, t_1^i\}, \{\gamma^i, \delta^i, t_2^i\}, \{\kappa^i, \pi^i, t_3^i\}$$

when the corresponding triple  $t_i$  is not in the exact cover. Conversely, since  $G$  has  $3p + 9m$  vertices, if  $G$  has a clique cover of size  $p + 3m$  it implies that each clique consists of exactly three vertices. (Recall that we do not have cliques of size 4 in  $G$ .) The corresponding exact cover is given by choosing those  $t_i \in C$  such that the triangles  $t_1^i, t_2^i, t_3^i$  are in the clique cover. Finally, we prove that the reduction is parsimonious. First note that if the triple triangle  $\{t_1^i, t_2^i, t_3^i\}$  is not chosen then we have to choose the triangles  $\{\alpha^i, \beta^i, t_1^i\}, \{\gamma^i, \delta^i, t_2^i\}, \{\kappa^i, \pi^i, t_3^i\}$  so as to cover the triple vertices. Second, once the triangle  $\{t_1^i, t_2^i, t_3^i\}$  is chosen there is exactly one way for the auxiliary nodes (and thus the element nodes) to be covered, namely, the lower triangle corresponding to the covering triple, i.e., choosing the triangles

$$\{\alpha^i, \beta^i, x_i\}, \{\gamma^i, \delta^i, y_i\}, \{\kappa^i, \pi^i, z_i\}.$$

These observations immediately imply that the reduction is parsimonious.

(2) PARTITION INTO CLAWS: The reduction is from X3C and is the same as the one given in [3]. The reduction consists of the following steps.

1. Construct the bipartite graph  $G(C \cup X, E)$  corresponding to the given instance  $I(X, C)$  of X3C.
2. As in [3], we assume that each element vertex appears in either two or three sets; i.e., the element vertices have a degree 2 or 3.
3. For each element of degree 3, we add an extra edge and for each element of degree 2, we add two extra edges. This is shown in Figure 4.4. Let  $G_1$  denote the resulting graph.

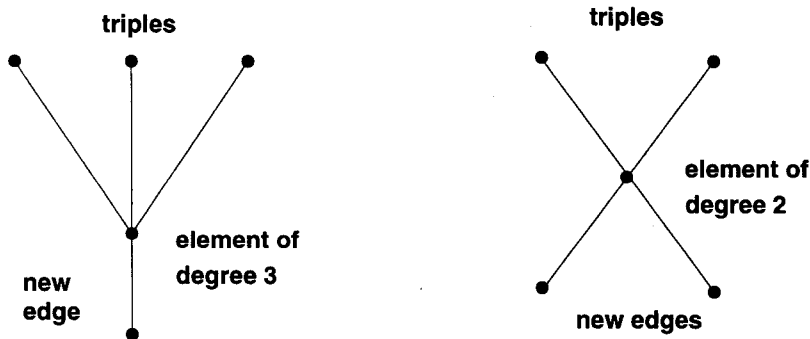


FIG. 4.4. Figure illustrating the reduction from X3C to PARTITION INTO CLAWS.

Clearly the reduction is planarity-preserving. We now recall the proof in [3] to show that the edges of  $G_1$  can be partitioned into a disjoint set of claws. Note that each element vertex is adjacent to either 1 or 2 vertices of degree 1, so it follows that each element node must be the center of at least one claw. But each such element node has degree 4 and hence can be the center of exactly one claw. After removing the claws from  $G_1$  the resulting graph  $G_2$  has the property that all element nodes have degree 1. This implies that the only way to partition  $G_2$  into claws is for each triple to have a degree of 0 or 3. Thus the triples with degree 3 induce a solution for the X3C in an obvious way. Conversely, given a solution for  $I$ , the above argument can be reversed to yield a partition of the edges in  $G_1$  into claws. The following observations immediately imply that the reduction is parsimonious.

1. There is a unique way to pick the claws in  $G_1$  with the element nodes as centers.
2. In  $G_2$ , each triple vertex has degree 3 or 0 and each element node has degree 1 in  $G_2$ .

(3) BIPARTITE DOMINATING SET: Reduction from X3C. The construction is similar to that in [3]. Let  $I(X, C)$  be an instance of PL-X3C with each element occurring in at most three triples. We first construct the bipartite graph  $G$  associated with  $I$ . Next, we attach a 2-claw ( $K_{1,2}$ ) to each triple vertex in  $G$  as shown in Figure 4.5. (In [3], they add a path of length 2.) Let  $G'$  denote the graph obtained as a result of the transformation. The construction is depicted in Figure 4.5. Since  $G$  is bipartite and we added a claw as shown in Figure 4.5, it follows that  $G'$  is also bipartite. Also note that the reduction is planarity-preserving and thus  $G'$  is planar. Let the number of triples be  $m$  and the number of elements be  $3p$ . Then we set  $k = p + m$ . Now by arguments similar to those in [3], it is easy to see that  $G'$  has a dominating set of size  $k$  if and only if  $I$  has an exact cover of size  $p$ . We prove the reduction is parsimonious. Consider a solution  $S(I)$  for  $I$ . Since each triple in the solution covers three distinct element nodes, these element nodes cannot be used to dominate the vertices in  $G'$  without increasing the cardinality of the solution for  $G'$ . This means that for each of the  $p$  triples chosen in the solution  $S(I)$ , we have exactly one node in  $G'$  that can be used in the dominating set so as to dominate all the element nodes. Moreover, due to the constraints on the size of the dominating set in  $G'$ , it follows that we can select exactly one vertex per claw (the vertex with degree 3 and marked  $a$ ) in the dominating set. These observations imply that the reduction is parsimonious.

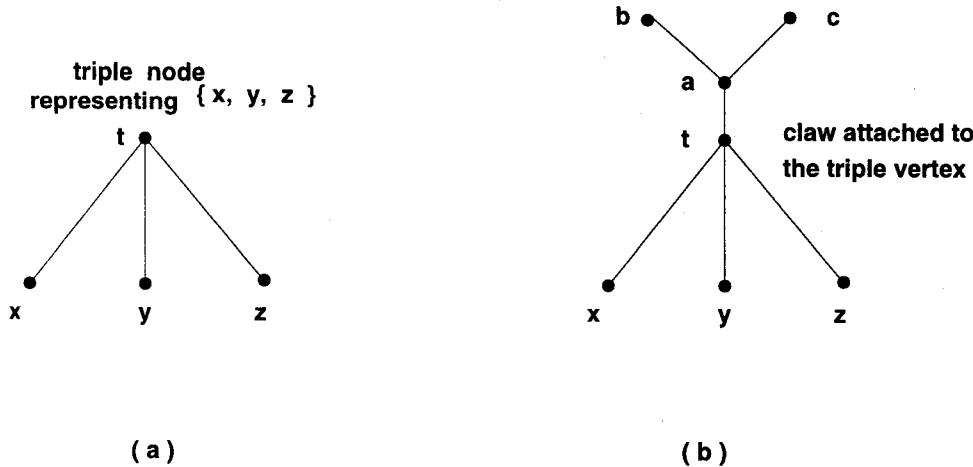


FIG. 4.5. Figure illustrating the parsimonious reduction from X3C to BIPARTITE DOMINATING SET. It is easy to see that the reduction preserves planarity of the graph.

(4) PARTITION INTO TRIANGLES: The reduction is from X3C and is the same as the reduction described in the proof of CLIQUE COVER. Given that the resulting graph has no cliques of size 4, the proof follows.

(5) MINIMUM HITTING SET: As given in [5], each instance of vertex cover can be seen to be an instance of a hitting set, in which every edge  $(u, v)$  corresponds to the set  $\{u, v\}$ . The elements of the set are simply the nodes of the graph. The result now follows by noting that there is a weakly parsimonious reduction from 3SAT to MINIMUM VERTEX COVER.  $\square$

COROLLARY 4.7. *The problems #PL-MINIMUM CLIQUE COVER, #PL-PARTITION INTO CLAWS, #PL-BIPARTITE DOMINATING SET, #PL-PARTITION INTO TRIANGLES, and #PL-MINIMUM HITTING SET are #P-complete.*

THEOREM 4.8. *Let  $\Pi$  be one of the problems in Table 1.1. It is #P-complete to count the number of solutions to  $\Pi$ , even when one is given an instance of  $\Pi$  and a solution which is guaranteed to satisfy  $\Pi$ .*

*Proof.* Starting from a 3CNF formula obtained in the proof of Theorem 3.4, we now do the same set of reductions discussed in the earlier theorems to obtain an instance of the problem  $\Pi$ . Since we know that the 3CNF formula is satisfiable, it follows that the instance of  $\Pi$  has a solution.  $\square$

**5. Unique and ambiguous planar problems.** Our parsimonious planar crossover box for 3SAT can also be used to show that additional problems for planar CNF formulas are *as hard as* the corresponding problems for arbitrary CNF formulas, with respect to polynomial time or random polynomial time reducibilities. We briefly describe these results. We first recall the definitions of  $D^P$  and random polynomial time reductions from [19, 27].

DEFINITION 5.1.  $D^P = \{L_1 - L_2 \mid L_1, L_2 \in NP\}$ . *Intuitively, a problem is in  $D^P$  if it can be solved by asking one question in NP and one question in Co-NP.*

DEFINITION 5.2. *Problem A is reducible to problem B by a randomized polynomial time reduction if there is a randomized polynomial time Turing machine T and a polynomial p such that*

- (1)  $\forall x, [x \notin A \rightarrow T[x] \notin B]$ ;

(2)  $\forall x, [x \in A \rightarrow T[x] \in B \text{ with probability at least } 1/p(|x|)]$ .  $\square$

THEOREM 5.3. UNIQUE-PL-3SAT is  $D^P$ -complete under randomized polynomial time reductions.

*Proof.* We modify the proof of the  $D^P$ -completeness of UNIQUE SAT in [27], so that whenever their reduction outputs a boolean formula  $f$ , our reduction outputs the planar formula  $PL(f)$  obtained by applying the parsimonious planar crossover box to  $f$ . The formula  $PL(f)$  has exactly the same number of satisfying assignments as  $f$ . In particular,  $PL(f)$  is uniquely satisfiable if and only if  $f$  is uniquely satisfiable.  $\square$

A second example is the following.

THEOREM 5.4. AMBIGUOUS-PL-3SAT is NP-complete.

*Proof.* Given an instance of an arbitrary 3CNF formula  $f$ , we first construct a new formula using the same construction as in Step 1 of Theorem 3.4. As pointed out in the proof of Theorem 3.4, the new formula is ambiguously satisfiable if and only if the original formula is satisfiable. We then do the same sequence of reductions as in Theorem 3.4 to obtain a planar formula that is ambiguously satisfied if and only if the original formula is satisfied.  $\square$

Using the ideas similar to those in the proof of Theorem 3.5, we can prove the following theorem.

THEOREM 5.5. AMBIGUOUS-1-VALID PL-3SAT is NP-complete.

COROLLARY 5.6. UNIQUE-1-VALID PL-3SAT is Co-NP-complete.

*Proof.* To prove the membership in Co-NP, consider an arbitrary formula  $F(x_1, \dots, x_n)$ , which is an instance of 1-VALID PL-3SAT. By the definition of 1-VALID formulas, an assignment  $\mathbf{v}$  to the variables such that  $\mathbf{v}[x_1] = \mathbf{v}[x_2] = \dots = \mathbf{v}[x_n] = 1$  satisfies  $F(x_1, \dots, x_n)$ . Now consider the formula  $H(x_1, \dots, x_n) = F(x_1, x_2, \dots, x_n) \wedge (\overline{x_1} \vee \overline{x_2} \dots \overline{x_n})$ .  $F$  is uniquely satisfiable if and only if  $H$  is unsatisfiable. To prove Co-NP-hardness, given a formula  $f(x_1, x_2, \dots, x_n)$ , we construct a formula  $g$  such that

$$g(x_1, \dots, x_{n+1}) = [f(x_1, x_2, \dots, x_n) \wedge x_{n+1}] \vee (\overline{x_1} \wedge \overline{x_2} \dots \overline{x_{n+1}}).$$

Now using ideas similar to those in the proof of Theorem 3.5, we obtain a planar formula  $g_1$  with the following properties:

- (1)  $g_1$  is 1-valid.
- (2)  $g_1$  is uniquely satisfiable if and only if  $f$  is unsatisfiable.  $\square$

Combining our parsimonious planar crossover box for 3SAT and the reductions to prove Theorem 3.8, we get that exact analogues of Theorems 5.3–5.5 hold for each of the problems EX3SAT, 1-3SAT, 1-EX3SAT, and 1-EX3MONOSAT. Thus, we have the following corollary.

COROLLARY 5.7. Let  $\Pi$  be one of the following problems: EX3SAT, 1-3SAT, 1-EX3SAT, or 1-EX3MONOSAT. Then the problem AMBIGUOUS-PL- $\Pi$  is NP-complete and the problem UNIQUE-PL- $\Pi$  is  $D^P$ -complete under randomized polynomial time reductions.

As a corollary of our parsimonious reductions, the unique versions of many graph problems are also  $D^P$ -complete.

COROLLARY 5.8. Let  $\Pi$  be one of the problems PL-PARTITION INTO TRIANGLES, PARTITION INTO CLAWS, or BIPARTITE DOMINATING SET. Then the problem AMBIGUOUS- $\Pi$  is NP-complete and the problem UNIQUE- $\Pi$  is  $D^P$ -complete under randomized polynomial reductions.

*Proof.* Given that each reduction in the sequence of reductions 3SAT  $\rightarrow$  PL-3SAT  $\rightarrow$  PL-EX1-3SAT  $\rightarrow$  PL-X3C is parsimonious, that UNIQUE 3SAT is  $D^P$ -complete,

and that the reduction from X3C to each of the problems mentioned above is parsimonious, the proof of the corollary is similar to the proof of Theorem 5.4.  $\square$

**5.1. Nonapproximability results for integer linear programming.** Next, we give an application of our result that AMBIGUOUS PL-3SAT is NP-complete and prove that it is not possible to approximate the optimal value of the objective function of an integer linear program.

An instance of an integer linear program consists of a system of linear inequalities and an objective function which is to be maximized (minimized); i.e., maximize (minimize)  $cx$  subject to the constraints  $Ax \leq b$ . The variables  $x$  are allowed to take only integer values. We say that a minimization problem  $\Pi$  is  $\epsilon$ -approximable,  $\epsilon > 1$  (or has an  $\epsilon$ -approximation) if there is a polynomial time algorithm that, given an instance  $I \in \Pi$ , finds a solution which is within a factor  $\epsilon$  of an optimal solution for  $I$ .

**THEOREM 5.9.** *Unless  $P = NP$ , given an instance of the integer linear program problem and a feasible solution, the maximum (minimum) value of the objective function is not polynomial time  $\epsilon$ -approximable for any  $\epsilon > 1$ , even when the bipartite graph associated with the set of constraints is planar.*

*Proof.* We prove the theorem for the maximization version of the problem. The proof for the minimization version is similar and hence omitted.

*Step 1.* Given a 3SAT formula  $f(x_1, x_2, \dots, x_n)$ , we construct a formula

$$g(x_1, \dots, x_{n+1}) = [f(x_1, x_2, \dots, x_n) \wedge x_{n+1}] \vee (\bar{x}_1 \wedge \bar{x}_2 \dots \bar{x}_{n+1}).$$

It follows that for any assignment  $\mathbf{v}$ ,  $\mathbf{v}[g(x_1, \dots, x_{n+1})] = 1$  if and only if either (i)  $\mathbf{v}[f(x_1, x_2, \dots, x_n)] = 1$  and  $\mathbf{v}[x_{n+1}] = 1$ , or (ii)  $\mathbf{v}[x_1] = \mathbf{v}[x_2] = \dots = \mathbf{v}[x_{n+1}] = 0$ .

*Step 2.* Starting from  $g(x_1, \dots, x_{n+1})$ , we construct a PL-3SAT formula  $\hat{g}(x_1, x_2, \dots, x_{n+1}, t_1, \dots, t_m)$  such that  $\hat{g}(x_1, x_2, \dots, x_{n+1}, t_1, \dots, t_m)$  is satisfiable if and only if  $g(x_1, x_2, \dots, x_{n+1})$  is satisfiable.

The construction can be carried out as in Step 2 in the proof of Theorem 3.4. We therefore omit the details here.

*Step 3.* Let  $\hat{g} = G_1 \wedge G_2 \dots G_r$ . Construct a new 1-EX3-MONOSAT formula  $h$  from  $\hat{g}$  such that  $\hat{g}$  is satisfiable if and only if  $h$  is satisfiable. Let  $h = C_1 \wedge C_2 \dots C_p$ . Replace each clause  $C_i = (x_{i_1} + x_{i_2} + x_{i_3})$  by the inequality  $(x_{i_1} + x_{i_2} + x_{i_3}) \geq 1$ . All the inequalities corresponding to the clauses make up the constraints. We also add constraints that  $\forall i, x_i \in \{0, 1\}$ . The objective function is now simply  $x_{n+1}$ . It is easy to verify that the maximum value of the objective function is exactly 1 if  $f(x_1, x_2, \dots, x_n)$  is satisfiable and is 0 otherwise. Hence it follows that unless  $P = NP$  the integer linear program problem has no polynomial time  $\epsilon$ -approximation algorithm for any  $\epsilon > 1$ .  $\square$

**6. Conclusions and open problems.** We showed that for many problems  $\Pi$  studied in the literature, the problems #PL- $\Pi$ , AMBIGUOUS-PL- $\Pi$ , and UNIQUE-PL- $\Pi$  are *as hard as* the respective problems # $\Pi$ , AMBIGUOUS- $\Pi$ , and UNIQUE- $\Pi$  with respect to polynomial time or random polynomial time reducibilities. We note that the problem #PL-HAMILTONIAN-CYCLE was proved to be #P-complete by Provan [20]. We can give an alternate proof of the #P-hardness of #PL-HAMILTONIAN-CYCLE by a reduction from a variant of RED1. The reduction is significantly more complicated than that in [20]. Consequently, we omit it here.

As corollaries of our results, we have shown that many planar problems are complete for the classes NP, #P, and  $D^P$ . Our results and their proofs provide the following general tools for proving hardness results for planar problems.



1. We have shown how parsimonious and weakly parsimonious crossover boxes can be used to prove the  $\#P$ -hardness of many planar counting problems. These ideas were used to prove the  $\#P$ -hardness of problem  $\#1$ -VALID PL-3SAT.

2. We extended the class of basic planar CNF satisfiability problems that are known to be NP-complete. Previously, only PL-3SAT [15] and PL-1-EX3SAT [3] were known to be NP-hard. We expect that the variants of the problem PL-3SAT shown to be NP-hard here will be useful in proving hardness results for many additional planar problems. In particular, we have already shown that the problem PL-1-3MONOSAT and its variant *RED1* are especially useful in proving the  $\#P$ -hardness of many planar graph problems.

3. We have shown that the problem AMBIGUOUS-PL-3SAT can be used to prove the nonapproximability of linear integer programming. Recently, there has been a lot of research in the area of approximability of graph and combinatorial problems, and the tools for showing negative results are few. Our proof of the nonapproximability of the minimum or maximum objective value of an integer linear program is direct and significantly different from the proof given in Kann [12] or Zuckerman [28]. Moreover, in [9], we show how to use the NP-completeness of AMBIGUOUS-PL-3SAT to show the nonapproximability of several constrained optimization problems even when restricted to *planar instances*. (The results in [12, 28] do not hold for planar instances.)

Finally, the results presented here and their proofs suggest a number of open problems including the following.

1. Can natural planar problems be found that are complete for additional complexity classes such as PSPACE,  $\#PSPACE$ , MAX SNP, MAX  $\Pi_1$ , etc.? (In recent papers [8, 9, 17, 10], we partially answer this question by showing that a number of problems are complete for the classes PSPACE,  $\#PSPACE$ , MAX-SNP, MAX  $\Pi_1$ , even when restricted to planar instances.)

2. Valiant [25] has shown that the problem  $\#2SAT$  is  $\#P$ -complete. How hard is the problem  $\#PL-2SAT$ ? We conjecture that the problem is  $\#P$ -complete, but it seems to us that different techniques than the ones used here are required to prove this. Recently Vadhan has affirmatively proved the conjecture [29].

3. We have shown that many unique satisfiability problems are complete for  $D^P$ , even when restricted to planar instances. Using our parsimonious reductions, we then proved the  $D^P$ -completeness of a number of graph problems for planar graphs. A number of such problems for planar graphs remain open. For example, how hard is the problem UNIQUE-PL-HAMILTONIAN CIRCUIT?

4. Do results similar to the ones proved in this paper hold for other restricted classes of graphs, e.g., intersection graphs of unit disks and squares? Such graphs have been studied extensively by [2, 7, 18] in the context of image processing, VLSI design, geometric location theory, and network design.

**Acknowledgments.** We thank the anonymous referee for invaluable suggestions. These suggestions significantly improved the quality of presentation and helped us in correcting a number of errors in the earlier draft. We thank Salil Vadhan for providing a draft of his paper [29].

#### REFERENCES

- [1] G. BRIGHTWELL AND P. WINKLER, *Counting linear extensions is  $\#P$ -complete*, in Proc. 23rd Annual ACM Symposium on Theory of Computing, 1991, pp. 175–181.
- [2] B.N. CLARK, C.J. COLBOURN, AND D.S. JOHNSON, *Unit disk graphs*, Discrete Math., 86 (1990), pp. 165–177.
- [3] M.E. DYER AND A.M. FRIEZE, *Planar 3DM is NP-complete*, J. Algorithms, 7 (1986), pp. 174–184.

- [4] M.E. DYER AND A.M. FRIEZE, *The complexity of partitioning graphs into connected components*, Discrete Appl. Math., 10 (1985), pp. 139–153.
- [5] M.R. GAREY AND D.S. JOHNSON, *Computers and Intractability. A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.
- [6] M.R. GAREY, D.S. JOHNSON, AND L. STOCKMEYER, *Simplified NP-complete problems*. Theoret. Comput. Sci., 1 (1976), pp. 237–267.
- [7] D.S. HOCHBAUM AND W. MAASS, *Approximation schemes for covering and packing problems in image processing and VLSI*, J. Assoc. Comput. Mach., 32 (1985), pp. 130–136.
- [8] H. B. HUNT III, R. E. STEARNS, AND M. V. MARATHE, *Local Reductions, Generalized Satisfiability, Complexity and Efficient Approximability: I*, in preparation. A preliminary version of the paper appears in Proc. 9th ACM Conference on Structure in Complexity Theory, 1994, pp. 356–366.
- [9] H. B. HUNT III, R. E. STEARNS, AND M. V. MARATHE, *Local Reductions, Generalized Satisfiability, Complexity and Efficient Approximability: II*, in preparation. A preliminary version of the paper appears in Proc. 9th ACM Conf. on Structure in Complexity Theory, 1994, pp. 356–366.
- [10] H. B. HUNT III, M. V. MARATHE, V. RADHAKRISHNAN, S. S. RAVI, D. J. ROSENKRANTZ, AND R.E. STEARNS, *Parallel approximation schemes for planar and near-planar satisfiability and graph problems*, Inform. and Comput., submitted, 1998. A preliminary version appears in Proc. 14th Foundations of Software Technology and Theoretical Computer Science (FST & TCS), Lecture Notes in Comp. Sci. 761, Springer-Verlag, New York, 1994, pp. 342–353. A complete version of the paper appears as Technical Report No. LA-UR-96-2723, Los Alamos National Laboratory, Los Alamos, NM, 1996.
- [11] H.B. HUNT III, V. RADHAKRISHNAN, M.V. MARATHE, AND R.E. STEARNS, *On the Complexity of Planar Graph/Hypergraph Coloring*, in preparation, 1997.
- [12] V. KANN, *Polynomially bounded minimization problems which are hard to approximate*, Nordic J. Comput., 1 (1994), pp. 317–331.
- [13] R.M. KARP, *Reducibility among combinatorial problems*, in Complexity of Computer Computations, R. E. Miller and J. W. Thatcher, eds., Plenum Press, New York, 1972, pp. 85–103.
- [14] R. KARP AND M. LUBY, *Monte Carlo algorithms for enumeration and reliability problems*, in Proc. 24th ACM Symp. on Theory of Computing, 1983, pp. 54–64. A complete version appears in J. Complexity, 1 (1985), pp. 45–64.
- [15] D. LICHTENSTEIN, *Planar formulae and their uses*, SIAM J. Comput., 11 (1982), pp. 329–343.
- [16] N. LINIAL, *Hard enumeration problems in geometry and combinatorics*, SIAM J. Algebraic Discrete Meth., 7 (1986), pp. 331–335.
- [17] M.V. MARATHE, H.B. HUNT III, R.E. STEARNS, AND V. RADHAKRISHNAN, *Complexity of Hierarchically and 1-Dimensional Periodically Specified Problems*, Technical Report No. LA-UR-95-3348, Los Alamos National Laboratory, Los Alamos, NM, August 1995. Also presented at the DIMACS Workshop on Satisfiability Problem: Theory and Applications, March 1996.
- [18] N. MEGGIDO AND K. SUPOWIT, *On the complexity of some common geometric location problems*, SIAM J. Comput., 13 (1984), pp. 182–196.
- [19] C. PAPADIMITRIOU, *Computational Complexity*, Addison-Wesley, Reading, MA, 1994.
- [20] J.S. PROVAN, *The complexity of reliability computations in planar and acyclic graphs*, SIAM J. Comput., 15 (1986), pp. 694–702.
- [21] J.S. PROVAN AND M. O. BALL, *On the complexity of counting cuts and of computing the probability that a graph is connected*, SIAM J. Comput., 12 (1983), pp. 777–788.
- [22] S. SALUJA, K. V. SUBRAMANIAM, AND M. THAKUR, *Descriptive complexity of #P functions*, J. Comput. System Sci., 50 (1995), pp. 493–505.
- [23] J.B. SAXE, *Two Papers on Graph Embedding Problems*, Technical Report CMU-CS-80-102, Dept. of Comp. Science, Carnegie Mellon University, 1980.
- [24] T.J. SCHAEFER, *The complexity of satisfiability problems*, in Proc. 10th ACM Symp. on Theory of Computing, 1978, pp. 216–226.
- [25] L.G. VALIANT, *The complexity of enumeration and reliability problems*, SIAM J. Comput., 8 (1979), pp. 410–421.
- [26] L.G. VALIANT, *The complexity of computing the permanent*, Theoret. Comput. Sci., 8 (1979), pp. 189–201.
- [27] L. G. VALIANT AND V. V. VAZIRANI, *NP is as easy as detecting unique solutions*, in Proc. 17th Annual ACM Symp. on Theory of Computing, 1985, pp. 458–463.
- [28] D. ZUCKERMAN, *NP-complete problems have versions that are hard to approximate*, in Proc. 8th Annual ACM Conf. on Structure in Complexity Theory, 1993, pp. 305–312. A complete version titled *On unapproximable versions of NP-complete problems* appears in SIAM J. Comput., 25 (1996), pp. 1293–1304.
- [29] S. VADHAN, *The Complexity of Counting in Sparse, Regular and Planar Graphs*, Technical Report, MIT, Cambridge, MA, 1997. Also available online from [www-math.mit.edu/~salil](http://www-math.mit.edu/~salil).