## Linear-Time Algorithm for the Paired-Domination Problem on Weighted Block Graphs

Ching-Chi Lin and Cheng-Yu Hsieh

Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung 20224, Taiwan lincc@mail.ntou.edu.tw Department of Computer Science and Information Engineering, National Taiwan University, Taipei 10617, Taiwan r01922114@ntu.edu.tw

#### Abstract

Given a graph G = (V, E), the domination problem is to find a minimum size vertex subset  $S \subseteq V(G)$  such that every vertex not in S is adjacent to a vertex in S. A dominating set S of G is called a paired-dominating set if the induced subgraph G[S] contains a per-The paired-domination problem fect matching. involves finding a paired-dominating set S of G such that the cardinality of S is minimized. Suppose that, for each  $v \in V(G)$ , we have a weight w(v) specifying the cost for adding v to S. The weighted paired-domination problem is to find a paired-dominating set S whose weight  $w(S) = \sum \{w(v) : v \in S\}$  is minimized. In this paper, we propose an O(n + m)-time algorithm for the weighted paired-domination problem on block graphs using dynamic-programming method. Moreover, the algorithm can be completed in O(n)time if the block-cut-vertex structure of G is given.

#### 1 Introduction

The domination problem has been extensively studied in the area of algorithmic graph theory for serval decades; see [2, 8, 10-14] for books and survey papers. Given a graph G = (V, E), the domination problem is to find a minimum size vertex subset  $S \subseteq V(G)$  such that every vertex not in S is adjacent to a vertex in S. The problem has many applications in the real world such as location problems, communication networks, and kernels of games [11]. Depending on the requirements of different types of applications, there are several variants of the domination problem, such as the independent domination, connected domination, total domination, and perfect domination problems [2, 8, 14, 20]. These problems have been proved to be NP-complete and have polynomialtime algorithms on some special classes of graphs. In particular, Haynes and Slater [9] introduced the concept of paired-domination motivated by security concerns. In a museum protection program, beside the requirement that each region has a guard in it or is in the protection range of some guard, the guards must be able back each other up.

In [9], Haynes and Slater showed that the paired-domination problem is NP-complete on general graphs and gave a lower bound of  $n/\Delta(G)$ for the cardinality of a paired-dominating set of G. Recently, many studies have been made for this problem in proving NP-completeness, providing approximation algorithms, and finding polynomial-time algorithms on some special classes of graphs. Here, we only mention some related results. For more detailed information regarding this problem, please refer to [15]. Chen *et* al. [5] demonstrated that the paired-domination problem is also NP-complete on bipartite graphs, chordal graphs, and split graphs. In [3], Chen *et* al. proposed an approximation algorithm with ratio  $\ln(2\Delta(G)) + 1$  for general graphs and showed that the problem is APX-complete, i.e., has no PTAS. Panda and Pradhan [18] strengthened the results in [5] by showing that the problem is also NP-complete for perfect elimination bipartite graphs.

Meanwhile, polynomial-time algorithms have been studied intensively on some special classes



Figure 1: (a) A block graph G. (b) The corresponding block-cut-vertex graph B for the block graph G in (a).

of graphs such as tree graphs [19], weighted tree graphs [3], inflated tree graphs [16], permutation graphs [6, 17], strongly chordal graphs [4], interval graphs [5,7] and circular-arc graphs [7]. Especially, Chen *et al.* [5] introduced an O(m+n)time algorithm for block graphs, a proper superfamily of tree graphs. In this paper, we propose an O(n+m)-time algorithm for the weighted paired-domination problem on block graphs using dynamic-programming method, where n = |V(G)|and m = |E(G)|. This strengthens the results in [3, 5]. Moreover, the algorithm can be completed in O(n) time if the block-cut-vertex structure of G is given. Notice that the block-cut-vertex structure of a block graph G can be constructed in O(n+m) time by the depth first search algorithm [1].

The remainder of this paper is organized as follows. Section 2 gives the block-cut-vertex structure of a block graph G, we employ the dynamicprogramming method to present an O(n)-time algorithm for finding a minimum-weight paireddominating set of G. Section 3, gives an efficient implementation of the algorithm proposed in Section 2. Section 4 contains some concluding remarks and future work.

### 2 The Proposed Algorithm for Block Graphs

In a graph G = (V, E), a vertex subset  $S \subseteq V(G)$  is said to be a *dominating set* of G if every vertex not in S is adjacent to a vertex in S. Let G[S] denote the subgraph of G induced

by a subset S of V(G). A dominating set S of G is called a *paired-dominating set* if the induced subgraph G[S] contains a perfect matching. The *paired-domination problem* involves finding a paired-dominating set S of G such that the cardinality of S is minimized. Suppose that, for each  $v \in V(G)$ , we have a weight w(v) specifying the cost for adding v to S. The weighted paireddomination problem is to find a paired-dominating set S whose weight  $w(S) = \sum \{w(v) : v \in S\}$  is minimized. Given the block-cut-vertex structure of a block graph G, we designed an O(n)-time algorithm that determines a minimum-weight paireddominating set of G using dynamic-programming method. Below, we introduce some preliminaries for block graphs.

For any connected graph G, a vertex  $x \in V(G)$ is called a *cut-vertex* of G, if G - x contains more than one connected component. A *block* is a maximal connected subgraph without a cut-vertex. A graph G is called a *block graph*, if every block in Gis a complete graph. Notice that block graphs are a proper superfamily of tree graphs and a proper subfamily of chordal graphs. Suppose G has blocks  $B_1, B_2, \ldots, B_x$  and cut vertices  $c_1, c_2, \ldots, c_y$ . We define the *block-cut-vertex graph*  $G^* = (V, E)$  of G, where

$$V(G^*) = \{B_1, B_2, \dots, B_x, c_1, c_2, \dots, c_y\}; \text{ and} E(G^*) = \{(B_i, c_j) \mid c_j \in B_j, 1 \le i \le x, 1 \le j \le y\}.$$

Consequently, graph  $G^*$  is a forest and the leaves in  $G^*$  are precisely the blocks with exactly one cut-vertex in G. A block containing exactly one cut-vertex in G is called an *end block*. It should be noted that, by using the depth first



Figure 2: A weighted block graph  $H = B \cup G_1 \cup G_2 \cup \ldots \cup G_k$ .

search algorithm, one can recognize the block graphs and construct the block-cut-vertex graphs  $G^*$ , both in O(n + m) time [1]. Figure 1 shows an illustrative example, in which Figure 1(b) depicts the corresponding block-cut-vertex graph  $G^*$ for the block graph G in Figure 1(a). Clearly, G has 7 blocks  $B_1, B_2, \ldots, B_7$  and 6 cut vertices  $c_1, c_2, \ldots, c_6$ . Moreover, the end blocks of G are  $B_3, B_4, B_5$ , and  $B_7$ .

#### 2.1 The algorithm

In this subsection, given the block-cut-vertex structure of a weighted block graph G, we propose an O(n)-time algorithm for finding a minimumweight paired-dominating set of G. Before describing the approach in detail, four notations D(H, u), P(H, u), P'(H, u), and  $\bar{P}(H, u)$  are defined below, where H is a subgraph of G and  $u \in V(H)$  is a cut vertex of G. The notations are introduced for the purpose of describing the recursive formulations used in developing dynamic-programming algorithms.

- D(H, u): A minimum-weight dominating set of  $H, u \in D(H, u)$  and H[D(H, u) - u]has a perfect matching.
- P(H, u): A minimum-weight paired-dominating set of H and  $u \in P(H, u)$ .
- P'(H, u): A minimum-weight paired-dominating set of H and  $u \notin P'(H, u)$ .
- $\overline{P}(H, u)$ : A minimum-weight paired-dominating set of H - u, and u is not dominated by  $\overline{P}(H, u)$ .

Clearly, either P(G, u) or P'(G, u) is a minimumweight paired-dominating set of G. For ease of subsequent discussion, D(H, u), P(H, u), P'(H, u), and P(H, u) are called a  $\kappa_1$ -paireddominating set,  $\kappa_2$ -paired-dominating set,  $\kappa_3$ paired-dominating set, and  $\kappa_4$ -paired-dominating set of H with respect to u, respectively.

Consider a weighted block graph  $H = B \cup$  $G_1 \cup G_2 \cup \ldots \cup G_k$ , where B is a block of H,  $G_i$  and  $G_j$  have disjoint vertex sets for  $i \neq j$ , and  $u_1, u_2, \ldots, u_k$  are the vertices of B such that  $V(B) \cap V(G_i) = u_i$  for  $1 \leq i \leq k$ . See Figure 2 for an illustrative example. In order to obtain a minimum-weight paired-dominating set of G, we use a dynamic-programming approach to iteratively determine  $D(H, u_1)$ ,  $P(H, u_1)$ ,  $P'(H, u_1)$ , and  $\bar{P}(H, u_1)$  in a bottom-up manner. Suppose that the dominating sets  $D(G_i, u_i), P(G_i, u_i),$  $P'(G_i, u_i)$ , and  $P(G_i, u_i)$  have been determined in the previous iterations and are assigned to vertex  $u_i$  for  $1 \leq i \leq k$ . If we know how to compute  $D(H, u_1), P(H, u_1), P'(H, u_1), \text{ and } P(H, u_1) \text{ in }$ O(k) time, we can then design an algorithm to solve the weighted paired-domination problem in O(n) time using dynamic-programming method.

Based on the above observation, the concept of our algorithm is as below. The algorithm first sets the current graph G' = G and the set of processed blocks  $W = \emptyset$ . Further, it initially assigns  $D(G[\{v\}], v) = \{v\}, P(G[\{v\}], v) = \Delta$ ,  $P'(G[\{v\}], v) = \Delta$  and  $\overline{P}(G[\{v\}], v) = \emptyset$  to each vertex  $v \in V(G)$ . Specially, we use  $\triangle$  to denote the empty set with a weight of infinity, i.e.,  $\triangle = \emptyset$ and  $w(\Delta) = \infty$ . The algorithm then iteratively processes blocks in the repeat loop. During each iteration of the loop, we remove an end block Bwith exactly one cut vertex in the current graph G' and determines the dominating sets D(H, u), P(H, u), P'(H, u), and P(H, u), where u is thecut vertex and H is the connected component containing u in  $G[B \cup W]$ . After the execution of the repeat loop, we have only one block left, i.e., the current graph G' is a block. With the information determined in the repeat loop, we now can find the two paired-dominating sets P(G, u) and P'(G, u), where u is an arbitrary vertex in G'. The output S is selected from P(G, u) and P'(G, u) based on the weights of the sets. The steps of the algorithm are detailed below.

 $\label{eq:algorithm1} \begin{array}{l} \mbox{Algorithm1} \mbox{ Finding a paired-dominating set on} \\ \mbox{weighted block graphs} \end{array}$ 

**Input:** A weighted block graph G with the blockcut-vertex structure  $G^*$  of G.

- **Output:** A minimum weighted paired-dominating set S of G.
- 1: let  $G' \leftarrow G, W \leftarrow \emptyset$ ;
- 2: for each  $v \in V(G)$  do
- 3: let  $D(G[\{v\}], v) \leftarrow \{v\}, P(G[\{v\}], v) \leftarrow \Delta;$
- 4: let  $P'(G[\{v\}], v) \leftarrow \triangle, \bar{P}(G[\{v\}], v) \leftarrow \emptyset;$
- 5: end for
- 6: repeat
- 7: arbitrarily choose a block B with exactly one cut vertex in G';
- 8: suppose that  $V(B) = \{u_1, u_2, \dots, u_k\}$ , where  $u_1$  is the cut vertex and  $G_i$  is the connected component in G[W] such that  $V(B) \cap V(G_i) = u_i$  for  $1 \le i \le k$ ;
- 9: find  $D(H, u_1), P(H, u_1), P'(H, u_1), \bar{P}(H, u_1),$ where  $H = B \cup G_1 \cup G_2 \cup \ldots \cup G_k$ ;
- 10: let  $G' \leftarrow G' \{u_2, \ldots, u_k\}, W \leftarrow W \cup B;$
- 11: **until** G' itself is a block
- 12: find P(G, u) and P'(G, u), where u is an arbitrary vertex in G';
- 13: let  $S \leftarrow P(G, u)$  if w(P(G, u)) < w(P'(G, u)), and  $S \leftarrow P'(G, u)$  otherwise;
- 14: return S.

In the next section, we will prove the correctness of the algorithm and give an O(n)-time implementation.

# 3 Efficient implementation of the algorithm

Let  $H = B \cup G_1 \cup G_2 \cup \ldots \cup G_k$  be a weighted block graph such that B is a block of H and  $V(G_i) \cap V(G_j) = \emptyset$  for  $i \neq j$ . Further, we suppose that  $u_1, u_2, \ldots, u_k$  are the vertices of B such that  $V(B) \cap V(G_i) = u_i$  for  $1 \leq i \leq k$ , refer to Figure 2 for an illustrative example. In this section, we provide an efficient implementation of algorithm 1. More concretely, given the dominating sets  $D(G_i, u_i), P(G_i, u_i), P'(G_i, u_i)$ , and  $\overline{P}(G_i, u_i)$  for  $1 \leq i \leq k$ , dynamic-programming procedures are proposed in Subsections 3.1–3.4, which can determine  $D(H, u_1), P(H, u_1), P'(H, u_1)$ , and  $\overline{P}(H, u_1)$  in O(k) time, respectively. Clearly, the proposed procedures ensure the correctness of the algorithm and imply that the algorithm can be implemented in O(n) time. Consequently, we can obtain the main result of this paper.

**Theorem 1** Given a weighted block graph G with the block-cut-vertex structure of G, a paireddominating set of G can be found in O(n) time.

#### **3.1** Determination of $D(H, u_1)$

First, some notations are introduced below, for the purpose of describing the procedures. For a set S of sets of vertices, F(S) denotes the set with minimum weight in S. Further, let  $S_i^*$  be the set of vertices such that  $S_i^* = F(\{D(G_i, u_i), P(G_i, u_i), P'(G_i, u_i), \bar{P}(G_i, u_i)\})$ for  $2 \leq i \leq k$ . We use r to denote the number of  $S_i^*$  such that  $S_i^* = D(G_i, u_i)$ , i.e.,  $r = |\{S_i^* \mid S_i^* = D(G_i, u_i) \text{ and } 2 \leq i \leq k\}|$ . Suppose  $\alpha$  is the index in  $\{2, 3, \ldots, k\}$  such that  $S_{\alpha}^* \neq$  $D(G_{\alpha}, u_{\alpha})$  and  $w(D(G_{\alpha}, u_{\alpha})) - w(S_{\alpha}^*)$  is minimized. Further, let  $\beta$  be the index in  $\{2, 3, \ldots, k\}$ such that we have  $S_{\beta}^* = D(G_{\beta}, u_{\beta})$  and  $w(F(\{P(G_{\beta}, u_{\beta}), P'(G_{\beta}, u_{\beta}), \bar{P}(G_{\beta}, u_{\beta})\})) - w(S_{\beta}^*)$ is minimized.

Recall that  $D(H, u_1)$  is a minimum-weight dominating set of H such that  $u_1 \in D(H, u_1)$ and  $H[D(H, u_1) - u_1]$  has a perfect matching. By the definition of  $D(H, u_1)$ , the only potential candidate for being a dominating set of  $G_1$  is  $D(G_1, u_1)$ . Hence, in order to obtain  $D(H, u_1)$ , we first construct a dominating set  $X = D(G_1, u_1) \cup$  $S_2^* \cup S_3^* \cup \ldots \cup S_k^*$ . We will show that if r is even, then S = X is a  $\kappa_1$ -dominating set of H with respect to  $u_1$ . Otherwise, for the purpose of satisfying the requirement that  $H[X - u_1]$ has a perfect matching with minimum cost, we can either replace  $S^*_{\alpha}$  with  $D(G_{\alpha}, u_{\alpha})$ , or replace  $S^*_{\beta}$  with  $F(\{P(G_{\beta}, u_{\beta}), P'(G_{\beta}, u_{\beta}), \overline{P}(G_{\beta}, u_{\beta})\}).$ For the former case, a dominating set  $X^+$  =  $(X - S^*_{\alpha}) \cup D(G_{\alpha}, u_{\alpha})$  is created. On the other hand, a dominating set  $X^- = (X - S^*_{\beta}) \cup$  $F(\{P(G_{\beta}, u_{\beta}), P'(G_{\beta}, u_{\beta}), \overline{P}(G_{\beta}, u_{\beta})\})$  is built for the latter case. The output  $S = F(\{X^+, X^-\})$  is selected from  $X^+$  and  $X^-$  based on the weights of the sets. Similarly, we will show that S is a  $\kappa_1$ -dominating set of H with respect to  $u_1$ . The following is a formal description of the procedure.

**Procedure 2** Finding  $D(H, u_1)$ 

**Input:** Dominating sets  $D(G_i, u_i)$ ,  $P(G_i, u_i)$ ,  $P'(G_i, u_i)$ , and  $\overline{P}(G_i, u_i)$  for  $1 \le i \le k$ . **Output:** A  $\kappa_1$ -paired-dominating set S of H with respect to  $u_1$ . 1: let  $X \leftarrow D(G_1, u_1) \cup S_2^* \cup S_3^* \cup \ldots \cup S_k^*$ ; 2: let  $X^+ \leftarrow (X - S^*_{\alpha}) \cup D(G_{\alpha}, u_{\alpha});$ 3: let  $X^- \leftarrow (X - S^*_\beta) \cup$  $F(\{P(G_{\beta}, u_{\beta}), P'(G_{\beta}, u_{\beta}), \overline{P}(G_{\beta}, u_{\beta})\});$ 4: if r is even, then let  $S \leftarrow X$ ; otherwise, let  $S \leftarrow F(\{X^+, X^-\});$ 

5: return S.

**Lemma 2** Given the dominating sets  $D(G_i, u_i)$ ,  $P(G_i, u_i), P'(G_i, u_i), and \overline{P}(G_i, u_i) \text{ for } 1 \leq i \leq k,$ Procedure 2 outputs a  $\kappa_1$ -paired-dominating set S of H with respect to  $u_1$  in O(k) time.

Proof. The procedure certainly runs in O(k)time. To prove that S is a  $\kappa_1$ -paired-dominating set of H with respect to  $u_1$ , it suffices to show that the output S is a minimum-weight dominating set of H such that  $u_1 \in S$  and  $H[S-u_1]$  has a perfect matching. By the definition of  $D(H, u_1)$ , the only potential candidate for being a dominating set of  $G_1$  is  $D(G_1, u_1)$ . Hence, we have  $D(G_1, u_1) \subseteq S$ . Since  $u_1 \in D(G_1, u_1)$  and B is a clique, all the three sets  $X, X^+$  and  $X^-$  are dominating set of H. Therefore, it remains to show that the weight w(S) of S is minimized subject to the condition that  $H[S - u_1]$  contains a perfect matching.

Notice that, for  $2 \leq i \leq k$ , both  $G_i[D(G_i, u_i)$  $u_i$ ] and  $G_i[P(G_i, u_i)]$  contain perfect matchings and  $u_i \notin P'(G_i, u_i) \cup \overline{P}(G_i, u_i)$ . Hence, if r is even, then  $H[X - u_1]$  contains a perfect matching and the weight w(X) of X is minimized, as an immediate consequence of the selections of  $S_i^*$ . Next, suppose that r is odd. In order to satisfy the condition that  $H[X - u_1]$  contains a perfect matching with minimum cost, it is natural to replace  $S^*_{\alpha}$  with  $D(G_{\alpha}, u_{\alpha})$ , or replace  $S^*_{\beta}$  with  $F(\{P(G_{\beta}, u_{\beta}), P'(G_{\beta}, u_{\beta}), \overline{P}(G_{\beta}, u_{\beta})\}).$ For the former case, a dominating set  $X^+$  =  $(X - S^*_{\alpha}) \cup D(G_{\alpha}, u_{\alpha})$  is created. On the other hand, a dominating set  $X^- = (X - S^*_\beta) \cup$  $F(\{P(G_{\beta}, u_{\beta}), P'(G_{\beta}, u_{\beta}), \overline{P}(G_{\beta}, u_{\beta})\})$  is built for the latter case. And, we select S from  $X^+$  and  $X^$ based on the weights of the sets. As a consequence of selections of  $S^*_{\alpha}$ ,  $S^*_{\beta}$ , and  $S^*_i$  for  $2 \leq i \leq k$ , one

can verify that  $S = F({X^+, X^-})$  is a minimumweight dominating set of H such that  $H[S - u_1]$ contains a perfect matching. П

#### 3.2**Determination of** $P(H, u_1)$

Notice that  $P(H, u_1)$  is a minimum-weight dominating set of H such that  $u_1 \in P(H, u_1)$ and  $H[P(H, u_1)]$  has a perfect matching. Therefore, either  $D(G_1, u_1) \subseteq P(H, u_1)$  or  $P(G_1, u_1) \subseteq$  $P(H, u_1)$  is a dominating set of  $G_1$ . In order to obtain  $P(H, u_1)$ , we construct the six dominating sets  $X, X^+, X^-, Y, Y^+$ , and  $Y^-$  of H. The dominating sets  $X, X^+$  and  $X^-$  are created for the situation when  $D(H, u_1)$  is a dominating set of  $G_1$ . Meanwhile, the dominating sets  $Y, Y^+$  and  $Y^-$  are built for the situation when  $P(G_1, u_1)$  is a dominating set of  $G_1$ , where  $Y = P(G_1, u_1) \cup S_2^* \cup S_3^* \cup$  $\cdots \cup S_k^*, Y^+ = (Y - S_\alpha^*) \cup D(G_\alpha, u_\alpha), \text{ and } Y^- =$  $(Y - S^*_{\beta}) \cup F(\{P(G_{\beta}, u_{\beta}), P'(G_{\beta}, u_{\beta}), \bar{P}(G_{\beta}, u_{\beta})\}).$ If r is even, then the induced subgraphs  $H[X^+], H[X^-]$  and H[Y] all contain perfect matchings. The output  $S = F(\{X^+, X^-, Y\})$  is selected from  $X^+, X^-$  and Y based on the weights of the sets. We will show that S is a  $\kappa_2$ -dominating set of H with respect to  $u_1$ . Similarly, if r is odd, then the induced subgraphs  $H[X], H[Y^+]$ and  $H[Y^{-}]$  all contain perfect matchings. And, we will show that the output  $S = F(\{X, Y^+, Y^-\})$  is a  $\kappa_2$ -dominating set of H with respect to  $u_1$  in this situation. The procedure is detailed in the next page.

**Procedure 3** Finding  $P(H, u_1)$ 

**Input:** Dominating sets  $D(G_i, u_i)$ ,  $P(G_i, u_i)$ ,  $P'(G_i, u_i)$ , and  $\overline{P}(G_i, u_i)$  for  $1 \leq i \leq k$ .

- **Output:** A  $\kappa_2$ -paired-dominating set S of H with respect to  $u_1$ .
- 1: find the dominating sets  $X, X^+$  and  $X^-$  as described in Procedure 2;
- 2: let  $Y \leftarrow P(G_1, u_1) \cup S_2^* \cup S_3^* \cup \cdots \cup S_k^*$ ;
- 3: let  $Y^+ \leftarrow (Y S^*_{\alpha}) \cup \tilde{D}(G^-_{\alpha}, u_{\alpha});$ 4: let  $Y^- \leftarrow (Y S^*_{\beta}) \cup$  $F(\{P(G_{\beta}, u_{\beta}), P'(G_{\beta}, u_{\beta}), \overline{P}(G_{\beta}, u_{\beta})\});$
- 5: if r is even, then let  $S \leftarrow F(\{X^+, X^-, Y\})$ ; otherwise, let  $S \leftarrow F(\{X, Y^+, Y^-\});$
- 6: return S.

**Lemma 3** Given the dominating sets  $D(G_i, u_i)$ ,  $P(G_i, u_i), P'(G_i, u_i), and \overline{P}(G_i, u_i) \text{ for } 1 \leq i \leq k,$  Procedure 3 outputs a  $\kappa_2$ -paired-dominating set S of H with respect to  $u_1$  in O(k) time.

*Proof.* The procedure certainly can be completed in O(k) time. To prove the correctness of the procedure, it suffices to show that the output S is a minimum-weight dominating set of H such that  $u_1 \in S$  and H[S] contains a perfect matching. Further, since  $v_1 \in D(G_1, u_1) \cap P(G_1, u_1)$  and B is a clique,  $X, X^+, X^-, Y, Y^+$ , and  $Y^-$  are all dominating sets of H. Thus, it remains to show that the weight w(S) of S is minimized subject to the condition that H[S] contains a perfect matching.

Notice that, for  $2 \leq i \leq k$ , both  $G_i[D(G_i, u_i)$  $u_i$  and  $G_i[P(G_i, u_i)]$  contain perfect matchings and  $u_i \notin P'(G_i, u_i) \cup \overline{P}(G_i, u_i)$ . We first consider the situation when r is even. For the case when  $D(G_1, u_1)$  is a dominating set of  $G_1$ , in order to satisfy the condition that H[X] contains a perfect matching with minimum cost, we can either replace  $S^*_{\alpha}$  with  $D(G_{\alpha}, u_{\alpha})$  or replace  $S^*_{\beta}$  with  $F(\{P(G_{\beta}, u_{\beta}), P'(G_{\beta}, u_{\beta}), \overline{P}(G_{\beta}, u_{\beta})\}).$ Thus,  $X^+$  and  $X^-$  are the two potential candidates for S when  $D(G_1, u_1) \subseteq P(H, u_1)$ . For the case when  $P(G_1, u_1)$  is a dominating set of  $G_1, H[Y]$  contains a perfect matching. We select  $S = F({X^+, X^-, Y})$  from  $X^+, X^-$  and Y based on the weights of the sets. As a consequence of selections of  $S^*_{\alpha}$ ,  $S^*_{\beta}$ , and  $S^*_i$  for  $2 \leq i \leq k$ , one can verify that the output S is a minimum-weight dominating set of H such that H[S] contains a perfect matching. Using a similar method of the above arguments, one can show that the correctness also holds for the situation when r is odd. 

#### **3.3 Determination of** $P'(H, u_1)$

Recall that  $P'(H, u_1)$  is a minimum-weight dominating set of H such that  $u_1 \notin P'(H, u_1)$ and  $H[P'(H, u_1)]$  has a perfect matching. Therefore, either  $P'(G_1, u_1) \subseteq P'(H, u_1)$  or  $\overline{P}(G_1, u_1) \subseteq$  $P'(H, u_1)$  is a dominating set of  $G_1$ . For ease of subsequent discussion, we consider the two cases  $P'(G_1, u_1) \subseteq P'(H, u_1)$  and  $\overline{P}(G_1, u_1) \subseteq$  $P'(H, u_1)$ , respectively, in the rest of this subsection. More concretely, a paired-dominating set  $Q_1$ is created for the former situation. Meanwhile, a paired-dominating set  $Q_2$  is built for the latter situation. Clearly,  $P'(H, u_1)$  can be selected from  $Q_1$ and  $Q_2$  based on the weights of the sets.

#### **3.3.1** Finding $Q_1$

Below we present an O(k)-time procedure for finding  $Q_1$ . The procedure solves the problem by considering eight cases  $C_1, C_2, \ldots, C_8$  depending on  $S_i^*$  and r. For  $1 \leq i \leq 8$ , the case  $C_i = (c_1, c_2, c_3, c_4, c_5)$  is an ordered 5-tuple. For  $1 \leq j \leq 5, c_j = 1$  if condition  $D_j$  holds, and  $c_j = 0$ otherwise. Further, we have  $c_j = "*"$  if condition  $D_j$  is known never to occur. The five conditions  $D_1, D_2, \ldots, D_5$  are defined as follows:

$$\begin{array}{l} D_1: \, S_i^* = P(G_i, u_i) \text{ for some } 2 \leq i \leq k. \\ D_2: \, r \text{ is odd.} \\ D_3: \, r \text{ is equal to } 1. \\ D_4: \, r \text{ is equal to } 0. \\ D_5: \, S_i^* = \bar{P}(G_i, u_i) \text{ for some } 2 \leq i \leq k. \end{array}$$

Then, we define the cases  $C_1 = (1, 1, *, *, *)$ ,  $C_2 = (1, 0, *, *, *)$ ,  $C_3 = (0, 1, 1, *, 1)$ ,  $C_4 = (0, 1, 1, *, 0)$ ,  $C_5 = (0, 1, 0, *, *)$ ,  $C_6 = (0, 0, *, 1, 1)$ ,  $C_7 = (0, 0, *, 1, 0)$ , and  $C_8 = (0, 0, *, 0, *)$ . For example, case  $C_1$  represents the situation when there exists an index  $\ell$  such that  $S_{\ell}^* = P(G_{\ell}, u_{\ell})$ with  $2 \leq \ell \leq k$  and r is an odd number. Further, case  $C_7$  represents the situation when there exists no index  $\ell$  such that  $S_{\ell}^* = P(G_{\ell}, u_{\ell})$ , or  $S_{\ell}^* = \bar{P}(G_{\ell}, u_{\ell})$  and r = 0, i.e.,  $S_i^* = P'(G_i, u_i)$ for  $2 \leq i \leq k$ . Moreover, one can verify that all the possible combinations of the five conditions have been considered.

Next, some notations and paired-dominating sets are introduced. Let  $\alpha'$  be the index in  $\{2,3,\ldots,k\} - \{\alpha\}$  such that  $S_{\alpha'}^* \neq D(G_{\alpha'}, u_{\alpha'})$ and  $w(D(G_{\alpha'}, u_{\alpha'})) - w(S_{\alpha'}^*)$  is minimized. Let  $\gamma$  be the index in  $\{2,3,\ldots,k\}$  such that  $S_{\gamma}^* \neq P(G_{\gamma}, u_{\gamma})$  and  $w(P(G_{\gamma}, u_{\gamma})) - w(S_{\gamma}^*)$  is minimized. Let  $I = \{i \mid S_i^* = \overline{P}(G_i, u_i) \text{ and } 2 \leq i \leq k\}$ . We define the following paired-dominating sets of H, which are the potential candidates for  $Q_1$ .

$$\begin{array}{l} Z_1 &= P'(G_1, u_1) \cup S_2^* \cup S_3^* \cup \ldots \cup S_k^*. \\ Z_1^+ &= (Z_1 - S_\alpha^*) \cup D(G_\alpha, u_\alpha). \\ Z_1^- &= (Z_1 - S_\beta^*) \cup \\ F(\{P(G_\beta, u_\beta), P'(G_\beta, u_\beta), \bar{P}(G_\beta, u_\beta)\}). \\ T_1 &= (Z_1 - S_\gamma^*) \cup P(G_\gamma, u_\gamma). \\ T_2 &= (Z_1 - S_\alpha^* - S_{\alpha'}^*) \cup D(G_\alpha, u_\alpha) \cup D(G_{\alpha'}, u_{\alpha'}). \\ T_3 &= (Z_1 - \cup_{i \in I} S_i^*) \cup (\cup_{i \in I} P'(G_i, u_i)). \\ T_4 &= (Z_1 - S_\beta^*) \cup P(G_\beta, u_\beta). \\ T_5 &= (Z_1 - S_\beta^*) \cup F(\{P(G_\beta, u_\beta), P'(G_\beta, u_\beta)\}). \\ T_6 &= (Z_1 - S_\gamma^* - S_\beta^*) \cup P(G_\gamma, u_\gamma) \cup \\ F(\{P'(G_\beta, u_\beta), \bar{P}(G_\beta, u_\beta)\}). \\ T_7 &= (Z_1 - S_\gamma^* - S_\beta^*) \cup P(G_\gamma, u_\gamma) \cup \bar{P}(G_\beta, u_\beta)]. \\ T_8 &= (Z_1 - \cup_{i \in I} S_i^* - S_\beta^*) \cup (\cup_{i \in I} P'(G_i, u_i)) \cup \\ P'(G_\beta, u_\beta). \end{array}$$

As mentioned earlier, we solve the problem by considering the eight cases  $C_1, C_2, \ldots, C_8$ . The relations between the cases  $C_1, C_2, \ldots, C_8$  and the dominating sets  $Z_1, Z_1^+, Z_1^-, T_1, \ldots, T_8$  are detailed in the following procedure. We will prove its correctness and analyze its running time in Lemma 4.

#### **Procedure 4** Finding $Q_1$

- Input: Dominating sets  $D(G_i, u_i)$ ,  $P(G_i, u_i)$ ,  $P'(G_i, u_i)$ , and  $\overline{P}(G_i, u_i)$  for  $1 \le i \le k$ .
- **Output:** A minimum-weight dominating set S of H such that  $u_1 \notin S$ , H[S] has a perfect matching, and  $P'(G_1, u_1) \subseteq S$ .
- 1: if  $C_1$  or  $C_5$  holds, then let  $S \leftarrow F(\{Z_1^+, Z_1^-\});$
- 2: if  $C_2$  or  $C_7$  or  $C_8$  holds, then let  $S \leftarrow Z_1$ ;
- 3: if  $C_3$  holds, then let  $S \leftarrow F(\{Z_1^+, T_4, T_6, T_8\});$
- 4: if  $C_4$  holds, then let  $S \leftarrow F(\{Z_1^+, T_5, T_7\});$
- 5: if  $C_6$  holds, then let  $S \leftarrow F(\{T_1, T_2, T_3\})$ ;
- 6: return S.

**Lemma 4** Given the dominating sets  $D(G_i, u_i)$ ,  $P(G_i, u_i)$ ,  $P'(G_i, u_i)$ , and  $\overline{P}(G_i, u_i)$  for  $1 \le i \le k$ , Procedure 4 outputs a minimum-weight dominating set S of H such that  $u_1 \notin S$ , H[S] has a perfect matching, and  $P'(G_1, u_1) \subseteq S$ . Moreover, the procedure can be completed in O(k) time.

**Proof.** Clearly, all the paired-dominating sets  $Z_1$ ,  $Z_1^+$ ,  $Z_1^-$ ,  $T_1, \ldots, T_8$  can be constructed in O(k) time. Hence, the procedure certainly runs in O(k) time. Further, one can verify that all the possible combinations of conditions  $D_1, D_2, \ldots, D_5$  have been considered in cases  $C_1, C_2, \ldots, C_8$ . Hence, to prove the correctness of the procedure, it suffices to show that each step of the procedure is correct.

First, we consider cases  $C_1$  and  $C_2$ . In both of these cases, there exists an index  $\ell$  such that  $S_{\ell}^* = P(G_{\ell}, u_{\ell})$ . Therefore,  $Z_1, Z_1^+$ , and  $Z_1^-$  are dominating sets of H. It follows that, if r is even, then  $Z_1$  is a minimum-weight dominating set of H such that  $u_1 \notin Z_1$  and  $H[Z_1]$  has a perfect matching due to the selections of  $S_i^*$  for  $2 \leq i \leq k$ . So, we have  $S = Z_1$  for case  $C_2$ . On the other hand, if r is odd, then in order to satisfy the condition that  $H[Q_1]$  contains a perfect matching with minimum cost, we can either replace  $S_{\alpha}^*$  with  $D(G_{\alpha}, u_{\alpha})$ , or replace  $S_{\beta}^*$  with  $F(\{P(G_{\beta}, u_{\beta}), P'(G_{\beta}, u_{\beta}), \overline{P}(G_{\beta}, u_{\beta})\})$ . This implies that we have  $S = F(\{Z_1^+, Z_1^-\})$  for case  $C_1$ .

Next, we consider cases  $C_3, C_4$ , and  $C_5$ . Notice that in all three cases, there exists no index  $\ell$  such that  $S_{\ell}^* = P(G_{\ell}, u_{\ell})$  and r is an odd number. Moreover, for any paired-dominating set  $Q_1$  of H, we have either  $S_i^* = P'(G_i, u_i)$  for  $2 \le i \le k$  or  $V(B) \cap V(Q_1) \neq \emptyset$ , where  $B = H[\{u_1, u_2, \dots, u_k\}].$ In case  $C_3$ , a paired-dominating set  $T_8$  is created for the former. Meanwhile, paired-dominating sets  $Z_1^+, T_4$ , and  $T_6$  are built for the latter. As a consequence of r = 1, in order to ensure  $H[Q_1]$  contains a perfect matching when  $V(B) \cap V(Q_1) \neq \emptyset$ , we replace  $S^*_{\alpha}$  with  $D(G_{\alpha}, u_{\alpha})$  in  $Z^+_1$ , replace  $S^*_{\beta}$ with  $P(G_{\beta}, u_{\beta})$  in  $T_4$ , and replace  $S^*_{\gamma}$  and  $S^*_{\beta}$ with  $P(G_{\gamma}, u_{\gamma})$  and  $F(\{P'(G_{\beta}, u_{\beta}), \overline{P}(G_{\beta}, u_{\beta})\})$ in  $T_6$ , respectively. Under the premise of minimizing weight, one can verify that  $Z_1^+$ ,  $T_4$ , and  $T_6$  are exactly the three potential candidates for  $Q_1$ . In case  $C_4$ , we have  $S^*_{\beta} = D(G_{\beta}, u_{\beta})$  and  $S_i^* = P'(G_i, u_i)$  for  $2 \le i \le k$  and  $i \ne \beta$ . Using a similar method of the above arguments, one can show that  $S = F(\{Z_1^+, T_5, T_7\})$  is true for case  $C_4$ . In case  $C_5$ , we have  $r \geq 3$ . Therefore, for the same reasons as case  $C_1$ , we have  $S = F(\{Z_1^+, Z_1^-\})$  for case  $C_5$ .

Finally, we consider cases  $C_6, C_7$ , and  $C_8$ . Notice that in all three cases, there exists no index  $\ell$ such that  $S_{\ell}^* = P(G_{\ell}, u_{\ell})$  and r is an even number. In case  $C_6$ , we have either  $S_i^* = P'(G_i, u_i)$ or  $S_i^* = \bar{P}(G_i, u_i)$ , where  $1 \le i \le k$ . To ensure  $H[Q_1]$  contains a perfect matching, we replace  $S^*_{\gamma}$ with  $P(G_{\gamma}, u_{\gamma})$  in  $T_1$ , replace  $S^*_{\alpha}$  and  $S^*_{\alpha'}$  with  $D(G_{\alpha}, u_{\alpha})$  and  $D(G_{\alpha'}, u_{\alpha'})$  in  $T_2$ , and replace  $S_i^*$ with  $P'(G_i, u_i)$  for all  $i \in I$  in  $T_3$ , respectively. Under the premise of minimizing the weight w(S), one can verify that  $T_1$ ,  $T_2$ , and  $T_3$  are exactly the three potential candidates for  $Q_1$ . Notice that, in case  $C_7$ ,  $S_i^* = P'(G_i, u_i)$  for  $2 \le i \le k$ . Further,  $r \geq 2$  is an even number in case  $C_8$ . Thus, we have  $S = Z_1$  for the same reasons as case  $C_2$ . 

#### **3.3.2** Finding $Q_2$

In the following, we present a procedure to find the paired-dominating set  $Q_2$ . Similar to Procedure 4, the procedure solves the problem by considering six cases  $C_9, C_{10}, \ldots, C_{14}$ . For  $9 \le i \le 14$ , the case  $C_i = (c_1, c_2, c_3, c_4)$  is an ordered 4-tuple. Further, the value of  $c_k$  has the same definition as before for  $1 \le k \le 4$ . Then, we define  $C_9 = (1, 1, *, *), C_{10} = (1, 0, *, *), C_{11} = (0, 1, 1, *), C_{12} = (0, 1, 0, *), C_{13} = (0, 0, *, 1), and C_{14} = (0, 0, *, 0)$ . Again, one can verify that all the possible combinations of the four conditions have been considered in cases  $C_9, C_{10}, \ldots, C_{14}$ . The paired-dominating sets  $Z_2, Z_2^+, Z_2^-, T_9, \ldots, T_{12}$  of H are defined below, which

are the potential candidates for  $Q_2$ .

$$\begin{split} &Z_2 = \bar{P}(G_1, u_1) \cup S_2^* \cup S_3^* \cup \ldots \cup S_k^*. \\ &Z_2^+ = (Z_1 - S_\alpha^*) \cup D(G_\alpha, u_\alpha). \\ &Z_2^- = (Z_1 - S_\beta^*) \cup \\ & F(\{P(G_\beta, u_\beta), P'(G_\beta, u_\beta), \bar{P}(G_\beta, u_\beta)\}). \\ &T_9 = (Z_2 - S_\gamma^*) \cup P(G_\gamma, u_\gamma). \\ &T_{10} = (Z_2 - S_\alpha^* - S_\alpha^*) \cup D(G_\alpha, u_\alpha) \cup D(G_{\alpha'}, u_{\alpha'}). \\ &T_{11} = (Z_2 - S_\beta^*) \cup P(G_\beta, u_\beta). \\ &T_{12} = (Z_2 - S_\gamma^* - S_\beta^*) \cup P(G_\gamma, u_\gamma) \cup \\ & F(\{P'(G_\beta, u_\beta), \bar{P}(G_\beta, u_\beta)\}). \end{split}$$

Moreover, the relations between the cases  $C_9, C_{10}, \ldots, C_{14}$  and the paired-dominating sets  $Z_2, Z_2^+, Z_2^-, T_9, \ldots, T_{12}$  are detailed in the following procedure.

**Procedure 5** Finding  $Q_2$ 

- Input: Dominating sets  $D(G_i, u_i)$ ,  $P(G_i, u_i)$ ,  $P'(G_i, u_i)$ , and  $\overline{P}(G_i, u_i)$  for  $1 \le i \le k$ .
- **Output:** A minimum-weight dominating set S of H such that  $u_1 \notin S$ , H[S] has a perfect matching, and  $\overline{P}(G_1, u_1) \subseteq S$ .
- if C<sub>9</sub> or C<sub>12</sub> holds, then let S ← F({Z<sub>2</sub><sup>+</sup>, Z<sub>2</sub><sup>-</sup>});
  if C<sub>10</sub> or C<sub>14</sub> holds, then let S ← Z<sub>2</sub>;
  if C<sub>11</sub> holds, then let S ← F({Z<sub>2</sub><sup>+</sup>, T<sub>11</sub>, T<sub>12</sub>});
  if C<sub>13</sub> holds, then let S ← F({T<sub>9</sub>, T<sub>10</sub>});

5: return S.

**Lemma 5** Given the dominating sets  $D(G_i, u_i)$ ,  $P(G_i, u_i)$ ,  $P'(G_i, u_i)$ , and  $\overline{P}(G_i, u_i)$  for  $1 \le i \le k$ , Procedure 5 outputs a minimum-weight dominating set S of H such that  $u_1 \notin S$ , H[S] has a perfect matching, and  $\overline{P}(G_1, u_1) \subseteq S$ . Moreover, the procedure can be completed in O(k) time.

**Proof.** Clearly, each step of the procedure can be completed in O(k) time. Therefore, the procedure runs in O(k) time. Further, one can verify that all the possible combinations of conditions  $D_1, D_2, D_3$ , and  $D_4$  have been considered in cases  $C_9, C_{10}, \ldots, C_{14}$ . Hence, to prove the correctness of the procedure, it suffices to show that each step of the procedure is correct.

First, we consider cases  $C_9$  and  $C_{10}$ . In both of these cases, there exists an index  $\ell$  such that  $S_{\ell}^* = P(G_{\ell}, u_{\ell})$ . Therefore, for the same reasons as cases  $C_1$  and  $C_2$  in Procedure 4, we have  $S = F(\{Z_2^+, Z_2^-\})$  for case  $C_9$  and  $S = Z_2$  for case  $C_{10}$ , respectively. Next, we consider cases  $C_{11}$ , and  $C_{12}$ . Notice that in both cases, there exists no index  $\ell$  such that  $S_{\ell}^* = P(G_{\ell}, u_{\ell})$  and r is an odd number. Since we have r = 1 in case  $C_{11}$ , in order to satisfy the condition that  $H[Q_2]$  contains a perfect matching with minimum cost, we replace  $S^*_{\alpha}$ with  $D(G_{\alpha}, u_{\alpha})$  in  $Z_2^+$ , replace  $S^*_{\beta}$  with  $P(G_{\beta}, u_{\beta})$ in  $T_{11}$ , and replace  $S^*_{\gamma}$  and  $S^*_{\beta}$  with  $P(G_{\gamma}, u_{\gamma})$  and  $F(\{P'(G_{\beta}, u_{\beta}), \bar{P}(G_{\beta}, u_{\beta})\})$  in  $T_{12}$ , respectively. Under the premise of minimizing weight, one can verify that  $Z_2^+$ ,  $T_{11}$ , and  $T_{12}$  are exactly the three potential candidates for  $Q_2$ . In case  $C_{12}$ , we have  $r \geq 3$ . Therefore, for the same reasons as case  $C_9$ , we have  $S = F(\{Z_2^+, Z_2^-\})$  for case  $C_{12}$ .

Finally, we consider cases  $C_{13}$ , and  $C_{14}$ . Notice that in both cases, there exists no index  $\ell$  such that  $S_{\ell}^* = P(G_{\ell}, u_{\ell})$  and r is an even number. In case  $C_{13}$ , either  $S_i^* = P'(G_i, u_i)$  or  $S_i^* = \overline{P}(G_i, u_i)$ for  $1 \leq i \leq k$ . Therefore, to satisfy the condition that  $H[Q_2]$  contains a perfect matching, we replace  $S_{\gamma}^*$  with  $P(G_{\gamma}, u_{\gamma})$  in  $T_9$ , and replace  $S_{\alpha}^*$ and  $S_{\alpha'}^*$  with  $D(G_{\alpha}, u_{\alpha})$  and  $D(G_{\alpha'}, u_{\alpha'})$  in  $T_{10}$ , respectively. Again, under the premise of minimizing the weight, one can verify that  $T_9$  and  $T_{10}$ are exactly the two potential candidates for  $Q_2$ . Notice that, in case  $C_{14}, r \geq 2$  is an even number. Thus, we have  $S = Z_2$  for the same reasons as case  $C_2$  in Procedure 4.

Combining Lemmas 4 and 5, we obtain the following result.

**Lemma 6** Given the dominating sets  $D(G_i, u_i)$ ,  $P(G_i, u_i)$ ,  $P'(G_i, u_i)$ , and  $\overline{P}(G_i, u_i)$  for  $1 \le i \le k$ , a  $\kappa_3$ -paired-dominating set  $P'(H, u_1)$  can be determined in O(k) time.

#### **3.4** Determination of $\overline{P}(H, u_1)$

Remember that  $\overline{P}(H, u_1)$  is a minimum-weight dominating set of  $H - u_1$  and  $u_1$  is not dominated by  $\overline{P}(H, u_1)$ . Hence, by the definition of  $\overline{P}(H, u_1)$ , the only composition is  $\overline{P}(H, u_1) =$  $\overline{P}(G_1, u_1) \cup P'(G_2, u_2) \cup \ldots \cup P'(G_k, u_k)$ . This implies that, given the dominating sets  $D(G_i, u_i)$ ,  $P(G_i, u_i), P'(G_i, u_i)$ , and  $\overline{P}(G_i, u_i)$  for  $1 \le i \le k$ , a  $\kappa_4$ -paired-dominating set  $\overline{P}(H, u_1)$  can be determined in O(k) time. Thus, we have the following result.

**Lemma 7** Given the dominating sets  $D(G_i, u_i)$ ,  $P(G_i, u_i)$ ,  $P'(G_i, u_i)$ , and  $\overline{P}(G_i, u_i)$  for  $1 \le i \le k$ , a  $\kappa_4$ -paired-dominating set  $\overline{P}(H, u_1)$  can be determined in O(k) time.

#### 4 Conclusion and Future Work

In this paper, we have presented an optimal algorithm for finding a paired-dominating set of a block graph G. The algorithm uses a dynamicprogramming approach to iteratively determine D(H, u), P(H, u), P'(H, u), and  $\overline{P}(H, u)$  in a bottom-up manner, where H is a subgraph of Gand  $u \in V(H)$  is a cut vertex of G. When the graph is given in an adjacency list representation, our algorithm runs in O(n + m) time. Moreover, the algorithm can be completed in O(n) time if the block-cut-vertex structure of G is given.

Below we present some open problems related to the paired-domination problem. It is known that distance-hereditary graphs is a proper superfamily of block graphs. Therefore, it is interesting to study the time complexity of paired-domination problem in distance-hereditary graphs. In [3], Chen *et al.* proposed an approximation algorithm with ratio  $\ln(2\Delta(G)) + 1$  for general graphs and showed that the problem is APX-complete, i.e., has no PTAS. Thus, it would be useful if we could develop an approximation algorithm for general graphs with constant ratio. Meanwhile, it would be desirable to show that the problem remains NPcomplete in planar graphs and design an approximation algorithm.

#### References

- A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The design and analysis of computer algorithms*. Addison-Wesley Publishing Co., Reading, Mass.-London-Amsterdam, 1974.
- [2] G. J. Chang. Algorithmic aspects of domination in graphs. In *Handbook of Combinatorial Optimization*, pages 339–405. Springer-Verlag, New York, second edition, 2013.
- [3] L. Chen, C. Lu, and Z. Zeng. Hardness results and approximation algorithms for (weighted) paireddomination graphs. *Theoret. Comput. Sci.*, 410(47-49):5063–5071, 2009.
- [4] L. Chen, C. Lu, and Z. Zeng. A linear-time algorithm for paired-domination problem in strongly chordal graphs. *Inform. Process. Lett.*, 110(1):20– 23, 2009.
- [5] L. Chen, C. Lu, and Z. Zeng. Labelling algorithms for paired-domination problems in block and interval graphs. J. Comb. Optim., 19(4):457–470, 2010.
- [6] T. C. E. Cheng, L. Kang, and E. Shan. A polynomial-time algorithm for the paired-

domination problem on permutation graphs. Discrete Appl. Math., 157(2):262–271, 2009.

- [7] T. C. E. Cheng, L. Y. Kang, and C. T. Ng. Paired domination on interval and circular-arc graphs. *Discrete Appl. Math.*, 155(16):2077–2086, 2007.
- [8] W. Goddard and M. A. Henning. Independent domination in graphs: a survey and recent results. *Discrete Math.*, 313(7):839–854, 2013.
- [9] T. Haynes and P. Slater. Paired-domination in graphs. *Networks*, 32:199–206, 1998.
- [10] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. Domination in Graphs: Advanced Topics. Marcel Dekker Inc., New York, 1998.
- [11] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. Fundamentals of Domination in Graphs. Marcel Dekker Inc., New York, 1998.
- [12] S. T. Hedetniemi and R. C. Laskar. Bibliography on domination in graphs and some basic definitions of domination parameters. *Discrete Math.*, 86(1-3):257–277, 1990.
- [13] S. T. Hedetniemi and R. C. Laskar, editors. *Top*ics on domination. Annals of Discrete Mathematics. North-Holland Publishing Co., Amsterdam, 1991.
- [14] M. A. Henning. A survey of selected recent results on total domination in graphs. *Discrete Math.*, 309(1):32–63, 2009.
- [15] L. Kang. Variations of dominating set problem. In Handbook of Combinatorial Optimization, pages 3363–3394. Springer-Verlag, New York, second edition, 2013.
- [16] L. Kang, M. Y. Sohn, and T. C. E. Cheng. Paireddomination in inflated graphs. *Theor. Comput. Sci.*, 320(2-3):485–494, 2004.
- [17] E. Lappas, S. D. Nikolopoulos, and L. Palios. An O(n)-time algorithm for the paired domination problem on permutation graphs. *European* J. Combin., 34(3):593–608, 2013.
- [18] B. S. Panda and D. Pradhan. Minimum paireddominating set in chordal bipartite graphs and perfect elimination bipartite graphs. J. Comb. Optim., 26(4):770–785, 2013.
- [19] H. Qiao, L. Kang, M. Cardei, and D.-Z. Du. Paired-domination of trees. J. of Global Optimization, 25(1):43–54, 2003.
- [20] C.-C. Yen and R. C. T. Lee. The weighted perfect domination problem and its variants. *Discrete Appl. Math.*, 66(2):147–160, 1996.