Parameterized tractability of some (efficient) Y-domination variants for planar graphs and t-degenerate graphs *

Preliminary version

Leizhen Cai   Ton Kloks

Department of Computer science and Engineering
The Chinese University of Hong Kong
Shatin, New Territories
Hong Kong, China
lcai@cse.cuhk.edu.hk,kloks@cs.vu.nl

Abstract

In this paper we show that the (efficient) Y-domination problem is fixed-parameter tractable when restricted to planar graphs. Furthermore, we show that the efficient domination problem, a special case of the efficient Y-domination problem, is fixed-parameter tractable when restricted to t-degenerate graphs with fixed t.

1 Introduction

Let $G = (V, E)$ be a graph. For a given a set Y of integers a Y-assignment is a function $f : V \to Y$ such that every closed neighborhood sum is at least one (i.e., $\sum_{y \in N[x]} f(y) \geq 1$ for every vertex $x \in V$, where $N[x]$ is the closed neighborhood of x). A Y-assignment is called *efficient* if every closed neighborhood sum is exactly one.

Well known, and most studied examples of Y-domination problems are the efficient domination assignment ($Y = \{0, 1\}$), the minus assignment ($Y = \{-1, 0, +1\}$) and the signed assignment ($Y = \{-1, +1\}$) problems.

An algorithm for a fixed-parameter problem (I, k), where I is an instance and k is the parameter, is *uniformly polynomial* if it runs in time $O(f(k)|I|^c)$, where $|I|$ is the size of I, for an arbitrary function $f(k)$ and a constant c independent of k. A fixed-parameter problem is *fixed-parameter tractable* if it admits a uniformly polynomial algorithm.

As the parameter for the problems we consider in this paper, we chose the *value*, i.e., the number of vertices with a *positive* number assigned to it. (We chose this parameter quite arbitrarily and for many cases it seems a natural choice. For many cases we could just as well choose the *weight* as the parameter, i.e., the sum over the vertices of all assigned values.)

We mention some definitions for the efficient domination, efficient minus domination and efficient signed domination in the next section as examples of Y-dominating problems. In this introduction we mention some of the latest results on the complexity of these special cases.

First of all, the efficient domination problem is W[1]-hard (and in W[2]) as was shown in [10]. In this book, the problem is referred to as the perfect code problem, and the authors express their believe that the hardness of the problem could be "between" W[1] and W[2]. At present, this is the only hardness result of this type we are aware of for the three mentioned special cases.

We turn to the NP-completeness results of the three variants for special graph classes, since this issue was well-studied over the last few years. We mention some results.

It is known that the efficient dominating set problem is NP-complete, even when restricted to chordal graphs [20], chordal bipartite graphs [16], planar bipartite graphs [16], and planar graphs of maximum degree 3 [13]. A polynomial algorithm for trees can be found in [2].

Notice, that the efficient domination problem is the same as the independent perfect domination problem. An $O(n^3)$ algorithm for AT-free graphs can be found in [6] (also for the the weighted version). (See also [18]).

*This work was partially supported by an Earmarked Research Grant from the Research Grants Council of Hong Kong.

In [8] an algorithm that runs in $O(n^2)$ for the subclass of cocomparability graphs (which are properly contained in AT-free graphs) can be found.

For complexity results on the minus domination problem, (i.e., $Y = \{-1, 0, +1\}$) we refer to [11]. For example, in this paper it is shown that the minus domination problem is NP-complete for chordal graphs and chordal bipartite graphs and a linear time algorithm is given for the minimum minus dominating function on trees.
In [17] it is shown that the question whether there exists an *efficient* minus domination problem is NP-complete, even when restricted to chordal graphs, chordal bipartite graphs, planar bipartite graphs, and planar graphs with maximum degree 4. The method they use to show NP-completeness is very simple and was used earlier in the paper [11] to show that the minus domination problem is NP-complete even when restricted to bipartite graphs and chordal graphs by a transformation from the dominating set problem. (The idea used in [17] is to attach a P_3 to each vertex of the graph and transform the problem to the efficient domination problem. The same method was used also in [11]) In their paper [17] the authors also show that, for trees this problem coincides with the "ordinary" efficient domination problem.

It is known that the efficient signed domination (i.e., $Y = \{-1, +1\}$) and the efficient minus domination problem can be solved in linear time for special cases of interval graphs (see [17]). In this paper linear time algorithms are given for these special classes of interval graphs (called "chain interval graphs" defined as those interval graphs for which every pair of minimal separators have an empty intersection).

As far as we know, the complexity of the efficient minus domination, the minus domination and the efficient signed domination problem are still open for AT-free graphs, even for interval graphs.

In Section 4 of this paper we show a general method that shows that the FP version of the efficient domination problem can also be solved for t-degenerate graphs for fixed values of t. This class of graphs is very general and contains for example bounded treewidth graphs, chordal graphs with bounded clique number, graphs with bounded bandwidth, cutwidth etc., general planar graphs, and many other classes.

We now turn to some remarks on the methods we use to solve the Y-domination problem for planar graphs.

Recently it was shown in [1], that if the domination number of a planar graph is at most k, then the treewidth can be bounded by $\delta\sqrt{k}$ for some reasonably small constant δ (for a detailed proof see [1]).
This result was used to improve a result by Downey and Fellows [10], who showed that the domination number for planar graphs is in FPT and can be solved in $O(11^k n)$ time.
The relation between planar domination and treewidth as expressed above was used to obtain a new uniformly polynomial time algorithm for the planar dominating set problem which runs in time $O(c^{\sqrt{k}} n)$ for some reasonable constant c [1] (for present best bounds on c we refer to [1]).
The same method (as will be described in [7]) gives a somewhat more direct method (at least to describe it) with a similar (small) worst case constant d, but runs in time $O(d^{\sqrt{k}} n^2)$. This method uses the result of [19] which gives an algorithm for the BRANCHWIDTH of a planar graph in $O(n^2)$ time. At the moment it is not clear which of these two algorithms could be of more practical use (if any of the two). Clearly, both algorithms are not hard to implement and both have as an output an approximate tree decomposition which is then used to solve the problem. The advantage of the latter one is that it gives a treewidth value which is sure to be close to the optimal one whereas the first algorithm gives only an approximate tree decomposition of which the approximation factor is not guaranteed.

In this paper we show that this result can be used also to obtain "fast" FP algorithms for the Y-dominating problem on planar graphs by simply applying the result stated above.

2 Preliminaries

For detailed information on fixed parameter complexity we refer to the recent work of Downey and Fellows [10]. Readers who are unfamiliar with treewidth can have a look in [15]. For more information on special classes of graphs and problems not defined here, we refer to [5].
In this section we define some of the problems that have been mostly studied as special cases of the Y-DOMINATION problem that were mentioned in the introduction.
But we start with the most important results for the problem in it's full generality.
For a graph $G = (V, E)$ and a vertex $x \in V$, we write $N(x)$ for the neighborhood of x and $N[x]$ for its *closed neighborhood*, i.e., $N[x] = N(x) \cup \{x\}$. By the way, in the rest of this paper, for convenience we turn to the modern abbreviated style of writing this as $N[x] = N(x) + x$. We use the same notation for sets, and also we use $A - x$ (if A is some set) for example instead of the more cumbersome notation $A \setminus \{x\}$.

If f is an assignment of integers to the vertices of a graph $G = (V, E)$ and $A \subseteq V$ then we use the notation $f(A) = \sum_{x \in A} f(x)$.

Definition 1 *Let Y be a finite set of integers. The Y-DOMINATION problem asks for an assignment $f : V \to Y$ set Y of integers such that for each vertex x, $f(N[x]) \geq 1$.*
The efficient Y-DOMINATION problem asks for such an assignment with $f(N[x]) = 1$ for all vertices $x \in V$.

These problems were first described investigated in the paper by [2]. We mention here the most important results of this paper.
Two Y-dominating functions are *equivalent* if they have the same closed neighborhood sum at every vertex. D. W. Bange in his paper proved that G has an efficient Y-dominating function if and only if all equivalent Y-dominating functions have the same weight. Furthermore, in this paper it is shown that if the closed neighborhood matrix of G is invertible then G has an efficient Y-dominating function for some set Y. This is quite obvious, but unfortunately, the solution is in most cases not integer. Special cases that would be interesting to look at also from a point of view in coding theory, are certain strongly regular graphs (see, e.g., [9]). An analysis of the spectrum of the closed neighborhood matrices should give some insight which of these graphs are worth some detailed analysis. (For the moment we only mention that strongly regular graphs for which the ordinary adjacency matrix has smallest eigenvalue at least -2 were characterized, and they all seem to have efficient dominating sets.) We don't go into cases where the set Y also has values which are real or fractional numbers. For some results of this type (i.e., real numbers) the reader is referred to [14].
The problem of finding Y-assignments is of theoretical interest at least and for various sets Y of great importance for many practical and important applications, as was pointed out for example in [2, 17, 20] and in many other papers.

The most researched cases of Y-assignments are listed in the following definition.

- An *efficient* assignment is a $\{0, 1\}$-assignment f of the vertices such that for every vertex $x \in V$, $f(|N[x]) = 1$.
- The *minus domination* problem asks for an $\{0, -1, +1\}$-assignment of the vertices of the graphs such that $f(N[x]) \geq 1$ for every vertex x.
- The *efficient minus domination* problem asks for an $\{0, +1, -1\}$-assignment f to the vertices such that $f(N[x]) = 1$ for every vertex x.
- The *efficient signed domination* problem asks for an $\{+1, -1\}$-assignment to all vertices such that $f(N[x]) = 1$ for every vertex x.

Notice that not every graph has an efficient assignment. A square may serve as a trivial example of this.
The parameterized version of these type of problems we consider in this paper asks for an assignment with *value* at most k, where k is the parameter of the FP problem.

Definition 2 *If Y is a set of integers then we define the* value *of the Y-assignment f as $\| \{x \mid f(x) > 0\} \|$. The* weight *of the Y-assignment is defined as $\sum_{x \in V} f(x)$.*

Clearly, for many cases there is a close relation between the weight and value of assignments. As a trivial example we mention that these values coincide for efficient and efficient signed domination.

Consider an Y-assignment f of a graph $G = (V, E)$ and let $D = \{x \mid f(x) > 0\}$. Clearly, by definition, D is a dominating set, since every vertex must have at least one vertex with a positive number assigned to it in it's closed neighborhood.
Unfortunately DOMINATING SET is $W[2]$-hard, (see, e.g., [10]) and hence it is not expected that this problem has a fixed parameter solution. However, when restricted to planar graphs, the problem is in FPT. A proof of this appears for example in [10], and the result was recently improved by [1], using graph separator techniques (for more information and results on graph separators, see the forthcoming book [7]).
The most important tool we use is the following theorem, which appeared in [1]. The technique used to prove it will appear in a more general setting in [7].

Theorem 3 *If a planar graph has a dominating set of size at most k, then the treewidth is bounded by $\delta\sqrt{k}$, with some small constant δ. A constructive algorithm finding an approximate tree decomposition with width $\delta\sqrt{k}$ can be given in linear time.*

At the moment the constant δ is already quite small (smaller than 50), but work on it is under construction. The proof of this theorem is based on methods that will appear in [7], and a preliminary full proof can be found in [1].

3 (Efficient) Y-domination on planar graphs

In this section we show a uniformly polynomial time algorithm for the efficient Y-assignment on planar graphs.

Let k be the parameter, i.e., we try to find an efficient Y-assignment f with value at most k in a planar graph, (i.e., the number of vertices which are assigned a positive value is at most k).
We mentioned already that if f is an Y-efficient assignment with value at most k, then clearly it also must have a dominating set with size at most k. Hence we can apply Theorem 3 to find a tree decomposition with width at most $\ell = \delta\sqrt{k}$ for some reasonable constant δ.
The following, somewhat surprising result shows that we don't have to worry about the actual *weight* of an efficient or efficient minus or efficient signed dominating set. The theorem below is due to [2] (and in fact a more general result was proved in this paper; we state it in this way, since it suffices for our purposes here).

Theorem 4 *A graph $G = (V, E)$ has an efficient Y-dominating assignment if and only if all equivalent Y-dominating functions have the same weight.*

Hence if our algorithm finds an efficient Y-dominating assignment then it's weight is fixed.

Our algorithm first computes an approximate tree decomposition of width at most $c\sqrt{k}$ (see, e.g., [1, 7]). For practical aplications we recommend to optimize the treewidth from this approximation as much as possible with ad-hoc methods, since this could improve the time bound of our algorithm considerably. At the moment we don't have a general method to improve the treewidth given by Theorem 3 but we feel that the exact treewidth could be considerably smaller than the approximate one given above. (Maybe for particular cases of the Y-domination problem even some definite improvements on the treewidth are possible.)

Let $(T, \mathcal{S})$ be a tree decomposition with bounded width ℓ (where $\ell = c\sqrt{k}$) as given output of the approximation algorithm mentioned in Theorem 3. (For the current best value of the constant c, we refer to [1].) Here $\mathcal{S}$ is a set of subsets S_i, such that there is a one-to-one correspondents with the vertices in the tree and the subsets of $\mathcal{S}$.
It is easy to see, (see, e.g., [15]) that we may assume that a rooted tree decomposition can be made binary and that the so-called "bags" S_i can be made of four different types.

- An introduce node i; The node i in the tree has only one child and the corresponding bag S_i has a "new" vertex x which does not appear in the bag corresponding with the child j of i. The rest of the bag $S_i - x$ is exactly the same as the bag corresponding with the bag S_j, i.e., $S_i = S_j + x$.
- A forget node i; The node i in the tree has only one child j and its bag S_i is exactly the same as the bag of its child except for one node that is missing; i.e., $S_i = S_j - x$.
- A join node i has two children p and q. The three corresponding bags are exactly the same.
- A leaf node i; which is simply a leaf of the rooted tree T. (If the root of T is also a pendant vertex, it is *not* considered as a leaf.)

We use this tree decomposition to find an efficient dominating set.

Definition 5 *A Y-assignment f_i to the vertices of a bag S_i is called* feasible *for a node i in the tree, if there exists an extension f of f_i to all vertices from all bags corresponding to the nodes in the subtree of i such that for every vertex x of a bag of a node in the subtree, which is not in S_i, $|f(N[x])| = 1$.*

We use dynamic programming and work our way up in the tree and keep track of all possible feasible assignments with their respective values. Let us start with the leaves. For a leaf node i we simply compute a table of all possible assignments of the vertices in S_i with elements from Y and keep track of it's value.

Now consider a join node i and let p and q be its children. An assignemt at i is feasible if and only if it is feasible for both it's children p and q. The value of a feasible assignment at i is simply the weight of the assignments of the assignments at p and q, not double counting the values at i of course.

Consider an introduce node i. Let x be the vertex being introduced. Consider all possible assignments of S_i which are feasible when restricted to S_i and which are feasible when restricted to the child j of i. If the assignment remains feasible within S_i then this is added to the table of feasible assignments of i. If x is assigned a positive value, then the value of this feasible assignment is updated.

Finally consider a forget node i and let x be the vertex which disappears from the child. Because we want the final set to be efficient, we now must insist that $f(N[x]) = 1$ for a feasible assignment f, since this value cannot be changed when we go further up in the tree. If this does not hold, the feasible assignment is not put in the table of feasible assignments at node i (otherwise it is again feasible for node i).

Our algorithm comes to an end when we reach the root r of the subtree. At the root, we only have to compute the feasible assignments, as described above, and make a final check if the value is at most k.

Theorem 6 *There exists an algorithm which runs in time $O(|Y|^{\ell} n)$ to check if a planar graph G has an efficient Y-assignment of value at most k. Here $\ell = c\sqrt{k}$ for some small constant k.*

Remark 7 *In this section we have given the exact description of the algorithm for the* EFFICIENT Y-DOMINATION *problem, but it should be immediately clear that only slight modifications are necessary to describe the algorithm for "ordinary"* y-DOMINATION.

4 Efficient domination of t-degenerate graphs

In this section we present a uniformly polynomial-time algorithm for solving the fixed parameter version of the efficient domination problem on t-degenerate graphs (t fixed).

Definition 8 *A graph G is t-degenerate if there is a linear ordering L of its vertices $v_1, v_2, \cdots, v_n$ such that each v_i, $1 \leq i \leq n$, is adjacent to at most t vertices that succeed v_i in L.*

Equivalently, G is t-degenerate if every induced subgraph has a vertex of degree at most t. The class of t-degenerate graphs contains many well-known classes of graphs; for example, graphs embeddable on some fixed surface, graphs of bounded treewidth, and graphs that are H-minor free. For some recent results on t-degenerate graphs, see [4].

The EFFICIENT DOMINATION k-SET problem asks whether an input graph G contains an efficient domination set with k vertices. Recall that for any vertex v of G, an efficient dominating set S contains exactly one vertex in the closed neighborhood $N_G[v]$. Furthermore, if a vertex v is in S then v is the only vertex of S inside the closed neighborhood $N_G[N_G[v]]$ of $N_G[v]$.

We now give our algorithm. First, we compute an ordering $L = v_1, v_2, \ldots, v_n$ of the vertices of G that satisfies the definition of a t-degenerate graph, which can be done easily in linear time. Then we build a search tree to search for an efficient dominating set S of size k according to this ordering.

The root of the search tree is labeled by $(*, G)$. Each of the remaining nodes in the search tree is labeled by a pair (v, G), where v is the vertex selected to be in S and G is the graph in which we search for the remaining vertices of S.

As we mentioned, for a vertex v_i, exactly one of the vertices in $N_G[v_i]$ belongs to S, and once a vertex v_i is in S no other vertices of $N_G[N_G[v]]$ are in S. Therefore we label the nodes of the search tree as follows: The children of the root are labeled by (v_i, G_i), where $v_i \in N_G[v_1]$ and $G_i = G - N_G[N_G[v_1]]$. For a node labeled by (v_i, G_i), we pick the first vertex u of G_i in the linear ordering L, and label its children by (v_j, G_j), where $v_j \in N_{G_i}[u]$ and $G_j = G - N_{G_i}[N_{G_i}[u]]$.

For each root-to-leaf path in the search tree, the set of vertices in the labels is a candidate for an efficient dominating set of size k. Therefore we need only build the search tree up to depth k. Once we build the search tree, we check for each such candidate if it is indeed an efficient dominating set of G.

Since each node of the search tree has at most $t+1$ children, each node takes $O(tn)$ time, and checking whether a candidate is an efficient dominating set takes $O(tn)$ time, the total running time of the algorithm is $O((t+1)^{k+1}n)$.

Theorem 9 *The* EFFICIENT DOMINATION k-SET *problem can be solved in $O((t+1)^{k+1}n)$ time for t-degenerate graphs.*

5 Concluding remarks

Notice that a vast amount of problems remain open. We only mention some of the most obvious ones.
As far as we know no complexity results are known for the EFFICIENT MINUS and EFFICIENT SIGNED domination problem for (classes of) AT-free graphs. It seems not likely that the method of [6, 18] can be used also to solve these problems in polynomial time, unless we require additionally that the separator sets have bounded size.

As mentioned, all equivalent Y-dominating functions have the same weight if and only if the graph has an efficient Y-dominating function (see [2]). It would be interesting to know if a similar statement can be made for the value of efficient Y-dominating functions.

A final interesting open problem is the complexity of the signed domination problem on planar graphs.

References

[1] Alber, J., H. Bodlaender, R. Niedermeier, and H. Fernau, Faster algorithms for domination of planar graphs, *Proc. 17th SWAT'00*, Springer-Verlag, LNCS 1851, pp. 97–110.

[2] Bange, D. W., A. Barkauskas, L. H. Post, and P. J. Slater, Generalized domination and efficient

domination in graphs, *Discrete Mathematics* **159**, (1996), pp. 1–11.

[3] Bodlaender, H., A tourist guide through treewidth, *Acta Cybernetica* **11**, (1993), pp. 1–23.

[4] Borodin, O. V., A. V. Kostochka, B. Toft, Variable degeneracy: extensions of Brooks' and Gallai's theorems, *Discrete Mathematics* **214**, (2000), pp. 101–112.

[5] Brandstaedt, A., V. B. Le, and J. P. Spinrad, *Graph classes–A Survey*, SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, 1999.

[6] Broersma, H. J., T. Kloks, D. Kratsch, and H. Müller, Independent sets in AT-free graphs, To appear in: *SIAM J. of Discr. Math.*.

[7] Broersma, Hajo and Ton Kloks, *Separators in graphs*. To appear.

[8] Chang, M.-S., Weighted domination of cocomparability graphs, *Discrete Applied Mathematics* **80**, (1997), pp. 135–148.

[9] Corneil, Derek G. and Rudi mathon (eds.), *Geometry and combinatorics: selected works of J. J. Seidel*, London Ac. Press, 1991.

[10] Downey, R. G., and M. R. Fellows, *Parameterized complexity*, Monographs in Computer Science, Springer-Verlag 1999.

[11] Dunbar Jean, Wayne Goddard, Stephen Hedetniemi, Alice McRae, and Michael A. Henning, The algorithmic complexity of minus domination in graphs, *Discrete applied mathematics* **68**, (1996), pp. 73–84.

[12] Dunbar, Jean, Stephen Hedetniemi, Michael A. Henning, and Allice McRae, Minus domination in graphs, *Discrete mathematics* **99**, pp. 35–47.

[13] Fellows, M. R. and M. N. Hoover, perfect dominations, *Australian Journal of Combinatorics* **3**, (1991), pp. 141–150.

[14] Goddard, W. and Michael A. Henning, Real and integer domination in graphs, *Discrete mathematics* **199**, (1999), pp. 61–75.

[15] Kloks, T., *Treewidth—Computations and Approximations*, Springer–Verlag, LNCS 842, 1994.

[16] Lu, C. L., *On the study of weighted efficient domination*, PhD-thesis, Department of Computer Science, National Tsing Hua University, Taiwan, 1998.

[17] Lu, C. L., S. -L. Peng and C. Y. Tang, Efficient minus and signed domination in graphs, To appear at ISAAC'00.

[18] Müller, Haiko, *Asteroidal sets in graphs*, Habilitation, Friedrich–Sciller–Universität Jena, 1999.

[19] Seymour, P. D. and R. Thomas, Call routing and the ratcatcher, *combinatorica* **14**, (1994), pp. 217–241.

[20] Yen, C. C. and R. C. T. Lee, The weighted perfect domination problem and its variants, *Discrete Applied Mathematics* **66**, (1996), pp. 147–160.