Parallel Approximation Schemes for a Class of Planar and Near Planar Combinatorial Optimization Problems

Harry B. Hunt III
Department of Computer Science, University at Albany—SUNY, Albany, New York 12222
E-mail: hunt@cs.albany.edu

Madhav V. Marathe
Los Alamos National Laboratory, P.O. Box 1663, MS M997, Los Alamos, New Mexico 87545
E-mail: marathe@lanl.gov

Venkatesh Radhakrishnan
Hewlett-Packard Company, MS 47LA, 19447 Pruneridge Avenue, Cupertino, California 95014
E-mail: rven@cup.hp.com

and

S. S. Ravi, Daniel J. Rosenkrantz, and Richard E. Stearns
Department of Computer Science, University at Albany—SUNY, Albany, New York 12222
E-mail: ravi@cs.albany.edu, djr@cs.albany.edu, res@cs.albany.edu

Received January 12, 1998

Defin a $(\delta, g)$-almost planar graph to be a graph $G(V, E)$ consisting of vertex set $V$ and a genus $g$ layout with at most $\delta \cdot |V|$ crossover nodes. We study a class of combinatorial optimization problems formulated as follows. Let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of variables each of which has a finite domain $D = \{0, 1, \ldots, \text{poly}(n)\}$. Also, let $S$ be a fixed finite set of finite arity relations $\{R_1, \ldots, R_q\}$. The optimization problem $\text{MAX-RELATION}(S)$ is the following: Given a set of terms $\{t_1, t_2, \ldots, t_m\}$, where each term $t_i$ is of the form $f(x_{i_1}, x_{i_2}, \ldots, x_{i_r})$ for some $f \in S$, assign values to each $x_i, 1 \leq i \leq n$, so as to maximize the number of satisfied terms. We show that for each fixed finite set $S$ and fixed $\delta, g \geq 0$, there is an NC-approximation scheme (NCAS) for the problem $\text{MAX-RELATION}(S)$ when restricted to instances whose bipartite graphs (that represent the variable-term relationship) are $(\delta, g)$-almost planar. This result in conjunction with approximation-preserving reductions to $\text{MAX-RELATION}(S)$ enables us to obtain NCASs for a number of graph theoretic and satisfiability problems when restricted to $(\delta, g)$-almost planar instances. Our results provide a characterization of a class of problems having an NCAS (and hence a PTAS).

Key Words: NC-approximation schemes; planar and almost planar graphs; bounded genus graphs; generalized CNF satisfiability MAX SNP.

1. INTRODUCTION

There has been extensive work on designing polynomial time approximation schemes and parallel $\epsilon$-approximations for NP-hard optimization problems restricted to planar instances [7, 15, 22, 34, 38, 43]. In this paper, we combine these lines of research and present a unifie approach for obtaining NC-approximation schemes for a large class of problems when restricted to graphs of fixed genus (which include planar graphs) and $\delta$-near-planar graphs (see Definition 7.1). Recall that an approximation algorithm for an optimization problem $\Pi$ provides a performance guarantee of $\rho$ if for every instance $I$ of $\Pi$, the value returned by the approximation algorithm is within a factor $\rho$ of the optimal value for $I$. A polynomial time approximation scheme (PTAS) for problem $\Pi$ is a family of algorithms $\mathcal{F}$ such that, for any fixed $\epsilon > 0$, there is a polynomial time algorithm $A \in \mathcal{F}$ in the family that for all $I \in \Pi$ returns a

1 Research partially supported by NSF under Grants CCR-94-06611 and CCR-97-34936 and the Department of Energy under Grant W-7405-ENG-36. Preliminary versions of portions of this paper have appeared as [25, 27].
solution which is within a factor $(1 + \epsilon)$ of the optimal value for $I$. An approximation scheme such that every algorithm in the family can be implemented in $NC$ (i.e., polylog time using a polynomial number of processors) is called an $NC$-approximation scheme (NCAS).

A graph is said to be \textit{planar} if it can be laid out in the plane (or equivalently, on the surface of a sphere) in such a way that there are no crossovers of edges. One direction along which the notion of planarity can be generalized is to allow a limited number of edge crossovers in a layout. This generalization leads to the notion of $\delta$-near-planar graphs \cite{45}. Informally, for each fixed $\delta \geq 0$, a $\delta$-near-planar graph is a graph with vertex set $V$ together with a planar layout with at most $\delta |V|$ crossovers of edges. Such graphs arise in a number of application areas such as wide-area networks \cite{12,50}, radio networks \cite{46}, and finite element analysis \cite{52}. Further, the interaction graphs of many optimization problems on planar graphs and several classes of geometric graphs are $\delta$-near-planar \cite{26,38}.

A second direction along which the notion of planarity can be generalized is to consider embeddings of graphs on other surfaces. This leads to the notion of graph \textit{genus}. The genus of a graph is the minimum integer $g \geq 0$ such that the graph can be embedded with no edge crossovers on the surface of a sphere with $g$ handles. (The genus of any planar graph is zero.) For the convenience of the reader, definitions related to genus and some results regarding genus that are used in this paper are given in the Appendix. Thomassen \cite{53} proved that the problem of determining whether a graph has genus $g$ is $NP$-complete.

As a result, for the rest of this paper, we will assume that we are given a graph and its layout of genus $g$. Graphs of bounded genus have several applications including VLSI layout via minimization and bounded thickness book embeddings (see \cite{16,17,53} and the references cited therein).

The classes of $\delta$-near-planar graphs and bounded genus graphs are in a sense orthogonal. Specifically, there are classes of graphs which are $\delta$-near-planar for small $\delta$ but whose genus is not bounded by any constant. For example, consider a family of graphs that for each positive integer $n > 0$, consist of a clique on $n^{1/4}$ nodes and a simple chain of length $(n - n^{1/4})$ attached to one of the vertices in the clique (Fig. 1a). These graphs have genus $\Theta(n^{1/2})$ since the genus of an $r$-clique is $\Theta(r^2)$ \cite{57}. However, the graphs are $1$-near-planar, since the number of crossovers is at most $n$. Similarly, there are graphs of bounded genus that are not $\delta$-near-planar for any fixed $\delta$. To illustrate this point, consider another family of graphs that for each positive integer $n > 0$ consist of a $\sqrt{n} \times \sqrt{n}$ torus as laid out in Fig. 1b. In this layout, the number of crossovers is $\Theta(n^2)$. Thus, the graphs with the given layouts are not $\delta$-near-planar for any constant $\delta$. On the other hand, it can be seen that the graphs have genus at most 1; that is, they can be laid out without any edge crossovers on a sphere with one handle.

2. SUMMARY OF RESULTS AND THEIR SIGNIFICANCE

The focus of this paper is on a class of optimization problems restricted to instances whose underlying graphs are a generalization of both $\delta$-near planar graphs and bounded genus graphs. We refer to such
graphs as \((\delta, g)\)-almost planar graphs. Formally, a \((\delta, g)\)-almost planar graph is a graph with vertex set \(V\) together with a genus \(g\) layout with at most \(\delta \cdot |V|\) crossover\(^2\) nodes. The class of \((\delta, g)\)-almost planar graphs contains both \(\delta\)-near-planar graphs and genus \(g\) graphs. To define the general class of optimization problems considered in this paper, let \(X = \{x_1, x_2, \ldots, x_n\}\) be a set of variables each of which has a finite domain \(D = \{0, 1, \ldots, \text{poly}(n)\}\). Also, let \(S\) be a \(\delta\)-finite set of \(\delta\)-finite arity relations \(\{R_1, \ldots, R_q\}\). The optimization problem \(\text{MAX-RELATION}(S)\) is the following: Given a set of terms \(\{t_1, t_2, \ldots, t_m\}\), where each term \(t_i\) is of the form \(f(x_{i1}, x_{i2}, \ldots, x_{ik})\) for some \(f \in S\), assign values to each \(x_{ij}\), \(1 \leq i \leq n\), so as to maximize the number of satisfied terms. It should be noted that the \(\text{MAX-k-CONSTRAINT SATISFACTION PROBLEMS (MAX-k-CSP)}\) studied in [13, 32, 51, 54], where \(k\) is a constant specifying the maximum number of variables appearing in any constraint, are \(\text{MAX-RELATION}(S)\) problems for appropriate sets \(S\) of relations.

For each \(\delta\), \(g \geq 0\), we present a general approach for devising NCASs for a class of optimization problems restricted to instances for which the bipartite graphs representing the variable-term relationship are \((\delta, g)\)-almost planar. The general approach involves two main steps.

First, by an extension of the ideas in Baker [7] we show that for each \(\delta\)-finite set \(S\) and \(\delta\)-finite \((\delta, g)\)-almost planar graphs, there is an NCAS for the problem \(\text{MAX-RELATION}(S)\) when restricted to instances whose bipartite graphs are \((\delta, g)\)-almost planar. In the next step, we show that a number of important classes of problems when restricted to \((\delta, g)\)-almost planar instances can be reduced to appropriate problems \(\text{MAX-RELATION}(S)\). The reductions devised have two important properties: (i) they can be carried out in NC and (ii) if a problem instance is \((\delta, g)\)-almost planar, then the instance of \(\text{MAX-RELATION}(S)\) obtained as a result of the reduction is \((\delta', g')\)-almost planar, where \(\delta'\) and \(g'\) are functions of \(\delta\) and \(g\) (independent of \(n\)). Thus, each of these problems has an NCAS when restricted to \((\delta, g)\)-almost planar instances. We refer to such reductions as \(\text{structure preserving NC-L-reductions}\).

Our results provide a syntactic (algebraic) class of problems, namely, \((0, 0)\)-almost planar \(\text{MAX-RELATION}(S)\), whose closure under L-reductions defines one characterization for problems that have an NCAS (and hence a PTAS). As will be seen, the algebraic model (characterization) is general enough to express the optimization versions of (i) the generalized satisfiability problems of Schaefer [49], (ii) a class of nonlinear optimization problems, and (iii) several well-known graph theoretic problems. A number of additional applications of our results are presented in Sections 8 and 9.

The results presented here extend the known results in the following ways. First, no PTASs were known for any of the problems considered here when restricted to either bounded genus graphs or \(\delta\)-near-planar graphs. Thus, our results significantly extend the results in [7, 38, 43]. The PTASs for various planar satisfiability problems have the same time versus performance trade-off as those of Baker [7]. Prior to this work, only \(\text{MAX 3SAT}\) restricted to planar instances was known to have a PTAS [43]. Moreover, since the PTAS is based on the planar separator theorem, it is a PTAS only in the asymptotic sense [29]. (To illustrate the asymptotic nature of these approximation schemes, it is observed in [7] that using the planar separator theorem [38] for obtaining an independent set which is at least half the size of an optimal solution requires graphs with at least \(2^{\Omega(n)}\) nodes. Similar numbers can be obtained for the satisfiability problems as well.)

Second, as mentioned earlier, we develop the technique of \emph{structure preserving NC-L-reductions}. Specifically, we aim at devising L-reductions that preserve the graph theoretic parameters of the underlying instance. The L-reductions proposed earlier to prove nonapproximability typically did not have this property. Furthermore, the results presented here demonstrate the use of L-reductions in devising PTASs and NCASs rather than in proving nonapproximability results. Trevisan [55] has also used L-reductions extensively in devising PTASs. Although some of the PTASs implied by our results were known in the sequential case [7], our general approach provides insights into the common structure of the problems. Moreover, the approach allows us to obtain NCASs for other problems (such as \(\text{MAX-SAT}(S)\) and hypergraph problems) for \((\delta, g)\)-almost planar graphs or hypergraphs for which no previous approximation schemes were known. As an additional property, we observe that structure preserving L-reductions are often very efficient in terms of time and space; thus, the resulting sequential and parallel algorithms use linear or near-linear total work.

\(^2\) A formal definition of the notion of crossover nodes was presented in [45] in the context of \(\delta\)-near-planar graphs. This definition appears in Section 7.
The main results of this paper first appeared as a technical report [24] and subsequently as a conference paper [25]. Some of the results were obtained independently by other researchers and are reported in [11, 31, 32].

The remainder of this paper is organized as follows. In Section 3 we discuss related work. Sections 4 and 5 contain definitions and preliminary results. Section 6 discusses NCASs for problems restricted to (0, g)-almost planar instances. Section 7 extends the results in Section 6 to obtain approximation schemes for problems restricted to (δ, g)-almost planar instances. In Sections 8 and 9 we discuss some applications of the results in the previous sections.

3. RELATED WORK

Several researchers have developed syntactic characterizations of decision and optimization problems to provide a uniform framework for solving such problems [8, 13, 31–33, 36, 49]. Building on our work presented in [25], Khanna and Motwani [31] have presented elegant methods that extend and generalize some of the results presented here. They defined three important syntactic classes, namely MPSAT, TMAX, and TMIN. As a corollary, they show that the problem Pl-MAX SAT has a PTAS, thus answering an open question in [25]. Independently, Trevisan [55] also obtained a PTAS for Pl-MAX SAT. The syntactic class TMAX extends our ideas for the AN3SAT problem (see Section 8.3). Jacob et al. [28] present further extensions of some results in [31]. Arora, Karger and Karpinski [4] have devised PTASs for a number of graph problems restricted to dense instances. Frieze and Kannan [19] extended this work and devised PTASs for a subclass of problems MAX-RELATION(S) when restricted to dense instances.

Lipton and Tarjan [38] devised PTASs for planar graph problems using the planar separator theorem. Baker [7] presented a technique to obtain PTASs with better performance-time trade-offs for planar graph problems. The idea is to decompose a given planar graph into a number of outerplanar subgraphs, solve the problem optimally for each subgraph and combine these optimal solutions to obtain a near-optimal solution for the whole graph. Only sequential approximation schemes were considered in [7, 38]. Also, approximation schemes for MAX-3SAT and other generalized satisfiability problems were not considered in [7]. NCASs for planar graph problems were considered independently in [15, 34]. Other researchers have also used a technique similar to that in [7] to design approximation schemes for problems restricted to geometric instances (see [23, 26, 56]). Recently Arora et al. devised PTASs for a number of problems (such as the traveling salesperson problem) when restricted to (i) planar graphs [3] and (ii) Euclidean plane [2].

The class of δ-near-planar graphs was introduced in Radhakrishnan et al. [45]. There it was shown that several path problems for δ-near-planar graphs can be solved in time bounded by linear functions of the best known time bounds for the corresponding planar graph problems. These algorithms were based on finding a good tree decomposition of the given δ-near-planar graph. In [45], the authors asked whether the ideas in [7] could be used to find efficient approximation schemes for NP-complete problems on δ-near-planar graphs. The results in this paper answer their question affirmatively.

Graphs of fixed genus have been studied extensively in the past. Many difficult problems for general graphs are solvable efficiently when restricted to graphs of fixed genus. For example, Miller [42] showed that the isomorphism problem can be solved efficiently for graphs of fixed genus. Djidjev and Reif [16] gave a linear time algorithm to find small separators in bounded genus graphs. Eppstein [17] gave linear time algorithms for solving the subgraph isomorphism problem for bounded genus graphs.

4. BASIC DEFINITIONS

In this section, we present a few basic definitions which enable us to discuss formulas and their values. We begin with some notation for describing terms and formulas.

For a set $D$ and a natural number $m$, the set of $m$-tuples of elements of $D$ are denoted by $D^m$. A subset $R$ of $D^m$ is called a $m$-ary relation over $D$; $m$ denotes the arity of $R$. A term is a string of the form $R(x_1, \ldots, x_k)$ where $x_i$ are variables or constants and $R$ is a $k$-ary relation over $D$. (The tuples of $R$, indicate the allowed values that the variables $\{x_1, \ldots, x_k\}$ can take.)
Let $S = \{R_1, \ldots, R_m\}$ be any finite set of finite arity relations. Any term $p = f(x_1, \ldots, x_k)$, where $f \in S$, is called an $S$-term; the size of $p$, denoted by $|p|$, is defined to be $k + 1$. $\text{VAR}(p)$ is defined to be the set of variables in the list $x_1, \ldots, x_k$. If $y$ is an assignment such that $\text{VAR}(y) \supseteq \text{VAR}(p)$ and $p$ is an $S$-term for some set $S$, we define $p[y]$ to be the value $f(d_1, \ldots, d_k)$ where $d_i$ is the value assigned to variable $x_i$ by $y$. When $P$ is a set of terms, we define $\text{VAR}(P) = \bigcup_{p \in P} \text{VAR}(p)$.

A formula $F$ is a pair $(V, P)$ where $V$ is a set of variables and $P$ is a set of terms such that $V \supseteq \text{VAR}(P)$. A formula in which each $P$ is an $S$-term is called an $S$-formula. Given a set $S$, where each $R_i \in S$ is specified by an explicit table, and an $S$-formula $F$, the problem $\text{MAX-RELATION}(S)$ is to determine an assignment to the variables of $F$ so as to maximize the number of terms satisfied. In this paper, we restrict our attention to variables $V = \{x_1, \ldots, x_n\}$ with domain $D = \{0, 1, \ldots, \text{poly}(n)\}$; thus, we allow the domain size to grow polynomially with the number of variables in the formula. When $D = \{0, 1\}$, the problem $\text{MAX-RELATION}(S)$ is denoted by $\text{MAX-SAT}(S)$. We note that $\text{MAX-SAT}(S)$ corresponds to maximization versions of the generalized satisfiability problems introduced by Schafer [49].

**Definition 4.1** [44]. Let $\Pi$ and $\Pi'$ be two optimization (maximization or minimization) problems. We say that $\Pi$ $L$-reduces to $\Pi'$ (denoted by $\Pi \leq_L \Pi'$) if there are two polynomial time algorithms $f$ and $g$ and constants $\alpha, \beta > 0$, such that for each instance $I$ of $\Pi$:

1. Algorithm $f$ produces an instance $I' = f(I)$ of $\Pi'$ such that the optima of $I$ and $I'$, $\text{OPT}(I)$ and $\text{OPT}(I')$, respectively, satisfy $\text{OPT}(I') \leq \alpha \text{OPT}(I)$.
2. Given any solution of $I'$ with cost $c'$, algorithm $g$ produces a solution of $I$ with cost $c$ such that $|c - \text{OPT}(I)| \leq \beta |c' - \text{OPT}(I')|$.

**Definition 4.2.** An NC-L-reduction is an L-reduction in which the functions $f$ and $g$ are NC computable.

As noted in [44], two important properties of L-reductions (and also of NC-L-reductions) are the following: (i) L-reductions compose; i.e., if $P \leq_L Q$ and $Q \leq_L R$ then $P \leq_L R$. (ii) If $P \leq_L Q$ and $Q$ has a PTAS, then $P$ has a PTAS.

**Definition 4.3.** Let $S$ be a finite set of finite arity relations over $D$. The bipartite graph of an $S$-formula $F$ (denoted by $BG(F)$) is defined as follows. The terms and variables in the formula $F$ are in one to one correspondence with the vertices of the graph. There is an edge between a term node and a variable node if and only if the variable appears (in complemented or uncomplemented form) in the term.

The interaction graph of an $S$-formula $F$ (denoted by $IG(F)$) is a graph $G(V, E)$ defined as follows. The variables in the formula $F$ are in one to one correspondence with the vertices of $G$. There is an edge $\{u, v\} \in E$ if and only if variables $u$ and $v$ appear together in some term of $F$.

We use standard graph theoretic definitions [57] and refer the reader to [20] for definitions of basic combinatorial problems in graph theory and logic.

**Definition 4.4** [1, 10, 48]. Let $G(V, E)$ be a graph. A tree-decomposition of $G$ is a pair $((X_i | i \in I), T = (I, \mathcal{F}))$, where $\{X_i | i \in I\}$ is a family of subsets of $V$ and $T = (I, \mathcal{F})$ is a tree with the following properties:

1. $\bigcup_{i \in I} X_i = V$.
2. For every edge $e = (u, v) \in E$, there is a subset $X_i, i \in I$, with $u \in X_i$ and $v \in X_i$.
3. For all $i, j, k \in I$, if $j$ lies on the path from $i$ to $k$ in $T$, then $X_i \cap X_k \subseteq X_j$.

The treewidth of a tree-decomposition $((X_i | i \in I), T)$ is $\max_{i \in I} |X_i| - 1$. The treewidth of a graph is the minimum over the treewidths of all tree decompositions.

We now clarify what we mean by the phrase problems restricted to almost-planar instances. For a graph problem $\Pi$, we use the phrase to mean a restriction of $\Pi$ to instances in which the graphs are almost-planar. For the problems $\text{MAX-RELATION}(S)$, we use the phrase to mean the restriction to instances whose corresponding bipartite graphs are almost-planar. Additionally, given a graph problem $\Pi$, we will use $(\delta, g)$-\Pi to denote the restriction of the problem $\Pi$ to instances in which the associated graph is $(\delta, g)$-almost-planar. Thus, for fixed $\delta, g \geq 0$, we use $(\delta, g)$-\text{MAX-RELATION}(S) to denote the problem
Given an undirected graph \( n \) weights can use Theorem 5.2 to bound the treewidth of a (possibly disconnected) graph into the maximum of the treewidths of its connected components, so that the solution to the whole graph.

When this is not the case, we can solve the problem for each connected component and union the solutions such that each induced subgraph is isomorphic to a given fixed graph \( H \).

\[ \text{MAX-H-MATCHING} \]

The shortest distance from any vertex \( v \) in any \( k \) consecutive levels has treewidth \( O(f(g + 1, D)) \), where the function \( f \) is independent of \( n \).

Let \( G \) be a graph of fixed genus \( g \) and assume that a level numbering of \( G \) has been carried out. Let \( k \geq 1 \) be a fixed integer. Consider the set of vertices of \( G \) in any \( k \) consecutive levels. Let \( G_k \) denote the subgraph of \( G \) induced on these vertices. Note that \( G_k \) may be disconnected. However, as mentioned earlier, for the problems considered in this paper, we can process each component separately. It is not difficult to see that the diameter of any connected component of \( G_k \) is at most \( 2k \). Since the treewidth of a (possibly disconnected) graph is the maximum of the treewidths of its connected components, we can use Theorem 5.2 to bound the treewidth of \( G_k \). The following theorem indicates this bound.

\[ \text{THEOREM 5.3.} \quad \text{For each fixed } g \geq 0, \text{ given an } n \text{-node graph } G \text{ of genus } g, \text{ the subgraph of } G \text{ induced by the vertices in any } k \text{ consecutive levels has treewidth } O(f(g + 1, 2k)). \]

Eppstein [18] has observed that for graphs of constant genus, the function \( f(g + 1, D) = O(gD) \) is linear.
PROPOSITION 5.1. Let $\mathcal{S}$ be a finite set of relations whose arity is bounded by a constant $B$. Let $F$ be a formula corresponding to an instance of the problem $\text{MAX-RELATION}(\mathcal{S})$ such that the treewidth of the bipartite graph $BG(F)$ is at most $k$. Then, the treewidth of the interaction graph corresponding to $F$ is at most $Bk$.

Proof. Since the maximum arity of any relation in $\mathcal{S}$ is $B$, each term in $F$ has at most $B$ variables. Given a tree-decomposition $((X_i | i \in I), T = (I, \mathcal{F}))$ of the bipartite graph $BG(F)$, the tree-decomposition of the interaction graph $IG(F)$ can be obtained as follows. For each vertex $v_i$ corresponding to a term $c$ that uses variables $u_1, u_2, \ldots, u_r$, we do the following. Let $v_i$ belong to the sets $X_{i1}, \ldots, X_{im}$. In each set $X_{ij}$, $1 \leq j \leq m$, replace $v_i$ by the set of vertices $v_1 \ldots v_r$, $1 \leq r \leq B$, that correspond to the variables $u_1, \ldots, u_r$. It can be seen that the resulting pair $((X'_i | i \in I), T = (I, \mathcal{F}))$ is a tree decomposition of the interaction graph $IG(F)$. Since the treewidth of $BG(F)$ is at most $k$, the treewidth of $IG(F)$ is at most $Bk$. ■

LEMMA 5.1. Let $\mathcal{S}$ be a finite set of finite arity functions. Let $F$ be a formula corresponding to an instance of the problem $\text{MAX-RELATION}(\mathcal{S})$ such that the treewidth of the bipartite graph $BG(F)$ is at most $k$, where $k$ is a fixed integer. Then there is an NC-algorithm to obtain an optimal solution to $F$.

Proof. We first prove the theorem for the problems $\text{MAX-SAT}(\mathcal{S})$. As shown in [10], for any fixed $k$, there is an NC algorithm that finds a binary tree decomposition (i.e., each node in the tree has at most two children) of a graph with treewidth $k$. Also note that Proposition 5.1 allows us to work with the interaction graph of $F$. We use dynamic programming to solve instances of $\text{MAX-SAT}(\mathcal{S})$ whose interaction graph has treewidth $\leq k$; the resulting algorithm (similar to that used in [10]) runs in $O(n \log n)$ time using $O(n)$ processors. Let $((X_i | i \in I), T = (I, \mathcal{F}))$ denote the tree-decomposition of the interaction graph of $F$. Any node $i$ of $I$, it suffices to keep a table of the assignments to the variables in $X_i$ and the maximum number of clauses satisfied for each of these assignments. Since the treewidth of $BG(F)$ is at most $k$, the number of assignments that need to be stored in the table at any node is at most $2^k$. The table for each node $i$ can be computed from the tables of its descendents in constant time. Since the depth of the tree is $O(\log n)$, the table for the root node can be computed in $O(\log n)$ time, and the maximum number of clauses satisfied can be computed in constant time from the table at the root. The total number of processors used is $O(n)$ because at each level of the binary tree $T$, the number of nodes is $O(n)$.

It is now easy to further extend the ideas to obtain an $n^{O(k)}$ algorithm for the problems $\text{MAX-RELATION}(\mathcal{S})$ when restricted to instances of bounded treewidth, when the size of each domain $D_i$ is polynomial in the size of the instance. This follows by simply observing that at each stage of the dynamic programming algorithm we need to keep a table of size $n^{O(k)}$ instead of a table of size $2^k$ as was done for binary valued variables. The remaining details are straightforward. ■

We end this section with results from [5, 40, 47] which point out the difficulty of approximating some of the optimization problems considered in this paper.

THEOREM 5.4 [5, 40, 47]. Unless $P = NP$, the following statements hold:

1. The problems $\text{MAX-Triangle Matching}$ and $\text{MAX-H-Matching}$ do not have polynomial time approximation schemes.

2. For any fixed $\epsilon > 0$, the problem $\text{MIN-Dominating Set}$ does not have an approximation algorithm with performance guarantee $(1 - \epsilon) \ln n$, where $n$ is the number of nodes.

6. AN NC-APPROXIMATION SCHEME FOR $((0, 0))-\text{MAX-RELATION}(\mathcal{S})$

INPUT: An instance $F$ of $((0, 0))-\text{MAX-SAT}(\mathcal{S})$ represented as a bipartite graph $BG(F)$. (The positive integer $k$ which determines the performance guarantee is fixed in advance.)

1. Using NC-BFS, perform a breadth-first-search on the planar graph $BG(F)$ starting at any node $v$ in $BG(F)$. The level number of each node $w$ is the length of the path (i.e., the number of nodes in the path including the end points) from $v$ to $w$ in the BFS tree. (Remark: Since $BG(F)$ is bipartite and
the level numbers are obtained by BFS, all the nodes at any given level will be either variable nodes or clause nodes.

2. For each $i$ ($0 \leq i \leq k$), obtain an assignment to the variables of $F$ as follows:

(a) Partition $BG(F)$ into subgraphs $G'_1, G'_2, \ldots, G'_r$, each of diameter at most $4k + 4$, by deleting clauses at levels $j \equiv i \mod(2k + 2)$. This is done only for those values of $j$ for which the corresponding level contains only clause nodes. Let $F_i$ denote the formula obtained as a result of this operation. (Remark: Each $G'_j$ (1 \leq j \leq r) is the interaction graph of a subformula $F'_j$ of $F$. The variable set $V_j$ for $F_j$ is the set of vertices of $G'_i$ and a clause $c$ is included in $F'_j$ if and only if each variable appearing in $c$ is also in $V_j$.)

(b) Using Lemma 5.1, obtain an assignment to the variables of the subformula $F'_j$. Let $\text{COST}(F'_j)$ denote the number of satisfied clauses in $F'_j$. (Remark: The interaction graph of $F'_j$ has treewidth $O(k)$, which is a constant for a fixed $k$. Thus, by Lemma 5.1, the assignment maximizes the number of satisfied clauses in $F'_j$.)

(c) The assignment to the variables of $F$ is the union of the assignments to the variables of each subformula $F'_j$.

(d) $\text{COST}(F_i) = \sum_{1 \leq j \leq r} \text{COST}(F'_j)$.

3. $\text{HEU}(F) = \max_{0 \leq i \leq k} \text{COST}(F_i)$.

Output: The assignment found in Step 3. (This assignment satisfies at least $\frac{k}{k+1} |\text{OPT}(F)|$ clauses; see Theorem 6.1.)

In this section, we first present our approximation schemes for $\text{MAX-RELATION}(S)$, when restricted to planar instances. The results are obtained by extending an elegant technique of Baker [7] for obtaining approximation schemes for certain graph theoretic problems. This technique is very similar to a technique devised independently by Hochbaum and Maass [23] in the context of approximating certain geometric packing and covering problems. Following the terminology of [23], we call this the shifting technique. Consider a problem $\Pi$ which can be solved by a divide-and-conquer approach with a performance guarantee of $\rho$. The shifting strategy allows us to bound the error of the simple divide-and-conquer approach by applying it iteratively and choosing the best solution among these iterations as the solution to $\Pi$. In the context of $\text{MAX-RELATION}(S)$, the two important properties needed to obtain such algorithms are as follows:

1. The ability to decompose the given formula into variable-disjoint subformulas such that an optimal solution to each subformula can be obtained in polynomial time.

2. The ability to obtain a near-optimal solution for the whole formula by merging the optimal solution obtained for each subformula.

$\text{HEU-(0, 0)-MAX-SAT}(S)$ shows the steps of our approximation scheme for $(0, 0)$-$\text{MAX-SAT}(S)$. The next theorem establishes the performance guarantee provided by this heuristic.

**Theorem 6.1.** Given an instance $I$ of $(0, 0)$-$\text{MAX-SAT}(S)$, let $\text{OPT}(I)$ and $\text{HEU}(I)$ denote the set of satisfied clauses in an optimal truth assignment and that in a truth assignment produced by $\text{HEU-(0, 0)-MAX-SAT}(S)$, respectively. For each fixed $k$, $\text{HEU-(0, 0)-MAX-SAT}(S)$ is an NC-algorithm and $|\text{HEU}(I)| \geq \frac{k}{k+1} |\text{OPT}(I)|$.

**Proof.** For each value of $i$, the solution obtained in Step 2(c) of Algorithm $\text{HEU-(0, 0)-MAX-SAT}(S)$ is optimal (i.e., an assignment satisfying the maximum number of clauses) for the subformula of $I$ consisting of variables at levels $j \neq i \mod(2k + 2)$. Let $l$ be the integer such that for the levels $j = l \mod(2k + 2)$, the optimal solution $\text{OPT}(I)$ contains the minimum number of clauses from subformulas induced by variables at levels $j$. Let $\text{IND}(I)$ be the clauses in levels $j = l \mod(2k + 2)$ of $\text{OPT}(I)$. Hence, $|\text{IND}(I)| \leq \frac{l}{k+1} |\text{OPT}(I)|$. Since $\text{HEU-(0, 0)-MAX-SAT}(S)$ computes an optimum solution for the formulas whose corresponding variables are not in levels $j = l \mod(2k + 2)$, we have $|\text{HEU}(I)| \geq |\text{OPT}(I)| - |\text{IND}(I)|$. Hence,

$$|\text{HEU}(I)| \geq |\text{OPT}(I)| - \frac{1}{k+1} |\text{OPT}(I)|.$$
That is,

$$|\text{HEU}(I)| \geq \frac{k}{k+1} |\text{OPT}(I)|.$$  

We sketch how the algorithm can be implemented in NC. First consider Step 2(a). Suppose that the NC-BFS was started at a variable node, and consider the iteration when $$i = 0$$. (The details for the other values of $$i$$ are similar.) Since $$BG(F)$$ is bipartite, all the nodes at even numbered levels are clause nodes. For $$i = 0$$, we delete the clause nodes at levels 0, 2, 4, 6, etc. Thus, $$G_0^1$$ is the subgraph induced on nodes at levels 1 through 2, $$G_0^2$$ is the subgraph induced on nodes at levels 2 through 4, and so on. Using one processor per node, it is easy to compute the subgraph to which a node belongs in $$O(1)$$ parallel time. Once the subgraphs are found, we can carry out Step 2(b) in NC using the implementation presented in the proof of Lemma 5.1. NC implementations of Steps 2(c) and 2(d) are straightforward. Since the loop runs only $$k + 1$$ times (which is a constant), we have an NC implementation of the algorithm.

For fixed $$g \geq 0$$, consider the problems $$\text{MAX-RELATION}(S)$$ when restricted to instances whose corresponding bipartite graphs are $$(0, g)$$-almost planar. Then by a direct combination of Theorem 6.1, Theorem 5.3, and Lemma 4, we get

**Theorem 6.2.** For each fixed $$g \geq 0$$, the problems $$(0, g)$$-$$\text{MAX-RELATION}(S)$$ have NCAS.

**Remark.** We note an important difference between the time versus performance trade-offs of our algorithms for $$\text{MAX-SAT}(S)$$ and $$\text{MAX-RELATION}(S)$$. For achieving a performance of $$(k+1)/k$$ for $$(0, 0)$$-$$\text{MAX-SAT}(S)$$, our sequential algorithms take time $$O(2^k n)$$ while they take time $$n^{O(k)}$$ for $$(0, 0)$$-$$\text{MAX-RELATION}(S)$$. Thus, for the maximization versions of the general Boolean CNF satisfiability problems, we have linear time (and almost linear parallel work) approximation schemes. An important corollary of this is that most of the problems reduced to $$\text{MAX-SAT}(S)$$ in this paper also have linear time (and almost linear work) approximation schemes. Thus, for the rest of this paper, we do not explicitly state the exact running times; they can be calculated from the observation made above and the time needed to perform the NC-L-reductions.

### 7. $$(\delta, g)$$-ALMOST-PLANAR Instances

Next, we show that several natural CNF satisfiability and graph problems considered in [44] when restricted to $$\delta$$-near-planar instances have NCASs. First, we recall a few basic definitions from [45].

**Definition 7.1.** Let $$G(V, E)$$ be a graph. A planar layout with crossovers (see Fig. 2) for $$G$$ is a planar graph $$G' = (V', E')$$ together with a set $$C$$ of crossover nodes and a function $$h : E' \rightarrow E$$ such that the following properties hold:

![Fig. 2](image-url) 

A graph $$G$$ and a planar layout for $$G$$ with crossovers. The set $$C$$ of crossover nodes = \{w, u, x, y, z\}. The function $$h : E' \rightarrow E$$ can be readily inferred from the planar layout. (For example, $$h((E, w)) = h((w, z)) = h((z, B)) = (E, B).$$)
1. \(C \cap V = \emptyset\).
2. \(V' = V \cup C\).
3. Each node of \(C\) has degree 4 in \(G'\).
4. For all \((a, b)\) in \(E\), \(\{e \in E' \mid h(e) = (a, b)\}\) is the set of edges on a simple path from \(a\) to \(b\) in \(G'\) involving no other nodes of \(V\).

Let \(\delta\) be a positive number. A \(\delta\)-near-planar graph \(G(V, E)\) is a graph with vertex set \(V\) together with a planar layout with \(\leq \delta \cdot |V|\) crossover nodes.

A crossover node \(c\) is associated with (distinct) edges \(e_1\) and \(e_2\) of \(E\) if and only if there are edges \(e_1'\) and \(e_2'\) in \(E'\) with endpoint \(c\) such that \(e_1 = h(e_1')\) and \(e_2 = h(e_2')\).

We begin with the definition of a crossover box which is used in our approximation schemes. This definition is similar in spirit to that of [37], where crossovers were first used to prove the NP-hardness of decision problems for planar instances.

**Definition 7.2.** 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 all the following conditions hold:

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 \(b \equiv b_1\).

**Theorem 7.1.** For each fixed \(\delta, g \geq 0\), each of the problems \((\delta, g)\)-MAX-RELATION(\(S\)) has an NCAS.

**Proof.** We first show that for all \(\delta \geq 0\), \((\delta, 0)\)-MAX-3SAT has an NCAS. This is done by specifying an NC-L-reduction from \((\delta, 0)\)-MAX-3SAT to \((0, 0)\)-MAX-SAT(\(S\)), where \(S\) will be defined later. Consider an arbitrary instance of \((\delta, 0)\)-MAX-3SAT given by the CNF formula \(F\) with the graph \(G'_{F}\) (the graph corresponding to \(F\) as in Definition 7.1) embedded in the plane. This layout is a planar graph with vertex set consisting of the variables of \(F\), the clauses of \(F\), and the crossover nodes. In this layout, we add a new variable node on the edge between two crossover nodes or between a crossover node and a clause node as shown in Fig. 3a. Formally, we modify \(G'\) in the following manner, so that the neighbors of all clause and crossover nodes are variable nodes. Consider an edge \(e = (c_i, p_j)\), where \(c_i\) is a crossover node and \(p_j\) is a clause node. Suppose \(h(e) = (p_j, v_1)\). Delete \((c_i, p_j)\), introduce the new

**FIG. 3.** (a) Introducing new variables (dark circles) in the bipartite graph for an instance of \((\delta, 0)\)-MAX-3SAT. (b) Our crossover box used to obtain NC-L-reduction. Here \(C_1, C_2, C_3,\) and \(C_4\) represent the clauses \((c_i \equiv [b_1 \oplus b_2]), (c_i \equiv [b_2 \oplus b_3]), (c_i \equiv [b_3 \oplus b_4]),\) and \((c_i \equiv [b_4 \oplus b_1])\), respectively.
variable \( v'_i \), introduce the two edges \((c_i, v'_i)\) and \((v'_i, p_j)\), and replace the occurrence of \( v_i \) in clause \( p_j \) with an occurrence of \( v'_i \). Now consider an edge \( e = (c_i, c_j) \) where \( c_i \) and \( c_j \) are crossover nodes. Without loss of generality, suppose \( i < j \). Delete \((c_i, c_j)\), introduce the new variable \( y'_i \), and introduce the two edges \((c_i, y'_j)\) and \((y'_i, c_j)\).

The resulting graph is bipartite where the first set of nodes consists of the variable nodes and the second set of nodes consists of the clause nodes and the crossover nodes. (Thus, each edge is between a variable node and a clause node or between a variable node and a crossover node.) Also, each crossover node has four distinct variables as neighbors. We now replace each crossover node with the crossover box outlined above locally replaces each crossover by new clauses consisting of local variables. With this following argument, \( \forall i \), Without loss of generality, suppose \( \forall i \), \( z_i \) are given distinct names in each replacement. Here, \( b_1, b_2, b_3, \) and \( b_4 \) are identified with the neighbors of the crossover node in cyclic order in the layout. The set of clauses used to replace each crossover node are given by \((z_i \equiv [b_1 \oplus b_2]), (z_i \equiv [b_2 \oplus b_3]), (z_i \equiv [b_1 \oplus b_4]), (z_i \equiv [b_4 \oplus b_1])\). Here, \( \oplus \) denotes exclusive or and \( \equiv \) denotes logical equivalence. The resulting \((0,0)-\text{MAX-SAT}(S)\) formula \( F' \) is a conjunction of all the newly added clauses and the old clauses in which some variables are replaced by new variables. The set \( S \) of relations is given by \( S = \{R_1(x_1, x_2, x_3), R_2(x_1, x_2, x_3)\} \), where \( R_1(x_1, x_2, x_3) = 1 \) if and only if \((x_1 \vee x_2 \vee x_3)\) is true and \( R_2(x_1, x_2, x_3) = 1 \) if and only if \((x_1 \equiv (x_2 \oplus x_3))\) is true. Observe that given the layout, the reduction outlined above locally replaces each crossover by new clauses consisting of local variables. With this observation, it is easy to see that the above transformation can be carried out in \text{NC}.

We now show that this reduction is an \text{L}-reduction. Since the number of crossover nodes is no more than \( \delta |V| \), it follows that the number of additional clauses in \( F' \) is \( 4\delta |V| \). Let \( C \) and \( C' \) denote the set of clauses in \( F \) and \( F' \), respectively. We now have the following inequalities:

\[
|C| \geq \frac{1}{2} |V| \\
\text{OPT}(F) \geq \frac{1}{8} |C| \\
\text{OPT}(F') \leq |C'| \leq 4\delta |V| + |C| \leq (12\delta + 1)|C| \leq 8(12\delta + 1)\text{OPT}(F).
\]

The first inequality holds because \( S \) is a finite set of Boolean relations with maximum arity 3. The second inequality holds because as shown in [44], there is an assignment that satisfies at least \( |C|/2^3 \) of the clauses of a \text{MAX-SAT}(S) formula, given that the maximum arity of a relation in \( S \) is 3. The third inequality follows by combining the first two and observing that an optimal solution value for \( F' \) cannot exceed the number of clauses in \( F' \). This completes the first requirement of an \text{L}-reduction.

Next, we prove that the second condition in the definition of \text{L}-reduction holds. To do this we prove the following assertion: Given an assignment to the variables of \( F' \) satisfying \( c' \) clauses, another assignment to the variables of \( F' \) can be obtained satisfying at least \( c' \) clauses of \( F' \) such that all the clauses corresponding to each crossover node are satisfied. Consider any assignment to \( F' \) which satisfies \( c' \) clauses. Let \( c_i \) be a crossover node with neighbors \( b_1, b_2, b_3, b_4 \) in cyclic order. We consider two cases depending on the assignment to the variables in \( F' \) on either side of a crossover node. In the following argument, \( v[x] \) denotes the truth value assigned to variable \( x \).

**Case 1.** \( v[b_1] = v[b_3] \) and \( v[b_2] = v[b_4] \). Thus setting \( z_i = b_1 \oplus b_2 \) results in all four clauses of \( c_i \) being satisfied.

**Case 2.** \( v[b_1] \neq v[b_3] \) or \( v[b_2] \neq v[b_4] \). In this case, it can be seen that no combination of values to the variables \( b_1, b_2, b_3, b_4, \) and \( z_i \), which satisfies the above condition can simultaneously satisfy all the four clauses corresponding to the crossover node. In particular, the number of clauses not satisfied equals the number of unequal pairs. By making a pair unequal, at most one clause of \( F \) can be satisfied incorrectly.

The above discussion implies that it does not pay to assign different values to copies of a given variable in \( F \). This completes the proof of the assertion. The above discussion also implies that there is an optimal assignment to the variables of \( F' \) that satisfies all the clauses introduced to eliminate the crossovers in \( F \). Thus, \( \text{OPT}(F') = \text{OPT}(F) + k \), where \( k \) is the number of clauses introduced to remove the crossovers in the layout of \( F \). Thus from a solution to \( F' \) satisfying \( c' \) clauses, we can obtain a
solution to $F$ satisfying $c$ clauses such that $c = c' - k$, and hence $\text{OPT}(F) - c = \text{OPT}(F') - c'$. Thus, the second condition for $L$-reduction is satisfied with $\beta = 1$.

To complete the proof, we extend the above result in three successive steps.

1. First consider the problems $(\delta, 0)$-MAX SAT$(S)$. The idea is to map instances of this problem into instances of $(0, 0)$-MAX SAT$(S')$, where $S'$ is chosen appropriately. The NC-L-reduction consists of the same sequence of steps carried out in the proof for $(\delta, 0)$-MAX 3SAT. The only difference is that here, $S' = S \cup \{R_1(x_1, x_2, x_3)\}$, where $R_1(x_1, x_2, x_3) = 1$ if and only if $(x_1 \equiv x_2 \oplus x_3)$ is true. The reduction, in conjunction with Theorem 6.1, proves the result.

2. Next, consider the problems $(\delta, 0)$-MAX-RELATION$(S)$. The basic idea is similar to (1) but with two differences: (i) We reduce the problem to an appropriately chosen problem MAX-RELATION$(S)$ restricted to planar instances; these problems have an NCAS by Theorem 6.2. (ii) Each crossover point is now replaced by an appropriate function. As before, let $b_1, b_2, b_3$, and $b_4$ denote in cyclic order the four variables around a given crossover point. We have a new function $q$ such that

$$q(b_1, b_2, b_3, b_4) = \begin{cases} 1 & \text{if } b_1 \neq b_3 \text{ and } b_2 \neq b_4 \\ 0 & \text{otherwise.} \end{cases}$$

Let $F'$ denote the new formula. We now briefly discuss why this constitutes an $L$-reduction. Since the number of crossover nodes is no more than $\delta |V|$, it follows that the number of additional terms in $F'$ is $4\delta |V|$. Let $C'$ denote the number of terms in $F'$. We now have the following inequalities where the constants are derived from the fact that the maximum arity of a relation in $S$ is 4.

$$\text{OPT}(F') \leq |C'| \leq 4\delta |V| + |C| \leq (16\delta + 1)|C| \leq 16(16 \delta + 1)\text{OPT}(F).$$

Thus, the first condition for an $L$-reduction is satisfied. To verify the second condition, note that it does not pay to make a clause in $F$ true at the expense of making one of the terms replacing a crossover point. The rest of the proof is similar to that for $(\delta, 0)$-MAX-3SAT.

3. Finally, consider the problems $(\delta, g)$-MAX-RELATION$(S)$. The proof follows the same sequence of steps as given in (2) above except that we reduce the problems to appropriate problems MAX-RELATION$(S)$ restricted to $(0, g)$-almost planar instances (see Theorem 9.2).

This completes the proof of the theorem. ■

8. APPLICATIONS

8.1. Basic Graph Problems

THEOREM 8.1. For all fixed $g$ and $\delta \geq 0$, the following results hold:

1. There are NCASs for the following MAX SNP-complete graph problems when restricted to $(\delta, g)$-almost planar graphs: MAX-INDEPENDENT SET (MIS), MIN-VERTEX COVER (MVC), MAX-$k$-COLORABLE SUBGRAPH (MCS), and MAX-CUT (MC).

2. There are NCASs for the following MAX SNP-complete problems when restricted to bounded degree $(\delta, g)$-almost-planar graphs: MIN-DOMINATING SET (MDS) and MAX-H-MATCHING (MHM).

Proof. It should be noted that the bounded degree assumption is needed to obtain NCASs for the MDS and MHM problems. (In Theorem 8.2, we give appropriate lower bounds demonstrating the inherent hardness of these problems without the bounded degree assumption.)

For the problems restricted to $(0, g)$-almost planar graphs, our NCASs do not reduce the problem to an appropriate $(\delta , g')$-almost planar MAX-RELATION$(S)$ problem. Instead, for each such problem, a PTAS can be developed as follows. We first carry out NC-BFS and assign level numbers to nodes as indicated in Section 5. Now, for any fixed $k$, the treewidth of the subgraph induced on any $k$ consecutive levels is a constant by Theorem 5.2. Therefore, the problem can be solved optimally in NC for any
subgraph induced on vertices in any \( k \) consecutive levels. As a consequence, we can obtain a PTAS for the problem along the same lines as \( \text{H EU}-(0, 0)-\text{MAX-SAT(S)} \) discussed in Section 6.

We describe the reduction for the \( \text{MAX}-k\text{-COLORABLE SUBGRAPH} \) problem restricted to \( \delta\text{-near-planar} \) graphs. The reductions for the other problems are similar and are therefore omitted. (For details, see [25].) Given an instance \( I(V, E) \) of the \( \text{MAX}-k\text{-COLORABLE SUBGRAPH} \) problem for \( \delta\text{-near-planar} \) graphs, we construct an instance \( I'(U, C) \) of \((\delta, 0)-\text{MAX SAT(S)}_{\text{MCS}} \). Note that \( k \), the number of colors, is fixed in this problem. Associated with each vertex in \( I \) there are \( k \) distinct variables in \( I' \). Each variable represents a possible color that can be assigned to the node. Thus \( U = \{ u^1_i, \ldots, u^k_i \mid v_i \in V \} \). The set \( S_{\text{MCS}} \) consists of one relation \( R \) of arity \( 2k \). Since \( k \) is fixed, the arity of the relation is also fixed. Each edge \((v_i, v_j)\) in \( E \) is replaced by the clause \( C_{ij} \), where \( C_{ij} \) is of the form \( R(u^1_i, \ldots, u^k_i, u^1_j, \ldots, u^k_j) \). We will describe the relation \( R \) in terms of a conjunction of \( \text{CNF} \) clauses; this can be readily transformed into a truth table representation of \( R \). Consider an edge \((v_i, v_j) \in E \). Then

\[
R(u^1_i, \ldots, u^k_i, u^1_j, \ldots, u^k_j) = D_1 \land D_2 \land D_3 \land D_4 \land D_5,
\]

where each \( D_i \) is defined as follows.

\[
D_1 = (u^1_i \lor u^2_i \cdots \lor u^k_i)
\]
\[
D_2 = \bigwedge_{l,m=1, l \neq m}^k (\overline{u^l_i} \lor \overline{u^m_j})
\]
\[
D_3 = (u^1_j \lor u^2_j \cdots \lor u^k_j)
\]
\[
D_4 = \bigwedge_{l,m=1, l \neq m}^k (\overline{u^l_j} \lor \overline{u^m_i})
\]
\[
D_5 = \bigwedge_{r=1}^k (\overline{u^r_i} \lor \overline{u^r_j})
\]

We now explain the meaning of each \( D_i \). \( D_1 \) represents the condition that at least one color is assigned to \( v_i \). \( D_2 \) represents the condition that at most one color is assigned to \( v_j \). \( D_3 \) and \( D_4 \) are the same as \( D_1 \) and \( D_2 \) except that they represent conditions concerning the variables corresponding to \( v_j \). \( D_5 \) represents the condition that the same color is not assigned to \( v_i \) and \( v_j \). Thus \( R \) is true if and only if \( v_i \) and \( v_j \) are assigned exactly one color each and the colors assigned to \( v_i \) and \( v_j \) are different. Thus if \( C_{ij} \) is true, then the edge \((v_i, v_j)\) will be in the \( k\)-colorable subgraph chosen. The formula \( I' \) can now be given as

\[
C = \bigwedge_{(v_i, v_j) \in E} C_{ij}.
\]

Figure 4 depicts the schematic diagram of the reduction. Observe that each crossover in \( I \) is now replaced by \( k^2 \) crossovers in \( I' \). Therefore, \( \delta(I') = k^2 \delta(I) \), and the bipartite graph of \( I' \) is \( k^2\delta\text{-near-planar} \).

We now prove that the reduction is an L-reduction. The first condition follows by observing that the maximum number of relations satisfied in \( I' \) is equal to the number of edges in the maximum \( k\)-colorable subgraph; i.e., \( |\text{OPT}(I)| = |\text{OPT}(I')| \).

Next, given an assignment to \( I' \) which satisfies \( c \) clauses, we can obtain a \( k \)-coloring of \( I \) whose \( k\)-colorable subgraph at least has \( c \) edges by coloring the vertices which appear in a satisfied clause with the same color and coloring other vertices arbitrarily. Such an assignment would guarantee that the edges corresponding to the satisfied clauses would be in the \( k\)-colorable subgraph. Thus \( |A(I)| \geq |A(I')| \). Hence, \( |\text{OPT}(I)| - |A(I)| \leq |\text{OPT}(I')| - |A(I')| \). This proves the second condition of an L-reduction with \( \beta = 1 \). \( \blacksquare \)

We now show that in general the above results cannot be extended significantly.
For any fixed \( \delta > 0 \), the following hold:

1. Unless \( P = \text{NP} \), for any fixed \( \epsilon > 0 \), there is no polynomial time approximation algorithm that provides a solution with a performance guarantee better than \((1 - \epsilon)\ln n\) for the \textsc{Min-Dominating Set} problem restricted to \( \delta \)-near-planar graphs.

2. Unless \( P = \text{NP} \), the problems \textsc{Max-Triangle Matching} and \textsc{Max-H-Matching} do not have \textsc{PTAS} when restricted to \( \delta \)-near-planar graphs.

**Proof.** The intuition behind the proof of Theorem 8.2 is that an arbitrary graph can be made \( \delta \)-near-planar by “padding” in an approximation preserving fashion. We now present the details of the padding arguments.

**\textsc{Min-Dominating Set}:** Suppose there is an approximation algorithm \( \mathcal{A} \) that provides a performance guarantee of \( \rho < (1 - \epsilon)\ln n \) for some fixed \( \epsilon > 0 \) for the \textsc{Min-Dominating Set} problem for \( \delta \)-near-planar graphs. We will show that \( \mathcal{A} \) can be used to derive an approximation algorithm with a performance guarantee of \((1 - \epsilon')\ln n \) for some \( \epsilon' > 0 \) for the \textsc{Min-Dominating Set} problem for arbitrary graphs. In view of the nonapproximability results for \textsc{Min-Dominating Set} given in [6, 47], the required result will follow.

Consider an arbitrary graph \( G \). Let \( G' = (V \cup C, E') \) be a layout of \( G \) with \( |C| \) crossovers. The padded graph \( G'' \) can be obtained as follows: \( G'' = (V \cup V', E \cup E') \) where \( V' \) is the set of vertices in a star graph with one vertex \( v_0 \in V' \) adjacent to all the other vertices in \( V' \), and \(|V'| = \lceil \frac{1}{\delta} |C| \rceil \). Thus, \( E' = \{u, v_0 \mid u \in V' - \{v_0\}\} \). The graph \( G'' \) has \(|V'| + \lceil \frac{1}{\delta} |C| \rceil \) vertices and \( |C| \) crossovers. Therefore, \( G'' \) is \( \delta \)-near-planar.

It can be seen that any minimum dominating set for \( G'' \) consists of a minimum dominating set for \( G \) plus the vertex \( v_0 \in V' \). Thus \(|\text{OPT}(G'')| = |\text{OPT}(G)| + 1 \). We may assume without loss of generality that \(|\text{OPT}(G)| \geq 1/\epsilon \) (since the condition “\(|\text{OPT}(G)| < 1/\epsilon \)” can be checked in polynomial time). Thus, \(|\text{OPT}(G'')| \leq (1 + \epsilon) \text{OPT}(G)\).

Let \( \mathcal{A}(G'') \) be the solution produced by \( \mathcal{A} \) for \( G'' \). Thus, \(|\mathcal{A}(G'')| \leq \rho |\text{OPT}(G'')| \). From \( \mathcal{A}(G'') \) we can construct a dominating set \( \text{App}(G) \) for \( G \) by simply deleting the vertices in \( V' \). Thus, \(|\text{App}(G)| < |\mathcal{A}(G'')| \). Therefore,

\[ |\text{App}(G)| < |\mathcal{A}(G'')| \leq \rho |\text{OPT}(G'')| \leq (1 + \epsilon)\rho |\text{OPT}(G)|. \]

In other words, given an approximation algorithm with performance guarantee \( \rho \) for the \textsc{MDS} problem for \( \delta \)-near-planar graphs, we can obtain an approximation algorithm with a performance guarantee of \((1 + \epsilon)\rho < (1 - \epsilon^2)\ln n\) for the \textsc{MDS} problem for arbitrary graphs. The proof now follows from the known lower bound on the approximability of \textsc{MDS} [6, 47].

**\textsc{Max-Triangle Matching}:** We use a very similar construction. That is, given \( G(V, E) \), we construct \( G'' \) by starting with an arbitrary layout for \( G \) with \(|C| \) crossovers and adding a star graph with \( \lceil \frac{1}{\delta} |C| \rceil \) nodes. Observe that optimal solutions for \( G \) and \( G'' \) are identical, because the construction does not add any triangles. Furthermore, it can be seen that any approximate solution \( \text{App}(G'') \) to the
maximum triangle matching problem for $G''$ can be used as an approximation $\text{App}(G)$ for $G$; that is, $|\text{App}(G'')| \leq |\text{App}(G)|$. The result follows.

**Max-H-Matching:** Follows from the hardness of Max-Triangle Matching.

### 8.2. D2-Problems

Given a problem $\Pi$, D2-$\Pi$ is the problem of solving $\Pi$ in the square of the given graph $G$. Let $\Pi$ be one of the problems: D2-Max-Independent Set, D2-Min-Dominating Set, D2-Max-$k$-Colorable Subgraph, D2-Max-Cut, and D2-Max-H-Matching. Our technique for approximating these problems is derived from the ideas discussed in the previous sections along with the following observations.

1. If a graph $G$ has maximum degree $\Delta$ and treewidth $k$, then $G^2$ (the square of $G$) has treewidth no more than $\Delta k$. Thus for graphs of bounded degree and bounded treewidth, the square graph is also of bounded treewidth.

2. If the graph is of bounded degree and $(\delta, g)$-almost planar, then for each of the problems $\Pi$ the transformation to an appropriate Max-Relation(S) increases $\delta$ by only a constant factor and preserves the genus $g$. (See Lemma A.2 in the Appendix.)

The following theorem is a direct consequence of the above observations and the results presented in previous sections.

**Theorem 8.3.** For all fixed $\delta$ and $g \geq 0$, there are NCASs for the following problems when restricted to bounded degree $(\delta, g)$-almost planar graphs: D2-Max-Independent Set, D2-Min-Dominating Set, D2-Max-$k$-Colorable Subgraph, D2-Max-Cut, and D2-Max-H-Matching.

### 8.3. NC-Approximation Scheme for (0, 0)-Max-An-3Sat

**Heu-Max-Pl-An-3Sat:**

**Input:** An instance $F$ of (0, 0)-Max-An-3Sat problem.

1. **repeat** Steps (a) and (b) below until no single literal clause is left in $F$:
   
   (a) Pick a single literal clause $\bar{v}$ and delete it. (**Remark:** The corresponding variable must be set to false in any valid assignment.)

   (b) Update the remaining clauses in $F$ by removing the literal $\bar{v}$ from every clause in which $\bar{v}$ occurs.

2. Let $F' = (V', C')$ denote the formula obtained at the end of Step 1.

3. Construct $F'' = (V', C' \cup C'')$ where $C'' = \{v_i | v_i \in V\}$. (**Remark:** $F''$ is planar—see proof of Theorem 8.4.)

4. Find a near optimal satisfying assignment for $F''$ using the NCAS outlined in Heu-Pl-Max SAT(S). Let $T$ denote the set of variables set to true.

5. The required solution is the assignment which sets all the variables in $T$ to true and all other variables to false.

**Output:** A near optimal solution for $F$.

The problem Maximum All Negated 3Sat (Max-An-3Sat) is the following: Given a 3CNF formula $F$ in which each clause contains only negated literals, find an assignment which satisfies all the clauses in $F$ and maximizes the number of variables assigned the value true. The generalization to $k$-CNF formulas is denoted by Max-An-$k$-SAT. Observe that there is a trivial assignment (namely, setting each variable to false) that satisfies all the clauses in $F$. Hence, it is not “hard” to find a feasible solution to the problem. Zuckerman [58] has shown that, in general, Max-An-$k$-SAT cannot be approximated to within a factor $n^{\epsilon}$ for any $\epsilon > 0$, unless $P=NP$. In contrast, our next theorem shows that for each fixed $k \geq 3$ and $\delta \geq 0$, $(\delta, g)$-Max-An-$k$-SAT has an NCAS.

4 Given a graph $G(V, E)$, the square graph $G^2(V', E')$ is obtained by adding an edge between two nodes $x$ and $y$ whenever there is a path of length at most 2 between $x$ and $y$ in $G$. $G^2$ does not include self-loops or multi-edges.
Theorem 8.4. For each fixed $k \geq 3$ and $\delta, g \geq 0$, there are $\text{NC}$-approximation schemes for the problems $(\delta, g)$-$\text{MAX-AN-kSAT}$.

Proof. We first prove the theorem for the problem $(0, 0)$-$\text{MAX-AN-3SAT}$. Consider a formula $F(V, C)$ which is an instance of $(0, 0)$-$\text{MAX-AN-3SAT}$. Assume without loss of generality that each variable occurs in some clause $C$. Our approximation scheme is shown as Heu-MAX-PI-AN-3SAT. By an inspection of the algorithm, it can be seen that any valid assignment must assign the value false to all the variables deleted from $F$ to obtain $F'$. Let $C'$ denote the set of clauses in $F'$. Let $\text{OPT}(F)$ and $\text{OPT}(F')$ denote optimal solutions for $F$ and $F'$, respectively. Then the above discussion implies that $|\text{OPT}(F')| = |\text{OPT}(F)|$. We can therefore focus on the second part of the algorithm.

Using the fact that $F'$ is a planar 3CNF formula, we first argue that the interaction graph of $F'$, namely $IG(F')$, is also planar. To see this, consider any planar layout of the bipartite graph $BG(F')$. Let $c$ be an arbitrary clause of $F'$ and let $x_1, x_2$, and $x_3$ denote the three variables appearing in $c$. A planar layout for $IG(F')$ can be obtained as follows. The edge $\{x_1, x_2\}$ in $IG(F')$ can be laid out along the two-edge path $x_1 - c - x_2$ in $BG(F')$. Similarly, the edges $\{x_2, x_3\}$ and $\{x_1, x_3\}$ in $IG(F')$ can be laid out along the two-edge paths $x_2 - c - x_3$ and $x_1 - c - x_3$, respectively, in $BG(F')$. Clearly, this procedure does not introduce any edge crossovers. So, $IG(F')$ is also planar.

Since $IG(F')$ is planar, the size of a maximum independent set in $IG(F')$ is at least $|V'|/4$. We claim that $|\text{OPT}(F')| \geq |V'|/4$. To see this, note that two variables occurring in the same clause cannot simultaneously belong to an independent set in $IG(F')$. Thus, even after setting the variables in the independent set of $IG(F')$ to true, an assignment that satisfies all the clauses in $F'$ can be obtained. Since $F'$ and $IG(F') = (V', E')$ are planar, we have the following inequalities.

\[|C'| \leq |E'|, \quad |E'| \leq 3|V'| \quad \text{and} \quad |C'| \leq 3|V'|.\]

The first inequality is true since each clause in $F'$ contributes at least one edge to $IG(F')$. The second inequality follows by Euler's formula (see Appendix) for planar graphs. The third inequality follows immediately from the first two inequalities. Thus, we have

\[|\text{OPT}(F'')| \leq |C''| + |C'| \leq |V'| + |C'| \leq 4 |V'| \leq 16 |\text{OPT}(F')|.

In other words, the first condition of an $L$-reduction is satisfied. To see why the second condition for an $L$-reduction is satisfied, consider an optimal solution $\text{OPT}(F'')$ to $F''$. It is easy to verify that $|\text{OPT}(F'')| = |\text{OPT}(F')| + |C'|$. Next, consider a solution $A(F'')$ for $F''$. We can obtain a solution $A(F')$ for $F'$ that satisfies $A(F'') - |C'|$ clauses. Thus $|A(F')| - |\text{OPT}(F')| = |A(F'')| - |C'| - |\text{OPT}(F'')| + |C'|$ or $|A(F')| - |\text{OPT}(F')| = |A(F'')| - |\text{OPT}(F'')|$. Thus, the reduction satisfies the second condition for an $L$-reduction. Since $(0, 0)$-$\text{MAX-3SAT}$ has an $\text{NC}$-approximation scheme, it follows that $(0, 0)$-$\text{MAX-AN-3SAT}$ also has an $\text{NC}$-approximation scheme.

The ideas discussed above can be easily extended to yield $\text{NCAs}$ for $(0, 0)$-$\text{MAX-AN-kSAT}$ and $(\delta, g)$-$\text{MAX-AN-kSAT}$ for any fixed $k$ and $\delta, g \geq 0$. The details are therefore omitted. \qed

8.4. Max-$k$-Set Splitting

The Max-$k$-Set Splitting problem is the following: Given a collection $C$ of subsets of a finite set $S$, where each subset has size $k$, find a partition of $S$ into two disjoint sets $S_1$ and $S_2$ such that the number of sets in $C$ that are not contained entirely in $S_1$ or $S_2$ is maximized. The bipartite graph of a $k$-Set Splitting instance $I$ (denoted by $BG(I)$) is defined as follows. The subsets in $C$ and the elements in $S$ are in one to one correspondence with the vertices of the graph. There is an edge between a set node $c \in C$ and an element node $s \in S$ if and only if $s \in c$. The problem $(0, 0)$-$\text{Max-k-set splitting}$ is the restriction of the problem Max-$k$-Set Splitting to instances whose bipartite graphs are planar. As pointed out in [14], Max-$k$-Set Splitting is equivalent to the Max-Cut problem for hypergraphs, where each hyperedge has arity $k$. Combining this fact with our result in Theorem 9.1 for the Max-Cut problem for planar hypergraphs, it is seen that $(0, 0)$-$\text{Max-k-set splitting}$ has an $\text{NC}$ for any fixed $k$. The result immediately extends to the $(0, g)$-$\text{Max-k-set splitting}$ problem. This result should be contrasted with the following two results from the literature.

PARALLEL APPROXIMATION SCHEMES 55
1. Kann et al. [30] show that the general version of \textsc{Max-k-Set Splitting} can be approximated to within \((1 - 2^{1/k})\) of the optimal value. Furthermore, they show that unless \(P = \text{NP}\), \textsc{Max-k-Set Splitting} does not have a \text{PTAS}.

2. As shown by Arora et al. [4], \textsc{Max-k-Set Splitting} has a \text{PTAS} when restricted to instances for which \(|C| = \Theta(|S|^k)|\). Note that for the \((\delta, g)\)-almost planar instances considered here, \(|C| = \Theta(|S|)\).

8.5. \((\delta, g)\)-\textsc{B-Min-k-Sat}

As another application, we consider the minimum satisfiability problem (\textsc{Min-Sat}): Given a set \(C = \{c_1, c_2, \ldots, c_m\}\) of \(m\) clauses made up of uncomplemented and complemented occurrences of variables from the set \(X = \{x_1, x_2, \ldots, x_n\}\), find a truth assignment to the variables that satisfies the minimum number of clauses. The problem \textsc{B-Min-k-Sat} is the restriction of the problem \textsc{Min-Sat} in which each clause contains at most \(k\) literals and each variable occurs in at most \(B\) clauses.

The \textsc{Min-Sat} problem was introduced in [35]. In [22, 41] it is shown that Min-Sat can be approximated to within a factor of 2. These references also show that Min-Sat is as hard to approximate as \textsc{Min-Vertex Cover}. Let \(I\) be an instance of \textsc{Min-Sat} consisting of the clause set \(C_I\) and variable set \(X_I\). The auxiliary graph \(G_I(V_I, E_I)\) corresponding to \(I\) is constructed as follows. The node set \(V_I\) is in one to one correspondence with the clause set \(C_I\). For any two nodes \(v_i, v_j\) in \(V_I\), the edge \(\{v_i, v_j\}\) is in \(E_I\) if and only if the corresponding clauses \(c_i, c_j\) are such that there is a variable \(x \in X_I\) that appears in uncomplemented form in \(c_i\) and in complemented form in \(c_j\). We also recall a lemma from [41].

**Lemma 8.1** [41]. Let \(I\) be an instance of \textsc{Min-Sat} with clause set \(C_I\) and let \(G_I\) be the corresponding auxiliary graph.

1. Given any truth assignment for which the number of satisfied clauses in \(I\) is equal to \(k\), we can find a vertex cover of size \(k\) for \(G_I\).

2. Given any vertex cover \(C'\) of size \(k\) for \(G_I\), we can find a truth assignment that satisfies at most \(k\) clauses of the \textsc{Min-Sat} instance \(I\).

Thus, an approximate vertex cover of size \(C\) in the auxiliary graph yields a solution that satisfies \(C\) clauses of the \textsc{Min-Sat} instance.

We now explain how for any fixed integer \(k\), an \textsc{NCAS} for \((0, 0)\)-\textsc{B-Min-k-Sat} can be obtained. Let \(I\) be an instance of \((0, 0)\)-\textsc{B-Min-k-Sat}. It can be shown that the auxiliary graph for \(I\) is \(\delta\)-near-planar, with \(\delta \leq (Bk)^2\). To see this, observe that each node in the auxiliary graph is adjacent to at most \(Bk\) other clauses. Also note that the maximum degree of a node in the auxiliary graph is bounded by \(Bk\). Thus, combining our result in Theorem 8.1 for the vertex cover problem for \(\delta\)-near-planar graphs, the result in Lemma 8.1, and the above discussion, we obtain that the problem \((0, 0)\)-\textsc{B-Min-k-Sat} has an \textsc{NCAS}.

The result immediately extends to \((\delta, g)\)-\textsc{B-Min-k-Sat}. This result should be contrasted with the fact that in general, \textsc{B-Min-k-Sat} is \textsc{Max-SNP}-hard when \(k\) is not fixed [22, 41].

9. OTHER RESULTS

9.1. Planar Hypergraphs

In this section, we present the extensions of our results in Theorem 8.1 to problems restricted to planar hypergraphs with hyperedges of bounded arity. We first recall the following additional definitions from [39].

**Definition 9.1.** A hypergraph \(H(V, E)\) consists of a collection of vertices \(V\) and hyperedges \(E\). Each hyperedge is a nonempty subset of \(V\).

Given a hypergraph \(H(V, E)\), the associated bipartite graph \(G_H(V_1, E_1)\) is defined as follows: The vertices in \(V_1\) are in one to one correspondence with hyperedges and vertices in \(H\). There is an edge between a hyperedge node and a vertex node if and only if the vertex appears in the hyperedge.

A hypergraph \(H\) is said to be planar if and only if its associated bipartite graph \(G_H\) is planar.
Given a hypergraph $H(V, E)$, the degree of a vertex $x \in V$ is the number of hyperedges containing $x$. The arity of a hyperedge $e \in E$ is the cardinality of $e$. We also note that the definition of MAX-INDEPENDENT SET, MIN-DOMINATING SET, MIN-VERTEX COVER, and MAX-CUT for graphs extend in a straightforward manner to apply to hypergraphs [39]. In the remainder of this section, for problems restricted to bounded arity hyperedges we use $B_e$ to denote the bound on the arity of the hyperedges. Similarly, for problems restricted to bounded degree hypergraphs, we let $B_v$ denote the bound on the degree of the vertices.

**Theorem 9.1.** There are NC-approximation schemes for the following problems when restricted to planar hypergraphs in which each hyperedge has bounded arity: bounded degree MAX-INDEPENDENT SET, bounded degree MIN-VERTEX COVER, bounded degree MIN-DOMINATING SET, MAX-CUT, and MAX-k-COLORABLE SUBGRAPH.

**Proof.** For each of the problems $\Pi$ stated in the theorem, we give an NC-L-reduction to an appropriate $(\delta, 0)$-MAX SAT($S_\Pi$), thereby showing that these problems have NC-approximation schemes. We assume that the planar hypergraph is specified as a bipartite graph, since it is straightforward to construct the bipartite graph in NC when the hypergraph is given as a collection of sets. For the sake of brevity we will describe our proof in detail only for one of the problems. The reductions for the other problems can be done in a similar manner.

For the remainder of this proof, we adopt the following convention. Let $H$ be the instance of the problem we start with and let $H'$ denote the satisfiability instance obtained through the reduction. Let $OPT(H')$ and $A(H')$ denote respectively an optimal solution and an approximate solution for $H'$. Similarly, let $OPT(H)$ and $A(H)$ respectively denote an optimal solution for $H$ and the approximate solution for $H$ obtained by transforming the solution $A(H')$ to a solution for $H$. We sometimes use the same symbol for a set and its size, and the intended meaning will be clear from the context. We also use $\delta(H)$ and $\delta(H')$ to denote the number of crossovers in the layouts of $H$ and $H'$, respectively.

**MAX-INDEPENDENT SET:** Given an instance $H(V, E)$ of MAX-INDEPENDENT SET problem for planar hypergraphs having bounded degree and bounded arity hyperedges, with $V = \{v_1, \ldots, v_n\}$ and $|E| = m$, we construct an instance $H'(U, C)$ of $(\delta, 0)$-MAX SAT($S$) as follows. The variables $U = \{u_1, \ldots, u_n\}$ are in one to one correspondence with the vertices in $V$. The clause set $C = C_1 \cup C_2$ where $C_1$ and $C_2$ are defined as follows. The set $C_2 = \{u_i \mid v_i \in V\}$. The intuition is that a variable will be true if its corresponding vertex is in the independent set. We now describe how the clause set $C_1$ is constructed. Without loss of generality, we assume that each hyperedge has exactly $B_v$ nodes. (Otherwise, we can add copies of one of the nodes to the hyperedge.) Consider a hyperedge $e_i = \{v_{i1}, v_{i2}, \ldots, v_{iB_v}\}$. Corresponding to this hyperedge, we produce a subformula $F_i$ (consisting of $B_v$ clauses) given by

$$F_i(u_{i1}', u_{i2}', \ldots, u_{iB_v}') = (u_{i1}' \Rightarrow (u_{i2}' \land u_{i3}' \land \cdots \land u_{iB_v}')) \land (u_{i2}' \Rightarrow (u_{i1}' \land u_{i3}' \land \cdots \land u_{iB_v}'')) \land \cdots \land (u_{iB_v}' \Rightarrow (u_{i1}' \land u_{i2}' \land \cdots \land u_{iB_v-1}'))$$

Intuitively, when $F_i(u_{i1}', u_{i2}', \ldots, u_{iB_v}')$ is true, at most one of the vertices in the hyperedge $e_{i1}$ is included in the independent set. The clause set $C_1$ is the union of the clauses created for each hyperedge. Note that the set $S$ has only two relations, one of arity 1 (for clauses in $C_2$) and the other of arity $B_v$ (for clauses in $C_1$). It can be seen that each variable appears in $B_vB_v$ clauses of $C_2$. Therefore, the number of crossovers in the bipartite graph associated with $H'$ is bounded by $(B_vB_v)^2|U|$. In other words, the resulting formula is $\delta$-near-planar, where $\delta = (B_vB_v)^2$.

We now argue that the reduction is indeed an L-reduction. Using the facts that each node occurs in at most $B_v$ hyperedges and that each hyperedge has arity at most $B_v$, it can be verified that $|OPT(H)| \geq |V|/(B_vB_v - 1)$ and $\delta = (B_vB_v)^2$. Also, since each node appears in at most $B_v$ hyperedges, the total number of hyperedges in $H$ is at most $B_v|V|$. Each hyperedge leads to $B_v$ clauses in $C_2$. Therefore,
\[ |C_2| \leq B_e |V| \text{.} \] Obviously, \(|C_1| = |V|\). Thus,

\[
\begin{align*}
|\text{OPT}(H')| & \leq |C_1| + |C_2| \\
& \leq (B_e B_e + 1)|V| \\
& \leq (B_e B_e + 1)(B_e (B_e - 1) + 1)\text{OPT}(H)|.
\end{align*}
\]

This verifies the first condition of an L-reduction. We now verify the second condition in the definition of an L-reduction. First, consider an optimal solution \(\text{OPT}(H)\) for \(H\). Then, an optimal solution \(\text{OPT}(H')\) for \(H'\) can be obtained by setting all the variables corresponding to the vertices in the independent set to true. The remaining variables are set to false. Such an assignment satisfies \(|\text{OPT}(H')|\) clauses in \(C_2\) and all the clauses in \(C_1\). Therefore, the value of an optimal solution \(|\text{OPT}(H')|\) for \(H'\) satisfies the condition \(|\text{OPT}(H')| \geq |\text{OPT}(H)| + |C_1|\). Conversely, consider a solution that satisfies \(|\text{OPT}(H')|\) clauses of \(H'\). Let this solution satisfy \(t_1\) clauses from \(C_1\) and \(t_2\) clauses from \(C_2\). Thus, \(|\text{OPT}(H')| = t_1 + t_2\).

This solution can be modified so that the new solution satisfies all the clauses in \(C_1\) in the following manner. Consider each unsatisfied clause in \(C_1\) and satisfy it by setting the variable to the left of the implication to false. Each time this is done, we gain a clause from \(C_1\) but lose a clause from \(C_2\). Thus, at the end, the modified solution will also satisfy \(|\text{OPT}(H')|\) clauses. In obtaining the modified solution, \(|C_1| - t_1\) variables were set to false. Thus, the number of variables that are true in the modified solution is equal to \(t_2 - (|C_1| - t_1) = t_1 + t_2 - |C_1| = |\text{OPT}(H')| - |C_1|\). The transformation \(g\) consists merely of selecting those vertices for which the corresponding variable in the modified solution is true. Therefore, the resulting solution value \(A(H)\) for \(H'\) satisfies the condition \(A(H) = A(H') - |C_1|\). As already observed, \(|\text{OPT}(H')| \geq |\text{OPT}(H)| + |C_1|\). Adding the last two inequalities and rearranging terms, we get \(\text{OPT}(H) - |A(H)| \leq |\text{OPT}(H')| - |A(H')|\). This verifies the second condition for an L-reduction with \(\beta = 1\). ■

### 9.2. Approximating \((0, g)-\text{MAX-SAT}\)

Theorem 6.1 does not immediately enable us to obtain an NCAS for \((0, g)-\text{MAX-SAT}\). This is because each clause in a \((0, g)\)-almost planar instance could potentially contain \(\Theta(n)\) literals. Thus, even though the bipartite graph corresponding to the \((0, g)\)-almost planar instance of MAX SAT has bounded treewidth, the interaction graph may have a clique of size \(\Theta(n)\) and hence be of treewidth \(\Theta(n)\). However, we can develop an NCAS for a restricted version of \((0, g)-\text{MAX-SAT}\), as shown below.

**Theorem 9.2.** For any constant \(c > 0\), the problem \((0, g)-\text{MAX-SAT}\) has an NC-approximation scheme if the instance has the property that \(|C| \geq c|V|\) where \(C\) is the clause set and \(V\) is the variable set.

**Proof.** For ease of exposition, we first give the proof for \((0, 0)-\text{MAX-SAT}\) and then indicate the necessary modifications for \((0, g)-\text{MAX-SAT}\).

We give an NC-L-reduction from \((0, 0)-\text{MAX-SAT}\) to \((0, 0)-\text{MAX-SAT}(S)\), where \(S\) will be specified subsequently. Given the planar layout of the instance \(I(C, V) = C_1 \land C_2 \land \ldots \land C_m\) of \((0, 0)-\text{MAX-SAT}(S)\), we replace each clause \(C_i = (l_1 \lor l_2 \ldots \lor l_p)\) by the following set of clauses \(C_i^j:\)

\[
C_i^j = [(l_1 \lor l_2) \equiv y_i^j] \land \left[(y_i^j \lor l_3) \equiv y_i^j_3\right] \land \cdots \left[(y_i^j \lor l_p) \equiv y_i^j_{p-1}\right] \land y_i^j_p.
\]

Here, “≡” denotes logical equivalence. It is easy to see that the clauses can be laid out so as to preserve planarity. The set \(S\) consists of a relation \(R_1(x, y, z)\) of the form \((x \lor y) \equiv z\) (i.e., \(R_1(x, y, z)\) is true if and only if \((x \lor y) \equiv z\) is true) and a relation \(R_2(x)\) of the form \([x]\) (i.e., \(R_2(x)\) is true if and only if \(x\) is true). The instance \(F(I)\) of \((0, 0)-\text{MAX-SAT}(S)\) is \(C = \bigwedge_{i=1}^{m} C_i\). Let \(G_1(V_1, E_1)\) denote the bipartite graph associated with \(I\).

Since the reduction consists of a local replacement of the original set of clauses, we can do the transformation in parallel for each clause. Thus the reduction can be carried out in NC. In the remaining part of the proof, we show that the reduction is an L-reduction. The total number of clauses \(|C|\) in \(F(I)\) is \(\sum_{i=1}^{m} p_i\). Observe that the number of clauses in \(F(I)\) is no more than the total number of edges in the bipartite graph associated with \(C\). By Euler’s formula (see Appendix), the number of edges in a
planar graph is no more than three times the number of nodes in the graph. Thus, we have the following inequalities.

\[ |C| \leq |E_1| \]
\[ |E_1| \leq 3(|V| + |C|) \]
\[ \text{OPT}(F(I)) \leq |C| \leq |E_1| \leq 3(|V| + |C|) \]

It is well known that for any instance of SAT, there is always a truth assignment that satisfies at least half the total number of clauses. Thus, \( \text{OPT}(I) \geq \frac{|C|}{2} \). Now, using the fact that \( |C| \geq c|V| \), we get

\[ \text{OPT}(F(I)) \leq \alpha \text{OPT}(I), \]

where \( \alpha = 6(1 + \frac{1}{c}) \). This proves the first condition of an L-reduction. To prove the second condition, observe that the auxiliary variables \( y_1, \ldots, y_{n-1} \) are functionally dependent on the literals \( l_1, \ldots, l_p \). Therefore, it can be seen that any assignment of values to the variables \( l_1, \ldots, l_p \) can be extended so as to satisfy all but one clause in \( C_i \). Moreover, \( C_i \) is true if and only if the original clause \( C_i \) is true. Using these facts it follows that the second condition in the definition of L-reduction is satisfied with \( \beta = 1 \).

For \((0, g)\)-MAX-SAT, the NC-L-reduction is identical to the above except that the value of \( \alpha \) associated with the reduction is different. For a graph of genus \( g \), the number of edges is at most \( 3|V| + 6g \) (see Lemma A.1 in the Appendix). Using this fact, it can be seen that the value of \( \alpha \) is \( 6(1 + 1/c + g) \). The value of \( \beta \) remains 1. The result of the L-reduction is an instance of \((0, g)\)-MAX-SAT(S), which has a PTAS by Theorem 6.2.

Remark 1. Observe that the reduction was done to a \((0, 0)\)-MAX-SAT(S) formula that was not a \((0, 0)\)-MAX-3SAT formula. This result points out the following advantages of solving \((0, 0)\)-MAX-SAT(S) problems. First, in many cases, the reduction and its correctness proof are easier to describe. Second, in cases such as the above, if we wanted to convert the resulting formula into an equivalent 3CNF formula, we would destroy planarity.

Remark 2. The restriction on the number of clauses is needed because in an instance of \((0, 0)\)-MAX-SAT some clause may have \( \Theta(n) \) literals. Consequently, even if the number of clauses in the instance \( I \) is small (say, \( O(1) \)), the formula \( F(I) \) may have \( \Theta(n) \) clauses. This would mean that \( |\text{OPT}(F(I))| \) and \( |\text{OPT}(I)| \) are not linearly related and hence the reduction would no longer be an L-reduction. In the preliminary version of this paper [25], we raised the question of devising a PTAS for \((0, 0)\)-MAX-SAT. As mentioned earlier, this question was answered affirmatively in [31, 55]. It is not clear what sort of general sets \( S \) of unbounded arity relations can be handled by the methods presented in these references. Some progress in this direction is reported in [28].

9.3. NC-Approximations for General Graphs

In previous sections, we have demonstrated that optimization versions of various satisfiability problems are useful in devising approximation schemes for problems restricted to \((\delta, g)\)-almost-planar instances. Here, we present some remarks concerning problems for arbitrary instances. First, we note that all the problems that can be expressed as MAX SNP predicates can be reduced in an approximation preserving manner to corresponding instances of MAX-SAT(S). (A similar result was obtained independently in [32].) To see this, let \( \Pi \in \text{MAX SNP} \), where the predicate for \( \Pi \) is of the form \( \exists \phi(\bar{x}, G, S_t) \). Then, there exists a finite set of finite-arity Boolean relations \( R_{\Pi} \) and NC-computable functions \( f \) and \( g \) with the following properties:

1. The function \( f \) maps an input \( G \) of \( \Pi \) to an \( R_{\Pi} \)-formula \( f(G) \) such that, for each structure \( T \), there is an assignment \( v \) of truth-values to the variables of \( f(G) \) for which the number of simultaneously satisfiable terms of \( f(G) \) under \( v \) equals \( |\{\bar{x} | \phi(\bar{x}, G, T)\}| \).

2. The function \( g \) maps an assignment \( v \) of truth-values to the variables of \( f(G) \) to a structure \( T \) such that the number of simultaneously satisfiable terms of \( f(G) \) under \( v \) equals \( |\{\bar{x} | \phi(\bar{x}, G, T)\}| \).
The functions \(f\) and \(g\) can be constructed in a manner very similar to the corresponding functions for the L-reduction from \(\Pi\) to MAX 3SAT given in [44].

A consequence of the above reduction to \text{MAX-SAT}(S) is that every problem in \text{MAX SNP} has an \text{NC}-approximation algorithm with a constant performance guarantee. (This result was obtained independently in [11].) To see how this result follows, let \(S\) be a finite set of finite-arity Boolean relations. Without loss of generality, we may assume that each relation in \(S\) has at least one element. Each instance of the problem \text{MAX-SAT}(S) is a Boolean formula \(F\) of the form

\[
F(x_1, x_2, \ldots, x_n) = \bigwedge_{i=1}^{m} F_i(x_{i_1}, x_{i_2}, \ldots, x_{i_{r_i}})
\]

for each \(F_i \in S\). Let \(k\) be the maximum degree of any of the \(F_i\). Then, each \(F_i\) \((1 \leq i \leq m)\) is a satisfiable Boolean function of at most \(k\) variables, for some constant \(k\) \((\text{i.e., } r_i \leq k \text{ for } 1 \leq i \leq m)\). Given a Boolean formula \(F\) of the above form, a random truth assignment to the variables \(x_1, x_2, \ldots, x_n\) will satisfy on the average at least \(m/2^k\) of the conjuncts [44]. That such an assignment can be obtained in \text{NC} is a direct consequence of Theorem 3.2 in [9]. Thus, there is an \text{NC}-approximation with a performance guarantee of \(2^k\). Recently, an \text{NC}-approximation algorithm with a better performance guarantee was presented in [51].

**APPENDIX: GENUS AND ASSOCIATED DEFINITIONS**

In this Appendix, we provide a brief overview of the concept of graph genus used throughout the paper. We also provide a proof of a result used in Section 8.4. For a more detailed treatment of the concept of genus, the reader is referred to [16, 21, 57].

We consider undirected graphs without multi-edges or self-loops. To define the notion of embedding a graph on a surface, it is first necessary to define Jordan curves. A Jordan curve on a surface is a continuous curve which does not intersect itself. A closed Jordan curve is a Jordan curve whose endpoints coincide. An embedding of a graph \(G\) on a surface is a mapping where each node of \(G\) is mapped to a distinct point on the surface and each edge of \(G\) is mapped to a Jordan curve on the surface such that the two endpoints of the Jordan curve correspond to the two nodes joined by the edge. In such an embedding, an edge crossing is said to occur if one of the following conditions hold:

1. The Jordan curves corresponding to two edges intersect at a point that does not correspond to a node of the graph.
2. A Jordan curve corresponding to edge \(\{u, v\}\) passes through a point that corresponds to a node \(z\) different from \(u\) and \(v\).

When a graph is embedded on a surface with no edge crossings, the faces of the embedding are the boundaries of the connected regions obtained by deleting the embedding of \(G\) from the surface.

As is well known [57], planar graphs are those that can be embedded in the plane or, equivalently, on the surface of a sphere, with no edge crossings. Nonplanar graphs can be embedded without edge crossings on the surface of a sphere with an appropriate number of handles. For example, the graph \(K_5\) (the complete graph on five nodes) and the graph \(K_{3,3}\) (the complete bipartite graph with three nodes on each side of the bipartition) can be embedded without edge crossings on the surface of a sphere with one handle. Such a surface is called a toroid. Both \(K_5\) and \(K_{3,3}\) are nonplanar and so neither of these can be embedded without edge crossings on a sphere. These graphs are said to have a genus of 1. In general, given a graph \(G(V, E)\), the minimum number of handles to be added to a sphere so that the graph \(G\) can be embedded without edge crossings on the resulting surface is called the genus of \(G\). It is known that the genus of a graph is a well-defined parameter [57]. An easy upper bound on the genus of a graph is its crossing number, that is, the minimum number of edge crossings in an embedding of the graph on the surface of a sphere.

Euler’s formula for planar graphs, which relates the number of nodes, edges, and faces of a planar graph, can be extended to graphs of any genus \(g\). The following lemma from [57] indicates this extension.

**Lemma A.1.** Let \(G(V, E)\) be a connected graph of genus \(g\). Let \(f\) denote the number of faces in an embedding of \(G\) on a sphere with \(g\) handles. Then, \(|V| - |E| + f = 2 - 2g\).
For the special case of a planar graph, where \( g = 0 \), the relationship in Lemma A.1 becomes \(|V| - |E| + f = 2\). An easy consequence of this relationship, using the fact that each face is bounded by at least three edges, is that for a planar graph \( G(V, E) \), \(|E| \leq 3|V| - 6\). Lemma A.1 and this fact about planar graphs are used in Sections 8.3 and 9.2.

Recall that for a graph \( G \), the square graph \( G^2 \) is obtained by adding an edge \( \{x, y\} \) whenever there is a path of length at most 2 between \( x \) and \( y \) in \( G \). The next lemma points out a property of degree bounded \((\delta, g)\)-almost planar graphs. This property is used in Section 8.2. In proving this property, we use the following terminology. For any node \( x \), the neighborhood of \( x \), denoted by \( N(x) \), is the set of all nodes \( y \) such that \( G \) contains the edge \( \{x, y\} \). For a node \( x \), the second neighborhood of \( x \), denoted by \( N_2(x) \), is the set of all nodes \( z \) such that there is a path of length exactly two between \( x \) and \( z \) in \( G \).

**Lemma A.2.** Let \( G(V, E) \) be a bounded degree \((\delta, g)\)-almost planar graph. Then, \( G^2(V, E') \) is a \((\delta', g')\)-almost planar graph where \( \delta' = c\delta \) for some constant \( c \).

**Proof.** Let \( \Delta \) denote the (constant) maximum node degree of \( G \). Consider a \((\delta, g)\)-almost planar layout of \( G \). We will show how a \((\delta', g')\)-almost planar layout of \( G^2 \) can be constructed from the layout for \( G \) by adding the necessary edges, and then provide a bound on \( \delta' \).

Consider any node \( v \in V \) and let \( w \in N(v) \). Further, let \( N(w) = \{w_1, w_2, \ldots, w_r\} \); note that \( r \leq \Delta \). The edges \( \{v, w_1\}, \{v, w_2\}, \ldots, \{v, w_r\} \) are all in \( G^2 \). We lay out the edge \( \{v, w_i\} \) along the two-edge path \( v - w_i - w_i' \) in the layout for \( G \), \( 1 \leq i \leq r \). This results in a bundle of \( r \) (partial) edges from \( v \) to \( w \); at \( w \), the bundle separates into individual edges, each such edge going to one of the nodes in \( N(w) \). We repeat this process for each node in \( N(v) \). When all nodes in \( V \) have been so considered, the layout for \( G^2 \) would be complete.

Note that the layout for \( G^2 \) has been created on the same surface as \( G \). So, the genus \( g \) is preserved.

To estimate the number of crossovers in the layout for \( G^2 \), we note that the new crossovers created can be divided into two groups. The first group consists of crossovers between pairs of bundles. The second group consists of crossovers between a new edge and the bundles created. We bound the sizes of each of these groups separately.

To estimate the number of new crossovers between pairs of bundles, consider a vertex \( v \) and let \( w \in N(v) \). As mentioned above, a bundle of at most \( \Delta \) edges has been added around the original edge \( \{v, w\} \). The bundle corresponding to \( \{v, w\} \) may, in the worst case, have crossovers with bundles corresponding to all the edges that had crossovers with \( \{v, w\} \) in the layout for \( G \). For each crossover in \( G \), there are at most \( \Delta^2 \) crossovers due to the corresponding pair of bundles in the layout for \( G^2 \). Since there are at most \( \delta|V| \) crossovers in the layout for \( G \), the total number of crossovers in the first group is at most \( \Delta^2 \delta |V| \).

To estimate the number of new crossovers between new edges and bundles, consider again an arbitrary vertex \( v \) and let \( w \in N(v) \). Further, let \( N(w) = \{w_1, w_2, \ldots, w_r\} \), where \( r \leq \Delta \). The edges \( \{v, w_1\}, \{v, w_2\}, \ldots, \{v, w_r\} \) are all in \( G^2 \). In the layout for \( G^2 \), the edge \( \{v, w_i\} \) may cross, in the worst case, all the edges in the bundle corresponding to \( \{v, w_i\} \) and all the bundles corresponding to \( \{v, w_i\} \), \( 1 \leq i \leq r \). Since the bundle corresponding to \( \{v, w_i\} \) has at most \( \Delta + 1 \) edges (including the edge \( \{v, w_i\} \)), the number of crossovers between the edge \( \{v, w_i\} \) and the bundle corresponding to \( \{v, w_i\} \) is at most \( \Delta + 1 \). Further, the edge \( \{v, w_i\} \) may cross, in the worst case, all the bundles corresponding to each of the edges \( \{v, w_1\}, \{v, w_2\}, \ldots, \{v, w_r\} \). Since each such bundle has at most \( \Delta + 1 \) edges, the total number of crossovers due to \( \{v, w_i\} \) is at most \( \Delta + 1 \). Since \( r \leq \Delta \), the last expression is bounded by \( (\Delta + 1)^2 \). Thus, the number of crossovers between all the edges \( \{v, w_1\}, \{v, w_2\}, \ldots, \{v, w_r\} \) and the bundles is at most \( r(\Delta + 1)^2 \leq \Delta(\Delta + 1)^2 \). The last expression is an upper bound on the number of new crossovers due to the second neighborhood of \( v \) through the neighbor \( w \). Since \( v \) has at most \( \Delta \) neighbors, the number of crossovers due to the new edges from \( v \) to all the nodes in \( N_2(v) \) is at most \( \Delta^2(\Delta + 1)^2 \leq 4\Delta^4 \). Since the number of nodes in \( G \) is \(|V|\), the total number of crossovers in the second group is at most \( 4\Delta^4|V| \).

So, the total number of new crossovers in the layout for \( G^2 \) is at most \((\Delta^2 \delta |V| + 4\Delta^4 |V|)\). Since the layout for \( G \) has at most \( \delta |V| \) crossovers, the total number of crossovers in the layout for \( G^2 \) is at most \( \delta' |V| \), where \( \delta' = (\delta + \Delta^2 + 4\Delta^4) \). Since \( \delta \) and \( \Delta \) are constants, so is \( \delta' \). So, we have a \((\delta', g)\)-almost planar layout for \( G^2 \) and this completes the proof.
REFERENCES


ACKNOWLEDGMENTS

We are grateful to the referees for providing many valuable suggestions that vastly improved the paper. We thank Professor Pierluigi Crescenzi, Professor David Eppstein, Professor Dorit Hochbaum, Riko Jacob, Professor Rajeev Motwani, Dr. Aravind Srinivasan, and Professor Luca Trevisan for several useful discussions, pointers to related literature, and making available some of their papers. We also thank the referees of the 14th FST & TCS conference for their valuable comments on an earlier draft of this paper.