On weighted 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*).

In this paper, we prove NP-completeness of the etd decision problem on the class of planar bipartite graphs of maximum degree 3. Furthermore, we give an efficient decision algorithm that works on the T_3-free chordal graphs. In the main part, we present three graph classes on which the weighted etd problem is polynomially solvable: claw-free graphs, odd-sun-free chordal graphs (including strongly chordal graphs) and graphs which only induce cycles of length divisible by four (including chordal bipartite graphs). In addition, claw-free etd graphs are shown to be perfect.

Keywords: weighted efficient total domination, weighted efficient total edge domination, efficient total domination, total domination

1. Introduction

Total domination has been introduced 1980 by Cockayne, Dawes and Hedetniemi in [1] and is intensively studied now. A good introduction to the theory of domination in graphs, giving a broad overview of the important results and applications, is given in [2]. 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 total dominating set, i.e. a set $X \subseteq V(G)$ 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*.

An important result is the following Theorem of Gavlas, Schultz and Slater:

Theorem 1 (See [2]). *Let G be an etd graph. Each etd set of G has cardinality $\gamma_t(G)$.*

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

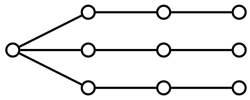

Figure 1: T_3.

ETD was shown to be $\mathcal{NP}$-complete in general in the paper [3]. Furthermore, ETD can be seen as a special type of the so-called generalized domination problem, studied in [4]. In particular, [4] shows that ETD is $\mathcal{NP}$-complete even if the instances are restricted to be chordal graphs. Furthermore, they give an $\mathcal{O}(n^3)$-time algorithm for ETD on interval graphs.

In this paper, we give an $\mathcal{NP}$-completeness result for ETD when restricted to planar bipartite graphs of maximum degree 3. Furthermore, we consider the time complexity of the weighted case of efficient total domination when restricted to certain graph classes, namely claw-free graphs, odd-sun-free chordal graphs and graphs which only induce cycles of length divisible by four, and prove polynomial solvability for each class. In addition, we give an efficient decision algorithm for ETD on T_3-free chordal graphs, which generalizes the result of [4] about interval graphs.

A *weighted graph* (G, c) is an ordered pair of a graph G and a function $c : V(G) \to \mathbb{R}$. An instance of the weighted efficient total domination problem *WETD* is a weighted graph (G, c) and we have to determine an etd set X of G minimizing $\sum_{x \in X} c(x)$ or decide that G is not etd. Clearly, an algorithm designed to compute a minimum weight etd set can also be used to compute γ_t for etd graphs. The weighted efficient domination problem has been examined in [5] and [6] among others. According to our knowledge, there is no literature on weighted efficient total domination yet.

Section 5.1 includes a characterization of all graphs G which are *efficiently total edge dominatable* (or *eted*) i.e. there is some edge set $D \subseteq E(G)$ such that each edge of G is incident to exactly one edge of D. This characterization implies polynomial solvability of the eted decision problem. In fact, one can even deal with the weighted case. In contrast, efficient edge domination was shown to be $\mathcal{NP}$-complete in the general case in [7].

1.1. Technical notations

Let G be a graph. $V(G)$ denotes its vertices, $E(G)$ its edges and $\overline{G}$ its complement. 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. The same goes for graph classes $\mathcal{G}$, i.e. a graph is $\mathcal{G}$-free if it is G-free for all $G \in \mathcal{G}$. A graph is *chordal* if all of its induced cycles have length 3. A graph is *chordal bipartite* iff all of its induced cycles have length 4. Let $n \geq 3$. An *n-sun* (or *sun*) is a chordal graph on $2n$ vertices whose vertex set can be partitioned into $W = \{w_1, \ldots, w_n\}$ and $U = \{u_1, \ldots, u_n\}$ such that W is stable and u_i

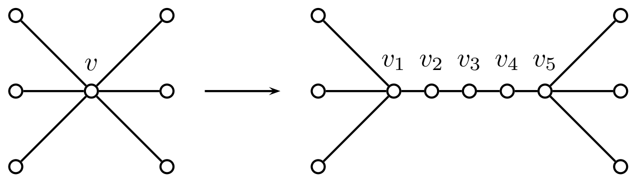

Figure 2: Stretching of v.

is adjacent to w_j iff $i = j$ or $i = j + 1 (\text{mod } n)$, for all $1 \le i, j \le n$. Note that the subgraph induced by U is not necessarily complete. An *odd sun* is an n-sun with odd n. A graph is *strongly chordal* if it is chordal and sun-free. A graph is *planar* if it can be drawn on the plane such that no two edges cross each other. A *leaf* is a vertex with degree 1. An *odd hole* is an induced cycle of odd length at least 5, an *odd antihole* is the complement of an odd hole. A *claw* is the complete bipartite graph $K_{1,3}$; 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 (see Fig. 1). For every vertex v, $N(v)$ denotes the set of the vertices adjacent to v, its *neighborhood*; sometimes it is useful to explicitly write $N_G(v)$ for the neighborhood of v in a graph denoted by G. If U is a set of vertices, $N(U) = \bigcup_{u \in U} N(u)$. Two adjacent vertices u, v are *adjacent twins* if $N(u) \setminus \{v\} = N(v) \setminus \{u\}$.

2. $\mathcal{NP}$-completeness of ETD

To show that ETD is $\mathcal{NP}$-complete on the class of planar bipartite graphs of maximum degree 3, we need the following construction: 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 each former neighbor of v to exactly one of the two endvertices of the path in an arbitrary way (see Fig. 2). We observe, 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 graphs of $\mathcal{G}$ with maximum degree 3.*

Proof. Let $\mathcal{G}$ be a graph class closed under the stretching operation. A polynomial reduction of ETD on $\mathcal{G}$ to ETD on the graphs of $\mathcal{G}$ with maximum degree 3 can be done in the following way. Iteratively choose a vertex v with $|N(v)| \ge 4$ and stretch 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. $\square$

Let $\mathbf{1}$ denote the vector with all components equal to 1 of suitable dimension. Graph classes on which ETD is $\mathcal{NP}$-complete can be obtained by reducing the Exact Cover decision problem (EC) to ETD. Given a 01-matrix A, EC asks for the existence of a 01-vector x such that $Ax = \mathbf{1}$. 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 matrix B_A by

$$B_A = \begin{pmatrix} 0 & 0 & 0 & A \\ 0 & 0 & I & I \\ 0 & I & 0 & 0 \\ A^t & I & 0 & 0 \end{pmatrix}. \tag{1}$$

and observe, that A is in EC iff B_A is in EC.

As is shown in [8], EC remains $\mathcal{NP}$-complete when restricted to the class of all 01-matrices A for which

$$\begin{pmatrix} 0 & A \\ A^t & 0 \end{pmatrix} \tag{2}$$

is the adjacency matrix of a planar graph. We observe, that the graph with adjacency matrix B_A is a bipartite planar graph being etd iff A is in EC. By applying Lemma 1, we obtain the following

Theorem 2. *ETD is $\mathcal{NP}$-complete on planar bipartite graphs of maximum degree 3.*

3. Graphs with balanced adjacency matrix

A 01-matrix is said to be *balanced* if it has no square submatrix (a_{ij}) of odd dimension $k \geq 3$ with

$$a_{ij} = \begin{cases} 1 & \text{if } i = j \text{ or } j = i+1 \text{ or } i = k, j = 1 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

up to permutation of rows and columns. All results on balanced matrices mentioned can be found in [9].

Let $\mathbf{0}$ denote the vector with all components equal to 0 of suitable dimension. For each balanced matrix A the *partitioning polytope* $P(A) = \{x \geq \mathbf{0} : Ax = \mathbf{1}\}$, if not empty, has only integral extreme points, as described in [9]. It is stated in [10], that a graph has a balanced adjacency matrix iff all of its induced cycles have length divisible by four. Such a graph is called *balanced*, according to [9]. If A is the adjacency matrix of a graph G, all integral points of $P(A)$ correspond to etd sets of G and vice versa. As linear programs are well known to be solvable in polynomial time, WETD restricted to the class of balanced graphs is solvable in polynomial time, too.

Therefore we have

Theorem 3. *WETD on balanced graphs is solvable in polynomial time.*

Since chordal bipartite graphs are balanced, Theorem 3 implies the following

Corollary 1. *WETD on chordal bipartite graphs is solvable in polynomial time.*

Conforti et al. state in [9] the following powerful characterization:

Theorem 4 (See [9]). *Let A be a balanced matrix. $Ax = \mathbf{1}$ holds for some 01-vector x iff there is no $(-1, 0, 1)$-vector y such that $y^t A \geq \mathbf{0}$ and $y^t \mathbf{1} < 0$.*

If transferred to our context, we obtain

Corollary 2. *A balanced graph G is etd iff there are no sets $X, Y \subseteq V(G)$ such that $|X| < |Y|$ and no vertex of G has more neighbors in Y than in X.*

Furthermore, it can be checked in polynomial time if a given graph is balanced, as shown in [9].

4. ETD and WETD on chordal graphs

We present a labeling procedure which works on arbitrary chordal graphs as input. Its output are two subsets of the vertices of the input graph, which satisfy certain properties presented below. These sets are then used to derive an efficient algorithm for the ETD problem on T_3-free chordal graphs and the WETD problem on odd-sun-free chordal graphs. The idea of Algorithm 1 is, that any edge between two vertices of an etd set of a chordal graph is necessarily a separating edge. This allows us to determine candidate vertices for possible etd sets of the graph. In the following, let $G = (V, E)$ be a chordal graph and let A, I be the output of Algorithm 1.

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 is a direct consequence of 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 $\{v, x\} \notin D$, $v \notin X$ 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. 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 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 from our assumption, that Algorithm 1 terminated and $A \cap I = \emptyset$.

Algorithm 1 Labeling algorithm

Require: A chordal graph $G = (V, E)$.
Ensure: Vertex sets $A, I \subseteq V$.

$A, I \leftarrow \emptyset$
$D \leftarrow \{e \in E : e \text{ lies on a triangle}\}$
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

To prove the second, let Z be a connected component of the subgraph $G(V \setminus (A \cup I))$. It is a classical result, that any chordal graph has a *simplicial vertex*, i.e. a vertex whose neighbors are mutually adjacent (see [11]). 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. □

Lemma 2. *Algorithm 1 needs $\mathcal{O}(n^3)$ time, where n is the number of vertices of the given graph.*

Proof. Let $G = (V, E)$ be a chordal graph with n vertices and m edges. The set $D = \{e \in E : e \text{ lies on a triangle}\}$ can 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. □

4.1. ETD on T_3-free chordal graphs

We now restrict our attention to T_3-free chordal graphs. Assume G is a T_3-free chordal graph and A and I are the output sets of Algorithm 1. Due to Observation 1, we may again assume that the procedure ended with $A \cap I = \emptyset$.

Observation 3. *Each unbalanced vertex has at most two unlabeled neighbors which are not leaves of G.*

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

Assume x is inactive and 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. The computation of this formula is done 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 ← {v ∈ V : v is unlabeled and no leaf of G }
W ← {v ∈ V : |N(v) ∩ U| = 2}
for all v ∈ W adjacent to an unlabeled leaf do
    f_v = ⋁_{u∈N(v)∩U} ¬x_u
end for
for all v ∈ W not adjacent to an unlabeled leaf do
    f_v = (⋁_{u∈N(v)∩U} x_u) ∧ (⋁_{u∈N(v)∩U} ¬x_u)
end for
return f = ⋀_{v∈W} f_v
```

$U \leftarrow \{v \in V : v \text{ is unlabeled and no leaf of } G\}$
$W \leftarrow \{v \in V : |N(v) \cap U| = 2\}$
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

$$X = A \cup \{v \in U : x_v = 1\}. \tag{4}$$

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

$$x_u = \begin{cases} 1 & \text{if } u \in X \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

and observe that f is satisfied by x. ☐

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

Theorem 5. *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 needs $\mathcal{O}(n^3)$ time, as is shown in Lemma 2. 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 [12].

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

4.2. WETD on odd-sun-free chordal graphs

Let $G = (V, E)$ be an odd-sun-free chordal graph. Let A and I be the output of Algorithm 1 to the input G. Assume $A \cap I = \emptyset$ and let $G' = (V, \{e \in E : e \not\subseteq I\})$.

Observation 5.
1. *Each etd set of G' disjoint to I is an etd set of G and vice versa.*
2. *If G is odd-sun-free, then G' is a balanced graph.*

Proof. The first claim follows from Observation 1.

To prove the second, let G be odd-sun-free.

Let C be an induced cycle of G' and v be an arbitrary vertex of C with neighbors $u, w \in C$. Assume $v \in A$. Thus, the two neighbors of u and w in C, u' and w', different to v are inactive, by Observation 2. By definition of G', $u, w \notin I$ and thus $u, w \notin A$ by Observation 2. Therefore u and w are unlabeled and thus not adjacent, by Observation 2. As C is a cycle in G, G is not chordal, in contradiction to the premise. Thus, $C \cap A = \emptyset$.

By definition of G', I is a stable set in G', and by Observation 2, $V \setminus (A \cup I)$ is a stable set in G'. Thus, C alternates between the two sets and therefore is an n-sun in G. As G is assumed to be odd-sun-free, n is even and hence C is a cycle of length divisible by four in G'. Therefore, G' is balanced. ☐

Now consider the adjacency matrix A of G'. By Observation 5, A is balanced. If we delete all columns corresponding to vertices of I, we obtain a balanced matrix A' again, since balancedness is closed under taking submatrices. By Observation 5, all integral points of $P(A')$ correspond to etd sets of G. This implies the following

Theorem 6. *WETD on odd-sun-free chordal graphs is solvable in polynomial time.*

By Theorem 6 and the definition of strongly chordal graphs, we obtain

Corollary 3. *WETD on strongly chordal graphs is solvable in polynomial time.*

5. Claw-free graphs

5.1. Line graphs

Let L be a line graph. From [13] we know that we can deduce an initial graph G whose line graph is L in linear time. We observe that L is etd iff G is eted. Using this observation, Algorithm 3 efficiently solves ETD on line graphs.

Algorithm 3 ETD decision on connected line graphs

Require: Connected line graph L.
Ensure: Decision if L is etd.
construct graph G whose line graph is L
if G is bipartite **then**
    construct color classes V_0 and V_1
    for $i = 0, 1$ **do**
        $G_i = (V_i, \{\{u, v\} : \text{ there is } w \in V_{1-i} \text{ with } N_G(w) = \{u, v\}\})$
    end for
    if G_0 or G_1 has a perfect matching M **then**
        return L is etd
    else
        return L is not etd
    end if
else
    return L is not etd
end if

The following lemma implies correctness of Algorithm 3:

Lemma 3. *Let G be a connected graph. G is eted iff G is bipartite and G_0 or G_1 has a perfect matching.*

Proof. Let $G = (V, E)$ be a connected graph with eted set D. We set

$$V_1 = \{v \in V : v \text{ is incident to exactly one edge of } D\} \tag{6}$$

and $V_0 = V \setminus V_1$. We observe

$$V_0 = \{v \in V : v \text{ is incident to either 0 or 2 edges of } D\}. \tag{7}$$

For each vertex $v \in V$, any incident edge has to be dominated by another edge of D. Hence, V_0 and V_1 form a bipartition of G. Since two incident edges of D, due to efficiency, do not have an incident edge of E in common, each incident pair $\{u,v\},\{v,w\} \in D$ corresponds to an edge $m \in E(G_1)$. Let $M \subseteq E(G_1)$ be the collection of these edges. By definition of V_1 and M, M is a perfect matching in G_1.

Now let $G = (V,E)$ be a connected bipartite graph with color classes V_0, V_1 and assume G_1 has a perfect matching M. Thus, each vertex of V_1 is incident to exactly one edge of M. By definition, for every edge $m = \{u,v\} \in M$ there is at least one pair of edges $\{u,w\},\{w,v\} \in E$ with $N_G(w) = \{u,v\} = m$. For each $m \in M$ we choose exactly one of these corresponding pairs and set D as the collection of all edges of these pairs. It is easy to see, that D is an eted set of G. □

Lemma 4. *ETD on line graphs is solvable in $\mathcal{O}(n^2)$ time, where n is the number of vertices of the given graph.*

Proof. Let L be a connected line graph on n vertices and $G = (V,E)$ be the deduced initial graph. The construction of G can be done in $\mathcal{O}(n^2)$ as described in [13]. G_0 and G_1 can be constructed in $\mathcal{O}(|E|^2)$. Perfect matchability can be tested in $\mathcal{O}(\sqrt{|V|}|E|)$ as presented in [14]. As $|E| = n$, we obtain a time complexity of $\mathcal{O}(n^2)$ in total. □

Corollary 4. *WETD on line graphs is solvable in $\mathcal{O}(n^3)$ time, where n is the number of vertices of the given graph.*

Proof. We can change Algorithm 3, using a minimum weighted perfect matching algorithm instead of a maximum cardinality matching algorithm. The weight of an edge e of G_0 or G_1 is the minimum of all sums of the weights of two edges of G corresponding to e. As the minimum perfect matching problem can be solved in $\mathcal{O}(|V|^3)$ (see [15]), we obtain a time complexity of $\mathcal{O}(n^3)$. □

5.2. Reduction of claw-free graphs to line graphs

In this section, we use the following characterization of line graphs given in [13]:

Lemma 5 (Roussopoulos [13]). *A graph is a line graph iff its edges can be partitioned into cliques, such that each vertex lies in at most two of these cliques.*

Lemma 6. *For each input graph G and output graph L of Algorithm 4 the following holds:*

1. *L is a line graph.*
2. *Each etd set of L that is disjoint to I is an etd set of G and vice versa.*

Proof. To 1: As G is claw-free, all vertices in A are contained in at most two inclusionwise maximal cliques and, by definition of A, one of them is a K_2. These inclusionwise maximal cliques are called *active*. As G is claw-free, each vertex of G is contained in at most two active cliques.

Algorithm 4 Reduction from claw-free graphs to line graphs

Require: Weighted claw-free graph $(G = (V, E), c)$.
Ensure: Weighted line graph (L, c'), vertex set $I \subseteq V(L)$ satisfying Lemma 6.
1: $A \leftarrow \{v \in V : v$ is incident to an edge not contained in a triangle $\}$
2: $I \leftarrow V \setminus A$
3: **for all** $e = \{u, v\} \in E$ **do**
4:     **if** $u, v \in I$ and u, v have no common neighbor in A **then**
5:         delete e
6:     **end if**
7: **end for**
8: **while** there are adjacent twins v_1, v_2 with a third neighbor **do**
9:     delete v_2
10: **end while**
11: return resulting weighted graph $(L, c|_{V(L)})$ and $I \subseteq V(L)$

Let F denote the resulting graph after step 7. We observe, that each etd set of G is disjoint to $I = V \setminus A$. Hence, F is etd iff G is etd. Furthermore, each edge of F belongs to an active clique. Thus, the neighborhood of each vertex is contained in at most two active cliques. The deletion of adjacent twins clearly preserves the latter two properties, since all adjacent twins in F belong to I (except for the case that they have no third neighbor).

Since there are no such twins in L, there are no two vertices contained in the same two active cliques. Thus, no two active cliques share an edge and we can therefore apply Lemma 5 to L with respect to the active cliques.

To 2: Assume G has an etd set X. As X is disjoint to I, no vertex of X and no edge incident to a vertex of X gets deleted during the procedure. Therefore, X is an etd set of L, too.

Assume L has an etd set X disjoint to I. Addition of adjacent twins of vertices of I has no effect on X being an etd set. Furthermore, adding edges between vertices of I does not have an effect on the efficiency of X, since X is disjoint to I. As G can be constructed from L by these two operations, X is an etd set of G, too. ☐

Theorem 7. *WETD on claw-free graphs is solvable in $\mathcal{O}(n^3)$ time, where n is the number of vertices of the given graph.*

Proof. We use Algorithm 4 to solve WETD on claw-free graphs in the following way. Given a claw-free graph G, Algorithm 4 computes a line graph L satisfying Lemma 6. We can now apply Algorithm 3 to L. Thereby, the edges corresponding to vertices of I must be neglected in the definition of the auxiliary graphs G_0 and G_1. Lemma 6 and Corollary 4 give correctness of the procedure.

To analyse the time complexity of the whole procedure, let $G = (V, E)$ have n vertices and m edges. Algorithm 4 computes the set A in $\mathcal{O}(mn)$ time by iteratively taking an edge and checking for the same adjacencies of its two incident vertices. The deletion of all edges between vertices of I without common neighbor in A and the detection of adjacent twins can both be done in the same time in an analog way. Now Algorithm 3 solves WETD on the output graph L in $\mathcal{O}(n^3)$ time. Since $m \leq n^2$, we obtain overall time complexity of $\mathcal{O}(n^3)$. $\square$

5.3. Perfectness of etd claw-free graphs

As Lemma 3 states, etd line graphs come from bipartite graphs only. Therefore, etd line graphs are perfect. We claim that the same holds for etd claw-free graphs and prove this claim in two steps.

Lemma 7. *An etd claw-free graph does not contain odd holes.*

Proof. Let G be a claw-free graph with etd set X. Assume C is an odd hole of G. Let $x \in X$. We show that x is adjacent to exactly 0 or 2 vertices of C, a contradiction to the oddity of C.

Assume x is adjacent to exactly one vertex v of C. Then $N(v) \cap C, x$ and v induce a claw, in contradiction to the premise. Assume x is adjacent to at least three vertices u, v and w of C. As C is induced, x cannot belong to C itself. As C is of length at least 5, u, v and w are not pairwise adjacent. Assume u is not adjacent to v. As G is claw-free, x has neighbor y in X not belonging to C. Due to efficiency, y is not adjacent to u or v and therefore $G(\{x, u, v, y\})$ is a claw, in contradiction to the premise. $\square$

In the case of odd antiholes we use *Ben Rebea's lemma*, as presented in [16]:

Lemma 8 (Ben Rebea's lemma, see [16])**.** *If a claw-free graph contains a stable set of size at least three and an odd antihole, then it contains an odd hole of length 5.*

Lemma 9. *An etd claw-free graph does not contain odd antiholes.*

Proof. Let G be a connected claw-free graph with etd set X. In the case of $|X| = 2$, the open neighborhood of each of the two dominating vertices is the disjoint union of two cliques, since G is claw-free. Hence, G does not contain an odd antihole. In the case of $|X| \geq 4$, G contains a stable set of size at least three. If G contained an odd antihole, it would also contain an odd hole by Lemma 8. This would be a contradiction to Lemma 7. $\square$

By the famous characterization of perfect graphs, we obtain

Theorem 8. *Etd claw-free graphs are perfect.*

References

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

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

[3] E.M. Bakker, J. van Leeuwen, *Some domination problems on trees and on general grahps*, technical report RUU-CS-91-22 (1991), Department of Information and Computing Sciences, Utrecht University.

[4] J. Kratochvíl, P.D. Manuel, M. Miller, *Generalized domination in chordal graphs*, Nordic Journal of Computing 2 (1995), pp. 41–50.

[5] G.J. Chang, C.P. Rangan, S.R. Coorg, *Weighted independent perfect domination on cocomparability graphs*, Discrete Applied Mathematics 63 (1995), pp. 215–222.

[6] C.L. Lu, C.Y. Tang, *Weighted efficient domination problem on some perfect graphs*, Discrete Applied Mathematics 117 (2002), pp. 163–182.

[7] D.L. Grinstead, P.J. Slater, N.A. Sherwani and N.D. Holmes, *Efficient edge domination problems in graphs*, Inform. Process. Lett. 8 (1993), pp. 221–228.

[8] M. E. Dyer and A. M. Frieze, *Planar 3DM is NP-complete*, Journal of Algorithms 7 (1986), pp. 174–184.

[9] M. Conforti, G. Cornuéjols and K. Vušković, *Balanced matrices*, Discrete Mathematics 306 (2006), pp. 2411–2437.

[10] A.E. Brouwer, P. Duchet, A. Schrijver, *Graphs whose neighborhoods have no special cycles*, Discrete Mathematics 47 (1983), pp. 177–182.

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

[12] 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.

[13] N.D. Roussopoulos, *A* $max\{m,n\}$ *algorithm for determining the graph* H *from its line graph* G, Inform. Process. Lett. 2 (1973), pp. 108–112.

[14] V.V. Vazirani, *A theory of alternating paths and blossoms for proving correctness of the* $O(\sqrt{V}E)$ *general graph maximum matching algorithm*, Combinatorica 14 (1994), pp. 71–109.

[15] H.N. Gabow, *Implementation of Algorithms for Maximum Matching on Nonbipartite Graphs*, Ph.D. Thesis, Stanford University, 1974.

[16] J.L. Fouquet, *A strengthening of Ben Rebea's lemma*, J. Comb. Theory Ser. B 59 (1993), pp. 35–40.