On efficient polynomial-time approximation schemes for problems on planar structures [1]

Liming Cai[2] , Michael Fellows [3] , David Juedes [4], and Frances Rosamond [5]

Abstract

This work examines the existence of efficient polynomial-time approximation schemes (EPTAS) for a variety of problems contained in the syntactic classes Planar TMIN, Planar TMAX, and Planar MPSAT as defined by Khanna and Motwani. Based on the recent work of Alber, Bodlaender, Fernau and Niedermeier and others, we describe subclasses of problems from Planar TMIN, Planar TMAX, and Planar MPSAT that have EPTAS. In contrast, we show that there exist W[1]-hard problems in Planar TMIN, Planar TMAX, and Planar MPSAT. It follows that these problems do not have efficient polynomial-time approximation schemes, unless W[1]=FPT. As our main result, we show that the existence of efficient polynomial-time approximation schemes for certain problems is closely connected their syntactic complexity: all problems that can be described by a collection of first order formulas with minterms of size 1 have efficient polynomial-time approximation schemes; in contrast, there exist W[1]-hard problems that can be described by collections of first order formula with minterms of size 4.

Classification: computational and structural complexity.

1 Introduction

Polynomial-time approximation provides various approaches for producing acceptably good solutions to intractable optimization problems. A general approach is the notion of polynomial-time approximation schemes (PTAS) that provide solutions with cost within $(1+\epsilon)$ of optimal, where the error ϵ can be chosen arbitrarily close to 0 [13]. While many problems do not have polynomial-time approximation schemes unless P=NP, for those that do, approximation is performed by algorithms that provide additional precision at the expense of additional running time. The running time for such algorithms is typically of the form $n^{O(1/\epsilon)}$ or $2^{O(1/\epsilon)}n^c$, where $c > 0$ is a constant. While both time bounds are exponential in $\frac{1}{\epsilon}$, the time bound $n^{O(1/\epsilon)}$ may become intolerable for moderate values of $n \geq 80$ and the reasonable error $\epsilon = 0.1$. For this reason, much work has been done on obtaining PTAS with more efficient running time such as $2^{O(1/\epsilon)}n$. For some optimization problems, worst case running time of existing PTAS has been improved through more sophisticated techniques or by more thorough analysis [2, 14, 10]. For many other problems, however, the existence of PTAS with more efficient running times remains a challenging open question.

Khanna and Motwani [18] introduced a number syntactic classes to characterize a large number of optimization problems on planar structures. Similarly, Hunt *et al.* [16, 17] investigated many problems with near planar structures and many problems on geometric graphs. All of these problems admit PTAS with running time $n^{O(1/\epsilon)}$. It is not known whether these problems admit PTAS with more efficient time such as $2^{O(1/\epsilon)}n$. In the present paper, we investigate complexity issues concerning PTAS for optimization problems on planar structures. Our study is based on the techniques and results recently developed in parameterized complexity theory.

[1]This work was supported in part by the National Science Foundation research grant CCR-000246

[2]School of EE & CS, Ohio University, Athens, OH 45701 USA,Email:cai@leon.cs.ohiou.edu, Fax:+1 740 593 0007

[3]Department of Computer Science and Software Engineering, University of Newcastle, University Drive, Callaghan 2308 Australia, mfellows@cs.newcastle.edu.au

[4]School of EE & CS, Ohio University, Athens, OH 45701 USA,Email:juedes@ohiou.edu, Fax:+1 740 593 0007

[5]Department of Computer Science and Software Engineering, University of Newcastle, University Drive, Callaghan 2308 Australia, fran@cs.newcastle.edu.au

Parameterized complexity was introduced by Downey and Fellows [11, 12] to study computational problems for which efficient algorithms for small or moderate parameters are natural and useful. Research results show that parameterized tractability and polynomial-time approximability are closed related. In particular, under a natural parameterization framework, Cai and Chen [7] observed that the existence of fully polynomial-time approximation schemes (FPTAS) implies parameterized tractability. Bazgan [4] and Cesati and Trevisan [9] independently improved this relation to *efficient polynomial-time approximation schemes* (EPTAS). According to [9], an optimization problem admits an EPTAS if it admits a PTAS of running time $f(1/\epsilon)p(n)$, where $f(k)$ is a function and $p(n)$ is some polynomial. Therefore, parameterized tractability becomes a necessary condition for an optimization problem to admit an EPTAS. In general, this condition is not sufficient since all MAX SNP-complete problems are parameterized tractable [7]. However, we show here that it is possible for parameterized tractability to be a part of a collection of sufficient conditions for obtaining EPTAS.

In the present paper we develop techniques to obtain upper and lower bounds on the parameterized complexity for various optimization problems on planar structures. These results have direct implications on the existence of EPTAS as well as upper and lower time bounds on EPTAS for these problems. Our research focuses on a large number of (non-weighted) optimization problems captured by the syntactic classes Planar MPSAT, Planar TMIN, and Planar TMAX which were introduced by Khanna and Motwani [18] to characterize PTAS. In particular, our work and technical results address the following issues.

(1) *The hardness of admitting an EPTAS.* Planar MPSAT, Planar TMIN, and Planar TMAX are natural extensions from the better-known classes Max SNP, Min $F^+\Pi_2$, and RMAX(2), respectively, by allowing a minterm (i.e., a conjunction of literals) in place of each literal. Our research shows that the existence of EPTAS for such problems is very sensitive to the size of each minterm. We are able to show that Planar TMIN_4 (Planar TMIN restricted to minterms of size four or smaller), Planar TMAX, and Planar MPSAT characterize $W[1]$-hard problems, and hence we exclude the possibility of an EPTAS for some problem in each of these classes. In contrast, problems such as Dominating Set on planar graphs are in the class Planar TMIN_1 and admit EPTAS. To achieve these results, we develop a number of non-trivial planarity construction techniques. We note here that the previous work of Cesati and Trevisan [9] gives examples of natural W[1]-hard problems in the syntactic class $O(1)$-Outerplanar MINMAX [18]. Hence, all the syntactic classes given by Khanna and Motwani [18] contain problems that do not admit EPTAS unless W[1]=FPT.

(2) *The application of parameterized tractability in obtaining EPTAS.* Polynomial-time approximation schemes for many optimization problems on planar graphs were developed based on the method of finding a small separator and then solving each separated components optimally [3]. Such an idea is generalized in [18] for problems in Planar TMIN, Planar TMAX, and Planar MPSAT. We observe that since solving each component can be done by a standard tree decomposition-based algorithmic schema [5, 6], parameterized algorithms via the standard tree decomposition-based schema can be used to solve each component optimally, thus leading to an EPTAS. In particular, we prove that essentially all problems in Planar TMIN_1, including Planar Vertex Cover and Planar Dominating Set, are solvable in time $2^{O(\sqrt{k})}n$, leading to $2^{O(1/\epsilon)}n$-time EPTAS for these problems. We also prove that all problems in Planar $\text{TMIN}^{polylog}$ (restricted to FOFs each with at most a polylogarithmic number of minterms) are parameterized tractable, implying EPTAS for each problem in this subclass.

We note that almost all of the above results in (1) and (2) apply to the classes Planar TMAX and Planar MPSAT. This paper is organized as follows. In section 2, we introduce the necessary concepts from parameterized complexity theory and establish the connection between EPTAS and parameterized tractability. In section 3, we give $W[1]$-hardness proofs for problems in Planar TMIN, Planar TMAX, and Planar MPSAT, thereby demonstrating the existence of problems in these classes that do not admit EPTAS unless $W[1]$=FPT. In section 4, time upper bounds for EPTAS are demonstrated for subclasses of problems in Planar TMIN, Planar TMAX, and Planar MPSAT. We conclude in section 5 with a brief discussion of open questions.

2 Preliminaries

Here we briefly introduce the necessary concepts concerning optimization problems and the theory of parameterized complexity. For additional information, we refer readers to the comprehensive text on parameterized complexity by Downey and Fellows [12] and the classic text on NP-completeness by Garey and Johnson [13].

A parameterized problem Π is defined over the set $\Sigma^* \times \mathbb{N}$, where Σ is a finite alphabet and $\mathbb{N}$ in the set of natural numbers. Therefore, each instance of the problem Π is a pair $\langle I, k\rangle$, where k is called the *parameter*.

Definition 2.1 *A problem* Π *is* parameterized tractable *if there is an algorithm running in time* $O(f(k)p(|I|))$ *that solves the parameterized problem* Π *for some polynomial* p *and some recursive function* f*.*

Parameterized tractability is usually stated in terms of decision problem, i.e., determining whether some instance $\langle I, k\rangle \in \Pi$. However, it is occasionally necessary to consider algorithms that produce witnesses. In this context, we consider algorithms that produce witnesses to the fact that $\langle I, k\rangle \in \Pi$, if such a witness exists, in $O(f(k)p(|I|))$ steps. In this case, we say that Π is parameterized tractable with witness, or that Π is solvable with witness in time $O(f(k)p(|I|))$. This slightly stronger definition of parameterized tractability will be used in section 4.

The complexity class FPT contains all parameterized tractable problems. Parameterized problems are classified into the W-hierarchy FPT $\subseteq W[1] \subseteq W[2] \subseteq \cdots$W$[P]$. Each class $W[t]$ contains problems that are not known to be parameterized tractable. Complete problems for these classes are defined based on a reductions that preserve parameterized tractability [12]. For example, Independent Set and Clique are $W[1]$-complete.

Many parameterized problems are naturally obtained from optimization problems through parameterizations. Following the earlier work of Cai and Chen [7], we use a standard parameterization of optimization problems. For each optimization problem Π, the *standard parameterized* (*decision*) *version* Π^* of Π is to determine, given an instance I of Π and an integer k, whether the optimal solution cost $OPT_\Pi(I)$ is $\geq k$ for maximization problems or $\leq k$ in the case of a minimization problems.

Definition 2.2 *[9] An optimization problem admits an efficient polynomial-time approximation scheme (EPTAS) if for each fixed error* $\epsilon > 0$*, there is an (uniform)* $f(1/\epsilon)p(|I|)$*-time approximation algorithm* A *that guarantees* $|A(I) - OPT(I)| \leq \epsilon OPT(I)$ *for every given instance* I*, where* $f(k)$ *is a function and* $p(n)$ *is some polynomial.*

The fact that the existence of an EPTAS implies parameterized tractability was first shown by Bazgan [4] and later (and independently) by Cesati and Trevisan [9]. The following proposition provides a detailed account of this result in terms of time complexity.

Proposition 2.1 *Let Π be an NP optimization problem. If Π admits an $O(f(1/\epsilon)n^c)$-time EPTAS, Π^* is solvable in time $O(f(2k)n^c)$.*

We next present our notation concerning the syntactic classes defined by Khanna and Motwani [18]. To begin, given a collection of variables, a *minterm* is simply a conjunction of literals. A literal is *positive* if it is x_i for some variable x_i. A literal is *negative* if it is $\neg x_j$ for some variable x_j. A minterm is *positive* (*negative*) if all of the literals are positive (negative). A first order formula (FOF) is a disjunction of minterms. An FOF F is *positive* (*negative*) if each minterm in F is positive (negative). The *width* of an FOF is the number of minterms in the formula. The *size* of a minterm is the number of literals in the minterm.

Definition 2.3 *Let $s(n), w(n)$ be two functions in n.*

(1) $\text{TMIN}_{s(n)}^{w(n)}$ *is the class of all NP optimization problems that can be expressed as follows: given a collection C of positive FOFs over n variables with width bounded by $w(n)$ and the maximum size of minterms bounded by $s(n)$, find a minimum weighted truth assignment T that satisfies all FOFs in C.*

(2) $\text{TMAX}_{s(n)}^{w(n)}$ *is the class of all NP optimization problems that can be expressed as follows: given a collection C of negative FOFs over n variables with width bounded by $w(n)$ and the maximum size of minterms bounded by $s(n)$, find a maximum weighted truth assignment T that satisfies all FOFs in C.*

(3) $\text{MPSAT}_{s(n)}^{w(n)}$ *is the class of all NP-optimization problems that can be expressed as follows: given a collection C of FOFs over n variables with width bounded by $w(n)$ and the maximum size of minterms bounded by $s(n)$, find a truth assignment T that maximizes the number of FOFs in C that are satisfied.*

Notice that width of any satisfiable minterm is bounded by n. Hence, we will only consider values of $s(n) \leq n$.

Given a collection C of FOFs over n variables, the *incidence graph* of C is the bipartite graph with edges between the set of FOFs and the set of variables such that there is an edge between a formula and a variable if and only the variable occurs in the formula. Let the syntactic class $\mathcal{C}$ be either MPSAT, TMIN, or TMAX. Then Planar $\mathcal{C}$ is the class of problems in $\mathcal{C}$ restricted to instances with planar incidence graphs.

We now define in the following subclasses of problems. Let $\mathcal{C} \in \{$ TMAX, TMIN, MPSAT$\}$. Planar $\mathcal{C}_{s(n)} = \bigcup_{w \in poly}$ Planar $\mathcal{C}_{s(n)}^{w(n)}$, Planar $\mathcal{C}^{w(n)} =$ Planar $\mathcal{C}_n^{w(n)}$, Planar $\mathcal{C} = \bigcup_{w \in poly}$ Planar $\mathcal{C}^{w(n)}$, and Planar $\mathcal{C}^{polylog} = \bigcup_{c \geq 0}$ Planar $\mathcal{C}^{\log^c n}$.

In our work in section 4, we use a *canonical problem* for each of these classes.

Definition 2.4 *Let $s(n)$ and $w(n)$ be two functions. Define*

(1) Planar KTmin$_{s(n)}$*: given a collection of FOFs over n variables with each minterm size bounded by $s(n)$, find a minimum weighted assignment that satisfies all FOFs.*

(2) Planar KTmin$^{w(n)}$*: given a collection of FOFs over n variables with each formula width bounded by $w(n)$, find a minimum weighted assignment that satisfies all FOFs.*

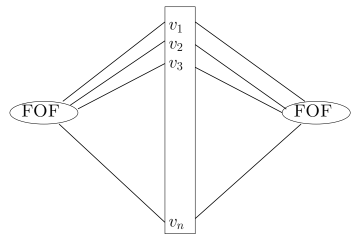

Figure 1: A block of n variables.

(3) PLANAR KTMIN^{polylog}*: given a collection of FOFs over nvariables with each formula width bounded by* $\log^c n$ *for some* $c \geq 0$*, find a minimum weighted assignment that satisfies all FOFs.*

(4) PLANAR KTMIN*: given a collection of FOFs over* n *variables, find a minimum weighted assignment to that satisfies all FOFs.*

Canonical problems for the subclasses of Planar TMAX and Planar MPSAT are defined in a similar fashion.

3 The hardness of achieving EPTAS

In this section, we show that there are problems in Planar TMIN_4, the subclasses of Planar TMIN with restriction to size 4 minterms, which do not admit EPTAS unless $W[1] = \text{FPT}$. We consider the canonical problem PLANAR KTMIN_4 in the class Planar TMIN_4 and prove $W[1]$-hardness for its parameterized complexity. Similar results hold for Planar TMAX and Planar MPSAT.

The $W[1]$-hardness proofs for these canonical problems rely on non-trivial parameterized reductions from CLIQUE. Very briefly, let $\langle G, k\rangle$ be an instance of CLIQUE. Assume that G has n vertices. From G and k, we construct a collection C of FOFs over $f(k)$ blocks of n variables. C will contain at most $2f(k)$ FOFs and the incidence graph of C will be planar. In the case of PLANAR KTMIN, each minterm in each FOF will contain at most 4 variables. The collection C is constructed so that G has a clique of size k if and only if C has a weight $f(k)$ satisfying assignment with exactly one variable set to true in each block of n variables. Here we have that $f(k) = O(k^4)$.

Theorem 3.1 PLANAR KTMIN_4 *is W[1]-hard.*

Proof: We show that CLIQUE is parameterized reducible to PLANAR KTMIN_4. Since CLIQUE is $W[1]$-complete, it will follow that PLANAR KTMIN_4 is $W[1]$-hard.

To begin, let $\langle G, k\rangle$ be an instance of CLIQUE. Assume that G has n vertices. From G and k, we will construct a collection C of FOFs over $f(k)$ blocks of n variables. C will contain at most $2f(k)$ FOFs and the incidence graph of C will be planar. Moreover, each minterm in each FOF will contain at most 4 variables. The collection C is constructed so that G has a clique of size k if and only if C has a weight $f(k)$ satisfying assignment with exactly one variable set to true in each block of n variables. Here we have that $f(k) = O(k^4)$. To maintain planarity in the incidence graph for C, we ensure that each block of n variables appears in at most 2 FOFs. If this condition is maintained, then we can draw each block of n variables as seen in Figure 1.

We describe the construction in two stages. In the first stage, we use k blocks of n variables and a collection C' of $k(k-1)/2 + k$ FOFs. In a weight k satisfying assignment for C', exactly one

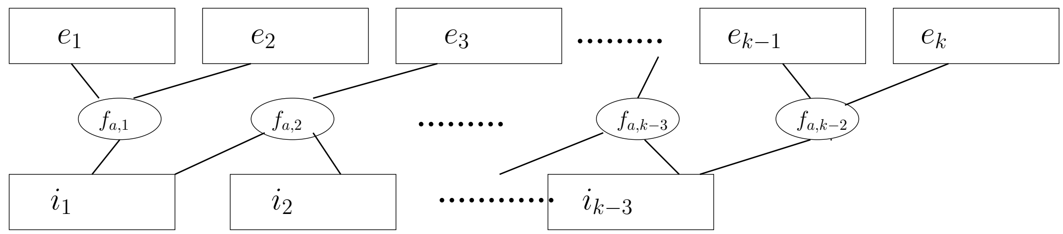

Figure 2: The widget A_k.

variable $v_{i,j}$ in each block of variables $b_i = [v_{i,1}, \dots, v_{i,n}]$ will be set to true. We interpret this event as "vertex j is the ith vertex in the clique of size k." The $k(k-1)/2 + k$ FOFs are described as follows. For each $1 \le i \le k$, let f_i be the FOF $\bigvee_{j=1}^{n} v_{i,j}$. This FOF ensures that at least one variable in b_i is set to true. For each pair $1 \le i < j \le k$, let $f_{i,j}$ be the FOF $\bigvee_{(u,v)\in E} v_{i,u}v_{j,v}$. Each FOF $f_{i,j}$ ensures that there is an edge in G between the ith vertex the clique and the jth vertex in the clique.

It is somewhat straightforward to show that $C' = \{f_1, \dots, f_k, f_{1,2}, \dots, f_{k-1,k}\}$ has a weight k satisfying assignment if and only if G has a clique of size k. To see this, notice that any weight k satisfying assignment for C' must satisfy exactly 1 variable in each block b_i. Each first order formula $f_{i,j}$ ensures that there is an edge between the ith vertex in the potential clique and the jth vertex in the potential clique. Notice also that, since we assume that G does not contain edges of the form (u,u), the FOF $f_{i,j}$ also ensures that the ith vertex in the potential clique is not the jth vertex in the potential clique. This completes the first stage.

This first stage can be drawn in the shape of the complete graph K_k on k vertices. To see this, place each block of variables b_i at vertex v_i of K_k and label each edge (i,j) in K_k by the FOF $f_{i,j}$. Finally, attach each FOF f_i to vertex v_i on the exterior. Notice that this drawing of incidence graph for this collection of FOFs is not planar. We fix this problem in the second stage.

In the second stage we achieve planarity by removing crossovers in incidence graph for C'. We use two types of widgets to remove crossovers while keeping the number of variables per minterm bounded by 4. The first widget A_k consists of $k + k - 3$ blocks of n variables and $k-2$ FOFs. This widget consists of $k-3$ internal and k external blocks of variables. Each external block $e_i = [e_{i,1}, \dots, e_{i,n}]$ of variables is connected to exactly one FOF inside the widget. Each internal block $i_j = [i_{j,1}, \dots, e_{j,n}]$ is connected to exactly two FOFs inside the widget. The $k-2$ FOFs are given as follows. The FOF $f_{a,1}$ is $\bigvee_{j=1}^{n} e_{1,j}e_{2,j}i_{1,j}$. For each $2 \le l \le k-3$, the FOF $f_{a,l}$ $= \bigvee_{j=1}^{n} i_{l-1,j}e_{l+1,j}i_{l,j}$. Finally, $f_{a,k-2} = \bigvee_{j=1}^{n} i_{k-3,j}e_{k-1,j}e_{k,j}$. These $k-2$ FOFs ensure that the settings of variables in each block is the same if there is a weight $2k-3$ satisfying assignment to the $2k-3$ blocks of n variables. The widget A_k is shown in Figure 2. Since each internal block is connected to exactly two FOFs, the incidence graph for this widget can be drawn on the plane without crossing any edges.

The second widget removes crossover edges from the first stage of the construction. In the first stage, crossovers in our drawing can occur in the incidence graph because two FOFs cross from one block of variables to another. To eliminate this, consider each edge i,j in K_k with $i < j$ as a directed edge from i to j. In the next stage of the construction, we send a copy of the variables in block i to block j. At each crossover point from the direction of blocks $u = [u_1, \ldots, u_n]$ and $v = [v_1, \ldots, v_n]$, insert a widget B that introduces 2 new blocks of n variables $u_1 = [u_{1_1} \ldots u_{1_n}]$ and $v_1 = [v_{1_1} \ldots v_{1_n}]$ and a FOF $f_B = \bigvee_{j=1}^{n} \bigvee_{l=1}^{n} u_j u_{1_j} v_l v_{1_l}$. The FOF f_B ensures that u_1 and v_1 are copies of u and v. As shown in Figure 3, the incidence graph for the widget B is also planar.

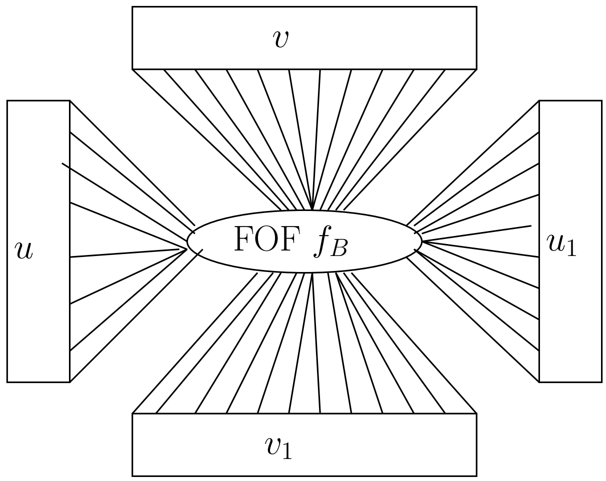

Figure 3: A planar drawing of the widget B.

To complete the construction, we replace each of the original k blocks of n variables from the first stage with a copy of the widget A_k. At each crossover point in the graph, we introduce a copy of widget B. We attach each f_i to its associated A_k. Finally, for each directed edge between block i and j, we insert the original FOF $f_{i,j}$ between the last widget B and the destination widget A_k. Since one of the new blocks of variables created by the widget B is a copy of block i, the effect of the FOF $f_{i,j}$ in this new collection of FOFs is the same as before.

Figure 4 provides a diagram that shows the full construction when $k = 5$. Since each the incidence graph of each widget in this drawing is planar, the entire collection C of first order formulas has a planar incidence graph.

Now, if we assume that there are $c(k) = O(k^4)$ crossover points in standard drawing of K_k, then our collection has $c(k)$ B widgets. Since each B widget introduces 2 new blocks of n variables, this gives $2c(k)$ new blocks. Since we have k A_k widgets, each of which has $2k-3$ blocks of n variables, this gives an additional $k(2k-3)$ blocks. So, in total, our construction has $f(k) = 2c(k)+2k^2-3k = O(k^4)$ blocks of n variables. Note also that there are $g(k) = k(k-1)/2++k+k(k-2)+c(k) = O(k^4)$ FOFs in the collection C.

As shown in our construction C has a weight $f(k)$ satisfying assignment (i.e., each block has exactly one variable set to true) if and only if the original graph G has a clique of size k. Since the incidence graph of C is planar and each minterm in each FOF contains at most four variables, it follows that this construction is a parameterized reduction from Clique to Planar KTmin$_4$. This completes the proof. □

Theorem 3.2 Planar KTmax *is W[1]-hard.*

Proof: The proof is very similar to the proof of Theorem 3.1. Given an instance $\langle G, k\rangle$ of Clique, convert that instance into the collection of FOFs C as described in the proof of Theorem 3.1.

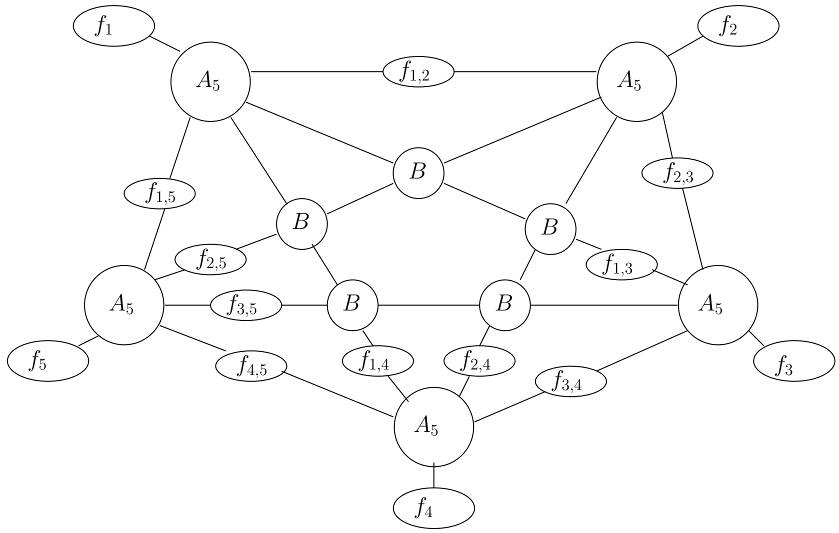

Figure 4: The construction when $k = 5$.

Replace each positive variable v_i in each minterm with a conjunction $\neg v_1 \neg v_2 \cdots \neg v_{i-1} \neg v_{i+1} \cdots \neg v_n$ of negated variables from the same block. The process converts each positive minterm to a negative minterm. Note that this new collection C' has weight $f(k)$ satisfying assignment if and only if G has a clique of size k. Since $\langle C', f(k) \rangle$ is an instance of Planar KTmax, this gives a parameterized reduction from Clique to Planar KTmax. □

Theorem 3.3 Planar KMpsat *is W[1]-hard.*

Proof: The proof is very similar to the proof of Theorem 3.1. Given an instance $\langle G, k \rangle$ of Clique, convert that instance into the collection of FOFs C as described in the proof of Theorem 3.1. For each FOF f_i in C, replace f_i with the formula

$$f_i' = \bigvee_{j=1}^{n} \neg v_{i,1} \neg v_{i,2} \ldots \neg v_{i,j-1} \neg v_{i,j+1} \ldots \neg v_{i,n},$$

which creates a new collection C' of FOFs. The incidence graph for this new collection of FOFs is still planar. Moreover, notice that each formula f_i' is true when at most one variable in each block is set to true. Therefore, there is a satisfying assignment to all the $g(k)$ FOFs in the collection if and only if G has a clique of size k. Since $\langle C', g(k) \rangle$ is an instance of Planar KMpsat_4, this gives a parameterized reduction from Clique to Planar KMpsat. □

Note that from the results of Khanna and Motwani [18], all three problems Planar KTmin_4, Planar KTmax, and Planar KMpsat have polynomial-time approximation schemes. However, as we show here, these problems do not have efficient polynomial-time approximation schemes unless $W[1] = \text{FPT}$.

Corollary 3.1 Planar KTmin_4*,* Planar KTmax*, and* Planar KMpsat *do not have efficient polynomial approximation schemes unless W[1] = FPT.*

Proof: By Proposition 2.1, the existence of an efficient polynomial-time approximation scheme for Planar KTmin_4, Planar KTmax, and Planar KMpsat implies that parameterized versions of these problems are in FPT. Since each of these problems is $W[1]$-hard, this implies that $W[1] =$ FPT. □

4 Upper bounds on the running time for EPTAS

In this section we show some positive results regarding the running time of EPTAS. The work in this section uses the following concepts.

Definition 4.1 *An outerplanar graph is a planar graph that has an embedding on the plane with all vertices appearing on the outer face. An r-outerplanar graph is an outerplanar graph when $r = 1$, or a $(r-1)$-outerplanar graph by deleting all vertices on the outer face when $r > 1$.*

The layer L_1 of an r-outerplanar graph consists of the vertices on the boundary of the outer face, and for $i > 1$, the layer L_i is the set of vertices that lie on the boundary of the outer face in the embedding of the subgraph $G - (L_1 \cup \cdots, \cup L_{i-1})$.

Definition 4.2 *A tree decomposition D of a graph $G = (V, E)$ consists of a tree $T = (I, F)$ and a collection of subsets of V, $\{X_i | i \in I\}$, one for each node in the tree T, that collectively satisfy the following conditions:*

1. *$\bigcup_{i \in I} X_i = V$, i.e., each $v \in V$ is in some X_i,*
2. *for each edge $(u, v) \in E$, there exists some i such that $u, v \in X_i$, and*
3. *for each $v \in V$, the set $\{i \in I | v \in X_i\}$ induces a subtree of T.*

Item (3) is often written as "for all $i, j, k \in I$, if j lies on the path between i and k, then $X_i \cap X_k \subseteq X_k$." Notice that this implies that set of vertices in the subtrees below some bag X_i are disjoint except for the vertices in X_i.

The *width* of a tree decomposition D is the maximum cardinality of any X_i in D. The *treewidth* of a graph G is the minimum width needed by every tree decomposition D of G. The notions of outerplanarity and treewidth are closely connected.

Proposition 4.1 *[6] An r-outerplanar graph has treewidth of at most $3r - 1$.*

Polynomial-time approximation schemes for many problems on planar structures (e.g., on planar graphs) can be developed based on the method of finding a small separator and then solving each separated component optimally [3, 18]. Such an idea has been generalized by Khanna and Motwani for problems in Planar TMIN, Planar TMAX, and Planar MPSAT as follows: given an t-outerplanar embedding of planar incidence graph, decompose the layers $L_1, \cdots, L_t$ of the t-outerplanar graph into r disconnected components, each consisting of at most c/ϵ layers, where $r = O(\epsilon t)$. Then, find an optimal solution for each component and assemble solutions for all components forming an $(1 + \epsilon)$-approximation for the original instance. Since each component is an $O(1/\epsilon)$-outerplanar graph, it has treewidth $O(1/\epsilon)$ by Proposition 4.1. A *standard tree decomposition-based algorithmic schema* [5, 6, 1, 18, etc.] is applied to obtain a PTAS. In general, for problems in Planar TMIN, Planar TMAX, and Planar MPSAT, the running time of these PTAS is $O(n^{O(1/\epsilon)})$.

On the other hand, many recent parameterized algorithms for problems on planar graphs, such as Planar Dominating Set [1], have been developed via the tree decomposition-based algorithmic schema. In particular, in order to give a positive answer to the relationship between $OPT(I)$ and the parameter k, the incidence graph has to have treewidth bounded by $w(k)$ for some function w. This observation leads to the following technical lemma.

Lemma 4.1 *Let Π be an optimization problem in Planar TMIN, Planar TMAX, or Planar MP-SAT. If Π^* in solvable with witness in time $O(f(w(k))p(n))$ via the standard tree decomposition-based algorithmic schema and a positive answer to the relationship between $OPT(I)$ and k implies that the incidence graph has treewidth bounded by $w(k)$, then Π has an EPTAS running in time*

$$O(\frac{1}{\epsilon}f(O(\frac{1}{\epsilon}))p(n))$$

for some polynomial p.

Proof: The proof of this theorem follows the approach given in the work of Baker [3] and Khanna and Motwani [18], with some minor modifications. Here, we give the proof for Planar TMIN. The proof for Planar TMAX and Planar MPSAT are similar. Note that the theorem holds only for the unweighted versions of these problems.

Let Π be a problem in Planar TMIN, and let I be an instance of Π. Then, I can be cast as a problem of determining a minimum weighted assignment to n variables that satisfies a collection C of m first order formulae. Since Π is in Planar TMIN, the incidence graph G for C is planar. Since G has $n+m$ vertices and each layer has at least one vertex, we have that G is t-outerplanar for some $t \leq n+m$. So, let $L_1, \ldots, L_t$ be the layers in G. Notice that it is possible to build G and then create the layers $L_1, \ldots, L_t$ in polynomial-time by first computing a planar embedding of G [15], and then iteratively removing the outermost layers of G.

We describe an efficient polynomial-time approximation scheme for Π. Let $p = \lceil \frac{1}{\epsilon} \rceil$, and notice that $\frac{1}{p} \leq \epsilon$. We break the layers of G into overlaping groups of size at most $2p+1$. Let S_j be the graph that consists of the layers $L_{2jp+2i-1}$ to $L_{2(j+1)p+2i}$ for some i, $1 \leq i \leq p$. The graphs S_j and S_{j+1} overlap at layers $L_{2(j+1)p+2i-1}$ and $L_{2(j+1)p+2i}$. For each S_j, build the subproblem C'_j consisting of just the FOFs in the layers $L_{2jp+2i-1}$ through $L_{2(j+1)p+2i}$ whose variables also appear in these same layers. This guarantees that all FOFs in layers L_{2jp+2i} through $L_{2(j+1)p+2i-1}$ appear in C'_j. We now solve each of these subproblems C'_j exactly using the parameterized algorithm mentioned in the statement of theorem. Since each S_j is $2p+1$-outerplanar graph, we know that it has treewidth of the incidence graph of C'_j (which is a subgraph of S_j) is at most $3(2p+1)-1$ by Proposition 4.1. Since Π^* is solvable in time $O(f(w(k))p(n))$ steps via the standard tree decomposition-based algorithm, we can solve the problem exactly in $O(f(t)p(n))$ steps, where t is the treewidth of the problem. Since each subproblem has treewidth $\leq 6p+2$, the minimum weighted value of the solution can be found in $O(f(6p+2)p(n))$ steps.

Since each minterm in the collection of FOFs C is positive, the union of the solutions to each C'_j provides a solution to the C. We repeat this process for each i ranging from 1 to p, and we return the minimum weighted solution. The total running time of this algorithm is

$$O(\frac{1}{\epsilon}f(O(\frac{1}{\epsilon})p(n))$$

To see that the solution is within $\frac{1}{p}$ of optimal, consider an optimal solution for C. Notice that this is a solution for each C'_j. So, since each C'_j was solved optimally, we have that the total cost of the solution produced by this algorithm is bounded below by the cost of the optimal solution plus the number of variables set in the overlapping layers (layers $L_{2jp+2i-1}$ and L_{2jp+2i}). By the pigeonhole principle, we know for some i, $1 \leq i \leq p$, that the total number of variables set to true in the overlapping layers $L_{2jp+2i-1}$ and L_{2jp+2i} is at most $\frac{OPT(C)}{p}$. Hence the solution produced by this algorithm has weight at most $OPT(C) + \frac{OPT(C)}{p} \leq (1+\epsilon)OPT(C)$. □

To obtain an EPTAS or time upper bounds on EPTAS for problems under consideration, it suffices to consider the parameterized tractability via the standard tree-decomposition algorithmic schema.

Lemma 4.2 *Let $G = (V_1 \cup V_2, E)$ be a planar bipartite graph such that there exists a $V' \subseteq V_1$ of size k which dominates every vertex in V_2. Then G has treewidth $O(\sqrt{k})$.*

Proof: Proof of the lemma is based on technical aspects of the proof for Theorem 10 in [1] to construct separators from building *upper*, *lower*, and *middle triples* for each layer in the graph decomposition. We briefly show the applicability of the construction to this lemma and refer the reader to [1] for many other details. Basically, there are three places in the proof for Theorem 10 [1] where the dominating number k or dominating vertices are used: (1) in defining triples, (2) in demonstrating a separation of layer L_{i-1} from L_{i+2}, and (3) in proving the size of separators.

Let D be a subset of V_1 with size k which dominate every vertex in V_2 in the bipartite graph. Let $D_i = D \cap L_i$. Then, just as in the case of a general planar graph, for each non-empty component of layer L_i, an upper, a lower, and a middle triple can be defined for each vertex in D_{i-1}, D_i, and D_{i+1}, respectively. Let S_i be the union of all upper, lower, and middle triples of L_i. We can also claim S_i separates layer L_{i-1} from L_{i+2}. The observation is, according to [1], that there is a vertex y on any path from a vertex in $L_{i+1} \cap D$ to a vertex in L_{i-1} such that y cannot be in D but it has to be included in either an upper, a lower, or a middle triple. It is clear that the observation applies to our bipartite case as well. For general planar graphs, the size of the separator S_i is obtained by establishing three technical lemmas similar to lemmas 1, 2, and 3 in [1]. Because the size of S_i is measured based on the sizes of D_{i-1}, D_i, and D_{i+1} and the number of non-empty components in layer L_{i+1}, the upper bound of the size still applies to the bipartite case. □

We next use Lemma 4.2 to show that subclasses of Planar TMIN are parameterized tractable. We apply Lemma 4.2 to show that the incidence graphs for all problems in Planar TMIN have small treewidth.

Theorem 4.1 *Planar $TMIN_1 \subseteq FPT$. Moreover, for every problem Π in Planar $TMIN_1$, Π has a $O(2^{O(\sqrt{k})}p(n)$ parameterized algorithm.*

Proof: Let Π be any problem in Planar TMIN_1. Let I be an instance of the problem and k be an integer. Then I can be expressed as a collection C of first-order formula. Let $x_1, \cdots, x_n$ be the set of variables in C and let $f_1, \cdots, f_m$ be the first-order formula in C. Then the incidence graph is a planar bipartite graph $G = (V_{var} \cup V_{for}, E)$, where $V_{var} = \{v_{x_1}, \cdots, v_{x_n}\}$ and $V_{for} = \{v_{f_1}, \cdots, v_{f_m}\}$.

If there is an weight k truth assignment that satisfies every formula f_j, $j = 1, \cdots, m$, then there exists a subset $D \subseteq V_{var}$ of size k in G that dominates every vertex in the set V_{for}. The vertices in D can be picked such that a vertex v_x is in D if and only the corresponding variable x is assigned the value "true." Because each formula f is satisfied (by the weight k truth assignment) and all $x_{i1}, \cdots, x_{it}$ occur in f as positive literals, at least one of these variable must be assigned with value "true" and that corresponding vertex dominates the vertex v_f.

By Lemma 4.2, the treewidth for the incidence graph is $O(\sqrt{k})$. Since the size of each minterm is one, we can construct a $2^{O(\sqrt{k})}N$-time algorithm to determine if there is an weight k truth assignment that satisfies every formula in C by using the standard dynamic programming approach [5]. For each node X_i in the tree-decomposition of the incidence graph, build a table with one entry for every possible setting of the variables and FOFs in X_i. The table stores the value of the minimum weighted satisfying assignment in the boolean formula constructed from the subtree rooted at X_i that is consistent with the given table entry. Note that since each minterm has size one, we can use true or false as a possible setting for each FOF. This gives a table of size at most $2^{O(\sqrt{k})}$. The minimum weighted satisfying assignment can be found by building the tables for each node in the tree decomposition in a bottom-up fashion. This approach is very similar to the algorithms given in the work of Khanna and Motwani [18]. □

The next theorem requires a somewhat straightforward result concerning parameterized tractability found in the work of Cai and Juedes [8].

Proposition 4.2 *If a parameterized problem* Π *is solvable in time* $O(2^{O(s(n)k)}p(n))$ *for some unbounded and nondecreasing function* $s(n) = o(\log n)$ *and some polynomial* p*, then it is parameterized tractable.*

Theorem 4.2 *Planar TMIN*$^{polylog} \subseteq$ *FPT.*

Proof: The proof is similar to the proof of Theorem 4.1. Given a problem Π in Planar TMIN^{polylog}, an instance I of Π can be represented as a collection of FOF C, where the width of each FOF in C is bounded by $(\log n)^c$.

As in the proof of the previous theorem, we can show that the treewidth of the incidence graph for C is $O(\sqrt{k})$. Since the width of each minterm is bounded by $(\log n)^c$, we can construct a $(\log n)^{cO(\sqrt{k})}N$-time algorithm to determine if there is an weight k truth assignment that satisfies every formula in C by using the standard dynamic programming approach [5]. For each node X_i in the tree-decomposition of the incidence graph, build a table with one entry for every possible setting of the variables and FOFs in X_i. The table stores the value of the minimum weighted satisfying assignment in the boolean formula constructed from the subtree rooted at X_i that is consistent with the given table entry. Note that since the width of each minterm is bounded by $(\log n)^c$, we can use the value of the minterm that satisfies it as a possible setting for each FOF. This gives a table of size at most $(\log n)^{cO(\sqrt{k})}$. The minimum weighted satisfying assignment can be found by building the tables for each node in the tree decomposition in a bottom-up fashion. This approach is very similar to the algorithms given in the work of Khanna and Motwani [18].

The running-time of this algorithm is $2^{\log(\log n)cO(\sqrt{k})}N$. Since $\log\log n = o(\log n)$, it follows from Proposition 4.2 that Π is in FPT. □

Similar results are given for Planar TMAX and Planar MPSAT.

Theorem 4.3 *Planar TMAX*$_1 \subseteq$ *FPT and Planar TMAX*$^{polylog} \subseteq$ *FPT.*

Proof: The proof follows the proofs of the previous theorems. Let Π be an optimization problem in Planar TMAX$_1$ or Planar TMAX^{polylog}. It suffices to show that the incidence graph for each instance of this problem has small treewidth. Let I be an instance of Π and assume that $OPT(I) = k$. Then I can be expressed by a collection C of m first-order formula, each a disjunction of negative minterms over n variables.

We first show that when the maximum weight of satisfying assignments is k, the incidence graph for C must be an $O(k)$-outerplanar graph. Without loss of generality, assume that in each formula f no variable occurs in every minterm for some FOF – otherwise, such a variable must be assigned the value "false" and the collection C can be simplified by eliminating such a variable. Now, consider the incidence graph of C. It must be r-outerplanar for some $r \leq n + m$. Let $L_1, \ldots, L_r$ be the layers in this r-outerplanar graph. Notice that, the first order formula in layers L_{i-1} and L_{i+2} must be separated in the sense that no variable appears in both a FOF in L_{i-1} and L_{i+2}. Also notice that, since no layer is empty and the incidence graph is bipartite, each layer, except possibly the last layer, contains at least one variable and at least one FOF.

The layers of the r-outerplanar incidence graph can thus be partitioned into groups $\{L_1, L_2, L_3\}$, $\{L_4, L_5, L_6\}$, $\cdots$, $\{L_{3t-2}, L_{3t-1}, L_{3t}\}$, where $t = \lceil r/3 \rceil$, such that layer the FOF in L_{3j+i} are separated from the FOF in layer $L_{3(j-1)+i}$ and layer $L_{3(j+1)+i}$ for each $i = 1, 2, 3$ and $j = 0, \cdots, t-1$. Consider some FOF f in layer L_{3j+i}. Since no variable occurs in every minterm in f, at least one variable x that appears in f can be assigned the value "true" such that all f is satisfied. The value of x has no effect on FOFs in layers $L_{3(j+1)+i}$ and $L_{3(j-1)+i}$. Now, if we set all other variables to be false, this is a satisfying assignment to the collection C of first order formula. Therefore, at least t variables can be assigned the value "true." This implies that $r \leq 3t \leq 3k$ since k is the maximum number of variables assigned the value "true." Therefore the incidence graph can be at most $3k$-outerplanar.

According to Proposition 4.1, the incidence graph has treewidth $9k - 1$ when the maximum weight for satisfying assignments is k. Using the associated tree decomposition in the standard dynamic programming algorithm gives the result. □

Theorem 4.4 *Planar MPSAT*$_1 \subseteq$ *FPT and Planar MPSAT*$^{polylog} \subseteq$ *FPT.*

Proof: The proof is similar to the proof for Theorem 4.3, except that we are trying to maximize the number of FOF that are satisfied. Let C be a collection of FOF, and assume that the maximum number of FOFs that can be satisfied is k for any the truth assignment to the variables in C. We will show that the incidence graph for C is at most $3k$ outerplanar.

As in the proof of Theorem 4.3, we can break the incidence graph into r layers $L_1, \ldots, L_r$. The first order formulae in layer L_{i-1} must be separated from the first order formula in layer L_{i+2} in the sense that they do not share variables. The layers of the r-outerplanar incidence graph can thus be partitioned into groups $\{L_1, L_2, L_3\}$, $\{L_4, L_5, L_6\}$, $\cdots$, $\{L_{3t-2}, L_{3t-1}, L_{3t}\}$, where $t = \lceil r/3 \rceil$, such that layer the FOF in L_{3j+i} are separated from the FOF in layer $L_{3(j-1)+i}$ and layer $L_{3(j+1)+i}$ for each $i = 1, 2, 3$ and $j = 0, \cdots, t-1$. Since each layer L_{3j+i} contains at least one FOF, pick one of them. Call this FOF f. Now, pick one minterm m in f and set the variables from m appropriately so that m evaluates to true. This assignment to the variables satisfies f. Now, since L_{3j+i} and $L_{3(j+1)+i}$ are separated, we can do this assignment for all layers L_{3j+i}. Thus we have created an assignment to the variables that satisfies at least t FOF.

This gives the bound $r \leq 3k$ on the outerplanarity of the incidence graph when the maximum number of formulae satisfied is k. According to Proposition 4.1, the incidence graph has treewidth $9k - 1$ when the maximum weight for satisfying assignments is k. Using the associated tree decomposition in the standard dynamic programming algorithm gives the result. □

For the above theorems and proofs, we have the following results concerning EPTAS.

Corollary 4.1 *All problems in the classes Planar TMIN*$_1$*, Planar TMAX*$_1$*, and Planar MPSAT*$_1$ *admit time* $O(2^{O(1/\epsilon)}n)$ *EPTAS.*

Proof: As shown in the proof of Theorem 4.1, for each problem Π in Planar TMIN$_1$, Π^* can be solved in time $O(2^t n)$ via the standard tree decomposition algorithmic schema, where t is the treewidth of the incidence graph of each collection C of FOFs. Therefore, by Lemma 4.1, there is an EPTAS for Π running in time

$$O(\frac{1}{\epsilon} 2^{O(\frac{1}{\epsilon})} n) = O(2^{O(\frac{1}{\epsilon})} n).$$

Similarly, for each problem Π in Planar TMAX$_1$, Π^* can be solved in time $O(2^t n)$ steps, where t is the treewidth of the given incidence graph. So, by Lemma 4.1, there is an EPTAS for Π running in time

$$O(\frac{1}{\epsilon} 2^{O(\frac{1}{\epsilon})} n) = O(2^{O(\frac{1}{\epsilon})} n).$$

The proof for Planar MPSAT$_1$ is similar. □

Corollary 4.2 *All problems in the classes Planar TMIN^{polylog}, Planar TMAX^{polylog}, and Planar MPSAT^{polylog} admit EPTAS.*

Proof: The proof employs Theorems 4.2, 4.3, and 4.4 and Lemma 4.1, and is similar to the proof of Corollary 4.1. □

5 Conclusion

Our results show that strong connection between the running time of PTAS for certain problems and the ability to describe these problems syntactically. If a problem can be described by a collection of FOFs with minterms of size 1, it must have an EPTAS. In contrast, we show that there exist problems that can be described by a collection of FOFs with minterms of size 4 (or smaller) that do not have EPTAS unless $W[1]$=FPT. This connection between the syntactic description of planar problems and the existence of EPTAS deserves further investigation. For instance, what is the complexity of problems in Planar TMIN_2 and Planar TMIN_3?

References

[1] J. Alber, H. Bodlaender, H. Fernau, and R. Niedermeier. Fixed parameter algorithms for planar dominating set and related problems. In *Proceedings of the 7th Scandinavian Workshop on Algorithm Theory (SWAT 2000)*, volume 1851 of *Lecture Notes in Computer Science*, pages 97–110. Springer-Verlag, 2000.

[2] S. Arora. Nearly linear time approximation schemes for Euclidian TSP and other geometric problems. In *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, pages 554–563. IEEE Computer Society Press, 1997.

[3] B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM*, 41:153–180, 1994.

[4] C. Bazgan. Schémas d'approximation et complexité paramétrée. Master's thesis, INRIA, Orsay, France, 1995.

[5] H. L. Bodlaender. Dynamic programming on graphs with bounded treewidth. In Timo Lepistö and Arto Salomaa, editors, *Automata, Languages and Programming, 15th International Colloquium*, volume 317 of *Lecture Notes in Computer Science*, pages 105–118, Tampere, Finland, 11–15 July 1988. Springer-Verlag.

[6] H. L. Bodlaender. Treewidth: Algorithmic techniques and results. In Igor Prívara and Peter Ruzicka, editors, *Mathematical Foundations of Computer Science 1997*, volume 1295 of *LNCS*, pages 19–36, Bratislava, Slovakia, 25–29 August 1997. Springer.

[7] L. Cai and J. Chen. On fixed-parameter tractability and approximability of NP op timization problems. *Journal of Computer and System Sciences*, 54(3):465–474, June 1997.

[8] L. Cai and D. W. Juedes. Subexponential parameterized algorithms collapse the W-hierarchy. In *Proceedings of the Twenty - Eighth International Colloquium on Automata, Languages and Programming (ICALP 2001)*, pages 273–284. Springer-Verlag, 2001.

[9] M. Cesati and L. Trevisan. On efficiency of polynomial time approximation schemes. *Information Processing Letters*, 64:165–171, 1997.

[10] W. F. de la Vega and M. Karpinski. Polynomial time approximation of dense weighted instances of MAX-CUT. *Random Structures and Algorithms*, 16:314–332, 2000.

[11] R. G. Downey and M. R. Fellows. Fixed-parameter intractability. In *Proceedings of the 7th Annual Conference on Structure in Complexity Theory*, pages 36–49, Boston, MA, USA, June 1992. IEEE Computer Society Press.

[12] R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer-Verlag, New York, 1999.

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

[14] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection with learning and approximation. In *37th IEEE Symposium on Foundations of Computer Science*, 1996.

[15] J. Hopcroft and R. Tarjan. Efficient planarity testing. *Journal of the ACM*, 21(4):549–568, 1974.

[16] H. B. Hunt III, M. V. Marathe, V. Radhakrishnan, S. S. Ravi, D. J. Rosenkrantz, and R. E. Sterns. Designing approximation schemes using L-reductions. In *Foundations of Software Technology and Theoretical Computer Science, 14th Conference*, pages 342–353, 1994.

[17] H. B. Hunt III, M. V. Marathe, V. Radhakrishnan, S. S. Ravi, D. J. Rosenkrantz, and R. E. Sterns. Nc-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. In *2nd European Symposium on Algorithms*, pages 468–477, 1994.

[18] S. Khanna and R. Motwani. Towards a syntactic characterization of PTAS. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing*, pages 329–337, Philadelphia, Pennsylvania, 22–24 May 1996.