Proceedings of International Conference
on Algorithms

On the Complexity of the Perfect Edge Domination Problem

Chin Lung Lu　　Chuan Yi Tang

Department of Computer Science
National Tsing Hua University
{cllu, cytang}@cs.nthu.edu.tw

Abstract

A perfect edge dominating set of $G = (V, E)$ is a subset D of E such that every edge not in D is dominated by exactly one edge in D. The perfect edge domination problem is to find a perfect edge dominating set with the minimum cardinality in G. In this paper, we show that the perfect edge domination problem is NP-complete on bipartite graphs. Moreover, we present linear-time algorithms for solving the perfect edge domination problem on trees and series-parallel graphs.

1. Introduction

Let $G = (V, E)$ be a *simple graph*, i.e., finite, undirected, and loopless graph without multiple edge. Denote n and m to be the number of vertices and edges, respectively. An edge $(u, v) \in E$ is said to *dominate* itself and any edge that has u or v as a vertex. A *perfect edge dominating set* of G is a subset $D \subseteq E$ such that every edge not in D is dominated by exactly one edge in D. A perfect edge dominating set D is *independent* if no two edges in D are adjacent. The (*independent*) *perfect edge domination problem* is to find an arbitrary (independent) perfect edge dominating set with the minimum cardinality in G. This minimum cardinality is called the (*independent*) *perfect edge domination number* of G. Denote $\delta(G)$ to be the perfect edge domination number of G.

Perfect edge domination problem is a variant of the edge domination problem, which has been extensively studied [12, 16, 8, 6, 4, 14], and has many interesting applications. The resource allocation problem in parallel processing system can be modeled as the independent perfect (vertex) domination problem [9, 15, 17, 18]. For example, a parallel processing system can be modeled by a graph $G = (V, E)$, where each vertex $u \in V$ represents a processing element and each edge $(u, v) \in E$ represents a direct communication link between the processing elements corresponding to u and v. Suppose that there are limited resources such as power sources, disks, I/O connections or software modules. It is desirable to allocate a minimum number of these resource units at the processing elements in such a way that every processing element has at most one resource unit and is within a distance one of exactly one resource unit. The solution of this problem represents an optimal situation in which there is neither duplication nor overlap. This problem can be also modeled as an (independent) perfect edge dominating problem.

Another application is related to the problem of finding a minimum set S of 1's in M such that any other 1 of M is in the same row or column with exactly an element of S, where M is a $p \times q$ (0,1)-matrix (i.e., each entry of M is either 0 or 1). Let us construct a bipartite graph $G = (A, B, E)$ by corresponding to every row a vertex in A, to every column a vertex in B and connecting a vertex in A to a vertex in B by an edge if and only if M has a one at the intersection of the corresponding row and column. It is easy to see that a minimum set S of 1's in M corresponds to a minimum perfect edge dominating set of $G = (A, B, E)$ and vice versa.

The concept of independent perfect edge domination in this paper is the same concept as efficient edge domination, defined by Grinstead et al. [3]. Grinstead et al. [3] proved that the efficient edge domination problem is NP-complete for general graphs and presented linear-time algorithms for computing the maximum number of edges that can be efficiently dominated on trees and series-parallel graphs. Pal [13] et al. proposed a linear-time algorithm for calculating an edge-packing with the maximum weight on interval graphs. Lu et al. showed that the efficient edge domination problem is NP-complete on bipartite graphs [10] and later gave an $O(n+\Delta m)$ time algorithm for solving the weighted efficient edge domination problem on bipartite permutation graphs [11], where Δ is the maximum degree of vertex in G.

In this paper, we show that the perfect edge domination problem is NP-complete on bipartite graphs in Section 2. Meanwhile, we also prove that the perfect (vertex) domination problem is NP-complete when G is restricted to the class of the line graphs of bipartite graphs, or, equivalently, the perfect claw-free graphs. In Sections 3 and 4, we present linear-time algorithms, which are optimal, for solving the perfect edge domination problem on trees and series-parallel graphs, respectively.

2. NP-completeness for bipartite graphs

In this section, we shall reduce the *exact cover problem*, which is known to be NP-complete [2], to the problem of determining whether there exists a perfect edge dominating set on a bipartite graph.

Exact Cover Problem

Instance: A family of sets $F = \{S_1, S_2, \ldots, S_n\}$.

Question: Does F contain an exact cover, i.e., a subfamily of pairwise disjoint sets whose union is equal to X, where $X = \bigcup_{1\le j\le n} S_j$?

Theorem 2.1 *The problem of determining whether there exists a perfect edge dominating set on a bipartite graph is NP-complete.*

Proof: Obviously, this problem is in NP. In the following, we show that the exact cover problem is polynomially reducible to this problem. Given an instance F of the exact cover problem, we construct a bipartite graph $G = (V, E)$ as follows. Let $F = \{S_1, S_2, \cdots, S_n\}$ and $X = \{x_1, x_2, \cdots, x_m\}$, where $X = \bigcup_{1\le j\le m} S_j$. At first, each element $x_i \in X$, where $1 \le i \le m$, is a vertex of G and each set $S_j \in F$, where $1 \le j \le n$, is also a vertex of G. There is a path of length two, say x_i—y_{ij}—S_j, between vertices x_i and S_j in G if and only if $x_i \in S_j$. Then, for each vertex S_j of G, we attach a path of length two, say S_j—a_j—b_j, at S_j. Furthermore, we add three vertices u, v and w to G in such a way that $(w, v) \in E$, $(v, u) \in E$ and all vertices x_i's, where $1 \le i \le m$, are adjacent to u. Finally, we add vertices $r_1, r_2, \cdots, r_{m+n}$, $z_1, z_2, \cdots, z_{m+n+1}$ to G such that vertices $r_1, r_2, \cdots, r_{m+n}$ are adjacent to v and vertices $z_1, z_2, \cdots, z_{m+n+1}$ are adjacent to w. More precisely,

$$\begin{aligned} V &= \{S_j, a_j, b_j | 1 \le j \le n\} \cup \{x_i | 1 \le i \le m\} \\ &\quad \cup \{y_{ij} | 1 \le i \le m, 1 \le j \le n \text{ and } x_i \in S_j\} \\ &\quad \cup \{w, v, u\} \cup \{r_k | 1 \le k \le m+n\} \\ &\quad \cup \{z_k | 1 \le k \le m+n+1\}, \\ E &= \{(S_j, a_j), (a_j, b_j) | 1 \le j \le n\} \\ &\quad \cup \{(x_i, y_{ij}) | 1 \le i \le m, 1 \le j \le n \text{ and } x_i \in S_j\} \\ &\quad \cup \{(y_{ij}, S_j) | 1 \le i \le m, 1 \le j \le n \text{ and } x_i \in S_j\} \\ &\quad \cup \{(w, v), (v, u)\} \cup \{(v, r_k) | 1 \le k \le m+n\} \\ &\quad \cup \{(w, z_k) | 1 \le k \le m+n+1\}. \end{aligned}$$

See Figure 1, for example, where $F = \{S_1, S_2, S_3\} = \{(x_1, x_3), (x_2, x_3, x_4), (x_2, x_4)\}$. Suppose that D is a perfect edge domination set of G. Let $U = \{(u, x_i) \in E | 1 \le i \le m\}$, $R = \{(v, r_k) \in E | 1 \le k \le m+n\}$ and $Z = \{(w, z_k) \in E | 1 \le k \le m+n+1\}$. Then, we claim that $\delta(G) \ge m+n+1$. Consider the following two cases.

Case 1: $Z \cap D \ne \emptyset$. Then, either $Z \cup \{(w, v)\} \subseteq D$ or $|Z \cap D| = 1$ by the definition of perfect edge domination. In the former case, we clearly have $|D| \ge m+n+2$. We consider the latter case in the following. Suppose $|Z \cap D| = 1$ and let $e \in Z \cap D$. Suppose that $(w, v) \in D$. Since $|Z \cap D| = 1$ and $|Z| \ge 3$, there is

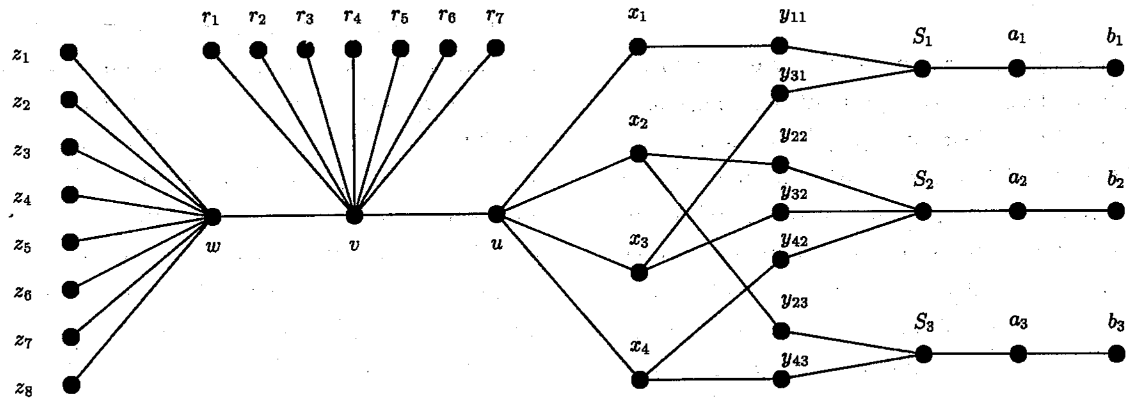

Figure 1: Bipartite graph G for $F = \{S_1, S_2, S_3\} = \{(x_1, x_3), (x_2, x_3, x_4), (x_2, x_4)\}$.

an edge e' in Z such that $e' \notin D$ and e' is dominated by edges e and (w, v) in D, a contradiction. Suppose that no edge of $R \cup \{(v, u)\}$ is in D. Then, all edges of R not in D are not dominated by any edge in D, a contradiction. Hence, there is exactly one edge e'' in $R \cup \{(v, u)\}$ such that $e'' \in D$. As a result, (w, v) not in D is dominated by two edges e and e'' in D, a contradiction. Therefore, $|Z \cap D| \neq 1$.

Case 2: $Z \cap D = \varnothing$. Clearly, (w, v) is in D to exactly dominate those edges in Z. Then, we have either $R \cup \{(v, u)\} \subseteq D$ or $(R \cup \{(v, u)\}) \cap D = \varnothing$. In the former case, we have $|D| \geq m + n + 2$. We consider the latter case in the following. Suppose that $U \cap D \neq \varnothing$ and let $e \in U \cap D$. Then, (v, u) not in D is dominated by edges (w, v) and e in D, a contradiction. For each vertex x_i of G, where $1 \leq i \leq m$, since (u, x_i) is not in D, there is exactly one edge e' in $\{(x_i, y_{ik}) \in E | 1 \leq k \leq n\}$ such that $e' \in D$. For each vertex S_j of G, where $1 \leq j \leq n$, suppose that $D \cap \{(S_j, a_j), (a_j, b_j)\} = \varnothing$. Then, (a_j, b_j) not in D is not dominated by any edge in D, a contradiction. Hence, D contains at least one edge in $\{(S_j, a_j), (a_j, b_j)\}$. Therefore, $|D| \geq m+n+1$.

As mentioned two cases above, we have $\delta(G) \geq m + n + 1$. In the case of $\delta(G) = m + n + 1$, an optimal solution must include (w, v), exactly one edge in $\{(x_i, y_{ik}) \in E | 1 \leq k \leq n\}$ for $1 \leq i \leq m$ and exactly one edge in $\{(S_j, a_j), (a_j, b_j)\}$ for $1 \leq j \leq n$.

Next, We claim that the exact cover problem has a positive answer (i.e., F has an exact cover F') if and only if $\delta(G) = m + n + 1$. First, suppose that F has an exact cover F'. Define $B \subseteq E$ as follows.

$$\begin{aligned} B &= \{(x_i, y_{ij}), (a_j, b_j) | S_j \in F' \text{ and } x_i \in S_j\} \\ &\quad \cup \{(S_j, a_j) | S_j \notin F'\} \cup \{(w, v)\}. \end{aligned}$$

It is easy to verify that B is a perfect edge dominating set of G with $\delta(G) = m + n + 1$. Conversely, suppose that $\delta(G) = m + n + 1$, i.e., there is a perfect edge dominating set D^* of size $m + n + 1$ in G. As mentioned above, D^* contains (w, v), exactly one edge in $\{(x_i, y_{ik}) \in E | 1 \leq k \leq n\}$ for $1 \leq i \leq m$ and exactly one edge in $\{(S_j, a_j), (a_j, b_j)\}$ for $1 \leq j \leq n$. Consider vertex x_i of G, where $1 \leq i \leq m$. Let $(x_i, y_{ij'}) \in D^*$. We claim that $B' \subseteq D^*$, where $B' = \{(x_k, y_{kj'}) \in E | 1 \leq k \leq m\}$. Suppose that $(x_{i'}, y_{i'j'}) \in B'$ and $(x_{i'}, y_{i'j'}) \notin D^*$. Then, we distinguish the following two cases.

Case 1: $(y_{ij'}, S_{j'}) \in D^*$. Observe that exactly one edge of $\{(S_{j'}, a_{j'}), (a_{j'}, b_{j'})\}$ is in D^*. Suppose that $(S_{j'}, a_{j'}) \notin D^*$ and $(a_{j'}, b_{j'}) \in D^*$. Then, $(S_{j'}, a_{j'})$ not in D^* is dominated by edges $(y_{ij'}, S_{j'})$ and $(a_{j'}, b_{j'})$ in D^*, a contradiction. Suppose that $(y_{i'j'}, S_{j'}) \notin D^*$. Then, $(y_{i'j'}, S_{j'})$ not in D^* is dominated by edges $(y_{ij'}, S_{j'})$ and $(S_{j'}, a_{j'})$ in D^*, a contradiction. Note that there is exactly an edge $e \neq (x_{i'}, y_{i'j'})$ in $\{(x_{i'}, y_{i'k}) \in E | 1 \leq k \leq n\}$ such that $e \in D^*$. As a result, $(x_{i'}, y_{i'j'})$ not in D^* is dominated by edges e and $(y_{i'j'}, S_{j'})$ in D^*, a contradiction.

Case 2: $(y_{ij'}, S_{j'}) \notin D^*$. Since $(y_{ij'}, S_{j'})$ not in D^* is dominated by edge $(x_i, y_{ij'})$, no edge in

$\{(y_{kj'}, S_{j'}) \in E | 1 \leq k \leq m$ and $k \neq i\} \cup \{(S_{j'}, a_{j'})\}$ belongs to D^*. As a result, $(y_{i'j'}, S_{j'})$ not in D^* is not dominated by any edge of D^*, a contradiction.

Let F' be defined by $S_j \in F'$ if and only if $(a_j, b_j) \in D^*$, where $1 \leq j \leq m$. Clearly, F' is a subfamily of pairwise disjoint sets whose union is equal to X. In other words, F' is an exact cover. ■

Theorem 2.2 *The perfect edge domination problem is NP-complete on bipartite graphs.*

It is easy to verify that a perfect edge dominating set of $G = (V, E)$ is a perfect (vertex) dominating set in $L(G) = (V', E')$, where $L(G)$ is the *line graph* of G with $V' = E$ and $E' = \{(e, f) | e$ and f are adjacent edges of $E\}$. However, not all line graphs of bipartite graphs are bipartite graphs (see Figure 2). Because we have proved that the perfect edge domination problem is NP-complete on bipartite graphs, it follows that the perfect (vertex) domination problem remains NP-complete even when G is restricted to the class of the line graphs of bipartite graphs. Observe that the line graph of a bipartite graph is both perfect and claw-free [5].

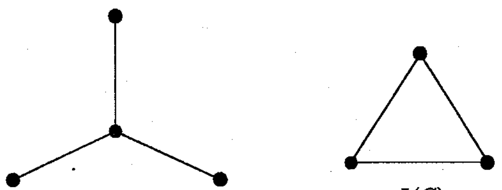

Figure 2: G is a bipartite (tree) graph, but $L(G)$ is not.

Corollary 2.1 *The perfect (vertex) domination problem is NP-complete when G is restricted to the class of the line graphs of bipartite graphs.*

Corollary 2.2 *The perfect (vertex) domination problem is NP-complete on perfect claw-free graphs.*

3. The algorithm for trees

The previous section has shown that it is hard to find a minimum perfect edge dominating set on bipartite graphs. In this section, however, we shall use the technique of dynamic programming to design a linear-time algorithm for solving the perfect edge domination problem on trees. Let $T = (V, E)$ be a rooted tree with root r, which abbreviated to (T, r). For any two rooted trees (T_1, r_1) and (T_2, r_2), define the *composition* of T_1 and T_2 to be a rooted tree (T, r_1) by adding an edge (r_1, r_2) to disjoint union T_1 and T_2 (see Figure 3). Note that a tree can be obtained from *trivial* graphs (i.e., graphs with just one vertex) by a sequence of tree compositions. For a rooted tree (T, r), we define the following notation for computing the minimum perfect edge dominating set in T.

- δ_0*-perfect edge dominating set* = a perfect edge dominating set D of T and no edge in D is incident with r.
- δ_1*-perfect edge dominating set* = a perfect edge dominating set D of T and exactly one edge in D is incident with r.
- δ_2*-perfect edge dominating set* = a perfect edge dominating set D of T and all edges in D are incident with r.
- δ_3*-perfect edge dominating set* = a perfect edge dominating set D of forest $T - r$, which obtained by removing r from T, and no edge in D is incident with r or any neighbor of r.
- $\delta_i(T, r)$ is the minimum size of δ_i-perfect edge dominating set of T, where $0 \leq i \leq 3$.

It is clear that $\min\{\delta_0(T, r), \delta_1(T, r), \delta_2(T, r)\}$ is the perfect edge dominating number $\delta(T)$ of T according to the definition. For any trivial rooted tree (T, r), the

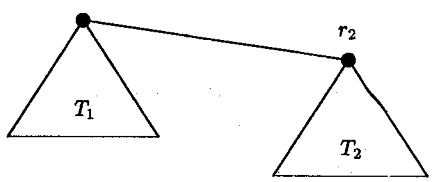

Figure 3: The composition of (T_1, r_1) and (T_2, r_2).

values of $\delta_i(T,r)$, where $0 \leq i \leq 3$, are initialized as follows.

$$\delta_0(T,r) = \delta_2(T,r) = \delta_3(T,r) = 0 \text{ and } \delta_1(T,r) = \infty.$$

Lemma 3.1 *Let* (T, r_1) *be the composition of rooted trees* (T_1, r_1) *and* (T_2, r_2). *Then,*

$$\delta_0(T,r_1) = \delta_0(T_1,r_1) + \delta_1(T_2,r_2).$$

Lemma 3.2 *Let* (T, r_1) *be the composition of rooted trees* (T_1, r_1) *and* (T_2, r_2). *Then,*

$$\delta_1(T,r_1) = \min\{\delta_1(T_1,r_1) + \delta_0(T_2,r_2), 1 + \delta_3(T_1,r_1) + \min\{\delta_2(T_2,r_2), \delta_3(T_2,r_2)\}\}.$$

Lemma 3.3 *Let* (T, r_1) *be the composition of rooted trees* (T_1, r_1) *and* (T_2, r_2). *Then,*

$$\delta_2(T,r_1) = 1 + \delta_2(T_1,r_1) + \min\{\delta_2(T_2,r_2), \delta_3(T_2,r_2)\}\}.$$

Lemma 3.4 *Let* (T, r_1) *be the composition of rooted trees* (T_1, r_1) *and* (T_2, r_2). *Then,*

$$\delta_3(T,r_1) = \delta_3(T_1,r_1) + \delta_0(T_2,r_2).$$

Based on the recursive functions of lemmas in this section, we design Algorithm *PEDP-T* to calculate the perfect edge domination number $\delta(T)$ of T using the technique of dynamic programming. Algorithm *PEDP-T* starts from the leaves of T and works inward to r. Reaching at vertex v, Algorithm *PEDP-T* computes all $\delta_i(u)$, where $0 \leq i \leq 3$ and u is the parent of v, according to Lemma 3.1, 3.2, 3.3 and 3.4, respectively. The detail of Algorithm *PEDP-T* is shown as follows. For convenience, notation $\delta_i(T,r)$, where $0 \leq i \leq 3$, is abbreviated to $\delta_i(r)$.

Since each vertex v of T is considered once and the computation of $\delta_i(u)$, where $0 \leq i \leq 3$, in Step 2 is done in constant time, the total time complexity of Algorithm *PEDP-T* is $O(n)$. With a slight modification, Algorithm *PEDP-T* cannot only compute $\delta(T)$, but also find the corresponding minimum perfect edge dominating set. Therefore, we have the following theorem.

Algorithm *PEDP-T*.
Input: A rooted tree T with root r.
Output: The perfect edge domination number $\delta(T)$.
1. /* **Initialization** */
  for each vertex v of T **do**
    $\delta_0(v) = \delta_2(v) = \delta_3(v) = 0$ and $\delta_1(v) = \infty$.
2. $T' = T$.
  while T' has more than one vertex **do**
    Choose a leave v of T' **do**.
    /* let u be the parent of v */
      $\delta_0(u) = \delta_0(u) + \delta_1(v)$.
      $\delta_1(u) = \min\{\delta_1(u) + \delta_0(v), 1 + \delta_3(u) + \min\{\delta_2(v), \delta_3(v)\}\}$.
      $\delta_2(u) = 1 + \delta_2(u) + \min\{\delta_2(v), \delta_3(v)\}\}$.
      $\delta_3(u) = \delta_3(u) + \delta_0(v)$.
      $T' = T' - v$.
3. Output $\min\{\delta_0(r), \delta_1(r), \delta_2(r)\}$.

Theorem 3.1 *The perfect edge domination problem can be solved in linear time on trees.*

4. The algorithm for series-parallel graphs

In this section, we shall present a linear-time algorithm for solving the perfect edge domination problem on (two-terminals) series-parallel graphs. Each series-parallel graph has two distinct vertices u and v to serve as its *left terminal* and *right terminal* respectively, and can be denoted by $(G,(u,v))$. A *series-parallel graph* is recursively defined as follows.

(1) The complete graph K_2 with two vertices u and v is a series-parallel graph $(K_2,(u,v))$.

(2) Let $(G_1,(u_1,v_1))$ and $(G_2,(u_2,v_2))$ be series-parallel graphs. Then, the graph G obtained by performing one of the following two operations on G_1 and G_2 is a series-parallel graph.

- *Series composition*: identify v_1 of G_1 with u_2 of G_2 to obtain $(G,(u_1,v_2))$ (see Figure 4(a)).
- *Parallel composition*: identify u_1 of G_1 with $u2$ of G_2 and v_1 of G_1 with v_2 of G_2 to obtain $(G,(u_1,v_1))$, or equivalently $(G,(u_2,v_2))$ (see Figure 4(b)). It is assumed

that no multiple edges will be created by this composition.

(3) Only graphs constructed by a finite number of applications of series and parallel compositions are series-parallel graphs.

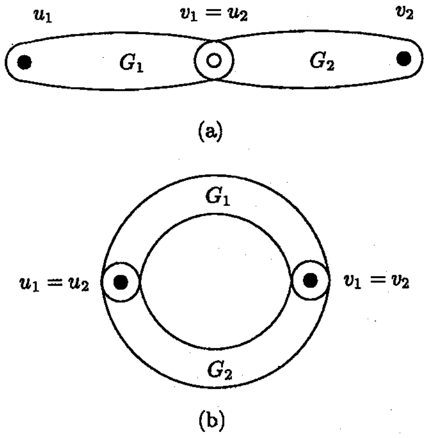

Figure 4: (a) Series composition. (b) Parallel composition.

Note that the class of series-parallel graphs is a subclass of planar graphs. Kikuno et al. [7] gave a linear-time algorithm to recognize whether a graph G is a series-parallel graph and constructed a parsing tree of G if so. A *parsing tree* T of a series-parallel graph $(G,(u,v))$ is defined as a binary tree in which each node of T represents a subgraph $(H,(u',v'))$ of G and labeled by (u',v'). Each leaf of T corresponds to an edge in G. Each internal node of T represents the subgraph of G obtained by applying a series or parallel composition to the subgraphs corresponding to its children. The root of T represents G itself. Figure 5 shows a series-parallel graph and its parsing tree. Note that the parsing tree of a series-parallel graph may be not unique. For $\alpha,\beta \in \{0,1,2,3\}$, we define (α,β)-*perfect edge dominating set* of a series-parallel graph $(G,(u,v))$ is a perfect edge dominating set D of

$$\begin{cases} G & \text{if } \alpha \neq 3 \text{ and } \beta \neq 3, \\ G\setminus\{u\} & \text{if } \alpha = 3 \text{ and } \beta \neq 3, \\ G\setminus\{v\} & \text{if } \alpha \neq 3 \text{ and } \beta = 3, \\ G\setminus\{u,v\} & \text{if } \alpha = 3 \text{ and } \beta = 3, \end{cases}$$

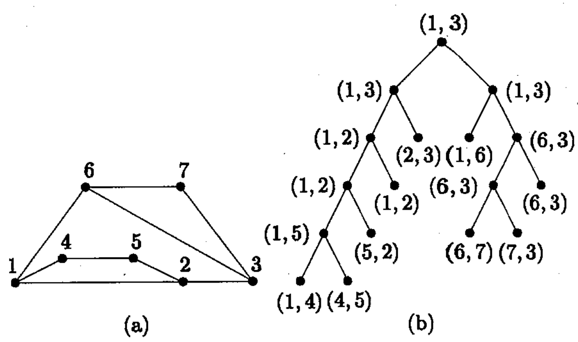

Figure 5: (a) A series-parallel graph G. (b) A parsing tree of G.

such that

$$\begin{cases} \text{no edge in } D \text{ is incident with } u \text{ if } \alpha = 0, \\ \text{exactly one edge in } D \text{ is incident with } u \text{ if } \alpha = 1, \\ D \text{ contains all edges which are incident with } u \\ \quad \text{if } \alpha = 2, \\ \text{no edge in } D \text{ is incident with } u \text{ or its neighbors} \\ \quad \text{if } \alpha = 3, \end{cases}$$

and

$$\begin{cases} \text{no edge in } D \text{ is incident with } v \text{ if } \beta = 0, \\ \text{exactly one edge in } D \text{ is incident with } v \text{ if } \beta = 1, \\ D \text{ contains all edges which are incident with } v \\ \quad \text{if } \beta = 2, \\ \text{no edge in } D \text{ is incident with } v \text{ or its neighbors} \\ \quad \text{if } \beta = 3. \end{cases}$$

The minimum size of an (α,β)-perfect edge dominating set of a series-parallel graph $(G,(u,v))$ is denoted by $\delta(G,u^\alpha,v^\beta)$. According to the definition of perfect edge domination, it can be easily verified that $\min\{\delta(G,u^\alpha,v^\beta)|\alpha,\beta \in \{0,1,2\}\}$ is the perfect edge domination number $\delta(G)$ of $(G,(u,v))$. For the graph $(K_2,(u,v))$, the values of $\delta(K_2,u^\alpha,v^\beta)$, where $\alpha,\beta \in \{0,1,2,3\}$, are initialized as follows.

- $\delta(K_2,u^1,v^1) = \delta(K_2,u^1,v^2) = \delta(K_2,u^2,v^1) = \delta(K_2,u^2,v^2) = 1$,
- $\delta(K_2,u^0,v^3) = \delta(K_2,u^3,v^0) = 0$,
- The values of all other cases are ∞.

Lemma 4.1 *Let $(G,(u_1,v_2))$ be a series-parallel graph obtained by applying a series composition to $(G_1,(u_1,v_1))$ and $(G_2,(u_2,v_2))$. Then, for any $\alpha,\beta \in \{0,1,2,3\}$, we have*

$$\delta(G,u_1^\alpha,v_2^\beta) = \min\left\{\begin{array}{l} \delta(G_1,u_1^\alpha,v_1^0)+\delta(G_2,u_2^0,v_2^\beta), \\ \delta(G_1,u_1^\alpha,v_1^3)+\delta(G_2,u_2^1,v_2^\beta), \\ \delta(G_1,u_1^\alpha,v_1^1)+\delta(G_2,u_2^3,v_2^\beta), \\ \delta(G_1,u_1^\alpha,v_1^2)+\delta(G_2,u_2^2,v_2^\beta) \end{array}\right\}$$

Lemma 4.2 *Let $(G,(u_1,v_1))$ be a series-parallel graph obtained by applying a parallel composition to $(G_1,(u_1,v_1))$ and $(G_2,(u_2,v_2))$. Then, we have*

(1)

$$\delta(G,u_1^0,v_1^0) = \delta(G_1,u_1^0,v_1^0)+\delta(G_2,u_2^0,v_2^0),$$

$$\delta(G,u_1^0,v_1^1) = \min\left\{\begin{array}{l} \delta(G_1,u_1^0,v_1^3)+\delta(G_2,u_2^0,v_2^1), \\ \delta(G_1,u_1^0,v_1^1)+\delta(G_2,u_2^0,v_2^3) \end{array}\right\},$$

$$\delta(G,u_1^0,v_1^2) = \delta(G_1,u_1^0,v_1^2)+\delta(G_2,u_2^0,v_2^2),$$

$$\delta(G,u_1^0,v_1^3) = \delta(G_1,u_1^0,v_1^3)+\delta(G_2,u_2^0,v_2^3).$$

(2)

$$\delta(G,u_1^1,v_1^0) = \min\left\{\begin{array}{l} \delta(G_1,u_1^1,v_1^0)+\delta(G_2,u_2^3,v_2^0), \\ \delta(G_1,u_1^3,v_1^0)+\delta(G_2,u_2^1,v_2^0) \end{array}\right\},$$

$$\delta(G,u_1^1,v_1^1) = \min\left\{\begin{array}{l} \delta(G_1,u_1^1,v_1^1)+\delta(G_2,u_2^3,v_2^3), \\ \delta(G_1,u_1^3,v_1^1)+\delta(G_2,u_2^1,v_2^3), \\ \delta(G_1,u_1^1,v_1^3)+\delta(G_2,u_2^3,v_2^1), \\ \delta(G_1,u_1^3,v_1^3)+\delta(G_2,u_2^1,v_2^1) \end{array}\right\},$$

$$\delta(G,u_1^1,v_1^2) = \min\left\{\begin{array}{l} \delta(G_1,u_1^1,v_1^2)+\delta(G_2,u_2^3,v_2^2), \\ \delta(G_1,u_1^3,v_1^2)+\delta(G_2,u_2^1,v_2^2) \end{array}\right\},$$

$$\delta(G,u_1^1,v_1^3) = \min\left\{\begin{array}{l} \delta(G_1,u_1^1,v_1^3)+\delta(G_2,u_2^3,v_2^3), \\ \delta(G_1,u_1^3,v_1^3)+\delta(G_2,u_2^1,v_2^3) \end{array}\right\}.$$

(3)

$$\delta(G,u_1^2,v_1^0) = \delta(G_1,u_1^2,v_1^0)+\delta(G_2,u_2^2,v_2^0),$$

$$\delta(G,u_1^2,v_1^1) = \min\left\{\begin{array}{l} \delta(G_1,u_1^2,v_1^1)+\delta(G_2,u_2^2,v_2^3), \\ \delta(G_1,u_1^2,v_1^3)+\delta(G_2,u_2^2,v_2^1) \end{array}\right\},$$

$$\delta(G,u_1^2,v_1^2) = \delta(G_1,u_1^2,v_1^2)+\delta(G_2,u_2^2,v_2^2),$$

$$\delta(G,u_1^2,v_1^3) = \delta(G_1,u_1^2,v_1^3)+\delta(G_2,u_2^2,v_2^3).$$

(4)

$$\delta(G,u_1^3,v_1^0) = \delta(G_1,u_1^3,v_1^0)+\delta(G_2,u_2^3,v_2^0),$$

$$\delta(G,u_1^3,v_1^1) = \min\left\{\begin{array}{l} \delta(G_1,u_1^3,v_1^1)+\delta(G_2,u_2^3,v_2^3), \\ \delta(G_1,u_1^3,v_1^3)+\delta(G_2,u_2^3,v_2^1) \end{array}\right\},$$

$$\delta(G,u_1^3,v_1^2) = \delta(G_1,u_1^3,v_1^2)+\delta(G_2,u_2^3,v_2^2),$$

$$\delta(G,u_1^3,v_1^3) = \delta(G_1,u_1^3,v_1^3)+\delta(G_2,u_2^3,v_2^3).$$

Based on the lemmas in this section, we design Algorithm *PEDP-SP* to calculate $\delta(G)$ of a series-parallel graph G using the technique of dynamic programming. Algorithm *PEDP-SP* starts from the leaves of a parsing tree T of G and works inward to root of T. Reaching at node (u,v), Algorithm *PEDP-SP* computes all $\delta(H,u^\alpha,v^\beta)$ according to Lemma 4.1 or 4.2, where H corresponds to the subgraph of G obtained by applying a series or parallel composition to the subgraphs corresponding to children of (u,v). The detail of Algorithm *PEDP-SP* is shown as follows.

Algorithm *PEDP-SP*.
Input: A series-parallel graph $(G,(t_1,t_2))$.
Output: The perfect edge domination number $\delta(G)$.
1. Construct a parsing tree T of G.
2. /* **Initialization** */
  for each leave (u,v) of T **do**
    $\delta(K_2,u^1,v^1)=\delta(K_2,u^1,v^2)=\delta(K_2,u^2,v^1)=\delta(K_2,u^2,v^2)=1$,
    $\delta(K_2,u^0,v^3)=\delta(K_2,u^3,v^0)=0$, and
    the values of other cases are ∞.
    mark leave (u,v).
3. **while** all nodes of T are not marked **do**
    choose an unmarked node (u,v) of T whose children are marked **do**
      case 1: Suppose that node (u,v) corresponds to the subgraph H of G obtained by applying a series composition to the subgraphs corresponding to its children. Then, compute all $\delta(H,u^\alpha,v^\beta)$, where $\alpha,\beta \in \{0,1,2,3\}$, according to Lemma 4.1.
      case 2: Suppose that node (u,v) corresponds to the subgraph H of G obtained by applying a parallel composition to the subgraphs corresponding to its children. Then, compute all $\delta(H,u^\alpha,v^\beta)$, where $\alpha,\beta \in \{0,1,2,3\}$, according to Lemma 4.2.
      mark node (u,v).
4. Output $\min\{\delta(G,t_1^\alpha,t_2^\beta)|\alpha,\beta \in \{0,1,2\}$.

Theorem 4.1 *The perfect edge domination problem can be solved in linear time on series-parallel graphs.*

5. Conclusions

In this paper, we considered the perfect edge domination problem in graphs. First, we proved that this problem is NP-complete on bipartite graphs. Meanwhile, we also showed the perfect (vertex) domination problem is NP-complete when graphs are restricted to the class of the line graphs of bipartite graphs, which equivalent to the perfect claw-free graphs. Finally, we gave optimal algorithms for solving the perfect edge domination problem on trees and series-parallel graphs using the techniques of dynamic programming. It is unknown that whether the perfect edge domination problem is polynomial or NP-complete on chordal graphs or planar graphs. For further research, we are interested in this problem for other classes of graphs, such as interval graphs and permutation graphs.

References

[1] M.C. Golumbic, *Algorithmic graph theory and perfect graphs*, Academic Press, New York, (1980).

[2] M.R. Garey and D.S. Johnson, *Computer and Intractability: A Guide to the Theory of NP-completeness*, W.H. Freeman, San Francisco (1979).

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

[4] S.F. Hwang and G.J. Chang, The edge domination problem, *Discuss. Math. – Graph Theory*, **15** (1995) 51-57.

[5] J.D. Horton and K. Kilakos, Minimum edge dominating set, *SIAM J. Disc. Math.*, Vol. 6, No. 3 (1993) 375–387.

[6] K. Kilakos, *On the complexity of edge domination*, Master's thesis, University of New Brunswick, New Brunswick, Canada, (1988).

[7] T. Kikuno, N. Yoshida and Y. Kakuda, A linear algorithm for the domination number of a series-parallel graph, *Discrete Applied Mathematics*, **5** (1983) 299-311.

[8] R. Laskar and K. Peters, Vertex and edge domination parameters in graphs, *Congr. Numer.*, **48** (1985) 291–305.

[9] M. Livingston and Q.F. Stout, Perfect dominating sets, *Congr. Numer.*, **79** (1990) 187–203.

[10] C.L. Lu and C.Y. Tang, The complexity of the efficient edge domination problem, *The 12th Workshop on Combinatorial Mathematics and Computation Theory*, Kaohsiung, R.O.C., November, (1995).

[11] C.L. Lu and C.Y. Tang, Solving the weighted efficient edge domination problem on bipartite permutation graphs, *The 13th Workshop on Combinatorial Mathematics and Computation Theory*, Taichung, R.O.C., June, (1996).

[12] S. Mitchell and S. Hedetniemi, Edge domination in trees, *Proc. Eighth Southeastern Conf. on Combinatorics, Graph Theory and Computing*, Utilitas Mathematica, Winnipeg, Canada, (1977) 489–509.

[13] M. Pal and G.P. Bhattacharjee, The parallel algorithms for determining edge-packing and efficient edge dominating sets in interval graphs, *Parallel Algorithms and Applications*, **7** (1995) 193–207.

[14] A. Srinivasan, K. Madhukar, P. Nagavamsi, C.P. Rangan and M.S. Chang, Edge domination on bipartite permutation graphs and cotriangulated graphs, *IPL*, **56** (1995) 165–171.

[15] C.C. Yen, *Algorithmic aspects of perfect domination*, Ph.D. Thesis, Department of Computer Science, National Tsing Hua University, Taiwan, R.O.C., (1992).

[16] M. Yannakakis and F. Gavril, Edge dominating sets in graphs, *SIAM J. Appl. Math.*, **38** (1980) 364–372.

[17] C.C. Yen and R.C.T. Lee, The weighted perfect domination problem, *IPL*, **35** (1990) 295–299.

[18] C.C. Yen and R.C.T. Lee, A linear time algorithm to solve the weighted perfect domination problem in series-parallel graphs, *European J. Oper. Res.*, **73** (1944) 192–198.