Discussiones Mathematicae
Graph Theory 42 (2022) 231–248
https://doi.org/10.7151/dmgt.2254

THE SEMITOTAL DOMINATION PROBLEM IN BLOCK GRAPHS

Michael A. Henning [1]

Department of Pure and Applied Mathematics
University of Johannesburg
Auckland Park, 2006 South Africa

e-mail: mahenning@uj.ac.za

Saikat Pal [2]

Department of Mathematics and Computing
Indian Institute of Technology (ISM), Dhanbad

e-mail: palsaikat67@gmail.com

and

D. Pradhan

Department of Mathematics and Computing
Indian Institute of Technology (ISM), Dhanbad

e-mail: dina@iitism.ac.in

Abstract

A set D of vertices in a graph G is a dominating set of G if every vertex outside D is adjacent in G to some vertex in D. A set D of vertices in G is a semitotal dominating set of G if D is a dominating set of G and every vertex in D is within distance 2 from another vertex of D. Given a graph G and a positive integer k, the semitotal domination problem is to decide whether G has a semitotal dominating set of cardinality at most k. The semitotal domination problem is known to be NP-complete for chordal graphs and bipartite graphs as shown in [M.A. Henning and A. Pandey, *Algorithmic aspects of semitotal domination in graphs*, Theoret. Comput. Sci. 766 (2019) 46–57]. In this paper, we present a linear time algorithm to compute a minimum semitotal dominating set in block graphs. On the other hand, we show that the semitotal domination problem remains NP-complete for undirected path graphs.

[1]Research supported in part by the University of Johannesburg.
[2]Corresponding author.

Keywords: domination, semitotal domination, block graphs, undirected path graphs, NP-complete.

2010 Mathematics Subject Classification: 05C69.

1. Introduction

A *dominating set* in a graph G is a set D of vertices of G such that every vertex in $V(G) \setminus D$ is adjacent to at least one vertex in D. The *domination number* of G, denoted by $\gamma(G)$, is the minimum cardinality of a dominating set of G. The concept of domination and its variations have been widely studied in theoretical, algorithmic and application aspects; a rough estimate says that it occurs in more than 6,000 papers to date. A thorough treatment of the fundamentals of domination theory in graphs can be found in the books [4, 5].

A *total dominating set*, abbreviated a TD-set, of a graph G with no isolated vertex is a set D of vertices of G such that every vertex in $V(G)$ is adjacent to at least one vertex in D. The *total domination number* of G, denoted by $\gamma_t(G)$, is the minimum cardinality of a TD-set of G. Total domination is now well studied in graph theory. The literature on the subject of total domination in graphs has been surveyed and detailed in the recent book [13].

A relaxed form of total domination called semitotal domination was introduced by Goddard, Henning and McPillan [3], and studied further in [6, 7, 8, 9, 10, 11, 12] and elsewhere. A set D of vertices in a graph G with no isolated vertices is a *semitotal dominating set*, abbreviated a semi-TD-set, of G if D is a dominating set of G and every vertex in D is within distance 2 of another vertex of D. The *semitotal domination number* of G, denoted by $\gamma_{t2}(G)$, is the minimum cardinality of a semi-TD-set of G. Since every TD-set is a semi-TD-set, and since every semi-TD-set is a dominating set, we have the following observation.

Observation 1 [3]. *For every isolate-free graph G, $\gamma(G) \le \gamma_{t2}(G) \le \gamma_t(G)$.*

As remarked in [3], by Observation 1 the semitotal domination number is squeezed between arguably the two most important domination parameters, namely the domination number and the total domination number. Goddard *et al.* [3] established tight upper bounds on the semitotal domination number of a connected graph in terms of its order. Henning [7] established tight upper bounds on the upper semitotal domination number of a regular graphs using edge weighting functions. Henning and Marcon [8] explored a relationship between the semitotal domination number and the matching number of a graph, and showed that the semitotal domination number of a connected graph is bounded above by the matching number plus one. Zhuang and Hao [15] established a lower bound on the semitotal domination number of trees and characterized the extremal trees. Semitotal domination in claw-free cubic graphs has been studied in [10].

Given a graph G and a positive integer k, the *semitotal domination* problem is to decide whether G has a semitotal dominating set of cardinality at most k. The semitotal domination problem is known to be NP-complete for general graphs [3]. Henning and Pandey [12] showed that the semitotal domination problem remains NP-complete for chordal bipartite graphs, planar graphs and split graphs. On the positive side, linear time algorithms exist to find a minimum semi-TD-set in trees [3, 11]. A polynomial time algorithm to compute a minimum cardinality semi-TD-set in interval graphs, a subclass of chordal graphs, is presented in [12].

In this paper, we design in Section 3 a linear time algorithm for computing a minimum semitotal dominating set in block graphs, a superclass of trees. On the other hand, we show in Section 4 that the semitotal domination problem remains NP-complete for undirected path graphs, a subclass of chordal graphs.

2. Terminology and Notation

For notation and graph theory terminology, we in general follow [13]. Specifically, let $G = (V, E)$ be a graph with vertex set $V = V(G)$ and edge set $E = E(G)$, and let v be a vertex in V. The *open neighborhood* of v is the set $N_G(v) = \{u \in V \mid uv \in E\}$ and the *closed neighborhood of* v is $N_G[v] = \{v\} \cup N_G(v)$. A vertex v is said to *dominate* a vertex u in G if $u \in N_G[v]$. The *open neighborhood* of a set S of vertices in G is the set $N_G(S) = \bigcup_{v \in S} N_G(v)$ and its *closed neighborhood* is the set $N_G[S] = N_G(S) \cup S$. The *degree* of a vertex v is $|N_G(v)|$ and is denoted by $d_G(v)$. For a set S of vertices in G, the subgraph induced by S in G is denoted by $G[S]$. Thus, the edge set of $G[S]$ consists of those edges of G with both ends in the set S. The set S is a *clique* of G, if $G[S]$ is a complete subgraph of G.

The *distance* between two vertices u and v in a connected graph G, denoted by $d_G(u, v)$, is the length of a shortest (u, v)-path in G. For a vertex v in G, the 2-*distance neighborhood* of v is the set $N_G^2(v) = \{u \mid 1 \le d_G(u, v) \le 2\}$ of all vertices at distance 1 or 2 from v in G, while the *closed* 2-*distance neighborhood* of v is $N_G^2[v] = N_G^2(v) \cup \{v\}$. A vertex in $N_G^2(v)$ is called a 2-*distance neighbor* of the vertex v in G.

A *rooted* tree is a tree T in which there is a designated vertex r named as *root*. For each vertex $v \neq r$ of T, the *parent* of v is the neighbor of v on the unique (r, v)-path, while a *child* of v is any other neighbor of v.

For a vertex v of G, the graph $G - v$ is the graph obtained from G by deleting v and deleting all edges of G incident with v. A vertex v is a *cut-vertex* of G if the number of components increases in $G - v$. A *block* of a graph G is a maximal connected subgraph of G has no cut-vertex of its own. Thus, a block is a maximal

2-connected subgraph of G. Any two blocks of a graph have at most one vertex in common, namely a cut-vertex. If a connected graph contains a single block, we call the graph itself a *block*. A *block graph* is a connected graph in which every block is a clique. A block containing exactly one cut-vertex is called an *end block*. A non-complete block graph has at least two end blocks.

We use the standard notation $[k] = \{1, 2, \ldots, k\}$. Let $G = (V, E)$ be a block graph, and let $\{B_1, B_2, \ldots B_r\}$ and $\{c_1, c_2, \ldots, c_s\}$ be the set of blocks and the set of cut-vertices of G, respectively. The *cut-tree* of G is the tree T_G defined by $V(T_G) = \{B_1, \ldots, B_r, c_1, \ldots, c_s\}$ and $E(T_G) = \{B_i c_j \mid c_j \in V(B_i), i \in [r], j \in [s]\}$. A block graph G and its associated cut-tree T_G is illustrated in Figure 1. The computation of blocks in a graph G and the construction of the cut-tree T_G can be done in $O(|V| + |E|)$ time by using *depth-first search* [1].

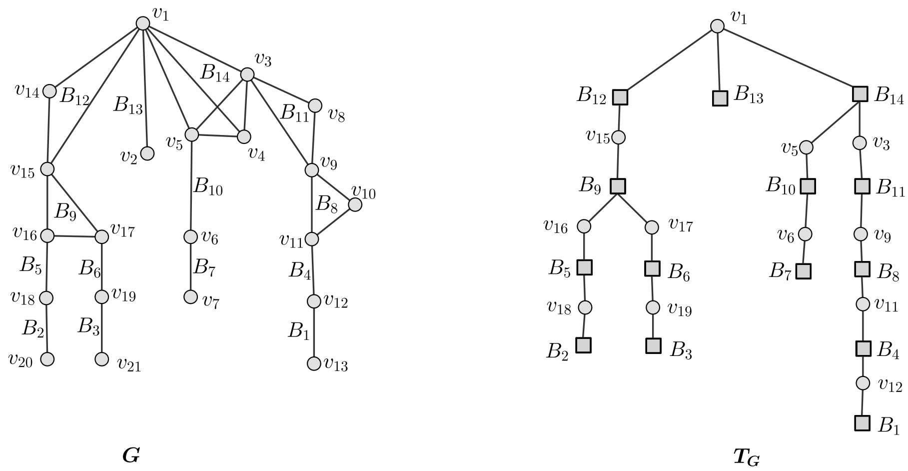

Figure 1. A block graph G and its corresponding cut-tree T_G.

3. Semitotal Domination in Block Graphs

In this section, we present a linear algorithm to compute a minimum semi-TD-set of a block graph G on at least two vertices. If G itself is a block, then the graph G is a complete graph. In this case, any two vertices in G form a semi-TD-set of G, implying that $\gamma_{t2}(G) = 2$. Hence it is only of interest for us to consider non-complete block graphs; that is, block graphs containing at least two blocks.

Let $G = (V, E)$ be a non-complete block graph. The algorithm we present to compute a minimum semi-TD-set in G runs in $O(|V| + |E|)$ time, and follows a certain ordering of the blocks. Let $\{B_1, B_2, \ldots, B_r\}$ and $\{c_1, c_2, \ldots, c_s\}$ be the set of blocks and the set of cut-vertices of G, respectively. Let T_G be the cut-tree associated with the graph G. Without loss of generality, we assume that

T_G is rooted at the cut-vertex c_s of G. Let $\sigma = (B_1, B_2, \ldots, B_r)$ be an ordering of blocks of G, where $\sigma^{-1} = (B_r, B_{r-1}, \ldots, B_1)$ is an ordering of blocks of G obtained by applying a breadth-first search starting at the root c_s of T_G. We call such an ordering of blocks of G as a RBFS-BLOCK-ORDERING of the blocks of G. For every $i \in [r]$, we define $F(B_i)$ as the parent of the block B_i in T_G. Further for every $i \in [r]$, we define

$$G_i = G\left[\bigcup_{\ell=i}^{r} V(B_\ell)\right].$$

We note that for every $i \in [r-1]$, the block B_i is an end block in the graph G_i with $F(B_i)$ as the unique cut-vertex in G_i that belongs to the block B_i. Since the G_r is the block B_r, we treat any vertex of the block B_r as $F(B_r)$. For the sake of simplicity, we denote the vertex $F(B_i)$ simply by F_i for $i \in [r]$. The following observation follows immediately from the fact that any two blocks of G have at most one vertex in common, namely a cut-vertex.

Observation 2. *For every $i \in [r-1]$ and every $k > i$, we have $V(B_i) \cap V(B_k) \subseteq \{F_i\}$.*

Before formally presenting our algorithm MSTDS-BLOCK(G), we discuss the main ideas of the algorithm. The algorithm constructs a set D which upon termination of the algorithm is a semi-TD-set of the non-complete block graph G. We assign to each vertex v of G a label $L(v) = (L_1(v), L_2(v))$ which we call its *L-label*. We call the labels $L_1(v)$ and $L_2(v)$ the L_1-label and L_2-label of v, respectively. The label $L_1(v)$ is used to determine whether the vertex v is already dominated or has yet to be dominated. Initially, $L_1(v) = L_2(v) = 0$ for every vertex v of G. As the algorithm progresses, the label of the vertex v changes. If the vertex v is not dominated by the current set D, then the label $L_1(v) = 0$ is unchanged; otherwise, $L_1(v) = 1$. The label $L_2(v)$ is used to determine whether the vertex v belongs to the current set D or not. If the vertex v does not belong to the current set D, then the label $L_2(v) = 0$ is unchanged. If the vertex v belongs to the current set D but has no 2-distance neighbor in D, then $L_2(v) = 1$. If the vertex v belongs to the current set D and has a 2-distance neighbor in D, then $L_2(v) = 2$.

At the i-th iteration, the algorithm systematically considers the vertices of the block B_i with respect to the RBFS-BLOCK-ORDERING $\sigma = (B_1, B_2, \ldots, B_r)$ of G and takes some action (either the algorithm selects new vertices or updates some of the vertices of the graph) based on the values of L_1 and L_2 assigned to the vertices that belong to $V(B_i) \setminus \{F_i\}$. If a vertex u is selected by the algorithm and added to the set D, then $L_1(u)$ is updated to 1, $L_2(u)$ is updated to 1 or 2, and $L(y)$ is made $(1, 0)$ for every neighbor y of u in G such that $L(y) = (0, 0)$. Upon termination of the algorithm, the set D consists precisely of the $(1,2)$-labeled vertices and forms a semi-TD-set of G. We now formally describe our algorithm to construct a semi-TD-set in a non-complete block graph.

Algorithm 1: MSTDS-BLOCK(G)

```
Input: A non-complete connected block graph $G = (V, E)$;
Output: A semi-TD-set $D$ of $G$;
Initialize $D = \emptyset$;
Initialize $L(u) = (0,0)$ for each vertex $u \in V$;
Compute a RBFS-BLOCK-ORDERING $\sigma = (B_1, B_2, \ldots, B_r)$ of the blocks of $G$;
$i = 1$;
while ($i < r$) do
    Let $F_i$ be the unique cut-vertex of $G_i$ present in $B_i$ and $\mathcal{C}(B_i) = V(B_i) \setminus \{F_i\}$;
    while ($\mathcal{C}(B_i) \neq \emptyset$) do
        Choose a vertex $v \in \mathcal{C}(B_i)$;
        if ($L(v) = (0,0)$) then
            if (there exists a vertex $u \in N_G[F_i]$ with $L_1(u) = 1$) then        /* Case 1 */
                $L(F_i) = (1,2)$ and $L_2(x) = 2$ for every vertex $x \in N_G(F_i)$ such that
                  $L_2(x) = 1$;
            else                                                                      /* Case 2 */
                $L(F_i) = (1,1)$;
            $L_1(x) = 1$ for every vertex $x \in N_G(F_i)$;
        else if ($L(v) = (1,0)$) then
            Let $A(v) = \{y \in N_G(v) \mid L_2(y) \neq 0\}$;
            if ($|A(v)| > 1$) then                                                    /* Case 3 */
                $L_2(x) = 2$ for every $x \in N_G(v)$ such that $L_2(x) = 1$;
            else if ($A(v) = \{u\}$ such that $L_2(u) = 1$ and $u \notin V(B_i)$) then   /* Case 4 */
                $L(F_i) = (1,2)$ and $L_1(x) = 1$ for every vertex $x \in N_G(F_i)$;
                $L_2(x) = 2$ for every vertex $x \in N_G(F_i) \cup \{u\}$ such that $L_2(x) = 1$;
        $\mathcal{C}(B_i) = \mathcal{C}(B_i) \setminus \{v\}$;
    $i = i + 1$;
$\mathcal{C}(B_r) = V(B_r)$;
while ($\mathcal{C}(B_r) \neq \emptyset$) do
    Choose a vertex $v \in \mathcal{C}(B_r)$;
    if ($L(v) = (0,0)$) then                                                          /* Case 5 */
        $L(c_j) = (1,2)$ for some cut-vertex $c_j$ of $G$ such that $c_j \in V(B_r)$;
        $L_1(x) = 1$ for every $x \in N_G(c_j)$;
        $L_2(x) = 2$ for every vertex $x \in N_G(c_j)$ such that $L_2(x) = 1$;
    else if ($L(u) = (1,1)$ for some $u \in N_G(v)$, where $v \in V(B_r)$) then
        Let $B(v) = \{y \in N_G(v) \mid L_2(y) \neq 0\}$;
        if ($|B(v)| > 1$) then                                                        /* Case 6 */
            $L_2(x) = 2$ for every $x \in N_G(v)$ such that $L_2(x) = 1$;
        else                                                                          /* Case 7 */
            $L(w) = (1,2)$ for some $w \in V(B_r) \setminus \{u\}$ and $L(u) = (1,2)$;
            $L_1(x) = 1$ for every vertex $x \in N_G(w)$;
            $L_2(x) = 2$ for every vertex $x \in N_G(w)$ such that $L_2(x) = 1$;
    $\mathcal{C}(B_r) = \mathcal{C}(B_r) \setminus \{v\}$;
return $D = \{u \in V \mid L(u) = (1,2)\}$;
```

In Table 1, we illustrate the different iterations of the algorithm MSTDS-BLOCK(G) on the graph G shown in Figure 1, where we only show the iterations of the algorithm in which some update has been done. Moreover, in the column "*Considered vertex* $v \in V(B_i)$ *with* $L(v)$" of Table 1, we have only shown those vertices of the block for which some update has been done. Upon termination of the algorithm, the resulting set $D = \{v_1, v_6, v_9, v_{12}, v_{15}, v_{18}, v_{19}\}$ a minimum semi-TD-set of the graph G shown in Figure 1.

Iteration i	**Considered block**	**Considered vertex** $v \in V(B_i)$ **with** $L(v)$	$\boldsymbol{F_i}$	$\boldsymbol{A(v)}$ **or** $\boldsymbol{B(v)}$	**Applied Case**	**Update**
1	B_1	$L(v_{13}) = (0,0)$	v_{12}	Not computed	**Case 2**	$L(v_{12}) = (1,1)$ $L(v_{11}) = (1,0)$ $L(v_{13}) = (1,0)$
2	B_2	$L(v_{20}) = (0,0)$	v_{18}	Not computed	**Case 2**	$L(v_{18}) = (1,1)$ $L(v_{16}) = (1,0)$ $L(v_{20}) = (1,0)$
3	B_3	$L(v_{21}) = (0,0)$	v_{19}	Not computed	**Case 2**	$L(v_{19}) = (1,1)$ $L(v_{21}) = (1,0)$ $L(v_{17}) = (1,0)$
7	B_7	$L(v_7) = (0,0)$	v_6	Not computed	**Case 2**	$L(v_6) = (1,1)$ $L(v_7) = (1,0)$ $L(v_5) = (1,0)$
8	B_8	(i) $L(v_{10}) = (0,0)$ (ii) $L(v_{11}) = (1,0)$	v_9	(i) Not computed (ii)$A(v) = \{v_9, v_{12}\}$	(i) **Case 1** (ii) **Case 3**	(i) $L(v_9) = (1,2)$ $L(v_{10}) = (1,0)$ $L(v_8) = (1,0)$ $L(v_3) = (1,0)$ (ii) $L(v_{12}) = (1,2)$
9	B_9	(i) $L(v_{16}) = (1,0)$ (ii) $L(v_{17}) = (1,0)$	v_{15}	(i) $A(v) = \{v_{18}\}$; $v_{18} \notin V(B_9)$ (ii) $A(v) = \{v_{15}, v_{19}\}$	(i) **Case 4** (ii) **Case 3**	(i) $L(v_{15}) = (1,2)$ $L(v_1) = (1,0)$ $L(v_{14}) = (1,0)$ $L(v_{18}) = (1,2)$ (ii) $L(v_{19}) = (1,2)$
13	B_{13}	$L(v_2) = (0,0)$	v_1	Not computed	**Case 1**	$L(v_1) = (1,2)$ $L(v_4) = (1,0)$ $L(v_2) = (1,0)$
14	B_{14}	$L(v_5) = (1,0)$	v_1	$B(v) = \{v_1, v_6\}$	**Case 6**	$L(v_6) = (1,2)$

Table 1. Illustration of the algorithm on the graph G shown in Figure 1.

Recall that in the i-th iteration of the algorithm MSTDS-BLOCK(G), the labels of all vertices in B_i are systematically considered. Furthermore, at the start of the i-th iteration, the labels $L(v)$ of all vertices v in B_j where $j < i$ are $(1,0)$, $(1,1)$ or $(1,2)$. We state this formally as follows.

Observation 3. *At the beginning of the i-th iteration of the algorithm* MSTDS-BLOCK*(G) where $i \geq 2$, we have $L(v) \in \{(1,0), (1,1), (1,2)\}$ for all $v \in V(B_j) \setminus \{F_j\}$ and $j \in [i-1]$.*

Let B_i be the block considered at the i-th iteration. If $L(v) = (1,0)$ for some $v \in V(B_i) \setminus \{F_i\}$, then the algorithm updates the L-labels of the neighbors of v. In particular, upon completion of the i-th iteration, there is no neighbor $u \in N_G(v) \setminus V(B_i)$ of v such that $L_2(u) = 1$. We state this observation formally as follows.

Observation 4. *Let B_i be the block considered at the i-th iteration and let $L(y) = (1, 0)$ for all $y \in V(B_i) \setminus \{F_i\}$. If $v \in V(B_i) \setminus \{F_i\}$ and there exists a vertex $u \in N_G(v) \setminus \{F_i\}$ with $L_2(u) \neq 0$, then $L(u) = (1, 2)$ upon completion of the i-th iteration of the algorithm.*

We note that the algorithm MSTDS-Block(G) has r iterations where r is the number of blocks in G. For $i \in [r] \cup \{0\}$, let D_i denote the set $\{u \in V(G) \mid L_2(u) \neq 0\}$ after the i-th iteration of the algorithm MSTDS-Block(G). We first prove that the set D_r is a semi-TD-set of G.

Lemma 5. *The set D_r is a semi-TD-set of G.*

Proof. Upon completion of the i-th iteration of the algorithm MSTDS-Block(G), by Observation 3, $L_1(x) = 1$ for all $x \in V(B_i) \setminus \{F_i\}$, where $i \in [r]$. This implies that D_r is a dominating set of G. To prove that D_r is a semi-TD-set of G, we show that for every $v \in D_r$, there exists a vertex $q \in D_r \setminus \{v\}$ such that $d_G(v, q) \leq 2$. Let $v \in D_r$ be arbitrary. Since G is a block graph, $v \in V(B_i)$ for some $i \in [r]$. We consider two cases.

Case 1. $i < r$. We first prove that if $v \in V(B_i) \setminus \{F_i\}$, then there exists a vertex $q \in D_r \setminus \{v\}$ such that $d_G(v, q) \leq 2$. Let $v \in V(B_i) \setminus \{F_i\}$. Since $i < r$, there exists a block B_j with $j > i$ such that $F_i \in V(B_j)$. Since $v \in V(B_i) \setminus \{F_i\}$, the vertex $v \in N_G(F_i)$. If $F_i \in D_r$, then taking $q = F_i$ the desired result holds. Hence we may assume that $F_i \notin D_r$. If $z \in D_r$ for some $z \in N_G(F_i)$, then $d_G(v, z) = d_G(v, F_i) + d_G(F_i, z) = 2$ and the desired result follows. Hence we may further assume that $z \notin D_r$ for every $z \in N_G(F_i)$. Thus, the set $\{y \in N_G(F_i) \mid y \in D_r\} = \{v\}$.

If $j = r$, then $B(F_i) = \{v\}$, where $B(u) = \{y \in N_G(u) \mid L_2(y) \neq 0\}$. In this case, the algorithm selects a vertex $w \in V(B_r) \setminus \{v\}$ (see Line 36 of the algorithm) at the r-th iteration. Notice that $d_G(v, w) \leq 2$.

If $j < r$, then $F_i \in V(B_j) \setminus \{F_j\}$ and $F_i \in N_G(v)$. Recall that $z \notin D_r$ for every $z \in N_G(F_i)$. In this case, $A(F_i) = \{v\}$, where $A(u) = \{y \in N_G(u) \mid L_2(y) \neq 0\}$. This implies that $A(F_i) = \{v\}$ at the beginning of the j-th iteration of the algorithm noting that $D_j \subseteq D_r$. In this case since $j < r$, the algorithm selects F_j (see Line 20 of the algorithm) at the j-th iteration. We note that $d_G(F_j, v) \leq 2$. In all the above cases, we have shown that if $v \in V(B_i) \setminus \{F_i\}$, then there exists a vertex $q \in D_r \setminus \{v\}$ such that $d_G(v, q) \leq 2$.

Now let $v = F_i$. Since $i < r$, we note that $v \in V(B_j)$ where $j > i$. If $j = r$, then the algorithm selects a vertex $w \in V(B_r) \setminus \{v\}$ (see Line 36 of the algorithm) at the r-th iteration. Since $d_G(v, w) \leq 2$, the desired result follows. If $j < r$, then

$v = F_i \in V(B_j) \setminus \{F_j\}$ where $j > i$. Thus by our earlier observations, there exists a vertex $q \in D_r \setminus \{v\}$ such that $d_G(v, q) \leq 2$. Therefore, D_r is a semi-TD-set of G.

Case 2. $i = r$. Suppose that there does not exist a vertex $q \in D_r \setminus \{v\}$ such that $d_G(v, q) \leq 2$. Since the algorithm does not select any vertex with L_2-label 1 at the r-th iteration, $v \in D_r$ implies that $v \in D_{r-1}$. Since G is a connected graph, $|V(B_r)| \geq 2$. Moreover, since there is no vertex $q \in D_r \setminus \{v\}$ such that $d_G(v, q) \leq 2$, at the beginning of the r-th iteration, we note that $L_2(v) = 1$. Thus in this case the algorithm selects a vertex $w \in V(B_r) \setminus \{v\}$ (see Line 36 of the algorithm) such that $d_G(v, w) \leq 2$. This is a contradiction to the fact that there does not exist a vertex $q \in D_r \setminus \{v\}$ such that $d_G(v, q) \leq 2$. Therefore there exists a vertex $q \in D_r \setminus \{v\}$ such that $d_G(v, q) \leq 2$, implying that D_r is a semi-TD-set of G. This completes the proof of Lemma 5. ■

We are now in a position to prove the following theorem.

Theorem 6. *The set D_r is a minimum semi-TD-set of G.*

Proof. Recall that for $i \in [r] \cup \{0\}$, the set D_i is the set $\{u \in V(G) \mid L_2(u) \in \{1, 2\}\}$ after the i-th iteration of the algorithm MSTDS-Block(G). By Lemma 5, the set D_r is a semi-TD-set of G. We prove next that D_r is a minimum semi-TD-set of G. For this purpose, we prove by induction on $i \geq 0$ that the set D_i is contained in some minimum semi-TD-set of G. If $i = 0$, then $D_0 = \emptyset$ and hence the set D_i is contained in every minimum semi-TD-set of G. This establishes the base case. Assume that $i \geq 1$ and that the set D_{i-1} is contained in some minimum semi-TD-set D' of G. We now show that D_i is contained in some minimum semi-TD-set of G. Recall that by our earlier assumptions, the graph G is a non-complete block graph. We proceed further with a series of claims. In each claim, we construct a minimum semi-TD-set of G containing D_i from the minimum semi-TD-set D' of G.

Claim 7. *If $i < r$ and $L(v) = (0, 0)$ for some vertex $v \in V(B_i) \setminus \{F_i\}$, then there is a minimum semi-TD-set of G containing $D_i = D_{i-1} \cup \{F_i\}$.*

Proof. By our induction hypothesis, the set D_{i-1} is contained in some minimum semi-TD-set D' of G. If $F_i \in D'$, then we are done. So we may assume that $F_i \notin D'$. Let u be a vertex in D' that dominates the vertex v. Since D' is a semi-TD-set of G, there is a vertex $u' \in D'$ such that $d_G(u, u') \in \{1, 2\}$. Since $L(v) = (0, 0)$, we note that $u \notin D_{i-1}$. If $u \in V(B_k) \setminus \{F_k\}$ where $k > i$, then by Observation 2, the vertex $u = F_i$ noting that $uv \in E(G)$. This is a contradiction since $F_i \notin D'$. Hence, $u \in V(B_k) \setminus \{F_k\}$ where $k \leq i$.

By Observation 3, $L(x) \in \{(1, 0), (1, 1), (1, 2)\}$ for every vertex $x \in V(B_j) \setminus \{F_j\}$ and all $j \in [i-1]$. Since $uv \in E(G)$, the vertex $v \in V(B_k)$. If $k < i$, then by

Observation 2, the vertex v is the vertex F_k. Notice that $\{z \in N_G[u] \mid L_1(z) = 0\} \subseteq N_G[F_i] \cup N_G[D_{i-1}]$, i.e., all the undominated vertices of $N_G[u]$ are dominated by $D_{i-1} \cup \{F_i\}$. Let $N_2(D', u) = \{x \mid x \in D' \cap N_G^2(u)\}$. If $d_G(F_i, x) \leq 2$ for every $x \in N_2(D', u)$, then $D'' = (D' \setminus \{u\}) \cup \{F_i\}$ is a minimum semi-TD-set of G containing $D_{i-1} \cup \{F_i\}$, as desired. Hence, we may assume that $d_G(F_i, x) > 2$ for some vertex $x \in N_2(D', u)$, for otherwise the desired result follows.

Let $p \in N_2(D', u)$ be an arbitrary vertex such that $d_G(F_i, p) > 2$. Thus, $p \in V(B_q)$ for some $q < i$ and $F_q \notin V(B_i)$. By Observation 4, either $L_2(p) = 0$ (hence $p \notin D_{i-1}$) or $L(p) = (1, 2)$. Let $S = \{x \in N_2(D', u) \mid d_G(F_i, x) > 2$ and $L_2(x) = 0\}$ and $S' = \{x \in S \mid N_G^2(x) \setminus \{u\} = \emptyset\}$. Notice that each element of S does not belong to D_{i-1} and belongs to the blocks that appear before i. Moreover, $N_G[S'] \subseteq N_G[D_{i-1}] \cup N_G[F_i]$. If $|S'| \geq 2$, then $(D' \setminus S') \cup \{F_i\}$ is a semi-TD-set of G of cardinality less than $|D'|$, contradicting the minimality of D'. Hence, $|S'| \leq 1$. If $|S'| = 1$, then $(D' \setminus S') \cup \{F_i\}$ is a minimum semi-TD-set of G containing $D_{i-1} \cup \{F_i\}$, as desired. Hence we may assume that $S' = \emptyset$.

If there is a vertex $q \in N_G[F_i]$ such that $L_1(q) = 1$, then $q \in D_{i-1}$ or $q' \in D_{i-1}$ where $qq' \in E(G)$. In this case, $(D' \setminus \{u\}) \cup \{F_i\}$ is a minimum semi-TD-set of G containing $D_{i-1} \cup \{F_i\}$ since $d_G(F_i, q) \leq 2$ or $d_G(F_i, q') \leq 2$. Hence, we may assume that $L_1(q) = 0$ for every vertex $q \in N_G[F_i]$, for otherwise the desired result follows. We now let $b \in N_G[F_i]$, and let b' be a vertex in D' that dominates the vertex b. Since $i < r$, we note that vertices b and b' exists. Further since $F_i \notin D'$, we note that $b' \neq F_i$. Since $L_1(q) = 0$ for all $q \in N_G[F_i]$, the vertex $b' \notin D_{i-1}$. Thus since $d_G(F_i, b') \leq 2$, the set $(D' \setminus \{u\}) \cup \{F_i\}$ is a minimum semi-TD-set of G containing $D_{i-1} \cup \{F_i\}$. This completes the proof of Claim 7. □

Recall that for each vertex $v \in V(B_i) \setminus \{F_i\}$ with $L(v) = (1, 0)$, the set $A(v) = \{y \in N_G(v) \mid L_2(y) \in \{1, 2\}\}$. If $|A(v)| \geq 2$, then for every $x \in A(v)$, there exists a vertex $y \in A(v)$ different from x such that $d_G(x, y) \leq 2$. So for every neighbor of v with L_2-label 1 (if such a neighbor of v exists), there is another neighbor of v with L_2-label 1 or 2. The following claim shows that if $A(v) = \{u\}$, $L_2(u) = 1$, and $u \notin V(B_i)$, then we can find a neighbor of v within distance 2 from u. Recall that D' is a minimum semi-TD-set of G and $D_{i-1} \subseteq D'$.

Claim 8. *Suppose that $i < r$ and $L(v) = (1, 0)$ for some vertex $v \in V(B_i) \setminus \{F_i\}$. If $A(v) = \{u\}$, where $L_2(u) = 1$ and $u \notin V(B_i)$, then there is a minimum semi-TD-set of G containing $D_{i-1} \cup \{F_i\}$.*

Proof. If $F_i \in D'$, then we are done. So we may assume that $F_i \notin D'$. By the choice of u and v, we note that $u \in V(B_k) \setminus \{F_k\}$ where $k < i$ as $u \notin V(B_i)$. Since $L(v) = (1, 0)$ and $L_2(u) = 1$, we have $u \in D_{i-1}$. Since D' is a semi-TD-set of G, there is a vertex $u' \in D'$ such that $d_G(u, u') \leq 2$. The fact that $L_2(u) = 1$ implies that $u' \notin D_{i-1}$. Let $u' \in V(B_\ell) \setminus \{F_\ell\}$ for some integer $\ell \geq 1$. If $\ell > i$, then since

$u \notin V(B_i)$ and $d_G(u, u') \leq 2$, Observation 2 implies that $u' = F_i$, contradicting the fact that $F_i \notin D'$. Hence, $\ell \leq i$.

We note that $\{z \in N_G[u'] \mid L_1(z) = 0\} \subseteq N_G[F_i] \cup N_G[D_{i-1}]$, i.e., all the undominated vertices of $N_G[u']$ are dominated by $D_{i-1} \cup \{F_i\}$. Let $N_2(D', u') = \{x \mid x \in D' \cap N_G^2(u')\}$. If $d_G(F_i, x) \leq 2$ or $d_G(u, x) \leq 2$ for every $x \in N_2(D', u')$, then $D'' = (D' \setminus \{u'\}) \cup \{F_i\}$ is a minimum semi-TD-set of G containing $D_{i-1} \cup \{F_i\}$. Hence, we may assume that $d_G(F_i, x) > 2$ and $d_G(u, x) > 2$ for some vertex $x \in N_2(D', u)$, for otherwise the desired result follows.

Let $p \in N_2(D', u)$ be an arbitrary vertex such that $d_G(F_i, p) > 2$ and $d_G(u, p) > 2$. Thus, $p \in V(B_q)$ for some $q < i$ and $F_q \notin V(B_i)$. By Observation 4, either $L_2(p) = 0$ (hence $p \notin D_{i-1}$) or $L(p) = (1, 2)$. Let $S = \{x \in N_2(D', u') \mid d_G(F_i, x) > 2,\ d_G(u, x) > 2 \text{ and } L_2(x) = 0\}$ and $S' = \{x \in S \mid N_G^2(x) \setminus \{u'\} = \emptyset\}$. Notice that each element of S' does not belong to D_{i-1} and belongs to the blocks that appear before i. Moreover, $N_G[S'] \subseteq N_G[D_{i-1}] \cup N_G[F_i]$. If $|S'| \geq 2$, then $(D' \setminus S') \cup \{F_i\}$ is a semi-TD-set of G of cardinality less than $|D'|$, contradicting the minimality of D'. Hence, $|S'| \leq 1$. If $|S'| = 1$, then $(D' \setminus S') \cup \{F_i\}$ is a minimum semi-TD-set of G containing $D_{i-1} \cup \{F_i\}$, as desired. Hence we may assume that $S' = \emptyset$. Since $d_G(u, F_i) \leq 2$, the set $(D' \setminus \{u'\}) \cup \{F_i\}$ is a minimum semi-TD-set of G containing $D_{i-1} \cup \{F_i\}$. This completes the proof of Claim 8. □

Claim 9. *If $i = r$ and $L(v) = (0, 0)$ for some vertex $v \in V(B_r)$, then there is a minimum semi-TD-set of G containing $D_{r-1} \cup \{c_j\}$, where $c_j \in V(B_r)$ is a cut-vertex of G.*

Proof. We once again consider the minimum semi-TD-set D' of G. Recall that $D_{r-1} \subseteq D'$. If $c_j \in D'$, then we are done. Hence we may assume that $c_j \notin D'$. Let u be a vertex in D' that dominates the vertex v. Since D' is a semi-TD-set of G, there is a vertex $u' \in D'$ such that $d_G(u, u') \leq 2$. Since $L(v) = (0, 0)$, we note that $u \notin D_{r-1}$. Further, we note that $L_2(x) = 0$ for all $x \in V(B_r)$. Moreover, since $c_j \in V(B_r)$ is an arbitrary cut-vertex of G, if $u \in V(B_r)$, then the vertex u is not a cut-vertex of G.

By Observation 3, $L_1(x) = 1$ for all $x \in V(B_j) \setminus \{F_j\}$, where $j \in [r-1]$. This implies that every vertex of $V(G) \setminus V(B_r)$ is dominated by D_{r-1}. We note that $\{z \in N_G[u] \cap V(B_r) \mid L_1(z) = 0\} \subseteq N_G[c_j] \cup N_G[D_{r-1}]$, i.e., the undominated vertices of $N_G[u]$ present in $V(B_r)$ are dominated by $D_{r-1} \cup \{c_j\}$. If $u \in V(B_r)$, then $(D' \setminus \{u\}) \cup \{c_j\}$ is a minimum semi-TD-set of G since u is not a cut-vertex of G and $d_G(x, c_j) \leq 2$ for every x such that $d_G(x, u) \leq 2$. Hence we may assume that $u \notin V(B_r)$, for otherwise the desired result follows.

Let $N_2(D', u) = \{x \mid x \in D' \cap N_G^2(u)\}$. If $d_G(c_j, x) \leq 2$ for every $x \in N_2(D', u)$, then $D'' = (D' \setminus \{u\}) \cup \{c_j\}$ is a minimum semi-TD-set of G containing $D_{r-1} \cup \{c_j\}$. Hence, we may assume that $d_G(c_j, p) > 2$ for some vertex $p \in N_2(D', u)$, for otherwise the desired result follows. Thus, $p \in V(B_q)$ for some

$q < r$ and $F_q \notin V(B_r)$. By Observation 4, either $L_2(p) = 0$ (hence $p \notin D_{r-1}$) or $L(p) = (1,2)$.

Let $S = \{x \in N_2(D', u) \mid d_G(c_j, x) > 2$ and $L_2(x) = 0\}$ and $S' = \{x \in S \mid N_G^2(x) \setminus \{u\} = \emptyset\}$. We note that each element of S' does not belong to D_{r-1} and belongs to the blocks that appear before r. Moreover, $N_G[S'] \subseteq N_G[D_{r-1}] \cup N_G[c_j]$. If $|S'| \geq 2$, then $(D' \setminus S') \cup \{c_j\}$ is a semi-TD-set of G of cardinality less than $|D'|$, contradicting the minimality of D'. Hence, $|S'| \leq 1$. If $|S'| = 1$, then $(D' \setminus S') \cup \{c_j\}$ is a minimum semi-TD-set of G containing $D_{r-1} \cup \{c_j\}$, as desired. Hence we may assume that $S' = \emptyset$. Since c_j is a cut-vertex of G and G is not complete, there must be a block B_k, where $k < r$, of G such that $V(B_k) \cap V(B_r) = \{c_j\}$. By Observation 3, $L_1(y) = 1$ for all $y \in V(B_k) \setminus \{F_k\}$. This implies that there is a vertex $y' \in N_G[y]$ such that $y' \in D_{r-1}$. We note that $d_G(c_j, y') \leq 2$, implying that $(D' \setminus \{u\}) \cup \{c_j\}$ is a minimum semi-TD-set of G containing $D_{r-1} \cup \{c_j\}$. This completes the proof of Claim 9. □

By Claim 9, if $L(v) = (0,0)$ for some vertex $v \in V(B_r)$, then the algorithm selects any cut-vertex $c_j \in V(B_r)$. Let B_k where $k < r$ be the block such that $V(B_r) \cap V(B_k) = \{c_j\}$. By Observation 3, $L(x) \in \{(1,0), (1,1), (1,2)\}$ for every $x \in V(B_k) \setminus \{c_j\}$. Thus there exists a vertex $y \in N_G(x)$ such that $L_2(y) \neq 0$. We note that $d_G(y, c_j) \leq 2$. The algorithm therefore assigns to c_j the label $L(c_j) = (1,2)$. If there exists a vertex $z \in N_G(u)$ for some $u \in V(B_r) \setminus \{v\}$ such that $L(z) = (1,1)$, then $d_G(c_j, z) = 2$. Let $L(v) = (1,0)$ for some $v \in V(B_r)$ and $B(v) = \{y \in N_G(v) \mid L_2(y) \neq 0\}$. If $|B(v)| > 1$, then for every $x \in B(v)$, there exists a vertex $y \in B(v)$ different from x such that $d_G(x, y) \leq 2$. Hence for every neighbor of v with L_2-label 1 (if such a neighbor of v exists), we can associate a vertex with L_2-label 1 or 2. If $|B(v)| = 1$ for any vertex $v \in V(B_r)$ with $L(v) = (1,0)$ that has a neighbor with label $(1,1)$, then the algorithm finds its 2-distance neighbor vertex by the following claim.

Claim 10. *Suppose that $L(v) = (1,0)$ for some vertex $v \in V(B_r)$, $L(u) = (1,1)$ for some $u \in N_G(v)$, and $B(v) = \{y \in N_G(v) \mid L_2(y) \neq 0\}$. If $|B(v)| = 1$, then there is a minimum semi-TD-set of G containing $D_{r-1} \cup \{w\}$, where $w \in V(B_r) \setminus \{u\}$.*

Proof. We once again consider the minimum semi-TD-set D' of G. Since $L(u) = (1,1)$ for some $v \in V(B_r)$, the vertex $u \in D_{r-1}$. Since D' is a semi-TD-set of G, there is a vertex $u' \in D'$ such that $d_G(u, u') \leq 2$. If $u' = w$, then we are done. Hence we may assume that $u' \neq w$. Since $L(u) = (1,1)$, we note that $u' \notin D_{r-1}$, and so $L_2(u') = 0$. Since $|B(v)| = 1$, there is no vertex $y \in N_G(v) \setminus \{u\}$ such that $L_2(y) \neq 0$. By Observation 3, $L_1(x) = 1$ for all $x \in V(B_j) \setminus \{F_j\}$, where $j \in [r-1]$. This implies that every vertex of $V(G) \setminus V(B_r)$ is dominated by D_{r-1}. We note that $\{z \in N_G[u'] \cap V(B_r) \mid L_1(z) = 0\} \subseteq N_G[w] \cup N_G[D_{r-1}]$, i.e., the undominated vertices of $N_G[u']$ present in $V(B_r)$ are dominated by $D_{r-1} \cup \{w\}$.

Let $N_2(D', u') = \{x \mid x \in D' \cap N_G^2(u')\}$ and $w \in V(B_r) \setminus \{u\}$. Let p be an arbitrary vertex in $N_2(D', u')$. If $d_G(p, u) \leq 2$ or $d_G(p, w) \leq 2$, then $(D' \setminus \{u'\}) \cup \{w\}$ is a minimum semi-TD-set of G containing $D_{r-1} \cup \{w\}$, as desired. Hence we may assume that $d_G(p, u) > 2$ and $d_G(p, w) > 2$. In this case, $p \in V(B_q)$ for some $q < k$, where $u' \in V(B_k) \setminus \{F_k\}$. We note that $L(x) = (1, 0)$ for every $x \in V(B_k) \setminus \{F_k\}$ since $L(u) = (1, 1)$. Thus by Observation 4, either $L_2(p) = 0$ (hence $p \notin D_{r-1}$) or $L(p) = (1, 2)$.

Let $S = \{x \in N_2(D', u') \mid d_G(x, w) > 2, d_G(x, u) > 2$, and $L_2(x) = 0\}$ and $S' = \{x \in S \mid N_G^2(x) \cap D' = \{u'\}\}$. We note that each element of S' does not belong to D_{r-1} and belongs to the blocks that appear before r. Moreover, $N_G[S'] \subseteq N_G[D_{r-1}] \cup N_G[w]$. If $|S'| \geq 2$, then $(D' \setminus S') \cup \{w\}$ is a semi-TD-set of G of cardinality less than $|D'|$, contradicting the minimality of D'. Hence, $|S'| \leq 1$. If $|S'| = 1$, then $(D' \setminus S') \cup \{w\}$ is a minimum semi-TD-set of G containing $D_{r-1} \cup \{w\}$, as desired. Hence we may assume that $S' = \emptyset$. Since $d_G(u, w) \leq 2$, the set $(D' \setminus \{u'\}) \cup \{w\}$ is a minimum semi-TD-set of G containing $D_{r-1} \cup \{w\}$. This completes the proof of Claim 9. □

We now return to the proof of Theorem 6. Recall that by the induction hypothesis, the set D_{i-1} is contained in some minimum semi-TD-set D' of G. Now assume that the algorithm is at the i-th iteration and let B_i be the block of G considered at the i-th iteration. If $L(v) = (0, 0)$ for some $v \in V(B_i) \setminus \{F_i\}$ and $i < r$, then the algorithm selects the vertex F_i (see Lines 11-13 of the algorithm MSTDS-Block(G) and notice that in the algorithm $L(F_i)$ is made $(1, 2)$ or $(1, 1)$). By Claim 7, $D_i = D_{i-1} \cup \{F_i\}$ is contained in some minimum semi-TD-set of G. If $L(v) = (1, 0)$ for some $v \in V(B_i) \setminus \{F_i\}$ and $i < r$, then the algorithm checks the set $A(v)$. If $|A(v)| > 1$, then the algorithm does not select any new vertex; rather it makes $L_2(x) = 2$ for the neighbor x of v if $L_2(x) = 1$. Hence, $D_i = D_{i-1}$ and therefore the set D_i is contained in the minimum semi-TD-set D' of G. If $|A(v)| = 1$, then the algorithm selects F_i (see Line 20 of the algorithm MSTDS-Block(G) and notice that $L(F_i)$ is made $(1, 2)$). By Claim 8, $D_i = D_{i-1} \cup \{F_i\}$ is contained in some minimum semi-TD-set of G. If $i = r$, then by Claim 9 and 10, the set D_i is contained in some minimum semi-TD-set of G. Therefore, by induction, D_r is a minimum semi-TD-set of G. This completes the proof of Theorem 6. ■

By Theorem 6, the algorithm MSTDS-Block(G) produces a minimum semi-TD-set of G. This establishes the correctness of the algorithm. We discuss next how a minimum semi-TD-set of a given block graph G can be computed in linear time. If G is complete, then as observed earlier, any two vertices in G form a semi-TD-set of G, implying that $\gamma_{t2}(G) = 2$. If G is not complete, then the algorithm MSTDS-Block(G) is used to compute a minimum semi-TD-set of G. We now show that the implementation of MSTDS-Block(G) can be done in linear time.

Suppose that G has blocks $B_1, B_2, \ldots, B_r$ and cut-vertices $c_1, c_2, \ldots, c_s$. A cut-tree T_G of G can be constructed in linear time [1]. Once a cut-tree is constructed, a RBFS-Block-Ordering of the blocks for G can be obtained in $O(r+s)$ time. The algorithm uses two dimensional array L on each vertex v of G. This two dimensional array can be seen as two arrays L_1 and L_2. Here, we use the array notation (.) instead of [.] for L_1 and L_2 to avoid confusion as we mean the same labels L_1 and L_2 used in the algorithm. Initially, $L_1(v) = 0 = L_2(v)$ for every vertex v of G. We also maintain an array F on each block of G, where F is defined with respect to the RBFS-Block-Ordering σ of the blocks for G. In particular, for $i \in [r-1]$, $F[i] = t$ if c_t is the cut-vertex common to the blocks B_i and B_{i+1}. At the i-th iteration, the algorithm considers the block B_i.

• If $L_1(v) = 0 = L_2(v)$ for some vertex $v \in V(B_i) \setminus \{F_i\}$, then $L_1(F_i)$ is made 1 and $L_2(F_i)$ is made 1 or 2. This takes at most $O(V(B_i) + d_G(F_i)) = O(d_G(F_i))$ time. Thus, $L_1(x)$ is made 1 for every vertex $x \in N_G[F_i]$, which takes $O(d_G(F_i))$ time.

• If $L_1(v) = 1$, and $L_2(v) = 0$ for some vertex $v \in V(B_i) \setminus \{F_i\}$, then $A(v)$ is computed which can be done in $O(d_G(v))$ time. If $|A(v)| > 1$, then $L_2(x)$ is made 2 for every $x \in N_G(v)$ such that $L_2(x) = 1$. This update can be done in $O(d_G(v))$ time. If $|A(v)| = 1$, then $L_2(F[i])$ is made 2 and $L_2(x)$ is made 2 for every $x \in N_G(c_t)$ such that $L_2(x) = 1$, where $F[i] = c_t$. This can be done in $O(d_G(v) + d_G(F_i))$ time.

For $i \in [r-1]$, at the i-th iteration, the algorithm takes

$$O\left(\sum_{v \in V(B_i)\setminus\{F_i\}} d_G(v) + d_G(F_i)\right) = O\left(\sum_{v \in V(B_i)} d_G(v)\right)$$

time. Now consider the r-th iteration of the algorithm. If $L_1(v) = 0 = L_2(v)$ for some $v \in V(B_r)$, then $L_2(c_j)$ is made 2 and the L-label of the neighbors of c_j is updated. This takes $O(d_G(c_j))$ time. If $L_1(u) = 1 = L_2(u)$ for some $u \in N_G(v)$, then a vertex w of $V(B_r)$ is chosen and the L-labels of the neighbors of w and v are updated. This takes $O(d_G(v) + d_G(w))$ time. So in total at the r-th iteration, the algorithm takes

$$O\left(\sum_{v \in V(B_r)} d_G(v)\right)$$

time. From the above discussion, we conclude that the algorithm takes at most $O(|V(G)| + |E(G)|)$ time. Therefore, we have the following theorem.

Theorem 11. *A minimum semitotal dominating set of a given block graph can be computed in linear time.*

4. NP-Completeness

In this section, we show that the semitotal domination problem is NP-complete for undirected path graphs, a subclass of chordal graphs. The semitotal domination problem is shown to be NP-complete for chordal graphs [14]. Let $\mathcal{F}$ be a finite family of nonempty sets. A graph $G = (V, E)$ is called an *intersection graph* for $\mathcal{F}$ if there exists a one-to-one correspondence between $\mathcal{F} = \{A_1, A_2, \ldots, A_n\}$ and $V = \{v_1, v_2, \ldots, v_n\}$ such that $v_iv_j \in E$ if and only if $A_i \cap A_j \neq \emptyset$. A graph G is called an *undirected path graph* if G is an intersection graph of a family of undirected paths of a tree.

Given a graph G and a positive integer k, the *domination* problem is to decide whether G has a dominating set of cardinality at most k. We describe next a polynomial time reduction from the domination problem to the semitotal domination problem. Given a graph $G = (V, E)$, we construct another graph $G' = (V', E')$, where $V' = V \cup \{x_i, y_i, z_i, p_i, q_i \mid i \in [n]\}$ and $E' = E \cup \{v_ix_i, x_iy_i, y_iz_i, y_ip_i, p_iq_i \mid i \in [n]\}$. The construction of the graph G' from the graph G is illustrated in Figure 2.

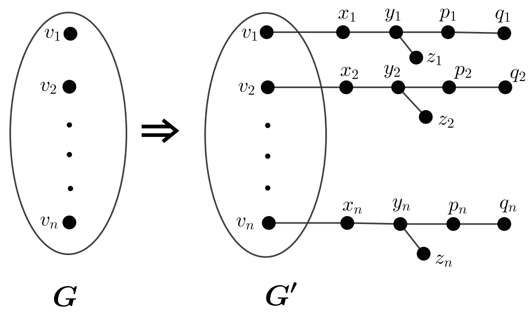

Figure 2. The constructed graph G' from the graph G.

Lemma 12. *The graph G has a dominating set of cardinality at most k if and only if the graph G' has a semi-TD-set of cardinality at most $k + 2n$.*

Proof. Let D be a dominating set of cardinality at most k. Consider the set $D' = D \cup \{y_i, p_i \mid i \in [n]\}$. We note that D' is a dominating set of G' with cardinality at most $k + 2n$. Since $d_G(y_i, p_i) = 1$ and $d_G(y_i, v_i) = 2$ for all $i \in [n]$, the set D' is a semi-TD-set of G'.

To prove the converse, we first show that there is a semi-TD-set D' of G of cardinality at most $k + 2n$ such that $y_i, p_i \in D'$ and $x_i, z_i, q_i \notin D'$ for all $i \in [n]$. Assume that D' is a minimum semi-TD-set of G' with cardinality at most $k+2n$. Since D' is a semi-TD-set, q_i or $p_i \in D'$ in order to dominate q_i and also z_i or $y_i \in D'$ in order to dominate z_i. Without loss of generality, we may assume that $y_i, p_i \in D'$ for each $i \in [n]$. Also we may assume that $q_i, z_i \notin D'$, for otherwise we can obtain another smaller semi-TD-set of G' of cardinality at most $k + 2n$

by removing q_i and z_i. Now suppose that $x_i \in D'$. We may assume that $v_i \notin D'$, for otherwise we get another semi-TD-set of G' of cardinality at most $k + 2n$ by removing x_i from D' as desired. With this assumption, the set $(D' \setminus \{x_i\}) \cup \{v_i\}$ is also a semi-TD-set of G with cardinality at most $k + 2n$. Hence without loss of generality, we assume that $x_i, z_i, q_i \notin D'$ for all $i \in [n]$. Consider the set $D'' = D' \setminus \{y_i, p_i \mid i \in [n]\}$. The resulting set D'' is a dominating set of G such that $|D''| \le k$. This completes the proof of the lemma. ∎

We now prove that the constructed graph G' is an undirected path graph. Suppose that G is an undirected path graph having n vertices. So by definition of undirected path graphs, there exists a tree T and a family $\mathcal{P}$ of paths of T such that G is the intersection graph of the family of paths $\mathcal{P}$ of T. Let T be a tree and $\mathcal{P} = \{P_{v_i} \mid i \in [n]\}$ be the family of distinct paths of T such that G is the intersection graph of the family of paths $\mathcal{P}$ of T. For each path P_{v_i} of T, let v_i^* be an end vertex of the path P_{v_i}. We construct two sets of paths by extending each P_{v_i} at v_i^*. We extend P_{v_i} at v_i^* to q_i and z_i by attaching paths $v_i^* u_i x_i y_i a_i p_i q_i$ and $v_i^* u_i x_i y_i z_i$, respectively. Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be the sets of paths obtained from each P_{v_i} where $i \in [n]$ by extending P_{v_i} at v_i^* to q_i and z_i, respectively. Suppose T' is the tree obtained from T by introducing the sets of paths $\mathcal{P}_1$ and $\mathcal{P}_2$. Let $P^*_{v_i} = P_{v_i} \cup \{v_i^* u_i\}$ for every $i \in [n]$ and let $\mathcal{P}^* = \{P^*_{v_i} \mid i \in [n]\}$. The graph G' is now the intersection graph of the family of paths $\mathcal{P}^* \cup \{x_i y_i a_i \mid i \in [n]\} \cup \{u_i x_i, a_i p_i, p_i q_i, y_i z_i \mid i \in [n]\}$ of T'. Therefore, G' is an undirected path graph. We note that the path $P^*_{v_i}$ in T' corresponds to the vertex v_i, the path $x_i y_i a_i$ in T' corresponds to the vertex y_i, and the paths $u_i x_i, a_i p_i, p_i q_i, y_i z_i$ in T' correspond to the vertices x_i, p_i, q_i, z_i, respectively.

The domination problem is shown to be NP-complete for undirected path graphs [2]. Therefore as an immediate consequence of Lemma 12, we have the following theorem.

Theorem 13. *The semitotal domination problem is NP-complete for undirected path graphs.*

5. Conclusion

In this paper, we considered the complexity of finding a minimum semi-TD-set in block graphs and present a linear time algorithm for this problem. On the other hand, we proved that the decision version of finding a minimum semi-TD-set is NP-complete in undirected path graphs, which is a superclass of block graphs. We note that strongly chordal graphs form a superclass of the block graphs. It would therefore be interesting to raise the problem to study the complexity of finding a minimum semitotal dominating set in strongly chordal graphs.

REFERENCES

[1] A.V. Aho, J.E. Hopcroft and J.D. Ullman, The Design and Analysis of Computer Algorithms (Addison-Wesley, Boston, 1974).

[2] K.S. Booth and J.H. Johnson, *Dominating sets in chordal graphs*, SIAM J. Comput. **11** (1982) 191–199.
https://doi.org/10.1137/0211015

[3] W. Goddard, M.A. Henning and C.A. McPillan, *Semitotal domination in graphs*, Util. Math. **94** (2014) 67–81.

[4] T.W. Haynes, S.T. Hedetniemi and P.J. Slater, Fundamentals of Domination in Graphs (Marcel Dekker Inc., New York, 1998).

[5] T.W. Haynes, S.T. Hedetniemi and P.J. Slater, Domination in Graphs: Advanced Topics (Marcel Dekker Inc., New York, 1998).

[6] T.W. Haynes and M.A. Henning, *Perfect graphs involving semitotal and semipaired domination*, J. Comb. Optim. **36** (2018) 416–433.
https://doi.org/10.1007/s10878-018-0303-9

[7] M.A. Henning, *Edge weighting functions on semitotal dominating sets*, Graphs Combin. **33** (2017) 403–417.
https://doi.org/10.1007/s00373-017-1769-4

[8] M.A. Henning and A.J. Marcon, *On matching and semitotal domination in graphs*, Discrete Math. **324** (2014) 13–18.
https://doi.org/10.1016/j.disc.2014.01.021

[9] M.A. Henning and A.J. Marcon, *Vertices contained in all or in no minimum semitotal dominating set of a tree*, Discuss. Math. Graph Theory **36** (2016) 71–93.
https://doi.org/10.7151/dmgt.1844

[10] M.A. Henning and A.J. Marcon, *Semitotal domination in claw-free cubic graphs*, Ann. Comb. **20** (2016)) 799–813.
https://doi.org/10.1007/s00026-016-0331-z

[11] M.A. Henning and A.J. Marcon, *Semitotal domination in graphs: Partition and algorithmic results*, Util. Math. **106** (2018) 165–184.

[12] M.A. Henning and A. Pandey, *Algorithmic aspects of semitotal domination in graphs*, Theoret. Comput. Sci. **766** (2019) 46–57.
https://doi.org/10.1016/j.tcs.2018.09.019

[13] M.A. Henning and A. Yeo, Total Domination in Graphs (Springer, New York, 2013).
https://doi.org/10.1007/978-1-4614-6525-6

[14] Z. Shao and P. Wu, *Complexity and approximation ratio of semitotal domination in graphs*, Commun. Comb. Optim. **3** (2018) 143–150.
https://doi.org/10.22049/CCO.2018.25987.1065

[15] W. Zhuang and G. Hao, *Semitotal domination in trees*, Discrete Math. Theoret. Comput. Sci. **20(2)** (2018) #5.
https://doi.org/10.23638/DMTCS-20-2-5

Received 19 April 2019
Revised 18 September 2019
Accepted 24 September 2019