On efficient total domination

Oliver Schaudt

Institut für Informatik, Arbeitsgruppe Faigle/Schrader, Universität zu Köln
Weyertal 80, 50931 Cologne, Germany

Abstract

An *efficiently total dominating set* of a graph G is a subset of its vertices such that each vertex of G is adjacent to exactly one vertex of the subset. If there is such a subset, then G is an *efficiently total dominatable graph* (G is *etd*).

We show that the corresponding etd decision problem is $\mathcal{NP}$-complete on $(1,2)$-colorable chordal graphs and on planar bipartite graphs of maximum degree 3 and obtain polynomial solvability on T_3-free chordal graphs, implying polynomial solvability on interval graphs and circular arc graphs.

Keywords: graph algorithms, computational complexity, total domination, efficient total domination

1. Introduction

Total domination has been introduced 1980 by Cockayne, Dawes and Hedetniemi in [2] and is intensively studied now. A good introduction to the theory of (total) domination, giving a broad overview of the important results and applications, is given in [5]. In the problem of total domination, one is interested in determining the value $\gamma_t(G)$ of a given graph G, defined as the smallest size of a subset $X \subseteq V(G)$ such that each vertex of G has at least one neighbor in X.

Let G be a simple undirected graph. A set $X \subseteq V(G)$ is said to be an *efficiently total dominating set* of G, or an *etd set*, if each $v \in V(G)$ is adjacent to exactly one vertex in X. G is then said to be an *efficiently total dominatable graph*, or G is *etd*. The corresponding decision problem is denoted by *ETD*. Let $\mathbf{1}$ denote the vector with all components equal to 1 of suitable dimension. ETD can alternatively be defined as the class of graphs whose neighborhood hypergraph has a perfect matching, as the class of graphs whose adjacency matrix A accepts the equation $Ax = \mathbf{1}$ for some 01-vector x, and as the class of graphs that have an induced matching such that each vertex is adjacent to exactly one matched vertex. There is some literature on efficient domination.

Email address: `schaudt@zpr.uni-koeln.de` (Oliver Schaudt)

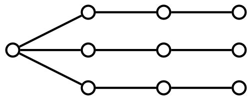

Figure 1: T_3.

In the case of efficient *total* domination however, only a few papers have been published so far (according to our knowledge).

An important result is the following

Theorem 1 (See [5]). *Let G be an etd graph. Each etd set X of G has cardinality $\gamma_t(G)$.*

We can therefore understand efficient total domination as an extremal case of total domination. Furthermore, understanding the structure of efficiently total dominatable graphs and the algorithmic complexity of the corresponding decision problem may put some light on total domination, too.

1.1. Preliminaries

Let G be a graph. $V(G)$ denotes its vertices and $E(G)$ denotes its edges. For each $U \subseteq V(G)$, $G(U)$ denotes the induced subgraph on the vertices of U. A graph is G-free if it does not contain G as induced subgraph. A graph is (p, q)-*colorable* if its set of vertices can be partitioned into $p + q$ parts, p parts being a clique and q parts being a stable set each. $(1, 1)$-colorable graphs are said to be *split* graphs. A graph is *chordal* if all of its induced cycles have length 3. A graph is *planar* if it can be drawn on the plane such that no two edges cross each other. A graph is an *interval graph* if it has an intersection model consisting of closed intervals on a line. A graph is a *circular arc graph* if it has an intersection model consisting of arcs on a circle. A *triangle* is the complete graph on 3 vertices, K_3. A T_3 is constructed in the following way. Start with three paths of length two, choose an endvertex of each path and connect these to a single new vertex r (see Fig. 1). For every vertex v, $N(v)$ denotes the set of the vertices adjacent to v, i.e. its open neighborhood; sometimes it is useful to explicitly write $N_G(v)$ for the open neighborhood of v in a graph denoted by G. A *leaf* is a vertex with degree 1.

All relevant graph classes and graph class inclusions are displayed in detail in [1].

2. $\mathcal{NP}$-completeness results

2.1. Stretching lemma

Given a graph G and a vertex $v \in V(G)$, a *stretching* of v is a graph obtained from G by substituting v by a path $(v_1, \ldots, v_5)$ of length four, and connecting

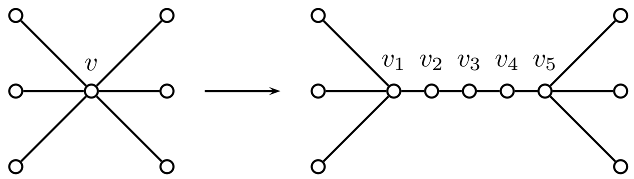

Figure 2: Stretching of v.

each former neighbor of v to exactly one of the two endvertices of the path in an arbitrary way (see Fig. 2). It is easy to see that a stretching of v is etd iff G is etd. Furthermore, stretching preserves bipartiteness.

Lemma 1. *Let $\mathcal{G}$ be a graph class closed under the stretching operation. If ETD is $\mathcal{NP}$-complete on $\mathcal{G}$ then ETD is $\mathcal{NP}$-complete on the class of graphs of $\mathcal{G}$ with maximum degree 3.*

Proof. Let $\mathcal{G}$ be a graph class closed under the stretching operation. The polynomial reduction can be done by iteratively choosing a vertex v with $|N(v)| \geq 4$ and stretching v to $(v_1, \ldots, v_5)$ in a way that connects exactly two former neighbors of v to v_1 and all other neighbors to v_5. □

2.2. ETD as matrix equation

Graph classes on which ETD is $\mathcal{NP}$-complete can be obtained by reducing the well known Exact Cover decision problem (EC) to ETD. Given an arbitrary 01-matrix A, EC asks for the 01-solvability of $Ax = \mathbf{1}$. Let I denote the identity matrix of suitable dimension. EC reduces to ETD in the following way: Given a 01-matrix A, we define a function

$$A(X) = \begin{pmatrix} X & 0 & 0 & A \\ 0 & 0 & I & I \\ 0 & I & 0 & 0 \\ A^t & I & 0 & 0 \end{pmatrix} \tag{1}$$

and observe for each X, that A is in EC iff $A(X)$ is in EC.

Let J denote the square matrix with all components equal to 1 of suitable dimension. $A(J-I)$ is the adjacency matrix of a $(1,2)$-colorable chordal graph, i.e. a chordal graph which can be partitioned into a clique and two independent sets, and $A(J-I)$ is in EC iff this very graph is in ETD. As EC is well known to be $\mathcal{NP}$-complete, we conclude $\mathcal{NP}$-completeness of ETD restricted to $(1,2)$-colorable chordal graphs. As the class of $(1,2)$-colorable chordal graphs is only slightly bigger than the class of split graphs (which are exactly the $(1,1)$-colorable graphs) and ETD restricted to split graphs is obviously trivial, we see that the gap of complexity between the two classes is big compared to their structural differences.

We conclude

Theorem 2. *ETD is $\mathcal{NP}$-complete on the class of $(1,2)$-colorable chordal graphs.*

As mentioned in [4], EC remains $\mathcal{NP}$-complete when restricted to the class of all 01-matrices A for which

$$A' = \begin{pmatrix} 0 & A \\ A^t & 0 \end{pmatrix} \tag{2}$$

is the adjacency matrix of a planar graph. Obviously, the graph with adjacency matrix $A(0)$ is a bipartite planar graph being etd iff A is in EC. By applying Lemma 1, we obtain the following

Theorem 3. *ETD is $\mathcal{NP}$-complete on planar bipartite graphs of maximum degree 3.*

3. Polynomial solvability on T_3-free chordal graphs

Each induced subgraph of a T_3-free chordal graph is T_3-free chordal again and each chordal graph has a *simplicial vertex*, i.e. a vertex whose neighbors form a clique (see [1]). We show that ETD is polynomially solvable on T_3-free chordal graphs, presenting our algorithm in two parts.

Algorithm 1 Labeling algorithm

Require: T_3-free chordal graph $G = (V, E)$.
Ensure: $A, I \subseteq V$ satisfying Observation 1, 2 and 3.

$A, I \leftarrow \emptyset$
$D \leftarrow \{e \in E : e \text{ lies on a triangle}\}$
labeling possible $\leftarrow$ $true$
while labeling possible **do**
  labeling possible $\leftarrow$ $false$
  if there is $v \in V \setminus I$ such that $\{\{v, u\} : u \in N(v) \setminus I\} \subseteq D$ **then**
    $I \leftarrow I \cup \{v\}$
    labeling possible $\leftarrow$ $true$
  else if there is $v \in V$ and $u \in N(v) \setminus A$ such that $N(v) \setminus \{u\} \subseteq I$ **then**
    $A \leftarrow A \cup \{u\}$
    $I \leftarrow I \cup (N(N(u)) \setminus \{u\})$
    labeling possible $\leftarrow$ $true$
  end if
  if $A \cap I \neq \emptyset$ **then**
    return A, I
  end if
end while
return A, I

Let $G = (V, E)$ be a T_3-free chordal graph and let A, I be the output of Algorithm 1. If a vertex $v \in V$ is in A (in I) it is said to be *active* (*inactive*). The vertices in $V \setminus (A \cup I)$ are said to be *unlabeled*. A vertex v is said to be *balanced* if $|N(v) \cap A| = 1$, *unbalanced* otherwise.

Observation 1. 1. *For each etd set X of G, $A \subseteq X$ and $X \cap I = \emptyset$.*
2. *If $A \cap I \neq \emptyset$, then G is not etd.*

Proof. The second claim follows easily from the first.

To prove the first, let X be an etd set of G. The proof is done by induction on the iterations of the procedure. Let I and A denote the constructed sets just before the next step. Let $v \in V \setminus I$ such that $\{\{v, u\} : u \in N(v) \setminus I\} \subseteq D$. By induction, $X \cap I = \emptyset$. Since v has a neighbor $x \in X$ and therefore $\{v, x\} \notin D$, $v \notin X$ due to efficiency of X. Let $v \in V$ and $u \in N(v) \setminus A$ such that $N(v) \setminus \{u\} \subseteq I$. Since $X \cap I = \emptyset$, $u \in X$ and therefore $N(N(u)) \setminus \{u\} \cap X = \emptyset$. □

Due to Observation 1, we may assume that the procedure ended with $A \cap I = \emptyset$ for the remainder of this section.

Observation 2. 1. *A vertex v is balanced iff $N(v) \cap A \neq \emptyset$ iff $N(v) \subseteq A \cup I$.*
2. *Each unlabeled vertex is balanced.*

Proof. The first claim follows easily from the definition of Algorithm 1 and $A \cap I = \emptyset$.

To prove the second, let Z be a connected component of the subgraph of $G(V \setminus (A \cup I))$. Let v be a simplicial vertex of Z. In the case of $|N_Z(v)| \geq 2$, each edge of Z incident to v lies on a triangle. Thus, v is inactive, in contradiction to the premise. If v has a single neighbor u in Z, then u must be active, in contradiction to the premise. Thus, v must be isolated in Z and therefore is balanced, by the first claim. □

Observation 3. *Each unbalanced vertex has at most two unlabeled neighbors which are no leaves of G.*

Proof. Let x be an unbalanced vertex. By Observation 2, $x \in A \cup I$ and $N(x) \cap A = \emptyset$.

Suppose x is inactive. Assume x has at least three unlabeled neighbors u, v and w. By Observation 2, u, v and w are balanced and pairwise not adjacent. Thus, u, v and w are adjacent to exactly one active vertex each (denoted by u', v' and w'). By chordality, these vertices are pairwise neither identical nor adjacent and all three cannot be adjacent to x. As u', v' and w' are all unbalanced, by Observation 2 they must each have another unlabeled vertex u'', v'' and w'' as neighbor, all different to x. By chordality again, u'', v'' and w'' are pairwise neither identical nor adjacent and even not adjacent to x. Furthermore, neither vertex u, v or w is adjacent to any of u'', v'' and w'', because of Observation 2. All in all, $G(\{x, u, v, w, u', v', w', u'', v'', w''\})$ is an induced T_3, in contradiction to the premise.

The assumption of active x having at least three unlabeled neighbors u, v and w not being leaves in G is dealt with in similar fashion. □

By Observation 3, the remaining problem can be interpreted as an instance f of 2-SAT, as computed by Algorithm 2.

Algorithm 2 Reduction to 2-SAT

Require: T_3-free chordal graph $G = (V, E)$ with A, I constructed by Alg. 1.
Ensure: 2-SAT formula f satisfying Observation 4.

$U \leftarrow \{v \in V : v$ is unlabeled and no leaf of G $\}$
$W \leftarrow \{v \in V : |N(v) \cap U| = 2\}$
for all $v \in W$ adjacent to an unlabeled leaf **do**
  $f_v = \bigvee_{u \in N(v) \cap U} \overline{x_u}$
end for
for all $v \in W$ not adjacent to an unlabeled leaf **do**
  $f_v = \left(\bigvee_{u \in N(v) \cap U} x_u\right) \wedge \left(\bigvee_{u \in N(v) \cap U} \overline{x_u}\right)$
end for
return $f = \bigwedge_{v \in W} f_v$

Observation 4. *The output formula f of Algorithm 2 is satisfiable iff G is etd.*

Proof. Let f be satisfiable. Then there is a Boolean function x which satisfies f. We set

$$X = A \cup \{v \in U : x_v = 1\}. \tag{3}$$

By the definition of f, all vertices have at most one neighbor in X. Furthermore, each vertex not adjacent to an unlabeled leaf has exactly one neighbor in X. For each vertex v with $N(v) \cap X = \emptyset$, we choose an arbitrary unlabeled leaf from the neighborhood of v and add it to X. Then, $|N(v) \cap X = 1|$ for all $v \in V$ and thus X is an etd set of G.

Let G be etd and X be an etd set of G, i.e. all vertices v satisfy $|N(v) \cap X| = 1$. For each $u \in U$ we set

$$x_u = \begin{cases} 1 & \text{if } u \in X \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

and observe that f is satisfied by x. □

We now come to the time complexity analysis of the presented algorithm.

Theorem 4. *ETD on the class of T_3-free chordal graphs is solvable in $\mathcal{O}(n^3)$ time, where n is the number of vertices of the given graph.*

Proof. Let $G = (V, E)$ be a T_3-free chordal graph with n vertices and m edges.

Algorithm 1: The set $D = \{e \in E : e$ lies on a triangle$\}$ can obviously be computed in $\mathcal{O}(mn)$. The conditions "there is $v \in V \setminus I$ such that $\{\{v, u\} : u \in N(v) \setminus I\} \subseteq D$" and "there is $v \in V$ and $u \in N(v) \setminus A$ such that $N(v) \setminus \{u\} \subseteq I$" can both be checked in $\mathcal{O}(n^2)$. Since in each iteration (except the last one) of the while sequence a vertex is added to A or to I, there are at most n iterations. As $m < n^2$, Algorithm 1 needs $\mathcal{O}(n^3)$ time.

Algorithm 2: Constructing the Boolean formula f and solving it takes $\mathcal{O}(n+m)$ steps, as the number of literals of f is linearly bounded by $n+m$ and solving a 2-SAT formula can be done in linear time, for example as explained in [3].

All in all, we obtain time complexity of $\mathcal{O}(n^3)$. □

Given an etd T_3-free chordal graph G, $\gamma_t(G)$ clearly equals

$$|A \cup \{v \in U : x_v = 1\}| + |\{v \in V : N(v) \cap (A \cup \{v \in U : x_v = 1\}) = \emptyset\}| \quad (5)$$

in the notation of Algorithm 2.

As interval graphs are well-known to be chordal and obviously T_3-free, we obtain the following

Corollary 1. *ETD on interval graphs is solvable in $\mathcal{O}(n^3)$ time, where n is the number of vertices of the given graph.*

Another implication is the following.

Corollary 2. *ETD on circular arc graphs is solvable in $\mathcal{O}(n^3)$ time, where n is the number of vertices of the given graph.*

Proof. Let $G = (V, E)$ be a circular arc graph with intersection model (C, S), i.e. $S = \{S_v\}_{v \in V}$ a set of arcs on a circle C and $E = \{\{u, v\}, S_u \cap S_v \neq \emptyset\}$.

Assume G is not an interval graph, i.e. $\bigcup S = C$. If each edge of G lies on a triangle or is incident to a leaf, it can be trivially decided if G is etd.

Assume there is an edge $e = \{u, v\}$ not belonging to a triangle, u and v not being leaves each. Thus the stretching G' of v (defined in section 2.1), connecting each former neighbor of v (but u) to one endvertex of the substituting path and u to the other endvertex, is a circular arc graph again. Let $v_1, \ldots, v_5$ be the new vertices. We define four graphs $G'_1, \ldots, G'_4$ obtained from G' by deleting either $\{v_1, v_2\}$, $\{v_2, v_3\}$, $\{v_3, v_4\}$, or $\{v_4, v_5\}$. Clearly, all four graphs are interval graphs.

It is easy to see that G is etd iff for some $1 \leq i \leq 4$, G'_i has an etd set disjoint to $\{v_i, v_{i+1}\}$. This can be decided by applying a simple modification of Algorithm 1 (starting with $I = \{v_i, v_{i+1}\}$ instead of $I = \emptyset$) and Algorithm 2 afterward. □

References

[1] A. Brandstädt, V.B. Le, J. Spinrad, *Graph classes: a survey*, SIAM Monographs on Discrete Math. Appl., Vol. 3, SIAM, Philadelphia, 1999.

[2] E.J. Cockayne, R.M. Dawes and S.T. Hedetniemi, *Total domination in graphs*, Networks 10 (1980), pp. 211–219.

[3] A. del Val, *On 2-SAT and Renamable Horn*, Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, pp. 279–284, July 30-August 03, 2000.

[4] M.R. Garey, D.S. Johnson, *Computers and Intractability–A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.

[5] T.W. Haynes, S.T. Hedetniemi, P.J. Slater, *Fundamentals of Domination in Graphs*, Marcel Dekker, Inc., New York, 1998.