Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Theoretical Computer Science 301 (2003) 381–397

Theoretical Computer Science

www.elsevier.com/locate/tcs

Efficient minus and signed domination in graphs☆

Chin Lung Lu[a], Sheng-Lung Peng[b], Chuan Yi Tang[c,*]

[a] Department of Biological Science and Technology, National Chiao Tung University, Hsinchu 300, Taiwan, ROC

[b] Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien 974, Taiwan, ROC

[c] Department of Computer Science, National Tsing Hua University, Hsinchu 300, Taiwan, ROC

Received 4 October 2001; accepted 15 August 2002
Communicated by D.-Z. Du

Abstract

An efficient minus (respectively, signed) dominating function of a graph $G = (V, E)$ is a function $f: V \to \{-1, 0, 1\}$ (respectively, $\{-1, 1\}$) such that $\sum_{u \in N[v]} f(u) = 1$ for all $v \in V$, where $N[v] = \{v\} \cup \{u \mid (u, v) \in E\}$. The efficient minus (respectively, signed) domination problem is to find an efficient minus (respectively, signed) dominating function of G. In this paper, we show that the efficient minus (respectively, signed) domination problem is NP-complete on chordal graphs, chordal bipartite graphs, planar bipartite graphs and planar graphs of maximum degree 4 (respectively, on chordal graphs). Based on the forcing property on blocks of vertices and automata theory, we provide a uniform approach to show that in a special class of interval graphs, every graph (respectively, every graph with no vertex of odd degree) has an efficient minus (respectively, signed) dominating function. We also give linear-time algorithms to find these functions. Besides, we show that the efficient minus domination problem is equivalent to the efficient domination problem on trees.

Keywords: Efficient minus domination; Efficient signed domination; Chordal graphs; Chordal bipartite graphs; Planar bipartite graphs; Chain interval graphs

☆ A preliminary version of this paper appears in the Proc. of the Eleventh Annual Internat. Symp. on Algorithms and Computation (ISAAC 2000). This research was partly supported by the National Science Council of the Republic of China under grant NSC 89-2213-E-259-010.

* Corresponding author.

E-mail address: cytang@cs.nthu.edu.tw (C.Y. Tang).

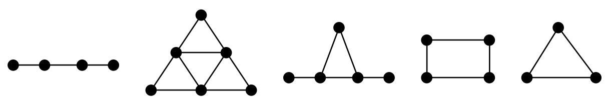
0304-3975/03/$ - see front matter
PII: S0304-3975(02)00594-7

1. Introduction

Let $G=(V,E)$ be a finite, undirected and simple graph, i.e., G has no multiple edges and no self-loops. For any vertex $v\in V$, the *open neighborhood* of v is $N(v)=\{u\in V|(u,v)\in E\}$ and the *closed neighborhood* of v is $N[v]=\{v\}\cup N(v)$. Let $f:V\to Y$ be a function which assigns to each $v\in V$ a value in Y, where Y is a subset of real numbers. To simplify notation, we let $f(S)=\sum_{u\in S} f(u)$ for any set $S\subseteq V$. We call $f(V)$ the *weight* of f. The function f is called an *efficient Y-dominating function* if $f(N[v])=1$ for every vertex $v\in V$ and Y is called the *weight set* of f. In particular, f is called an *efficient* (respectively, *efficient minus* and *efficient signed*) *dominating function* if the weight set Y is $\{0,1\}$ (respectively, $\{-1,0,1\}$ and $\{-1,1\}$). In [1], Bange et al. showed that if f_1 and f_2 are any two efficient Y-dominating functions of G, then $f_1(V)=f_2(V)$. In other words, all efficient Y-dominating functions of G have the same weight. Hence, the *efficient Y-domination problem* is the problem of finding an efficient Y-dominating function of G. The efficient minus and signed domination problems have applications in sociology, electronics and facility location of operation research [7–9,14,15]. Note that not every graph has an efficient (minus, signed) dominating function (see Fig. 1 for examples). By the definition, an efficient (signed) dominating function is also an efficient minus dominating function, but the converse is not true.

There is an extensive number of papers concerning the algorithmic complexity of the efficient domination problem in several graph classes [2,4–6,10,17–21]. The most frequently used algorithmic technique for solving the efficient domination problems is dynamic programming based on the *forcing property* on vertices, i.e., the value 1 assigned to a vertex v forces the other vertices in $N[v]$ to be assigned the value 0. However, for the efficient minus and signed domination problems, this forcing property does not work because of the "neutralization" of values -1 and 1. Hence, the techniques used for the efficient domination problem cannot be applied to these two problems. To date, the only known result is that the efficient signed domination problem is NP-complete on general graphs [1].

In this paper, we show that the efficient minus domination problem is NP-complete on chordal graphs, chordal bipartite graphs, planar bipartite graphs and planar graphs of maximum degree 4; the efficient signed domination problem is NP-complete on



EDF	Yes	No	No	No	Yes
EMDF	Yes	Yes	Yes	No	Yes
ESDF	No	Yes	No	No	Yes

Fig. 1. Each entry means whether the graph has *EDF*, *EMDF* or *ESDF*, where *EDF*, *EMDF* and *ESDF* stand for efficient dominating function, efficient minus dominating function and efficient signed dominating function, respectively.

chordal graphs. We find that a special class of interval graphs, which we call chain interval graphs, can be represented as a sequence of blocks, where a *block* is a set of vertices in which all vertices have the same closed neighborhood. According to clique and block structures, the chain interval graphs can be described by a formal language $\mathscr{L}$. By applying the forcing property on blocks, we create a finite state automaton which exactly accepts $\mathscr{L}$. As a result, every chain interval graph has an efficient minus dominating function. Similarly, we show that every chain interval graph with no vertex of odd degree has an efficient signed dominating function. In addition, we give linear-time algorithms to find them. For trees, we show that the efficient minus domination problem coincides with the efficient domination problem. According to Bange et al. [2], we can hence find an efficient minus dominating function of a tree in linear time.

2. NP-completeness results

A graph is *chordal* if every cycle of length greater than 3 has a *chord*, i.e., an edge between two non-consecutive vertices of the cycle [13]. *Chordal bipartite graphs* are bipartite graphs in which every cycle of length greater than 4 has a chord [13]. Note that the class of chordal bipartite graphs is not the intersection of the classes of chordal graphs and bipartite graphs.

Efficient domination problem (*EDP*)

Instance: A graph $G=(V,E)$.

Question: Does G have an efficient dominating function?

Efficient minus domination problem (*EMDP*)

Instance: A graph $G=(V,E)$.

Question: Does G have an efficient minus dominating function?

It is known that *EDP* is NP-complete even when restricted to chordal graphs [21], chordal bipartite graphs [18,19], planar bipartite graphs [18,19] and planar graphs of maximum degree 3 [10]. In the following, we show that *EMDP* on chordal graphs, chordal bipartite graphs, planar bipartite graphs and planar graphs of maximum degree 4 is NP-complete by reducing from *EDP*.

Theorem 1. *EMDP is NP-complete on chordal graphs, chordal bipartite graphs, planar bipartite graphs and planar graphs of maximum degree* 4.

Proof. It is not difficult to see that *EMDP* on chordal graphs (chordal bipartite graphs, planar bipartite graphs and planar graphs of maximum degree 4) is in NP. Hence, we only show that this problem can be reduced from *EDP* on the same graphs in polynomial time.

Given a graph $G=(V_G,E_G)$, we construct the graph $H=(V_H,E_H)$ by adding a path of length 3, say v–v_1–v_2–v_3, to each vertex v of G. That is, $V_H=V_G\cup(\bigcup_{v\in V_G}\{v_1,v_2,v_3\})$ and $E_H=E_G\cup(\bigcup_{v\in V_G}\{(v,v_1),(v_1,v_2),(v_2,v_3)\})$. Then, H is a chordal graph (respectively, chordal bipartite graph, planar bipartite graph and planar graph of maximum degree 4) if G is a chordal graph (respectively, chordal bipartite graph, planar bipartite graph and planar graph of maximum degree 3). Clearly, the construction of H can be done in polynomial time.

Now, we show that G has an efficient dominating function f if and only if H has an efficient minus dominating function g. First, suppose that G has an efficient dominating function f. Note that for each $v \in V_G$, there are four corresponding vertices v, v_1, v_2 and v_3 in V_H. Define a function $g: V_H \to \{-1, 0, 1\}$ of H as follows. Let $g(v) = f(v)$ for each $v \in V_G$. Furthermore, if $g(v) = 0$, then let $g(v_1) = 0, g(v_2) = 1$ and $g(v_3) = 0$; otherwise, let $g(v_1) = g(v_2) = 0$ and $g(v_3) = 1$. It can be verified that $g(N[v]) = g(N[v_1]) = g(N[v_2]) = g(N[v_3]) = 1$. Hence, g is an efficient minus dominating function of H.

Conversely, suppose that H has an efficient minus dominating function g. We then claim that $g(v) \geqslant 0$, $g(v_1) = 0$, $g(v_2) \geqslant 0$ and $g(v_3) \geqslant 0$ for each $v \in V_G$. If $g(v_3) = -1$, then $g(N[v_3]) \leqslant 0$, which contradicts the fact that $g(N[v_3]) = 1$. If $g(v_2) = -1$, then $g(N[v_3]) \leqslant 0$, a contradiction again. If $g(v_1) = -1$, then $g(N[v_2]) = 1$ implies that $g(v_2) = g(v_3) = 1$, which leads to $g(N[v_3]) = 2$, a contradiction. If $g(v_1) = 1$, then $g(N[v_2]) = 1$ implies that $g(v_2) = g(v_3) = 0$, which leads to $g(N[v_3]) = 0$, a contradiction. If $g(v) = -1$, then $g(N[v_1]) = 1$ and $g(N[v_2]) = 1$ imply that $g(v_1) = g(v_2) = 1$ and $g(v_3) = -1$, which contradicts the fact that $g(v_3) \geqslant 0$. Define a function $f: V_G \to \{0, 1\}$ of G by letting $f(v) = g(v)$ for every $v \in V_G$. It is not hard to see that f is an efficient dominating function of G since $f(N[v]) = 1$ for all v in V_G. □

One-in-three 3SAT problem

Instance: A set U of n boolean variables and a collection $\mathscr{C}$ of m clauses over U such that each clause has exactly three literals.

Question: Is there a truth assignment $t: U \to \{true, false\}$ for $\mathscr{C}$ such that each clause in $\mathscr{C}$ has exactly one true literal?

Efficient signed domination problem (*ESDP*)

Instance: A graph $G = (V, E)$.

Question: Does G have an efficient signed dominating function?

Next, we will show that one-in-three 3SAT problem, which is known to be NP-complete [11], is reducible to *ESDP* on chordal graphs in polynomial time.

Theorem 2. *ESDP is NP-complete on chordal graphs.*

Proof. It is not hard to see that *ESDP* on chordal graphs is in NP. We now show that one-in-three 3SAT problem is polynomially reducible to this problem. Let $U = \{u_1, u_2, \ldots, u_n\}$ and $\mathscr{C} = \{C_1, C_2, \ldots, C_m\}$ be an instance of one-in-three 3SAT problem, where each clause C_j, $1 \leqslant j \leqslant m$, contains three literals $l_{j,1}, l_{j,2}$ and $l_{j,3}$. We assume that no clause contains both a literal and its negation because this clause is always true and can be omitted. Let $U' = \{u_i, \bar{u}_i | 1 \leqslant i \leqslant n\}$. We construct a chordal graph $G = (V, E)$ as follows.

(1) For each variable u_i, $1 \leqslant i \leqslant n$, we construct the subgraph G_{u_i} of G as shown in Fig. 2(a).

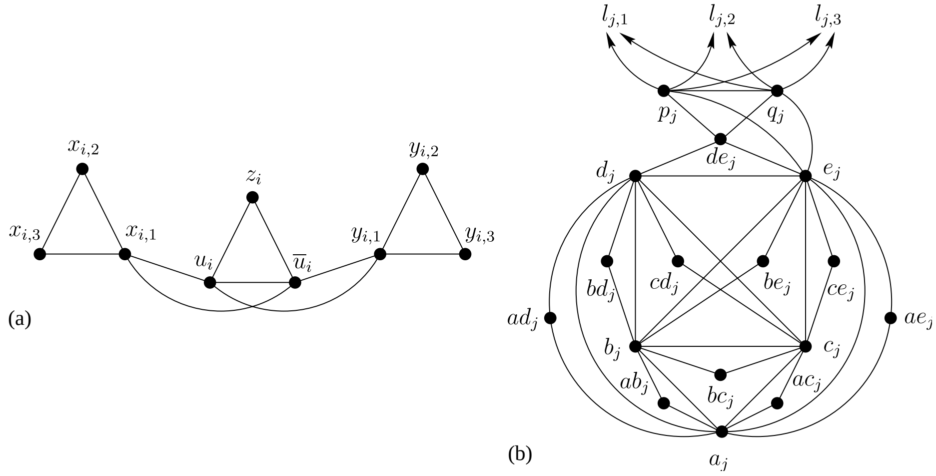


Fig. 2. The subgraph: (a) G_{u_i}, (b) G_{C_j}.

(2) For each clause C_j, $1 \leqslant j \leqslant m$, we construct the subgraph G_{C_j} of G as shown in Fig. 2(b), where G_{C_j} is connected to the three vertices corresponding to the three literals in clause C_j.
(3) We add all possible edges in $G_{U'}$ such that $G_{U'}$ forms a complete subgraph, i.e., any two vertices of U' are adjacent in G.

Note that $l_{j,1}, l_{j,2}, l_{j,3} \in U'$. Since the subgraphs G_{u_i}, $1 \leqslant i \leqslant n$, and G_{C_j}, $1 \leqslant j \leqslant m$, are all chordal and the subgraph $G_{U'}$ is complete, G is a chordal graph and can be constructed in polynomial time.

Let $f : V \to \{-1, 1\}$ be an efficient signed dominating function of G. We discuss some properties of f as follows. Clearly, for a vertex v of degree $2k$, where k is a positive integer, f must assign $k+1$ vertices of $N[v]$ values of 1 and k vertices of $N[v]$ values of -1. Consider each subgraph G_{C_j}, $1 \leqslant j \leqslant m$. Suppose that $f(a_j) = -1$. Then, $f(ab_j) = f(b_j) = f(ac_j) = f(c_j) = f(ad_j) = f(d_j) = f(ae_j) = f(e_j) = 1$ (since ab_j, ac_j, ad_j and ae_j are vertices of degree 2) and hence $f(N[a_j]) = 8 - 1 > 1$, a contradiction. Therefore, $f(a_j) = 1$ and similarly, $f(b_j) = f(c_j) = 1$. Suppose that $f(d_j) = -1$. Then, $f(ad_j) = f(bd_j) = f(cd_j) = 1$ and $f(d_j) + f(de_j) + f(e_j) \geqslant -3$. As a result, $f(N[d_j]) \geqslant 6 - 3 > 1$, a contradiction. Hence, $f(d_j) = 1$ and similarly, $f(e_j) = 1$. For each vertex v of degree 2 in G_{C_j}, $f(v) = -1$ since $f(a_j) = f(b_j) = f(c_j) = f(d_j) = f(e_j) = 1$. Note that $f(N[d_j]) = 1$ implies that $f(de_j) = -1$ and then $f(N[e_j]) = 1$ implies that $f(p_j) + f(q_j) = 0$. Consider each G_{u_i}, $1 \leqslant i \leqslant n$. Since $N[y_{i,1}] = N[y_{i,2}] \cup \{u_i, \bar{u}_i\}$ and $f(N[y_{i,1}]) = f(N[y_{i,2}]) = 1$, $f(u_i) + f(\bar{u}_i) = 0$ and hence $f(z_i) = 1$. Note that u_i (respectively, $\bar{u}_i$) is adjacent to p_j if and only if u_i (respectively, $\bar{u}_i$) is adjacent to q_j, and for each $1 \leqslant i' \leqslant n$ with $i' \neq i$, both $u_{i'}$ and $\bar{u}_{i'}$ are adjacent to u_i (respectively, $\bar{u}_i$). Hence, $f(N[u_i]) = 1 + f(x_{i,1}) + f(y_{i,1}) = 1$, which means that $f(x_{i,1}) + f(y_{i,1}) = 0$. Moreover, $f(y_{i,1}) + f(y_{i,2}) + f(y_{i,3}) = 1$ and $f(x_{i,1}) + f(x_{i,2}) + f(x_{i,3}) = 1$ since $f(N[y_{i,2}]) = 1$ and $f(N[x_{i,2}]) = 1$, respectively.

Now, we show that $\mathscr{C}$ has a satisfying truth assignment if and only if G has an efficient signed dominating function. First, suppose that f is an efficient signed dominating function of G. Let $1 \leqslant i \leqslant n$ and $1 \leqslant j \leqslant m$. Since $f(p_j)+f(q_j)=0$, $f(de_j)=-1$ and $f(e_j)=1$, $f(N[p_j])=1$ imply that there is exactly one of $l_{j,1}, l_{j,2}$ and $l_{j,3}$ whose function value is -1. Let $t: U \to \{true, false\}$ be defined by $t(u_i)=true$ if and only if $f(u_i)=-1$. Since $f(u_i)+f(\bar{u}_i)=0$, $t(u_i)$ is *true* if and only if $t(\bar{u}_i)$ is *false*. Hence, t is a one-in-three satisfying truth assignment for $\mathscr{C}$.

Conversely, suppose that $\mathscr{C}$ has a satisfying truth assignment. Then, we can identify an efficient signed dominating function f of G according to the mention above. In particular, $f(u_i)=-1$ if u_i is assigned *true*; otherwise, $f(u_i)=1$. □

3. Interval graphs

A graph $G=(V,E)$ is an *interval graph* if there exists a one-to-one correspondence between V and a family F of intervals such that two vertices in V are adjacent if and only if their corresponding intervals overlap [3,13]. We call F an *interval representation* of G. $S \subseteq V$ is a *clique* if any two vertices of S are adjacent in G. A clique is *maximal* if there is no clique properly containing it as a subset. Gilmore and Hoffman [12] showed that for an interval graph G, its maximal cliques can be linearly ordered such that for every vertex v of G, the maximal cliques containing v occur consecutively. We use $G=(C_1, C_2, \ldots, C_s)$ to denote the interval graph G with s linearly ordered maximal cliques and call it the *clique structure* of G (see Fig. 3(c)). Note that if G is connected, then $C_i \cap C_{i+1} \neq \emptyset$ for any $1 \leqslant i < s$.

The efficient domination problem on interval graphs can be solved in linear time using dynamic programming, which based on the forcing property on vertices (i.e., if we assign 1 to a vertex v, then we have to assign 0 to all vertices in $N(v)$; if we assign 0 to v, then we have to assign 1 to exactly one vertex of $N(v)$ and 0 to the remaining vertices in $N(v)$). However, this technique seems not be applied for both the efficient minus and signed domination problems on interval graphs because the forcing property on vertices does not work due to the "neutralization" of values -1 and 1.

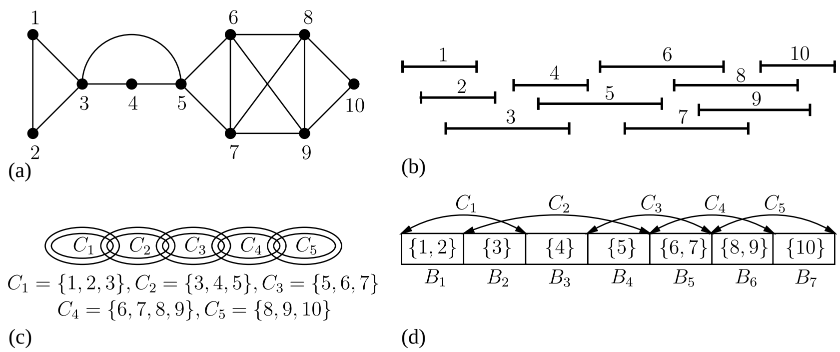


Fig. 3. (a) A chain interval graph G; (b) an interval representation of G; (c) the clique structure of G; and (d) the block structure of G.

In this section, we consider a special class of interval graphs, which we call the class of chain interval graphs. A *chain interval graph* $G=(C_1,C_2,\ldots,C_s)$ is an interval graph in which $C_{i-1}\cap C_i\cap C_{i+1}=\emptyset$ for any $1<i<s$ (see Fig. 3). It is worth mentioning that chain interval graphs contain proper ptolemaic interval graphs (i.e., the intersection of the classes of proper interval graphs and ptolemaic graphs) as a subclass [3]. Without loss of generality, we assume that $G=(C_1,C_2,\ldots,C_s)$ is a connected chain interval graph in the following. By the definition, any clique C_i of G, $1\leqslant i\leqslant s$, can be partitioned into three subsets $B_{i,l}=C_{i-1}\cap C_i$, $B_{i,r}=C_i\cap C_{i+1}$ and $B_{i,m}=C_i\backslash(B_{i,l}\cup B_{i,r})$, where $C_0=C_{s+1}=\emptyset$. We call these subsets *blocks* and say that C_i *contains* blocks $B_{i,l},B_{i,m}$ and $B_{i,r}$. Note that block $B_{i,m}$ might be empty. Let $bn(C_i)$ be the number of non-empty blocks of C_i.

Remark 3. If $s\geqslant 2$, then both C_1 and C_s contain exactly two blocks, i.e., $bn(C_1)=bn(C_s)=2$.

Remark 4. For any two consecutive cliques C_i and C_{i+1} of G, $B_{i,r}=B_{i+1,l}$. For a vertex $v\in B_{i,r}$, $N[v]=C_i\cup C_{i+1}=C_i\cup(C_{i+1}\backslash B_{i+1,l})$, i.e., $N[v]$ can be partitioned into C_i and $C_{i+1}\backslash B_{i+1,l}$.

Remark 5. Let C_i be a clique with $bn(C_i)=3$. For a vertex $v\in B_{i,m}$, $N[v]=C_i=B_{i,l}\cup B_{i,m}\cup B_{i,r}$.

Based on the clique structure of G, we can represent G by linearly ordered blocks $B_1,B_2,\ldots,B_t$, $t\geqslant s$, such that each clique contains either consecutive two or three blocks. We call this representation the *block structure* of G and denote it by $G=(B_1,B_2,\ldots,B_t)$ (see Fig. 3(d)). Note that $B_1,B_2,\ldots,B_t$ is a partition of V. We define the *block-number string* $bs(G)$ of G to be the string $bn(C_1)bn(C_2)\cdots bn(C_s)$. For example, the block-number string of G shown in Fig. 3 is 23222. For convenience, if G contains only one clique, we define $bn(G)=2$. Let $\mathscr{L}$ be the language consisting of the block-number strings of all chain interval graphs. Then, we have the following lemma immediately.

Lemma 6. *$\mathscr{L}$ is a regular language and its regular expression is* $2+2(2+3)^*2$.

3.1. The efficient minus domination problem

Let $f:V\to\{-1,0,1\}$ be an efficient minus dominating (*EMD* for short) function of $G=(B_1,B_2,\ldots,B_t)$. We call f a *simple EMD function* of G if $f(B_i)\in\{-1,0,1\}$ for all B_i, $1\leqslant i\leqslant t$. We will show later that any chain interval graph affirmatively admits a simple *EMD* function. A clique C_i, $1\leqslant i\leqslant s$, is P type if $bn(C_i)=2$ and Q type if $bn(C_i)=3$. The clique C_i is called a $P_{(a,b)}$ clique if C_i is P type, $f(B_{i,l})=a$ and $f(B_{i,r})=b$, and a $Q_{(a,b,c)}$ clique if C_i is Q type, $f(B_{i,l})=a, f(B_{i,m})=b$ and $f(B_{i,r})=c$. Note that C_1 and C_s are always a P-type clique by Remark 3. According to Remarks 4 and 5, respectively, we have the following two lemmas immediately.

Lemma 7. *Let f be an EMD function of G and let C_i and C_{i+1} be two consecutive cliques of G. Then, $f(C_i) + f(C_{i+1} \backslash B_{i+1,l}) = 1$.*

Lemma 8. *Let f be an EMD function of G and C_i be a Q type clique. Then, $f(B_{i,l}) + f(B_{i,m}) + f(B_{i,r}) = 1$.*

Lemma 9. *Let f be a simple EMD function of G. If $s \geqslant 2$, then C_1 and C_s are either a $P_{(0,1)}$ or $P_{(1,0)}$ clique.*

Proof. Let $v \in B_{1,l}$. Clearly, $N[v] = C_1 = B_{1,l} \cup B_{1,r}$. $f(N[v]) = 1$ implies that either (1) $f(B_{1,l}) = 0$ and $f(B_{1,r}) = 1$, or (2) $f(B_{1,l}) = 1$ and $f(B_{1,r}) = 0$. That is, C_1 is either a $P_{(0,1)}$ or $P_{(1,0)}$ clique. Similarly, C_s is either a $P_{(0,1)}$ or $P_{(1,0)}$ clique. □

Lemma 10. *Let f be a simple EMD function of G and $1 < i < s$. If clique C_i is Q type, then C_i is either a $Q_{(0,0,1)}, Q_{(0,1,0)}, Q_{(1,0,0)}$ or $Q_{(1,-1,1)}$ clique.*

Proof. By Lemma 8, we have $f(B_{i,l}) + f(B_{i,m}) + f(B_{i,r}) = 1$. Since f is a simple *EMD* function, $f(B_{i,l}), f(B_{i,m}), f(B_{i,r}) \in \{-1, 0, 1\}$. We claim that $f(B_{i,r}) \neq -1$ and $f(B_{i,l}) \neq -1$. Suppose that $f(B_{i,r}) = -1$. Then, $f(B_{i,l}) + f(B_{i,m}) = 2$. If C_{i+1} is Q type, then by Lemma 7, $f(B_{i,l}) + f(B_{i,m}) + f(B_{i,r}) + f(B_{i+1,m}) + f(B_{i+1,r}) = 1$. As a result, $f(B_{i+1,l}) + f(B_{i+1,m}) + f(B_{i+1,r}) = -1$, which contradicts Lemma 8. In other words, C_{i+1} is P type. By Lemma 7, we have $f(B_{i+1,r}) = 0$ and hence C_{i+1} is a $P_{(-1,0)}$ clique. According to Lemma 9, $i+1 < s$. If C_{i+2} is P type, then since $f(B_{i+2,r}) \in \{-1, 0, 1\}$, $f(B_{i+1,l}) + f(B_{i+1,r}) + f(B_{i+2,r}) \leqslant 0$, a contradiction to Lemma 7. If C_{i+2} is Q type, then $f(B_{i+1,l}) + f(B_{i+1,r}) + f(B_{i+2,m}) + f(B_{i+2,r}) = 1$ by Lemma 7. As a result, $f(B_{i+2,l}) + f(B_{i+2,m}) + f(B_{i+2,r}) = 2$, which contradicts Lemma 8. Therefore, $f(B_{i,r}) \neq -1$. Similarly, we have $f(B_{i,l}) \neq -1$.

By the above discussion, $f(B_{i,l}), f(B_{i,r}) \in \{0, 1\}$, $f(B_{i,m}) \in \{-1, 0, 1\}$ and $f(B_{i,l}) + f(B_{i,m}) + f(B_{i,r}) = 1$. We have the following two cases.

Case 1: $f(B_{i,r}) = 0$. Either $f(B_{i,l}) = 0$ and $f(B_{i,m}) = 1$, or $f(B_{i,l}) = 1$ and $f(B_{i,m}) = 0$. That is, C_i is a $Q_{(0,1,0)}$ or $Q_{(1,0,0)}$ clique.

Case 2: $f(B_{i,r}) = 1$. Either $f(B_{i,l}) = 0$ and $f(B_{i,m}) = 0$, or $f(B_{i,l}) = 1$ and $f(B_{i,m}) = -1$. That is, C_i is a $Q_{(0,0,1)}$ or $Q_{(1,-1,1)}$ clique. □

Lemma 11. *Let f be a simple EMD function of G and $1 < i < s$. If clique C_i is P type, then C_i is either a $P_{(0,0)}, P_{(0,1)}$ or $P_{(1,0)}$ clique.*

Proof. We first claim that $f(B_{i,r}) \neq -1$ and $f(B_{i,l}) \neq -1$. Suppose that $f(B_{i,r}) = -1$. According to Lemma 10, C_{i+1} is not Q type. Hence, C_{i+1} is P type and $f(B_{i,l}) = f(B_{i+1,r}) = 1$ since $f(B_{i,l}) + f(B_{i,r}) + f(B_{i+1,r}) = 1$ by Lemma 7. That is, C_i and C_{i+1} are $P_{(1,-1)}$ and $P_{(-1,1)}$ cliques, respectively. By Lemma 9, $i+1 < s$. If C_{i+2} is Q type, then according to Lemma 10, C_{i+2} must be a $Q_{(1,0,0)}$ or $Q_{(1,-1,1)}$ clique. As a result, $f(B_{i+1,l}) + f(B_{i+1,r}) + f(B_{i+2,m}) + f(B_{i+2,r}) = 0$, a contradiction to Lemma 7. In other words, C_{i+2} is P type and $f(B_{i+2,m}) = 1$ by Lemma 7. Hence, C_{i+2} is a $P_{(1,1)}$

clique and $i+2<s$. Similarly, C_{i+3} is P type and $f(B_{i+3,r})=-1$. In other words, C_{i+3} is a $P_{(1,-1)}$ clique which is the same as C_i. Continuing this way, we will find that C_s is either a $P_{(1,-1)}, P_{(-1,1)}$ or $P_{(1,1)}$ clique, which contradicts Lemma 9. Therefore, $f(B_{i,r})\neq -1$. Similarly, we have $f(B_{i,l})\neq -1$.

Next, we claim that C_i cannot be a $P_{(1,1)}$ clique. Suppose that C_i is a $P_{(1,1)}$ clique, i.e., $f(B_{i,r})=f(B_{i,l})=1$. If C_{i+1} is Q type, then $f(B_{i,l}) + f(B_{i,r}) + f(B_{i+1,m}) + f(B_{i+1,r})=1$ by Lemma 7. As a result $f(B_{i+1,l}) + f(B_{i+1,m}) + f(B_{i+1,r})=0$, a contradiction to Lemma 8. Hence, C_{i+1} is P type. By Lemma 7, we have $f(B_{i+1,r})=-1$, a contradiction, too.

As mentioned above, C_i may be either a $P_{(0,0)}, P_{(0,1)}$ or $P_{(1,0)}$ clique. □

Lemma 12. *Let f be a simple EMD function of G and $1\leqslant i<s$.*

(1) *If C_i is a $P_{(0,0)}$ clique, then C_{i+1} is either a $P_{(0,1)}$, $Q_{(0,0,1)}$ or $Q_{(0,1,0)}$ clique.*
(2) *If C_i is a $P_{(0,1)}$ clique, then C_{i+1} is either a $P_{(1,0)}$, $Q_{(1,0,0)}$ or $Q_{(1,-1,1)}$ clique.*
(3) *If C_i is a $P_{(1,0)}$ clique, then C_{i+1} is a $P_{(0,0)}$ clique.*
(4) *If C_i is a $Q_{(0,0,1)}$ clique, then C_{i+1} is either a $P_{(1,0)}$, $Q_{(1,0,0)}$ or $Q_{(1,-1,1)}$ clique.*
(5) *If C_i is a $Q_{(0,1,0)}$ clique, then C_{i+1} is a $P_{(0,0)}$ clique.*
(6) *If C_i is a $Q_{(1,0,0)}$ clique, then C_{i+1} is a $P_{(0,0)}$ clique.*
(7) *If C_i is a $Q_{(1,-1,1)}$ clique, then C_{i+1} is either a $P_{(1,0)}$, $Q_{(1,0,0)}$ or $Q_{(1,-1,1)}$ clique.*

Proof. (1) Let C_i be a $P_{(0,0)}$ clique. If C_{i+1} is P type, then by Lemma 7, $f(B_{i,l}) + f(B_{i,r}) + f(B_{i+1,r})=1$ and hence $f(B_{i+1,r})=1$, i.e., C_{i+1} is a $P_{(0,1)}$ clique. If C_{i+1} is Q type, then $f(B_{i,l}) + f(B_{i,r}) + f(B_{i+1,m}) + f(B_{i+1,r})=1$ and hence $f(B_{i+1,m}) + f(B_{i+1,r})=1$. According to Lemma 10, C_{i+1} may be a $Q_{(0,0,1)}$ or $Q_{(0,1,0)}$ clique.

(2) Similar to (1).

(3) Let C_i be a $P_{(1,0)}$ clique. If C_{i+1} is Q type, then $f(B_{i,l})+f(B_{i,r})+f(B_{i+1,m})+f(B_{i+1,r})=1$ by Lemma 7. Note that $B_{i,r}=B_{i+1,l}$. As a result, $f(B_{i+1,l})+f(B_{i+1,m})+f(B_{i+1,r})=0$, a contradiction to Lemma 8. In other words, C_{i+1} is exactly P type. It is not hard to see that C_{i+1} is a $P_{(0,0)}$ clique.

(4) Let C_i be a $Q_{(0,0,1)}$ clique. If C_{i+1} is P type, then by Lemma 7, $f(B_{i,l}) + f(B_{i,m})+f(B_{i,r})+f(B_{i+1,r})=1$ and hence $f(B_{i+1,r})=0$, i.e., C_{i+1} is a $P_{(1,0)}$ clique. If C_{i+1} is Q type, then $f(B_{i,l}) + f(B_{i,m}) + f(B_{i,r}) + f(B_{i+1,m}) + f(B_{i+1,r})=1$ and $f(B_{i+1,m}) + f(B_{i+1,r})=0$. By Lemma 10, C_{i+1} may be a $Q_{(1,0,0)}$ or $Q_{(1,-1,1)}$ clique.

(5) Let C_i be a $Q_{(0,1,0)}$ clique. If C_{i+1} is P type, then by Lemma 7, $f(B_{i,l}) + f(B_{i,m})+f(B_{i,r})+f(B_{i+1,r})=1$ and hence $f(B_{i+1,r})=0$, i.e., C_{i+1} is a $P_{(0,0)}$ clique. If C_{i+1} is Q type, then $f(B_{i,l}) + f(B_{i,m}) + f(B_{i,r}) + f(B_{i+1,m}) + f(B_{i+1,r})=1$. As a result, $f(B_{i+1,l}) + f(B_{i+1,m}) + f(B_{i+1,r})=0$, a contradiction to Lemma 8.

(6) Similar to (5).

(7) Similar to (4). □

According to Lemmas 9 and 12, we can create a directed graph $H=(V_H,E_H)$, where $V_H=\{P_{(0,0)},P_{(0,1)},P_{(1,0)},Q_{(0,0,1)},Q_{(0,1,0)},Q_{(1,0,0)},Q_{(1,-1,1)}\}$ and $E_H=\{\overrightarrow{uv}\,|\,u,v\in V$ and u,v satisfy one of the conditions of Lemma 12$\}$. We add a *start* node in H such

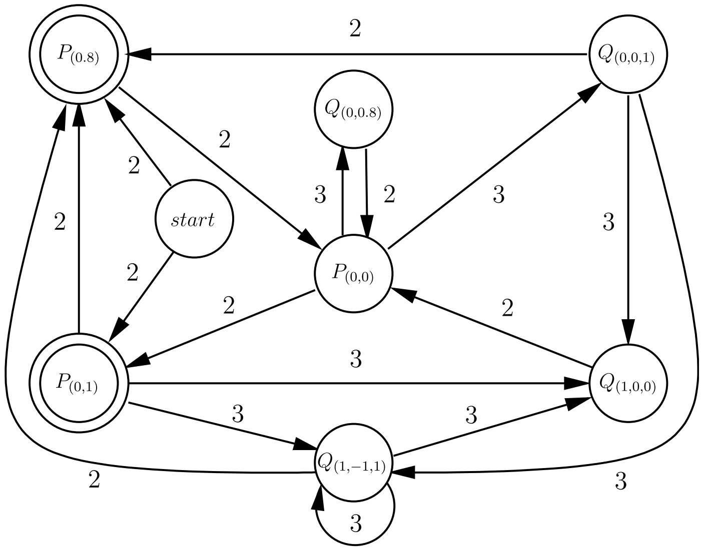


Fig. 4. The non-deterministic finite state automaton $\mathcal{M}$.

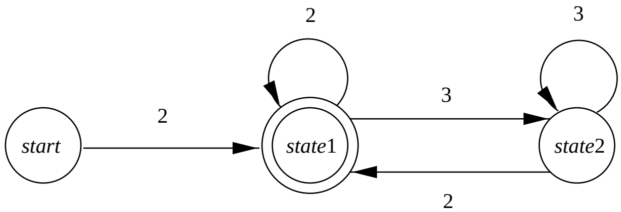


Fig. 5. The refined deterministic finite state automaton $\mathcal{M}''$.

that there are two edges from *start* node to $P_{(0,1)}$ and $P_{(1,0)}$. For each edge $\overrightarrow{uv}$, if $v \in \{P_{(0,0)}, P_{(0,1)}, P_{(1,0)}\}$, then we label $\overrightarrow{uv}$ with 2; otherwise, we label $\overrightarrow{uv}$ with 3. By letting $P_{(0,1)}$ and $P_{(1,0)}$ be two termination nodes, H becomes a non-deterministic finite state automaton and we denote it as $\mathcal{M}$. That is, $\mathcal{M}$ has a *start* node and two termination nodes $P_{(0,1)}$ and $P_{(1,0)}$, and each edge is labeled with a single symbol from $\Sigma = \{2, 3\}$ (see Fig. 4). In $\mathcal{M}$, each path p from *start* node to a termination node specifies a string $str(p)$ by concatenating the characters of Σ that label the edges of p. Clearly, $str(p)$ is a string accepted by $\mathcal{M}$. Furthermore, $\mathcal{M}$ can be reduced into a deterministic finite state automaton $\mathcal{M}''$ with the minimum states (see Fig. 5) using the following methods:

(1) Convert $\mathcal{M}$ into a deterministic finite state automaton $\mathcal{M}'$ (refer to [16, pp. 22–24]).
(2) Minimize the states of $\mathcal{M}'$ (refer to [16, pp. 65–71]).

(3) Simplify $\mathcal{M}'$ into $\mathcal{M}''$ by removing all non-start states of $\mathcal{M}'$ without input edge.

According to $\mathcal{M}''$, it is not hard to see that the accepted language of $\mathcal{M}''$ is $2+2(2+3)^*2$. Hence, we have the following lemma.

Lemma 13. *The accepted language of* $\mathcal{M}$ *is* $2+2(2+3)^*2$.

According to Lemmas 6 and 13, $\mathcal{L}$ is accepted by $\mathcal{M}$. That is, for any chain interval graph G, we can find a path $p=(start, n_1, n_2, \ldots, n_r)$ from *start* node to a termination node n_r in $\mathcal{M}$ such that $str(p)$ is equal to $bs(G)$. The existence of path p also implies that G affirmatively admits a simple *EMD* function $f: V \to \{-1, 0, 1\}$ which is defined as follows. Let $f(B_{i,l}) = a$ and $f(B_{i,r}) = b$ if $n_i = P_{(a,b)}$, and let $f(B_{i,l}) = a$, $f(B_{i,m}) = b$ and $f(B_{i,r}) = c$ if $n_i = Q_{(a,b,c)}$. Furthermore, for each block B_j of G, we first randomly choose one vertex u of B_j and let $f(u) = f(B_j)$. Then, for each vertex $v \in B_i \backslash \{u\}$, we let $f(v) = 0$. It is not hard to see that f is a simple *EMD* function of G. For example, considering the graph G as shown in Fig. 3, we have $p = (start, P_{(0,1)}, Q_{(1,-1,1)}, P_{(1,0)}, P_{(0,0)}, P_{(0,1)})$. Then, we can find a simple *EMD* function $f: V \to \{-1, 0, 1\}$ of G by letting $f(B_1) = 0, f(B_2) = 1, f(B_3) = -1, f(B_4) = 1$, $f(B_5) = 0$, $f(B_6) = 0, f(B_7) = 1$, $f(1) = f(2) = f(6) = f(7) = f(8) = f(9) = 0$, $f(3) = f(5) = f(10) = 1$ and $f(4) = -1$. Since a simple *EMD* function is an efficient minus dominating function, we have the following theorem.

Theorem 14. *Every chain interval graph has an efficient minus dominating function.*

The remaining work is to efficiently find a path p of G in $\mathcal{M}$ such that $str(p) = bs(G)$. We propose the following algorithm to find it. First of all, according to the clique structure $G = (C_1, C_2, \ldots, C_s)$ and $\mathcal{M}$, we construct a directed acyclic graph $DAG(G)$ with s layers as follows. For simplicity, we use V_i to denote the set of nodes in layer i and E_i to denote the set of the directed edges $\overrightarrow{uv}$ with $u \in V_i$ and $v \in V_{i+1}$.

(1) $V_1 = \{P_{(0,1)}, P_{(1,0)}\}$.
(2) Suppose that $V_i = \{u_1, u_2, \ldots, u_k\}$ for $1 \leqslant i < s-1$. Then, $E_i = \bigcup_{1 \leqslant j \leqslant k} \{\overrightarrow{u_j v} | \overrightarrow{u_j v}$ is an edge of $\mathcal{M}$ with label $bn(C_{i+1})\}$ and $V_{i+1} = \{v | \overrightarrow{uv} \in E_i\}$.
(3) $E_{s-1} = \{\overrightarrow{uv} | u \in V_{s-1},\ v \in \{P_{(0,1)}, P_{(1,0)}\}$ and $\overrightarrow{uv}$ is an edge of $\mathcal{M}\}$ and $V_s = \{v | \overrightarrow{uv} \in E_s\}$.

For example, considering the graph G of Fig. 3, the $DAG(G)$ is depicted in Fig. 6. After constructing $DAG(G)$, p can be easily found by starting a node in layer s and backtracking to node in layer 1. Since $s = O(|V|)$ and $|V_i| \leqslant 7$ for all i, $1 \leqslant i \leqslant s$, the construction of $DAG(G)$ and the finding of p can be done in $O(|V|)$ time. Hence, we have the following theorem.

Theorem 15. *The efficient minus domination problem can be solved in linear time on chain interval graphs.*

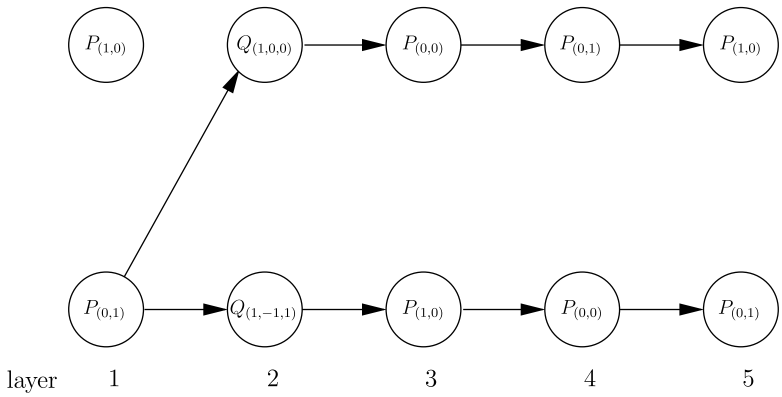


Fig. 6. The directed acyclic graph $DAG(G)$.

3.2. The efficient signed domination problem

In this subsection, we show that every chain interval graph with no vertex of odd degree has an efficient signed dominating function.

Remark 16. Any graph with a vertex of odd degree has no efficient signed dominating function.

By Remark 16, the graphs such as trees, cubic graphs and k-regular graphs, where k is odd, have no efficient signed dominating function. Therefore, the efficient signed domination problem is only considered on the graphs with no vertex of odd degree. In the following, we assume that G is a connected chain interval graph with no vertex of odd degree. We still make use of the forcing property on blocks to deal with the efficient signed domination problem. However, we consider the size of a block instead of the function value of a block. Note that for each vertex v of G, the degree of v is $|N(v)|$. For a vertex set W, we define $odd(W)=1$ if $|W|$ is odd; otherwise, $odd(W)=0$.

Remark 17. For every vertex v of G, $odd(N[v])=1$.

Let $B_1, B_2, \ldots, B_t$ be the block structure of G and $os(G)$ be the string $odd(B_1)odd(B_2)\cdots odd(B_t)$. A clique C_i of G is called an $S_{(a,b)}$ clique if $bn(C_i)=2$, $odd(B_{i,l})=a$ and $odd(B_{i,r})=b$, and an $L_{(a,b,c)}$ clique if $bn(C_i)=3$, $odd(B_{i,l})=a$, $odd(B_{i,m})=b$ and $odd(B_{i,r})=c$. By Remarks 4 and 17, we have the following lemma.

Lemma 18. *For any two consecutive cliques C_i and C_{i+1} of G, $odd(C_i \cup (C_{i+1} \setminus B_{i+1,l}))=1$.*

By Remarks 5 and 17, we have the following lemma.

Lemma 19. *Let* $bn(C_i)=3$. *Then*, $odd(B_{i,l}\cup B_{i,m}\cup B_{i,r})=1$.

Lemma 20. *If* $s\geqslant 2$, *then* C_1 *and* C_s *are either an* $S_{(0,1)}$ *or* $S_{(1,0)}$ *clique.*

Proof. By Remark 3, $bn(C_1)=bn(C_s)=2$. Let $v\in B_{1,l}$. By the block structure of G, we have $N[v]=B_{1,l}\cup B_{1,r}$ and $B_{1,l}\cap B_{1,r}=\emptyset$. Hence, according to Remark 17, we have either $odd(B_{1,l})=0$ and $odd(B_{1,r})=1$, or $odd(B_{1,l})=1$ and $odd(B_{1,r})=0$, which means that C_1 is either an $S_{(0,1)}$ or $S_{(1,0)}$ clique. Similarly, C_s is either an $S_{(0,1)}$ or $S_{(1,0)}$ clique. □

Lemma 21. *For each clique* C_i, $1<i<s$, *if* $bn(C_i)=3$, *then* C_i *is either an* $L_{(0,0,1)}$, $L_{(0,1,0)}, L_{(1,0,0)}$ *or* $L_{(1,1,1)}$ *clique.*

Proof. By Lemma 19, we have $odd(B_{i,l}\cup B_{i,m}\cup B_{i,r})=1$. Note that $B_{i,1}, B_{i,m}$ and $B_{i,r}$ is a partition of C_i. Hence, C_i is either an $L_{(0,0,1)}, L_{(0,1,0)}, L_{(1,0,0)}$ or $L_{(1,1,1)}$ clique. □

Lemma 22. *For each clique* C_i, $1<i<s$, *if* $bn(C_i)=2$, *then* C_i *is either an* $S_{(0,0)}, S_{(0,1)}$ *or* $S_{(1,0)}$ *clique.*

Proof. We only show that C_i cannot be an $S_{(1,1)}$ clique. Suppose that C_i is an $S_{(1,1)}$ clique. Clearly, $bn(C_{i+1})$ is either 2 or 3. In the case of $bn(C_{i+1})=3$, C_{i+1} must be either an $L_{(1,0,0)}$ or an $L_{(1,1,1)}$ clique according to Lemma 21. As a result, we have $odd(C_i\cup(C_{i+1}\backslash B_{i+1,l}))=0$, a contradiction to Lemma 18. In the case of $bn(C_{i+1})=2$, we have $odd(C_i\cup B_{i+1,r})=1$ by Lemma 18, which means that $odd(B_{i+1,r})=1$. That is, C_{i+1} is an $S_{(1,1)}$ clique. Continuing this way, we will find that C_s is an $S_{(1,1)}$ clique, which contradicts to Lemma 20. In other words, C_i is not an $S_{(1,1)}$ clique. Hence, C_i is either an $S_{(0,0)}, S_{(0,1)}$ or $S_{(1,0)}$ clique. □

Lemma 23. *Let* $1\leqslant i<s$.

(1) *If* C_i *is an* $S_{(0,0)}$ *clique, then* C_{i+1} *is either an* $S_{(0,1)}$, $L_{(0,0,1)}$ *or* $L_{(0,1,0)}$ *clique.*
(2) *If* C_i *is an* $S_{(0,1)}$ *clique, then* C_{i+1} *is either an* $S_{(1,0)}$, $L_{(1,0,0)}$ *or* $L_{(1,1,1)}$ *clique.*
(3) *If* C_i *is an* $S_{(1,0)}$ *clique, then* C_{i+1} *is an* $S_{(0,0)}$ *clique.*
(4) *If* C_i *is an* $L_{(0,0,1)}$ *clique, then* C_{i+1} *is either an* $S_{(1,0)}$, $L_{(1,0,0)}$ *or* $L_{(1,1,1)}$ *clique.*
(5) *If* C_i *is an* $L_{(0,1,0)}$ *clique, then* C_{i+1} *is an* $S_{(0,0)}$ *clique.*
(6) *If* C_i *is an* $L_{(1,0,0)}$ *clique, then* C_{i+1} *is an* $S_{(0,0)}$ *clique.*
(7) *If* C_i *is an* $L_{(1,1,1)}$ *clique, then* C_{i+1} *is either an* $S_{(1,0)}$, $L_{(1,0,0)}$ *or* $L_{(1,1,1)}$ *clique.*

Proof. In the following, we just show statements (1) and (3) because the others can be proved in a similar way.

(1) Let C_i be an $S_{(0,0)}$ clique. If $bn(C_{i+1})=2$, then by Lemma 18, we have $odd(C_i\cup B_{i+1,r})=1$ and hence $odd(B_{i+1,r})=1$. That is, C_{i+1} is an $S_{(0,1)}$ clique. If $bn(C_{i+1})=3$, then we have $odd(C_i\cup B_{i+1,m}\cup B_{i+1,r})=1$ and hence $odd(B_{i+1,m}\cup B_{i+1,r})=1$. That is, C_{i+1} is an $L_{(0,0,1)}$ or $L_{(0,1,0)}$ clique.

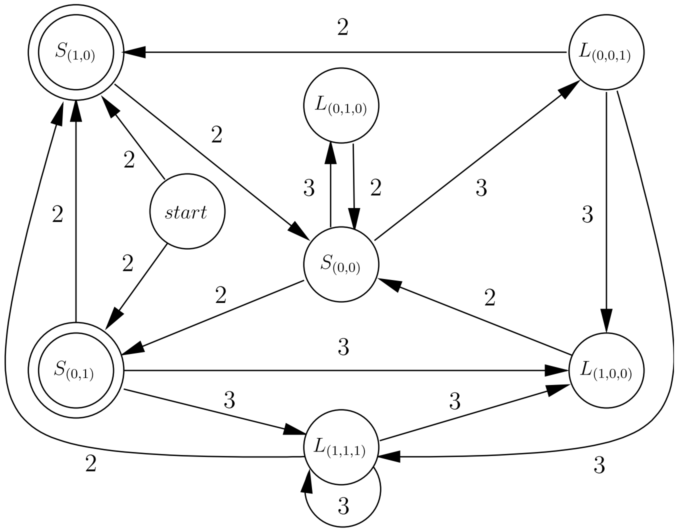


Fig. 7. The finite state automaton $\mathcal{N}$.

(3) Let C_i be an $S_{(1,0)}$ clique. If $bn(C_{i+1})=2$, then by Lemma 18, we have $odd(C_i \cup B_{i+1,r})=1$ and hence $odd(B_{i+1,r})=0$. That is, C_{i+1} is an $S_{(0,0)}$ clique. If $bn(C_{i+1})=3$, then we have $odd(C_i \cup B_{i+1,m} \cup B_{i+1,r})=1$ and hence $odd(B_{i+1,m} \cup B_{i+1,r})=0$. As a result, $odd(B_{i+1,l} \cup B_{i+1,m} \cup B_{i+1,r})=0$, a contradiction to Lemma 19. □

By Lemmas 20 and 23, we can create a non-deterministic finite state automaton $\mathcal{N}$ as shown in Fig. 7, where $S_{(0,1)}$ and $S_{(1,0)}$ are termination nodes. It is interesting that $\mathcal{N}$ is equivalent to $\mathcal{M}$. Moreover, each $S_{(a,b)}$ node of $\mathcal{N}$ corresponds to a $P_{(a,b)}$ node of $\mathcal{M}$ and each $L_{(a,b,c)}$ node corresponds to a $Q_{(a,b,c)}$ node except $L_{(1,1,1)}$ corresponds to $Q_{(1,-1,1)}$. As discussed in the previous subsection, there is a simple *EMD* function f for $G=(B_1,B_2,\ldots,B_t)$. It is not hard to see that for each block B_i of G, if $f(B_i)=0$, then the size of B_i must be even (i.e., $odd(B_i)=0$); otherwise, the size of B_i must be odd (i.e., $odd(B_i)=1$). This fact implies that f can be easily modified into an efficient signed dominating function of G as follows:

- If $f(B_i)=0$, then $|B_i|=2k$ and hence we assign $+1$ to k vertices in B_i and -1 to the remaining k vertices.
- If $f(B_i)=1$, then $|B_i|=2k+1$ and hence we assign $+1$ to $k+1$ vertices in B_i and -1 to the remaining k vertices.
- If $f(B_i)=-1$, then $|B_i|=2k+1$ and hence we assign -1 to $k+1$ vertices in B_i and $+1$ to the remaining k vertices.

In other words, every chain interval graph G with no vertex of odd degree has an efficient signed dominating function, which can be found just according to $os(G)$, instead of using the method similar to the one of finding the efficient minus dominating function of G. Because, from $\mathscr{N}$ and $\mathscr{M}$, we can find that the number of the consecutive ones in $os(G)$ is $2k+1$ (i.e., odd) and the function values of the corresponding blocks are $+1,-1,+1,-1,\ldots,+1$ (totally $2k+1$ values). Hence, we can easily determine the function value of each block in G just from $os(G)$ and then assign the value of each vertex in this block using the above method. Considering the graph G of Fig. 3 for an example, we have $os(G)=0111001$. Then, according to the method described above, we can find an efficient signed dominating function f of G such that $f(B_1)=0$, $f(B_2)=1$, $f(B_3)=-1$, $f(B_4)=f(B_5)=0$ and $f(B_6)=1$. Hence, we have $f(1)=1$, $f(2)=-1$, $f(3)=1$, $f(4)=-1$, $f(5)=1$, $f(6)=1$, $f(7)=-1$, $f(8)=1$, $f(9)=-1$ and $f(10)=1$.

Theorem 24. *For every chain interval graph G with no vertex of odd degree, G has an efficient signed dominating function f. Furthermore, f can be found in linear time.*

4. Trees

According to Remark 16, trees have no efficient signed dominating function since they contain leaves. In [2], Bange et al. proposed a linear-time algorithm for solving the efficient domination problem on trees. In the following, we will show the efficient minus domination problem is equivalent to the efficient domination problem on trees.

Lemma 25. *If a tree T has an efficient minus dominating function f, then $f(v)\geqslant 0$ for every vertex v of T.*

Proof. Suppose that f is an efficient minus dominating function of T and there is a vertex u in T with $f(u)=-1$. For convenience, we consider T as a rooted tree with root u. For each vertex v in T, let $\mathscr{P}(v)$ be the parent of v and $\mathscr{C}(v)$ be the set of all children of v. Clearly, $f(N[v])=f(v)+f(\mathscr{P}(v))+f(\mathscr{C}(v))=1$. Note that if v is a leaf of T, then $f(v)\geqslant 0$; otherwise we have $f(N[v])\leqslant 0$, a contradiction. Since u is the root and $f(u)=-1$, we have $f(\mathscr{C}(u))=2$, which means that there are at least two children of u, say x_1 and y_1, with $f(x_1)=f(y_1)=1$ and $f(\mathscr{C}(x_1))=f(\mathscr{C}(y_1))=1$. Then, $f(\mathscr{C}(x_1))=1$ implies that there is at least a child of x_1, say x_2, with $f(x_2)=1$ and $f(\mathscr{C}(x_2))=-1$; $f(\mathscr{C}(x_2))=-1$ implies that there is at least a child of x_2, say x_3, with $f(x_3)=-1$ and $f(\mathscr{C}(x_3))=1$; $f(C(x_3))=1$ implies that there is at least a child of x_3, say x_4, with $f(x_4)=f(x_1)$ and $f(\mathscr{C}(x_4))=f(\mathscr{C}(x_1))$. In this way, since T is finite, we will finally see that there is at least a leaf x_l of T whose $(f(x_l),f(\mathscr{C}(x_l)))$ is either $(1,1),(1,-1)$ or $(-1,1)$. Since x_l is a leaf and $\mathscr{C}(x_l)=\emptyset$, however, $(f(x_l),f(\mathscr{C}(x_l)))$ is either $(0,0)$ or $(1,0)$, which leads to a contradiction. Hence, $f(v)\geqslant 0$ for every vertex v of T. □

According to Lemma 25, we have the following theorem immediately.

Theorem 26. *The efficient minus domination problem is equivalent to the efficient domination problem on trees.*

5. Conclusions

When restricted to (proper) interval graphs, many NP-complete problems are solvable in polynomial or even linear time. However, we still do not know whether the efficient minus and signed domination problems are polynomially solvable because the forcing property on vertices and blocks cannot be applied. It is worth mentioning that chain interval graphs is the first and largest class of graphs which we find so far such that every chain interval graph has an efficient minus dominating function and every chain interval graph with no vertex of odd degree has an efficient signed dominating function. In graph theory, it is interesting to look for another class of graphs possessing such properties.

References

[1] D.W. Bange, A.E. Barkauskas, L.H. Host, P.J. Slater, Generalized domination and efficient domination in graphs, Discrete Math. 159 (1996) 1–11.
[2] D.W. Bange, A. Barkauskas, P.J. Slater, Efficient dominating sets in graphs, in: R.D. Ringeisen, F.S. Roberts (Eds.), Application of Discrete Mathematics, SIAM, Philadelphia, PA, 1988, pp. 189–199.
[3] A. Brandstädt, V. Le, J. Spinrad, Graph Classes—A Survey, SIAM Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia, 1999.
[4] G.J. Chang, C.P. Rangan, S.R. Coorg, Weighted independent perfect domination on cocomparability graphs, Discrete Appl. Math. 63 (1995) 215–222.
[5] M.S. Chang, Weighted domination of cocomparability graphs, Discrete Appl. Math. 80 (1997) 135–148.
[6] M.S. Chang, Y.C. Liu, Polynomial algorithm for the weighted perfect domination problems on chordal graphs and split graphs, Inform. Process. Lett. 48 (1993) 205–210.
[7] P. Damaschke, Minus domination in small-degree graphs, in: J. Hromkovic, O. Sýkora (Eds.), Proc. 24th Internat. Workshop on Graph-Theoretic Concepts in Computer Science I (W6'98), Lecture Notes in Computer Science, vol. 1517, Springer, Berlin 1998, pp. 17–25.
[8] J. Dunbar, W. Goddar, S. Hedetniemi, A. McRae, M.A. Henning, The algorithmic complexity of minus domination in graphs, Discrete Appl. Math. 68 (1996) 73–84.
[9] J. Dunbar, S.T. Hedetniemi, M.A. Henning, P.J. Slater, Signed domination in graphs, in: Y. Alari, A. Schwenk (Eds.), Proc. 7th Internat. Conf. on the Theory and Applications of Graphs, Wiley, New York, 1995, pp. 311–321.
[10] M.R. Fellows, M.N. Hoover, Perfect domination, Australas. J. Combin. 3 (1991) 141–150.
[11] M.R. Garey, D.S. Johnson, Computers and Intractability—A Guide to the Theory of NP-Completeness, Freeman, San Francisco, CA, 1979.
[12] P.C. Gilmore, A.J. Hoffman, A characterization of comparability graphs and of interval graphs, Canad. J. Math. 16 (1964) 539–548.
[13] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, New York, 1980.
[14] J.H. Hattingh, M.A. Henning, P.J. Slater, The algorithmic complexity of signed domination in graphs, Australas. J. Combin. 12 (1995) 101–112.
[15] T.W. Haynes, S.T. Hedetniemi, P.J. Slater, Fundamentals of Domination in Graphs, Marcel Dekker, New York, 1998.
[16] J.E. Hopcroft, J.D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, Reading, MA, 1979.
[17] Y.D. Liang, C.L. Lu, C.Y. Tang, Efficient domination on permutation graphs and trapezoid graphs, in: T. Jiang, D.T. Lee (Eds.), Proc. 3rd Annual Internat. Computing and Combinatorics Conf. (COCOON'97), Lecture Notes in Computer Science, vol. 1276, Springer, Berlin, 1997, pp. 232–241.
[18] C.L. Lu, On the study of weighted efficient domination, Ph.D. Thesis, Department of Computer Science, National Tsing Hua University, Taiwan, 1998.

[19] C.L. Lu, C.Y. Tang, Weighted efficient domination problem on some perfect graphs, Discrete Appl. Math. 117 (2002) 163–182.
[20] C.C. Yen, Algorithmic aspects of perfect domination, Ph.D. Thesis, Department of Computer Science, National Tsing Hua University, Taiwan, 1992.
[21] C.C. Yen, R.C.T. Lee, The weighted perfect domination problem and its variants, Discrete Appl. Math. 66 (1996) 147–160.