Algorithms **2013**, *6*, 43-59; doi:10.3390/a6010043

OPEN ACCESS

algorithms

ISSN 1999-4893

www.mdpi.com/journal/algorithms

Article

Computational Study on a PTAS for Planar Dominating Set Problem

Marjan Marzban and Qian-Ping Gu *

School of Computing Science, Simon Fraser University, Burnaby BC, V5A 1S6 Canada;
E-Mail: mmarzban@cs.sfu.ca

* Author to whom correspondence should be addressed; E-Mail: qgu@cs.sfu.ca;
Tel.: +1-778-782-6705; Fax: +1-778-782-3045.

Received: 2 November 2012; in revised form: 10 January 2013 / Accepted: 13 January 2013 / Published: 21 January 2013

Abstract: The dominating set problem is a core NP-hard problem in combinatorial optimization and graph theory, and has many important applications. Baker [JACM 41,1994] introduces a k-outer planar graph decomposition-based framework for designing polynomial time approximation scheme (PTAS) for a class of NP-hard problems in planar graphs. It is mentioned that the framework can be applied to obtain an $O(2^{ck}n)$ time, c is a constant, $(1+1/k)$-approximation algorithm for the planar dominating set problem. We show that the approximation ratio achieved by the mentioned application of the framework is not bounded by any constant for the planar dominating set problem. We modify the application of the framework to give a PTAS for the planar dominating set problem. With k-outer planar graph decompositions, the modified PTAS has an approximation ratio $(1 + 2/k)$. Using $2k$-outer planar graph decompositions, the modified PTAS achieves the approximation ratio $(1+1/k)$ in $O(2^{2ck}n)$ time. We report a computational study on the modified PTAS. Our results show that the modified PTAS is practical.

Keywords: dominating set problem; PTAS; branch-decomposition based algorithms; planar graphs; computational study

1. Introduction

An important research area in graph theory and networks is domination; it has been energetically investigated for many years due to its large number of real-world applications, such as resource allocation [1,2] and voting [3]. Haynes *et al.* In their books, [4,5] provide a good survey on domination problems. Let G be a simple undirected graph with the set of vertices $V(G)$ and the set of edges $E(G)$. We denote $|V(G)|$ by n. The r-dominating set D of G is a subset of $V(G)$ containing r vertices, such that for every vertex v in $V(G)$, either $v \in D$ or v is adjacent to a vertex in D. The minimum integer r for which G has a r-dominating set is called the *domination number* of G and is denoted by $\gamma(G)$. The dominating set problem is to decide that given a graph G and an integer r, whether $\gamma(G) \leq r$. The optimization version of this problem is to find a minimum dominating set.

The dominating set problem is a core NP-hard problem in combinatorial optimization and graph theory [6]. There is a long history of research on the approximation and exact algorithms to tackle the intractability of the problem. A minimization problem P is $\alpha-$approximable $(\alpha \geq 1)$ if there is an algorithm which gives a solution for any instance of P in polynomial time in the instance size with solution value at most αOPT, where OPT is the value of an optimal solution for the instance of P. If P is $(1+\epsilon)-$approximable for any fixed $\epsilon > 0$ then P has a polynomial time approximation scheme (PTAS). The dominating set problem for general graphs is $(1+\log n)-$approximable [7], however, it is not approximable within a factor $(1-\epsilon)\ln n$ for any $\epsilon > 0$ unless $NP \subseteq DTIME(n^{\log\log n})$ [8]. The dominating set problem has been widely studied on an important class of graphs, the *planar* graphs. A graph is planar if it can be drawn on the sphere with no crossing edges. The dominating set problem in planar graphs (planar dominating set problem) remains NP-hard [6] but admits a PTAS [9].

The fixed parameter algorithms have played a central role in exact algorithms for the planar dominating set problem. A minimization problem P is fixed-parameter tractable if given a parameter r, whether OPT of P is at most r can be decided in $f(r)n^{O(1)}$ time, where $f(r)$ is a computable function depending only on r [10]. Such an algorithm is called a *fixed parameter tractable* (FPT) algorithm. Readers may refer to [11] for a survey on new techniques for developing exact algorithms for NP-hard problems. It is shown in [10] that for general graphs, the dominating set problem is not fixed-parameter tractable unless some collapses occur between parametrized complexity classes. However, the planar dominating set problem is fixed-parameter tractable [10]. The planar dominating set problem also admits a linear size kernel [12].

Recent progresses in FPT algorithms result in subexponential time exact algorithms for the planar dominating set problem [13–15]. These algorithms use the tree-/branch-decomposition based approach and have running time $O(2^{c\sqrt{\gamma(G)}}n + n^{O(1)})$, c is a constant. The branch-decomposition based algorithm by Fomin and Thilikos (called FT algorithm in what follows) [14] achieves a smallest constant c in the exponent of the running time. The notion of branch-decomposition of graphs is introduced by Robertson and Seymour [16]. Informally, a branch-decomposition of a graph G is a collection of vertex cut sets of G that decomposes G into subgraphs with each edge of G a minimal subgraph. The width of a branch-decomposition is the maximum size of the vertex cuts in the collection. The branchwidth of G, denoted by $\mathrm{bw}(G)$, is the minimum width of all possible branch-decompositions of G. Given a graph G and a branch-decomposition of G with width β, FT algorithm finds an optimal solution in $O(2^{(3\log_4 3)\beta}n)$ time for the dominating set problem.

For a planar graph G, it is known that a branch-decomposition of minimum width $\mathrm{bw}(G)$ can be computed in $O(n^3)$ time [17,18] and $\mathrm{bw}(G) \leq 3\sqrt{4.5\gamma(G)}$ [14,19]. Alber *et al.* [12] give an $O(n^3)$ time algorithm which computes a subgraph H (kernel) of G such that H has $O(\gamma(G))$ vertices, $\gamma(H) \leq \gamma(G)$, and a minimum dominating set of G can be constructed from a minimum dominating set of H in linear time. Notice that for a subgraph H of G, $\mathrm{bw}(H) \leq \mathrm{bw}(G)$. From the above, the FT algorithm solves the planar dominating set problem in $O(2^{(3\log_4 3)\mathrm{bw}(G)}\gamma(G) + n^3)$ and $O(2^{15.13\sqrt{\gamma(G)}}\gamma(G) + n^3)$ time (The running time of FT algorithm can be further improved to $O(2^{11.98\sqrt{\gamma(G)}}\gamma(G) + n^3)$ using fast matrix multiplication in the dynamic programming step of the algorithm [20]. However, this improvement is only of theoretical interest because the fast matrix multiplication is not practical [21]).

For graphs with small treewidth/branchwidth, an FPT algorithm may be efficient to find an optimal solution, however, for graphs with large treewidth/branchwidth, one may have to rely on approximation algorithms for the planar dominating set problem. A PTAS is highly desired if the solution values are required to be close to optimal with a guaranteed approximation ratio. Baker introduces a framework to obtain PTAS for a class of NP-hard problems [9]. This framework is based on decomposing a planar graph into k-outer planar subgraphs.

A graph G is called *outer planar* or *1-outer planar* if it has a planar embedding such that all vertices of G are incident to a same face (called outer face). For $k > 1$, G is a *k-outer planar graph*, if it has a planar embedding such that removing the vertices of G incident to the outer face will result in a $(k-1)$-outer planar graph. A k-outer planar graph G has a branchwidth of at most $2k$. Baker's PTAS framework for a problem P in a planar graph G is to decompose G into a collection of k-outer planar subgraphs, find an optimal partial solution of P in each subgraph by an exact algorithm, and take the union of the optimal partial solutions as a solution of P in G. When the framework is used for a minimization problem, G is decomposed in such a way that every two "neighbor" k-outer planar subgraphs share "one-level" vertices. Baker shows that the framework gives a $2^{O(k)}n$ time $(1 + 1/k)$-approximation algorithm for the vertex cover problem in planar graphs and mentions that the framework can be applied to obtain $2^{O(k)}n$ time $(1 + 1/k)$-approximation algorithms for many other minimization problems, including the planar dominating set problem [9]. For a maximization problem like the independent set problem in planar graphs, Baker's framework gives a $2^{O(k)}n$ time $k/(k + 1)$-approximation algorithm.

We show that the approximation ratio of Baker's framework is not bounded by any constant for the planar dominating set problem when two "neighbor" k-outer planar subgraphs share only "one-level" of vertices. To get a PTAS for the planar dominating set problem, the application of the framework has to be modified. We modify the application of the framework by decomposing G into k-outer planar subgraphs such that every two "neighbor" subgraphs share "two-levels" of vertices. Let $O(2^{ck}n)$, c is a constant, be the running time of Baker's framework with "one-level" of overlapping vertices for the planar dominating set problem. We show that the modified application of the framework gives a PTAS with approximation ratio $(1 + 2/k)$ for the planar dominating set problem. By decomposing G into $2k$-outer planar subgraphs with "two-level" overlapping vertices, the modified PTAS achieves the approximation ratio $(1 + 1/k)$ in $O(2^{2ck}n)$ time.

In addition to the theoretical progresses in the algorithms for the dominating set problem, the practical performance of algorithms for the problem has received much attention. A computational study of an exact algorithm (FT algorithm) for the planar dominating set problem is reported in [21]. The study shows that the FT algorithm is practically efficient for graphs with small branchwidth. Heuristic algorithms for the dominating set problem have also been well investigated and a computational study of heuristic algorithms is reported in [22]. However, the practical performance of a PTAS is not known for the planar dominating set problem. One hurdle in evaluating the practical performance of a PTAS is the implementation of the algorithm. We conduct a computational study to evaluate the practical performance of the modified PTAS for the planar dominating set problem. In our implementation, the FT algorithm is used to compute an optimal partial solution in each k-outer planar subgraph. Our results show that the PTAS finds solutions with values very close to optimal in a practical time and much better than those given by well used heuristic algorithms. The computational study gives a concrete example on using a PTAS for solving important NP-hard problems in planar graphs and shows that the PTAS is practical for the planar dominating set problem. This work provides a tool for computing solutions close to optimal for the planar dominating set problem.

The next section gives preliminaries of the paper. In Section 3, we review Baker's framework, show that the approximation ratio of the framework is not bounded by a constant with "one-level" of overlapping vertices for the planar dominating set problem, and modify the application of the framework to give a PTAS for the problem. In Section 4, we report the computational study results. The final section concludes the paper.

2. Preliminaries

A graph G consists of a set $V(G)$ of vertices and a set $E(G)$ of edges, where each edge e of $E(G)$ is a subset of two elements from $V(G)$. For edge $e = \{u, v\} \in E(G)$, we say that vertices u and v are adjacent. The node degree of a vertex u is the number of vertices adjacent to u. Vertex u is dominated by vertex v if u and v are adjacent or $u = v$. Vertex u is dominated by a set D if u is dominated by a vertex of D. Edge e is covered by a vertex u if $u \in e$. For a subset $U \subseteq V(G)$ and a subset $A \subseteq E(G)$, we denote by $G[U]$ and $G[A]$ the subgraphs of G induced by U and A, respectively. For a subset $A \subseteq E(G)$, we denote by $E(G) \setminus A$ by $\overline{A}$ when G is clear from the context. A separation of graph G is a pair $(A, \overline{A})$ of subsets of $E(G)$. For each $A \subseteq E(G)$, we denote by $\partial(A)$ the vertex set $V(A) \cap V(\overline{A})$. The *order* of separation $(A, \overline{A})$ is $|\partial(A)| = |\partial(\overline{A})|$.

A graph G is planar if G has a planar embedding (a draw on a sphere without edge crossing). We call a planar embedding of G a plane graph. A face of a plane graph G is a connected region of the sphere bounded by edges and vertices of G and containing no edge or vertex of G in its interior. For a plane graph G and a face f of G, let $V_G(f)$ be the set of vertices in $V(G)$ incident to f. Given a plane graph G and a face f_o (called outer face) of G, let $V_1 = V_G(f_o)$. For $i \geq 1$, let $U_i = \cup_{j=1}^{i} V_j$, $G_i = G[V(G) \setminus U_i]$, f_i be the face of G_i such that $f_o \subseteq f_i$, and $V_{i+1} = V_{G_i}(f_i)$. We call the vertices of V_i *level* i vertices of G. Intuitively, G_i is the plane graph obtained from removing vertices of levels $1, 2, ..., i$ from G. V_{i+1} is the vertices of G_i incident to the outer face of G_i.

Branch-decomposition based algorithms play a central role in the PTAS studied in this paper. The notion of branch-decomposition is introduced by Robertson and Seymour [16]. A *branch-decomposition* of graph G is a pair (ϕ, T) where T is a tree each internal node of which has degree 3 and ϕ is a bijection from the set of leaves of T to $E(G)$. Consider a link e of T and let L_1 and L_2 denote the sets of leaves of T in the two respective subtrees of T obtained by removing e. We say that the separation $(\phi(L_1), \phi(L_2))$ is induced by this link e of T. We define the width of the branch-decomposition (ϕ, T) to be the largest order of the separations induced by links of T. The *branchwidth* of G, denoted by

$\mathrm{bw}(G)$, is the minimum width of all branch-decompositions of G. In the rest of this paper, we identify a branch-decomposition (ϕ, T) with the tree T, regarding each leaf of T as an edge of G.

Given a branch-decomposition T of G, an optimization problem P in G may be solved by the dynamic programming method as follows: convert T into a rooted binary tree by replacing a link $\{x, y\}$ of T with three links $\{x, z\}, \{y, z\}, \{z, r\}$, where z and r are new nodes to T, and r is the root of T. For a link $e = \{u, v\}$ of T, assume u is the end node reachable from root r by passing through e. Let A_e be the set of leaves of T reachable from r by passing through e. Link $e = \{u, v\}$ is called a leaf link if u is a leaf node, otherwise an internal link. An internal link e has two child links e_1 and e_2 covered by u. Notice that $A_e = A_{e_1} \cup A_{e_2}$. For a leaf link e, all partial solutions of P in the subgraph $G[A_e]$ can be computed by enumeration. For an internal link e, assume that all partial solutions of P in the subgraph $G[A_{e_1}]$ and those in $G[A_{e_2}]$ have been computed. Then all partial solutions of P in the subgraph $G[A_e]$ are computed by merging the partial solutions in $G[A_{e_1}]$ and those in $G[A_{e_2}]$. The merging process is performed in a bottom-up way, from each leaf link to the link $\{z, r\}$, to find an optimal solution of P in G.

The FT algorithm is a branch-decomposition-based algorithm for the planar dominating set problem. In FT Algorithm, the number of partial solutions in $G[A_e]$ is $3^{|\partial(A_e)|}$. To compute a partial solution in $G[A_e]$, every pair (s_1, s_2) is checked, where s_1 and s_2 are partial solutions in $G[A_{e_1}]$ and $G[A_{e_2}]$, respectively. Notice that each of $|\partial(A_e)|, |\partial(A_{e_1})|, |\partial(A_{e_2})|$ is at most the width of the given branch-decomposition T. When an optimal branch-decomposition T (of width $\mathrm{bw}(G)$) is given, the FT algorithm takes $O(2^{(3\log_4 3)\mathrm{bw}(G)})$ time and $O(3^{\mathrm{bw}(G)}\gamma(G))$ memory space to compute the partial solutions in $G[A_e]$. A planar graph G can be reduced to a kernel of size $O(\gamma(G))$ in $O(n^3)$ time and there are $O(\gamma(G))$ merging steps for the kernel. An optimal branch-decomposition of the kernel can be computed in $O((\gamma(G))^3)$ time. The FT algorithm solves the planar dominating set problem in $O(2^{(3\log_4 3)\mathrm{bw}(G)}\gamma(G) + n^3)$ time [14]. For many other NP-hard problems, branch-decomposition-based algorithms usually have exponential time and memory space in the width of a given branch-decomposition. The exponential time and memory space are often a bottle-neck in applying branch-decomposition-based algorithms in practice.

3. PTAS for Planar Dominating Set Problem

3.1. Baker's Framework for Minimization Problem

We review Baker's PTAS framework for minimization problems. We define the terminology for describing the framework. Given a plane graph G with m levels of vertices, for integers $2 \leq k < m$ and $2 \leq s \leq k + 1$, let $r = \lceil (m - s)/k \rceil$. We define $U(0, s) = \cup_{j=1}^{s} V_j$; $U(i, s) = \cup_{j=0}^{k} V_{(i-1)\times k+s+j}$ for $1 \leq i < r$; and $U(r, s) = \cup_{j=(r-1)\times k+s}^{m} V_j$. Then $G[U(0, s)]$ is the plane subgraph of G induced by the vertices of G with levels $1, ..., s$ and is s-outer planar; each $G[U(i, s)]$ is the subgraph of G induced by the vertices of G with levels $(i - 1) \times k + s, ..., i \times k + s$ and is $(k + 1)$-outer planar for $1 \leq i < r$; and $G[U(r, s)]$ is the subgraph of G induced by the vertices of G with levels $(r - 1) \times k + s, ..., m$ and is t-outer planar, where $t = m - [(r - 1) \times k + s] + 1 \leq k + 1$. Below is Baker's PTAS framework for minimization problems.

1. Let G be a plane graph with m levels of vertices for an outer face and $k \geq 2$ be an integer. Compute the vertex sets $V_1, ..., V_m$.
2. for $s = 2, ..., k+1$
   - (a) Compute subgraphs $G[U(i,s)]$ for $i = 0, 1, ..., r$.
   - (b) For every subgraph $G[U(i,s)]$, find an optimal solution $S(i,s)$ by an exact algorithm.
   - (c) Let $S_s = \cup_{i=0}^{r} S(i,s)$.
3. Let S be a set of $S_2, ..., S_{k+1}$ with the minimum cardinality.

Baker [9] gives a proof that the above framework achieves a $(1 + 1/k)$-approximation ratio for the minimum vertex cover problem in plane graph G: find a minimum subset C of $V(G)$ such that every edge of G is covered by a vertex in C. We review Baker's proof of the approximation ratio for the vertex cover problem. This proof gives a base on our later argument for the planar dominating set problem.

Given a plane graph G, let C be a minimum vertex cover of G. Given integer k, let $S(i,s)$ be a minimum vertex cover of $G[U(i,s)]$ and let $C(i,s) = C \cap U(i,s)$, $s = 2, ..., k+1$ and $0 \leq i \leq r$. Since no vertex of G in $V(G) \setminus U(i,s)$ covers any edge of $G[U(i,s)]$, $C(i,s)$ is a vertex cover of subgraph $G[U(i,s)]$. From this and the fact that $S(i,s)$ is a minimum vertex cover of $G[U(i,s)]$, $|S(i,s)| \leq |C(i,s)|$. Therefore, $S_s = \cup_{i=0}^{r} S(i,s)$ is a vertex cover of G and

$$|S_s| \leq \sum_{i=0}^{r} |S(i,s)| \leq \sum_{i=0}^{r} |C(i,s)| \tag{1}$$

Since the vertices of $V_{i \times k+s}$ appear in both subgraphs $G[U(i,s)]$ and $G[U(i+1,s)]$, $0 \leq i < r$,

$$\sum_{i=0}^{r} |C(i,s)| = |C| + \sum_{i=0}^{r-1} |C \cap V_{i \times k+s}| \tag{2}$$

Notice that

$$\min_{s=2}^{k+1} \{ \sum_{i=0}^{r-1} |C \cap V_{i \times k+s}| \} \leq \frac{|C|}{k} \tag{3}$$

Let S be a S_s with a minimum cardinality. Then from Inequalities (1), (2), and (3), we have $|S| \leq |C| + \frac{|C|}{k}$, that is, the solution produced by Baker's framework has the approximation ratio $(1 + 1/k)$ for the minimum vertex cover problem in planar graphs.

3.2. Modified Framework for Planar Dominating Set Problem

In [9], it is mentioned that the framework in the previous section can be applied to obtain a $(1+1/k)$-approximation algorithm for the planar dominating set problem. We show that this is not true. Recall that for the vertex cover problem, no vertex of G in $V(G) \setminus U(i,s)$ can cover any edge of $G[U(i,s)]$. This implies that for a minimum vertex cover C, $C(i,s) = C \cap U(i,s)$ is a vertex cover of subgraph $G[U(s,i)]$ and a minimum vertex cover $S(i,s)$ of $G[U(i,s)]$ has the property $|S(i,s)| \leq |C(i,s)|$. However, for the planar dominating set problem, the intersection of a minimum dominating set of G and $U(i,s)$ may not be a dominating set of $G[U(i,s)]$ because a vertex of G in $V(G) \setminus U(i,s)$ can dominate a vertex of $G[U(i,s)]$. More specifically, let D be a minimum dominating set of G, $D(i,s) = D \cap U(i,s)$ and $S(i,s)$

be a minimum dominating set of $G[U(i, s)]$. Then $D(i, s)$ may not be a dominating set of $G[U(i, s)]$ and $|S(i, s)| \leq |D(i, s)|$ may not hold. Below we show by an example that the approximation ratio of the mentioned application of Baker's framework is not bounded by any constant for the planar dominating set problem.

Let G be a plane graph with 4 levels of vertices shown in Figure 1. Let X_i be the set of vertices of G with labels $(i, 1), ..., (i, x)$, $1 \leq i \leq 6$. The subgraph $G[X_i]$ is a chain and there is a unique vertex in G dominating all vertices of X_i. For a large x, G has a unique minimum dominating set D with its six vertices shown as black squares in the figure. Let $k = 2$. For $s = 2$, the subgraphs $G[U(0, 2)]$ and $G[U(1, 2)]$ are shown in Figure 2 (a) and (b), respectively. Let $D(0, 2) = D \cap U(0, 2)$. Then $D(0, 2)$ (the set of vertices denoted by black squares) is not a dominating set of $G[U(0, 2)]$. On the other hand, a minimum dominating set $S(0, 2)$ of $G[U(0, 2)]$ contains a fraction of vertices in X_3 and $|S(0, 2)| > |D(0, 2)|$ for large $x = |X_3|$. Similarly, a minimum dominating set $S(1, 2)$ of $G[U(1, 2)]$ contains a fraction of vertices in X_2 and $|S(1, 2)| > |D(1, 2)|$. Let $S_2 = S(0, 2) \cup S(1, 2)$. Then $|S_2|/|D|$ is not bounded by any constant for non-constant x.

Figure 1. A plane graph G with four levels of vertices.

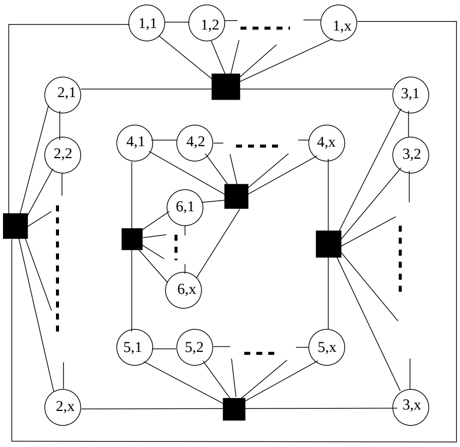
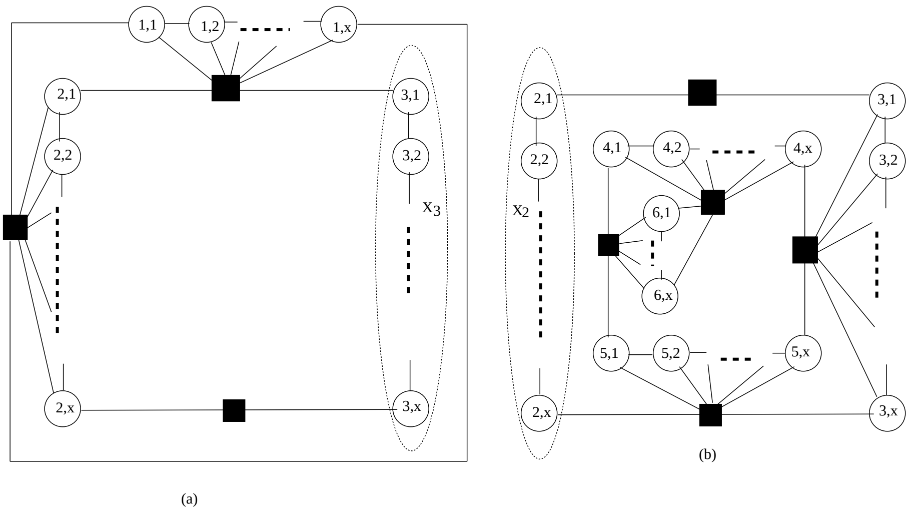
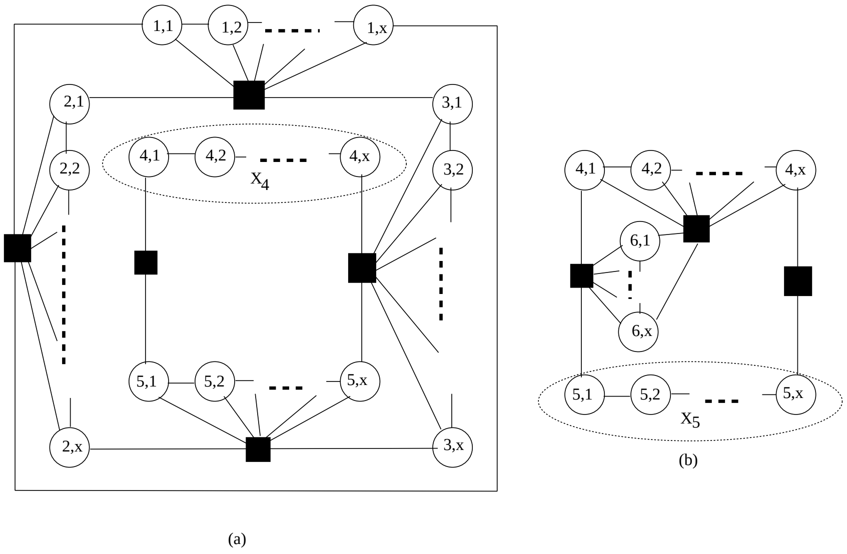


Figure 2. (**a**) Subgraph $G[U(0, 2)]$ and (**b**) Subgraph $G[U(1, 2)]$ of G.



For $s = 3$, the subgraphs $G[U(0, 3)]$ and $G[U(1, 3)]$ are shown in Figure 3 (a) and (b), respectively. Then a minimum dominating set $S(0, 3)$ of $G[U(0, 3)]$ contains a fraction of vertices in X_4 and a minimum dominating set $S(1, 3)$ of $G[U(1, 3)]$ contains a fraction of vertices in X_5. Let $S_3 = S(0, 3) \cup S(1, 3)$. Then $|S_3|/|D|$ is not bounded by any constant for non-constant $x = |X_4|$. Therefore, for a set S of S_2 and S_3 with the minimum cardinality, $|S|/|D|$ is not bounded by any constant for non-constant x.

Figure 3. (**a**) Subgraph $G[U(0, 3)]$ and (**b**) Subgraph $G[U(1, 3)]$ of G.



We modify the application of Baker's framework to get a PTAS for the planar dominating set problem. The idea for the modification is that instead of decomposing G into $(k + 1)$-outer planar subgraphs with two neighbor subgraphs $G[U(i, s)]$ and $G[U(i+1, s)]$ overlapping on one level of vertices, we decompose G into $(k+2)$-outer planar subgraphs with two neighbor subgraphs overlapping on two levels of vertices. For each subgraph, we find a minimum set which dominates only k levels of vertices in the subgraph. The formal modification is described below.

Let G be a plane graph with m levels of vertices. For integers $2 \leq k < m$ and $2 \leq s \leq k + 1$, let $r = \lceil (m - s)/k \rceil$. We define $W(0, s) = \cup_{j=1}^{s} V_j$; $W(i, s) = \cup_{j=-1}^{k} V_{(i-1)\times k+s+j}$ for $1 \leq i < r$; and $W(r, s) = \cup_{j=(r-1)\times k+s-1}^{m} V_j$. $G[W(0, s)]$ is the subgraph of G induced by the vertices of G with levels $1, ..., s$ and is s-outer planar; each $G[W(i, s)]$ is the subgraph of G induced by the vertices of G with levels $(i - 1) \times k + s - 1, ..., i \times k + s$ and is $(k + 2)$-outer planar for $1 \leq i < r$; and $G[W(r, s)]$ is the subgraph of G induced by the vertices of G with levels $(r - 1) \times k + s - 1, ..., m$ and is t-outer planar, where $t = m - [(r - 1) \times k + s] + 2 \leq k + 2$. We call the vertices of $G[W(0, s)]$ with level s, the vertices of each subgraph $G[W(i, s)]$ $(1 \leq i < r)$ with levels $(i - 1) \times k + s - 1$ and $i \times k + s$, and the vertices of $G[W(r, s)]$ with level $(r - 1) \times k + s - 1$ the *vertices on boundary*; and call the other vertices the *interior vertices*.

1. Let G be a plane graph with m levels of vertices for an outer face and $k \geq 2$ be an integer. Compute the vertex sets $V_1, ..., V_m$.
2. for $s = 2, ..., k + 1$

(a) Compute subgraphs $G[W(i,s)]$ for $i = 0, 1, ..., r$.

(b) For subgraph $G[W(0,s)]$, find a minimum subset $S(0,s)$ of $W(0,s)$ that dominates every vertex of $\cup_{j=1}^{s-1} V_j$ (every interior vertex).
For every subgraph $G[W(i,s)]$, $i = 1, ..., r-1$, find a minimum subset $S(i,s)$ of $W(i,s)$ that dominates every vertex of $\cup_{j=(i-1)\times k+s}^{i\times k+s-1} V_j$ (every interior vertex).
For subgraph $G[W(r,s)]$, find a minimum subset $S(r,s)$ of $W(r,s)$ that dominates every vertex of $\cup_{j=(r-1)\times k+s}^{m} V_j$ (every interior vertex).

(c) Let $S_s = \cup_{i=0}^{r} S(i,s)$.

3. Let S be a set of $S_2, ..., S_{k+1}$ with the minimum cardinality.

Theorem 3.1 *The modified application of Baker's framework gives an* $O(2^{(6\log_4 3)(k+2))} kn)$ *time* $(1 + 2/k)$*-approximation algorithm for the planar dominating set problem.*

Proof: We first show the approximation ratio of the framework. Notice that $S_s = \cup_{i=0}^{r} S(i,s)$ dominates every vertex of

$$(\cup_{j=1}^{s-1} V_j) \cup [\cup_{i=1}^{r-1} (\cup_{j=(i-1)\times k+s}^{i\times k+s-1} V_j)] \cup (\cup_{j=(r-1)\times k+s}^{m} V_j) = \cup_{j=1}^{m} V_j = V(G)$$

that is, S_s is a dominating set of G. Let D be a minimum dominating set of G and let $D(i,s) = D \cap W(i,s)$, $s = 2, ..., k+1$ and $0 \le i \le r$. Since no vertex of G in $V(G) \setminus W(i,s)$ can dominate any interior vertex of $G[W(i,s)]$ and D dominates every vertex of G, $D(i,s)$ dominates every interior vertex of $G[W(i,s)]$. From this and the fact that $S(i,s)$ is a minimum subset of $W(i,s)$ dominating every interior vertex of $G[W(i,s)]$, $|S(i,s)| \le |D(i,s)|$. From this, we have

$$|S_s| \le \sum_{i=0}^{r} |S(i,s)| \le \sum_{i=0}^{r} |D(i,s)| \tag{4}$$

Since the vertices of $V_{i\times k+s-1}$ and $V_{i\times k+s}$ appear in subgraphs $G[W(i,s)]$ and $G[W(i+1,s)]$, $0 \le i < r$,

$$\sum_{i=0}^{r} |D(i,s)| \le |D| + \sum_{i=0}^{r-1} |D \cap V_{i\times k+s-1}| + |D \cap V_{i\times k+s}| \tag{5}$$

Notice that

$$\min_{s=2}^{k+1} \{\sum_{i=0}^{r-1} |D \cap V_{i\times k+s-1}| + |D \cap V_{i\times k+s}|\} \le \frac{2|D|}{k} \tag{6}$$

Let S be an S_s with a minimum cardinality. Then from Inequalities (4), (5), and (6), we have $|S| \le |D| + \frac{2|D|}{k}$. that is, the solution produced by the modified algorithm has the approximation ratio $(1 + 2/k)$ for the planar dominating set problem.

Given a planar graph G, a planar embedding of G can be computed in linear time [23]. It is obvious that Step 1 and Step 3 can be computed in linear time. Step 2 (a) and (c) can be computed in $O(kn)$ time. Recall that FT Algorithm (by Fomin and Thilikos [14]) is the most efficient known exact algorithm for the planar domінaing set problem. We use FT Algorithm for Step 2 (b). Given a graph G and a branch-decomposition of G with width β, FT Algorithm finds an optimal solution for the dominating set problem in $O(2^{(3\log_4 3)\beta} n)$ time. Each subgraph $G[W(i,s)]$ is $(k+2)$-outer planar and has branchwidth at most $2(k+2)$. A branch-decomposition of $G[W(i,s)]$ with width at most $2(k+2)$ can be computed in linear time [24]. Each vertex of G appears in at most two subgraphs for a specific value s. Therefore, Step 2 (b) takes

$$\sum_{s=2}^{k+1}\sum_{i=0}^{r} O(2^{(6\log_4 3)(k+2)}|W(i,s)|) = \sum_{s=2}^{k+1} O(2^{(6\log_4 3)(k+2)}n) = O(2^{(6\log_4 3)(k+2)}kn)$$

time which is the dominating part of the modified application's running time. []

Notice that if G is decomposed into $(2k+2)$-outer planar subgraphs, the modified PTAS achieves the approximation ratio $(1+1/k)$ and has running time $O(2^{(12\log_4 3)(k+1)}kn)$.

We conclude this section by comparing the running time of the modified PTAS with that of the application of Baker's framework in Section 3.1 for the planar dominating set problem. We assume that the most efficient FT Algorithm is used in Step 2 (b) of both algorithms. Assume that G is decomposed into $(k+1)$-outer planar subgraphs in Baker's framework in Section 3.1. Then a branch-decomposition of $G[U(i,s)]$ with width at most $2(k+1)$ can be computed in linear time and Step 2 (b) takes

$$\sum_{s=2}^{k+1}\sum_{i=0}^{r} O(2^{(6\log_4 3)(k+1)}|U(i,s)|) = O(2^{(6\log_4 3)(k+1)}kn)$$

time which is the dominating part of the framework's running time. The constant in the exponent of the running time of the modified PTAS for the approximation ratio $(1+1/k)$ is as twice as that of Baker's framework in Section 3.1.

4. Computational Study of PTAS

We study the practical performance of the PTAS for the planar dominating set problem. The PTAS is implemented in C++ and its performance is tested for four different classes of graphs including the Delaunay triangulations of point sets taken from TSPLIB [25] (Class (1)), triangulations (Class (2)) and intersection graphs (Class (3)) generated by LEDA [26] and Gabriel graphs (Class (4)) generated using the points uniformly distributed in a two-dimensional plane. Those classes of graphs are well used in previous computational studies and the branchwidth of the graphs increases in the size of graphs (for classes of graphs with small branchwidth such as the maximal random planar graphs from LEDA [26], the FT algorithm can find optimal solutions efficiently [21] and thus they are not interesting in this study). The computer used for testing has an AMD Athlon(tm) 64 X2 Dual Core Processor 4600+ (2.4 GHz) and 3 GByte of internal memory. The operating system is SUSE Linux 10.2.

We use the FT algorithm to compute an optimal solution for each $(k+2)$-outer planar subgraph. There are three major steps of FT Algorithm:

1. Compute a linear size kernel H of the subgraph using the $O(n^3)$ time kernelization algorithm by Alber *et al.* [12].
2. Compute an optimal branch-decomposition of H by the $O(n^3)$ time algorithm [17,18].
3. Find an optimal solution for H by dynamic programming based on the branch-decomposition of H and compute an optimal solution for the subgraph from the optimal solution for H.

Step 3 has exponential time complexity and memory complexity in the width of the branch-decomposition used, and is the dominating part in the running time and used memory of the FT algorithm. Thus we include the kernelization in the FT algorithm because, for a kernel H of a graph G, $\mathrm{bw}(H) \leq \mathrm{bw}(G)$ and it often happens that $\mathrm{bw}(H) < \mathrm{bw}(G)$ for a kernel computed. Also, the effort for computing an optimal branch-decomposition reduces the running time and used memory in practice. For a planar graph G, the FT algorithm implemented runs in $O(2^{(3\log_4 3)\mathrm{bw}(H)}\gamma(G) + n^3)$ time and uses $O(3^{\mathrm{bw}(H)}\gamma(G))$ memory space. Readers may refer to [21] for more details on the practical performance of the FT algorithm.

Table 1 shows the computational results of the PTAS for the planar dominating set problem. For every instance, we calculate the approximated solutions for two different values of k, 3 and 4, and, for every value of k, we calculate the $(k+2)$-outer planar decomposition for every face of the instance. We choose the best value for an approximated solution. For some instances with small branchwidth, we also include the optimal solutions computed by the FT algorithm and reported in [21] in the column of ”Exact Alg.”. The size of a minimum dominating set of graph G, computed by the FT algorithm, is indicated by $\gamma(G)$ in Table 1, and for every value of k, D_{PTAS} is the size of dominating set computed by the PTAS. In the table, bw is the branchwidth of G, β is the branchwidth of a kernel H of G in the FT algorithm and the largest branchwidth of a kernel H of a $(k+2)$-outer planar subgraph in PTAS. The running time is in seconds. For two large instances rand16000 and rand20000, we only compute $\gamma(G)$ but not the minimum dominating sets by the FT algorithm due to the memory constraint. These values of $\gamma(G)$ are identified by “*”.

In order to compare the size of dominating sets obtained from the PTAS with the optimal solutions, we include some instances with small branchwidth for every class of graphs, such that a minimum dominating set can be computed by FT Algorithm. The Exact Alg. column shows the results of FT Algorithm reported in [21]. We use two values for k to decompose the instances into $(k+2)$-outer planar component. Notice that the branchwidth of every $(k+2)$-outer planar graph is at most $2(k+2)$. Hence, by increasing k the size of subgraphs and their branchwidth will increase. Theoretical results suggest that increasing k gives smaller approximated solutions for minimization problems. Our computing results confirm the theoretical analysis of the k-outer planar decomposition method. For example, for $k = 4$, every instance can be decomposed into subgraphs with a branchwidth of at most 12. This is the largest value of branchwidth that can be processed on our computational platform in a practical time.

Since the theory of NP-completeness has reduced hopes that NP-hard problems can be solved in polynomial time, heuristic and approximation algorithms have attracted more attentions. These algorithms compute near optimal solutions within a reasonable time for problems of practical size. We compare the performance of the PTAS with the performance of three different heuristic algorithms introduced in [22] for the planar dominating set problem. In what follows we briefly explain these heuristic algorithms (for more details please refer to [22]).

In [22], six heuristic algorithms for the dominating set problem are studied. We test the performance of these six methods, but only report three of them with better performances. The three reported heuristics are described below. Let D be a dominating set computed by these heuristics.

Table 1. Computational results (time in seconds) of PTAS for the planar dominating set problem.

	Graph	$\lvert E(G)\rvert$	*bw*	**Exact Alg.**			**k = 3**			**k = 4**		
	G			$\gamma(G)$	β	**time**	D_{PTAS}	β	**time**	D_{PTAS}	β	**time**
(1)	kroB150	436	10	23	10	10	28	8	2.07	-	-	-
	pr299	864	11	47	11	37	56	10	11.42	-	-	-
	tsp225	622	12	37	12	110	46	9	5.21	-	-	-
	a280	788	13	43	13	337	53	10	8.40	51	12	12.09
	rd400	1183	17	-	-	-	75	10	35.30	74	12	351.93
	pcb442	1286	17	-	-	-	79	10	10.46	78	10	10.86
	d657	1958	22	-	-	-	123	10	64.89	120	12	604.10
	pr1002	2972	21	-	-	-	190	10	115.65	182	12	1253.9
(2)	tri2000	5977	8	321	7	198	361	7	175.59	-	-	-
	tri4000	11969	9	653	7	1903	724	7	733.06	-	-	-
	tri6000	17979	9	975	8	3576	1136	8	1994.53	-	-	-
	tri8000	23975	9	1283	7	7750	1430	7	2858.63	-	-	-
	tri10000	29976	9	1606	7	16495	1804	7	4977.06	-	-	-
	tri11000	32972	14	-	-	-	1987	8	5910.8	1958	8	12341.1
	tri12000	35974	14	-	-	-	2164	7	5370.18	2132	7	6865.08
	tri14000	41974	15	-	-	-	2514	7	8220.49	2434	7	9208.72
	tri16000	47969	16	-	-	-	2920	7	10060.1	2885	7	12794.4
(3)	rand6000	10293	11	1563	9	150	1658	8	104.85	-	-	-
	rand10000	17578	13	2535	10	869	2850	8	535.87	2692	9	432.23
	rand15000	26717	14	3758	12	2769	4144	10	1313.14	-	-	-
	rand16000	28624	13	4002*	13	5917	4379	10	2443.27	4295	11	2027.7
	rand20000	35975	14	4963*	14	13993	5465	10	4241.65	5368	12	5017.02
	rand25000	40378	16	-	-	-	7101	8	6407.91	6632	12	9470
(4)	Gab500	949	13	115	12	238	136	10	18.02	129	10	18.95
	Gab600	1174	14	135	14	3074	164	10	26.05	156	10	22.10
	Gab700	1302	14	162	14	5710	187	10	22.81	183	10	24.30
	Gab800	1533	17	-	-	-	225	10	51.82	205	12	24.30
	Gab900	1758	17	-	-	-	243	10	48.39	231	12	344.50
	Gab1000	1901	18	-	-	-	260	10	49.69	259	12	781.89
	Gab1500	2870	21	-	-	-	402	10	116.37	385	12	960.71

Greedy: Initially, D is empty. In each iteration, a vertex which dominates a maximum number of vertices in $V(G) \setminus D$ is added to D.

Greedy-Rev: Initially $D = V(G)$. In each iteration, a vertex is removed from D, such that the resulting set remains a dominating set of G. A vertex is chosen to be removed, by ordering the vertices of D in increasing node degree, and removing the first vertex that does not dominate any vertex uniquely.

Greedy-Vote: Initially, D is empty. This algorithm does not include a vertex u in D only based on the number of vertices which are dominated by u. It uses a more complex voting scheme to select a vertex to be included. We omit the details of the selection scheme and readers may refer to [22] for details.

Table 2. Computational results for heuristic algorithms and PTAS for the planar dominating set problem (time in seconds).

	Graph	$\|E(G)\|$	$\gamma(G)$	**Greedy Alg.**		**Greedy-Rev Alg.**		**Greedy-Vote Alg.**		**PTAS**	
	G			D_{Gr}	**time**	D_{Rev}	**time**	D_{Vote}	**time**	D_{PTAS}	**time**
(1)	kroB150	436	23	27	0.002	31	0.01	31	0.002	28	2.08
	pr299	864	47	54	0.003	63	0.032	62	0.005	56	11.42
	tsp225	622	37	49	0.153	54	0.02	50	0.003	46	5.21
	a280	788	43	51	0.004	62	0.025	62	0.006	51	12.09
	rd400	1183	-	78	0.007	92	0.032	90	0.009	74	351.93
	pcb442	1286	-	76	0.908	90	0.063	87	0.01	78	10.86
	d657	1958	-	126	0.016	146	0.128	143	0.021	120	604.10
	pr1002	2972	-	190	0.032	236	0.328	194	0.04	182	1253.9
(2)	tri2000	5977	321	365	0.116	379	1.119	464	0.168	361	175.59
	tri4000	11969	653	729	0.183	765	1.792	787	0.544	724	733.06
	tri6000	17979	975	1118	0.418	1166	4.14	1306	0.541	1136	1994.53
	tri8000	23975	1283	1449	0.715	1522	7.003	1653	0.918	1430	2858.63
	tri10000	29976	1606	1819	1.117	1906	11.524	2302	1.572	1804	4977.06
	tri11000	32972	-	2040	1.375	2116	14.092	3431	2.561	1958	12341.1
	tri12000	35974	-	2186	1.607	2278	16.538	2741	2.243	2132	6865.08
	tri14000	41974		2576	2.462	2664	22.976	3317	3.163	2434	9208.72
	tri16000	47969	-	2917	2.839	3033	30.694	3684	4.005	2885	12794.4
(3)	rand6000	10293	1563	1932	0.748	2166	4.517	2908	1.206	1658	104.85
	rand10000	17578	2535	3197	2.06	3618	13.33	4164	2.878	2692	432.23
	rand15000	26717	3758	4698	4.861	5402	29.487	7277	7.641	4144	1313.14
	rand16000	28624	4002*	5039	5.176	5744	35.589	7552	10.327	4295	2027.7
	rand20000	35975	4963*	6273	8.053	7168	55.948	8571	11.903	5398	5017.02
	rand25000	45327	-	7772	12.467	8942	91.039	11865	20.615	6632	9470
(4)	Gab500	949	115	146	0.006	173	0.039	160	0.007	129	18.95
	Gab600	1174	135	168	0.007	199	0.051	171	0.009	156	22.10
	Gab700	1302	162	200	0.01	242	0.072	238	0.012	183	24.30
	Gab800	1533	-	227	0.012	270	0.097	307	0.019	205	24.30
	Gab900	1758	-	254	0.016	303	0.103	323	0.022	231	344.50
	Gab1000	1901	-	280	0.019	344	0.146	423	0.03	259	781.89
	Gab1500	2870	-	426	0.042	507	0.335	496	0.051	385	960.71

We study the performances of the above heuristic algorithms for the four classes of planar graphs that are used in the study of the PTAS. These heuristic algorithms are implemented in C++. Table 2 shows the computational results of these heuristic algorithms and the PTAS. In Table 2, D_{Gr}, D_{Rev}, and D_{Vote} are the sizes of dominating sets computed by the heuristic algorithm Greedy, Greedy-Rev, and Greedy-Vote, respectively. For every graph instance, if the size of the instance allows the application of the FT algorithm, we include the size of the minimum dominating set of the instance, as well. For the PTAS, we include the best result D_{PTAS} for every instance from Table 1. Time in the table is in seconds.

The results in the table show that the heuristic algorithms are always faster than the PTAS. However, the size of dominating sets computed by the heuristics are larger than those by the PTAS for most of instances.

Based on our computational results, the Greedy algorithm gives the smallest dominating sets compared to other heuristic algorithms. Table 3 shows the results of our computational study for the FT algorithm, Greedy (the best heuristic method) and PTAS for graph instances whose branchwidths are small enough to run the FT algorithm.

Table 3. Computational results for Exact, Greedy and PTAS algorithms for small instances (time in seconds).

	Graph	$\lvert E(G)\rvert$	**Exact Alg.**		**Greedy Alg.**		**PTAS**	
	G		$\gamma(G)$	**time**	D_G	**time**	D_{PTAS}	**time**
(1)	kroB150	436	23	10	27	0.002	28	2.08
	pr299	864	47	37	54	0.032	56	11.42
	tsp225	622	37	110	49	0.153	46	5.21
	a280	788	43	337	51	0.004	51	12.09
(2)	tri2000	5977	321	198	365	0.116	361	175.59
	tri4000	11969	653	1903	729	0.183	724	733.06
	tri6000	17979	975	3576	1118	0.418	1136	1994.53
	tri8000	23975	1283	7750	1449	0.715	1430	2858.63
	tri10000	29976	1606	16495	1819	1.117	1804	4977.06
(3)	rand6000	10293	1563	150	1932	0.748	1658	104.85
	rand10000	17578	2535	869	3197	2.06	2692	432.23
	rand15000	26727	3758	2769	4698	4.861	4144	1313.14
	rand16000	28624	4002*	5917	5039	5.176	4295	2027.7
	rand20000	35975	4963*	13993	6273	8.053	5398	5017.02
(4)	Gab500	949	115	238	146	0.006	129	18.95
	Gab600	1174	135*	3074	168	0.007	156	22.10
	Gab700	1302	162*	5710	200	0.01	183	24.30

Since the branchwidth of graphs in Class(1) grow quickly in the size of graphs, we have only included small instances of this class in Table 3. From the results of the table, we recommend the FT algorithm for optimal solutions if the branchwidth of a graph in Class (1) is smaller than 14. For the instances of Class (2), FT Algorithm is time consuming. If the running time is the driving factor, we suggest the Greedy algorithm for this class of graphs. For the instances of Classes (3) and (4), as the results in Table 3 suggest, the sizes of dominating sets computed by Greedy are considerably bigger than those computed by the PTAS. Moreover, the FT algorithm is time consuming, rendering the PTAS a better choice. For instance, for graph instance rand20000 with 35,975 edges, the FT algorithm takes almost four hours to compute the size of an optimal dominating set (not the set itself), while the PTAS computes a dominating set of a slightly larger size than the optimal value in less than two hours.

Table 4 shows the computational results for the instances that the FT algorithm is not able to find an optimal solution in practical time and memory space. The computational results show that for all of these instances, except one, the D_{PTAS} is smaller than D_{Gr}. In summary, for applications with running-time priority, Greedy is a better choice to compute an approximated dominating set, and if the running-time is not a big concern, the PTAS is a better option for instances whose optimal dominating set cannot be computed by the FT algorithm in a practical time.

Table 4. Computational results for Greedy and PTAS for large instances (time in seconds).

	Graph	$\vert E(G)\vert$	**Greedy Alg.**		**PTAS**	
	G		D_{Gr}	**time**	D_{PTAS}	**time**
(1)	rd400	1183	78	0.007	74	351.93
	pcb442	1286	76	0.908	78	10.86
	d657	1958	126	0.016	120	604.10
	pr1002	2972	190	0.032	182	1253.9
(2)	tri11000	32972	2040	1.375	1958	12341.1
	tri12000	35974	2186	1.607	2132	6865.08
	tri14000	41974	2576	2.462	2434	9208.72
	tri16000	47969	2917	2.839	2885	12794.4
(3)	rand25000	45327	7772	12.467	6632	9470
(4)	Gab800	1533	227	0.012	205	24.30
	Gab900	1758	254	0.016	231	344.50
	Gab1000	1901	280	0.019	259	781.89
	Gab1500	2870	426	0.042	385	960.71

5. Concluding Remarks

It is mentioned that Baker's k-outer planar graph decomposition framework can be applied to obtain a PTAS for the planar dominating set problem. We show that, in order to get a PTAS for the planar dominating set problem, the mentioned application needs some modification. We modify the application and give a PTAS for the planar dominating set problem. We also report a computational study on the modified PTAS. Computational studies on exact algorithms and heuristic algorithms for the planar dominating set problem have already been conducted, but no report on PTAS has yet been given. Our study on the PTAS makes the computational study of planar dominating set problem more comprehensive. For larger k, the PTAS gives better solutions, but is more time/memory consuming. Due to the computation platform limitation, we only evaluated the PTAS for small k. It would be interesting to test the practical performances of the PTAS for larger k on more powerful computation platforms. The practical performances of PTASes for other optimization problems in planar graphs are worth investigation.

Acknowledgements

The authors thank the anonymous reviewers for their constructive comments.

References

1. Berge, C. *Graphs and Hypergraphs*; American Elsevier: New York, NY, USA, 1973.
2. Liu, C. *Introduction to Combinatorial Mathematics*; McGraw-Hill: New York, NY, USA, 1963.
3. Norman, R.; Harary, F.; Cartwright, D. *Structural Models: An Introduction to the Theory of Directed Graphs*; Wiley: Weinheim, Germany, 1966.
4. Haynes, T.W.; Hedetniemi, S.T.; Slater, P.J. Domination in Graphs. In *Monographs and Textbooks in Pure and Applied Mathematics*; Marcel Dekker: New York, NY, USA, 1998.
5. Haynes, T.W.; Hedetniemi, S.T.; Slater, P.J. Fundamentals of Domination in Graphs. In *Monographs and Textbooks in Pure and Applied Mathematics*; Marcel Dekker: New York, NY, USA, 1998.
6. Garey, M.R.; Johnson, D.S. *Computers and Intractability, a Guide to the Theory of NP-Completeness*; Freeman: New York, NY, USA, 1979.
7. Johnson, D.S. Approximation algorithms for combinatorial problems. *J. Comput. Syst. Sci* **1974**, *9*, 256–278.
8. Fiege, U. A threshold of $\ln n$ for approximating set cover. *J. ACM* **1998**, *45*, 634–652.
9. Baker, B.S. Approximation algorithms for NP-complete problems on planar graphs. *J. ACM* **1994**, *41*, 153–180.
10. Downey, R.G.; Fellows, M.R. Parameterized Complexity. In *Monographs in Computer Science*; Springer-Verlag: Berlin/Heidelberg, Germany, 1999.
11. Fomin, F.V.; Grandoni, F.; Kratch, D. Some new techniques in design and analysis of exact (exponential) algorithms. *Bull. EATCS* **2005**, *87*, 47–77.
12. Alber, J.; Fellows, M.R.; Niedermeier, R. Polynomial time data reduction for dominating set. *J. ACM* **2004**, *51*, 363–384.
13. Alber, J.; Bodlaender, H.L.; Fernau, H.; Kloks, T.; Niedermeier, R. Fixed parameter algorithms for dominating set and related problems on planar graphs. *Algorithmica* **2002**, *33*, 461–493.
14. Fomin, F.V.; Thilikos, D.M. Dominating sets in planar graphs: Branch-width and exponential speed-up. *SIAM J. Comput.* **2006**, *36*, 281–309.
15. Kanj, I.A.; Perkovic, L. Improved Parameterized Algorithms for Planar Dominating Set. In *Proceedings of the 27th Mathematical Foundations of Computer Science. LNCS 2420*, Warsaw, Poland, Augaust, 2002; pp. 399–410.
16. Robertson, N.; Seymour, P.D. Graph minors X. Obstructions to tree decomposition. *J. Comb. Theory Ser. B* **1991**, *52*, 153–190.
17. Gu, Q.; Tamaki, H. Optimal branch-decomposition of planar graphs in $O(n^3)$ time. *ACM Trans. Algorithm* **2008**, *4*, 30:1–30:13.
18. Seymour, P.D.; Thomas, R. Call routing and the ratcatcher. *Combinatorica* **1994**, *14*, 217–241.
19. Fomin, F.V.; Thilikos, D.M. New upper bounds on the decomposability of planar graphs. *J. Graph Theory* **2006**, *51*, 53–81.
20. Dorn, F. Dynamic Programming and Fast Matrix Multiplication. In *Proceedings of the 14th Annual European Symposium on Algorithms (ESA2006) LNCS 4168*, Zurich, Switerland, September, 2006; pp. 280–291.

21. Marzban, M.; Gu, Q.; Jia, X. Computational study on planar dominating set problem. *Theor. Comput. Sci.* **2009**, *410*, 5455–5466.
22. Sanchis, L.A. Experimental analysis of heuristic algorithms for the dominating set problem. *Algorithmica* **2002**, *33*, 3–18.
23. Hopcroft, J.; Tarjan, R. Efficient planarity testing. *J. ACM* **1974**, *21*, 549–568.
24. Tamaki, H. A linear Time Heuristic for the Branch-decomposition of Planar Graphs. In *Proceedings of the 11th Annual European Symposium*, Budapest, Hungary, 16–19 September 2003; pp. 765–775.
25. Reinelt, G. TSPLIB-A traveling salesman library. *ORSA J. Comput.* **1991**, *3*, 376–384.
26. Library of Efficient Data Types and Algorithms, Version 5.2. Available online: http://www.algorithmic-solutions.com/leda/index.html (accessed on 1 July 2007).