Discrete Mathematics 80 (1990) 21–40
North-Holland

ON TWO DUAL CLASSES OF PLANAR GRAPHS

Ehab S. El-MALLAH
Department of Computing Science, University of Alberta, Edmonton, Canada

Charles J. COLBOURN
Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Canada

Received 29 January 1987
Revised 12 January 1988

A planar graph G is delta-wye "Δ-Y" reducible if G can be reduced to an edge by a sequence of Δ-Y, series, parallel and degree-1 reductions. Politof characterizes Δ-Y reducible graphs in terms of forbidden homeomorphic subgraphs. A wye-delta "Y-Δ" reducible graph is one that can be reduced to an edge by a sequence of Y-Δ, series, parallel and degree-1 reductions. Y-Δ reducible graphs are all partial 3-trees. Recently, Arnborg and Proskurowski have shown confluent reductions which are both necessary and sufficient for the recognition of partial 3-trees.

In this paper we note that Δ-Y graphs are the planar duals of Y-Δ graphs. We exploit this duality and the known reduction rules for partial 3-trees to characterize both classes of graphs using forbidden minors. The result yields a shorter proof of Politof's result. In addition, we give linear time algorithms for recognizing such graphs and for embedding any Δ-Y graph in a 4-tree. These algorithms complement many known linear time algorithms for solving some hard network problems on graphs given their embedding in a *k-tree* for some fixed k.

1. Introduction

In an undirected graph G a *delta* "Δ" is a cycle of three edges and a *wye* "Y" is a vertex of degree 3 and its incident edges. Fig. 1 illustrates the *series* (S), *parallel* (P), Δ-Y and Y-Δ reductions used throughout this paper. A *degree*-1 reduction is simply the deletion of a vertex of degree 1. The idea of solving network problems by recursively performing these reductions appears in the early work of Akers [2] and Lehman [15]. In particular, Akers devised series, parallel, Δ-Y and Y-Δ transformations that preserve the value of the maximum flow and the shortest path in a multiterminal network. Later, Lehman obtained approximate Δ-Y and Y-Δ reliability transformations of an undirected graph. Both Akers and Lehman conjectured that any connected planar graph can be reduced to a single edge using the above reductions. Epifanov [8] proved this conjecture. Recently, Feo [9] has given an $O(n^2)$ time implementation of such reductions on planar graphs, where $n = |V(G)|$.

The above results suggest that subclasses of planar graphs defined using these reductions possess interesting algorithmic properties. Since some network problems are NP-complete on planar graphs (see, for instance [10]) it is interesting to

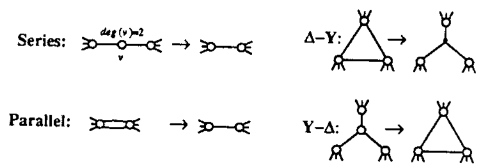


Fig. 1. Series, parallel, Δ-Y and Y-Δ reductions.

identify such subclasses. In particular, we focus on graphs that can be reduced to an edge by recursively applying a subset of the reductions mentioned above.

The class of *Series-Parallel* graphs (e.g. [18], [19]) is a well known member of such families of graphs. A graph G is a series-parallel graph if G can be reduced to an edge using series, parallel and degree-1 reductions.

Recently, Politof [16] introduced the class of Δ-Y reducible graphs (Δ-Y graphs for short) in the context of computing the all-terminal network reliability of a probabilistic network. Here one allows degree-1, series, parallel and Δ-Y reductions. Politof has shown that the class of Δ-Y graphs is exactly the subset of planar graphs with no subgraph homeomorphic to $C_{4,4}$ (the 3-dimensional cube) or to $W_{2,5}$, illustrated in Fig. 2.

In this paper, we introduce Y-Δ reducible graphs as follows. A planar graph G is Y-Δ reducible (or a Y-Δ graph for short) if G can be reduced to an edge using degree-1, S, P and Y-Δ reductions. This class of graphs has an intimate relation to the class of *k-trees*, for $k = 3$.

The class of k-trees is defined recursively as follows. The complete graph on k vertices, K_k, is a *k-tree*. Furthermore, if G is a *k-tree* then so is the graph obtained from G by adjoining a new vertex, and making it adjacent to every vertex in a subgraph H of G which is isomorphic to K_k. Note that trees are 1-trees. A *partial* k-tree is a subgraph of a k-tree. Note that partial k-trees are precisely the graphs of *tree-width* $\leq k$ (see [17]). In Section 2, we show that Y-Δ graphs are all partial 3-trees. Arnborg and Proskurowski [4] present a set of graph rewriting rules which are sufficient to recognize partial 3-trees in $O(n \log n)$ time.

One goal of this paper is to focus on the duality between Δ-Y graphs and planar partial 3-trees (Y-Δ graphs). This duality leads to a unified approach for

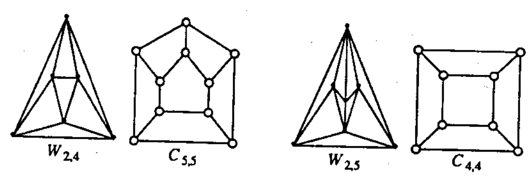


Fig. 2. (a) $W_{2,4}$ and $C_{5,5}$ (b) $C_{4,4}$ and $W_{2,5}$.

characterizing these two classes of graphs in terms of forbidden minors. A recent result of [5] characterizes partial 3-trees using four forbidden minors. Two of these minors are nonplanar, our result shows that the other two suffice to characterize Y-Δ graphs. In addition, our proof requires fewer cases than the more general proof of [5], and is significantly shorter than its counterpart in [16]. A particular improvement in the proof is obtained by exploiting the connectivity properties of these graphs. This latter observation allows us to use a theorem due to Halin on minimally k-connected graphs. In particular, we prove the following theorems.

Theorem 1. *A planar graph is a Y-Δ reducible graph if and only if it has no minor isomorphic to $W_{2,4}$ or $C_{5,5}$.*

Theorem 1 (dual). *A planar graph is a Δ-Y reducible graph if and only if it has no minor isomorphic to $C_{4,4}$ or $W_{2,5}$.*

A second objective is to devise linear time algorithms for recognizing both classes of graphs and to embed any Δ-Y graph in a k-tree, for $k \leq 4$. The fact that Δ-Y graphs are all partial 4-trees has been shown previously in [7]. The existence of such linear time algorithms is interesting since many NP-complete problems can be solved in linear time on partial k-trees given their embedding in a k-tree, for some fixed k, as part of the problem (see, for instance [3]).

The rest of this paper is organized as follows. Section 2 introduces most of the definitions and the background material needed throughout the paper. Section 3 develops a number of intermediate results needed to prove the sufficiency part of Theorem 1. Sections 4 and 5 present the proofs of Theorem 1 and its dual. Section 6 describes the recognition algorithms and analyzes the running time. Section 7 describes an algorithm for embedding Δ-Y graphs in k-trees, $1 \leq k \leq 4$. Finally, we draw some conclusions in Section 8.

2. Graphs, connectivity and k-trees

Most of the graph theoretic definitions used here appear in [6] and [13]. Some basic definitions follow. A graph $G = (V(G), E(G))$ is loopless and undirected. All graphs considered hereafter are connected. A graph is *simple* if it has no multiple (parallel) edges. A graph G_1 is *smaller* than another graph G_2 if $|V(G_1)| < |V(G_2)|$ or if $|V(G_1)| = |V(G_2)|$ and $|E(G_1)| < |E(G_2)|$. If X is a subset of vertices (edges) then $G \setminus X$ (or $G - X$) denotes the graph obtained from G by removing X. The notations "$\subseteq$" and "$\cong$" refer to the subgraph relationship and the isomorphism relationship, respectively.

The set of vertices adjacent to a vertex v is denoted $N_G(v)$; its degree is denoted $\deg_G(v)$. If X is a subset of V then $N_G(X) = \bigcup_{x \in X} N_G(x) \setminus X$. Let X and

Y be two subsets of V. The set of edges with one end vertex in X and the other in Y is denoted by $\delta_G(X, Y)$. We abbreviate $\delta_G(X, V\setminus X)$ by $\delta_G(X)$ and $\delta_G(\{x\})$ by $\delta_G(x)$. Subscripts and qualifiers of a variable to a certain graph are at times omitted when no confusion can arise.

K_r denotes the complete graph on r vertices, also called an r-clique. Two dual planar families of graphs are given special notations. $W_{2,k}$ denotes the graph obtained by adjoining two nonadjacent vertices to a cycle of length k and making them adjacent to every vertex in the cycle. The planar dual of $W_{2,k}$, $k \geq 3$, is denoted $C_{k,k}$. Fig. 2 illustrates two members of each family.

An edge $e = (u, v)$ is said to be *subdivided* if it is deleted and replaced by two new edges (u, w) and (w, v) incident to a new vertex w. A graph G' is said to be *homeomorphic* to a graph G if G' can be obtained from G by a possibly empty sequence of edge subdivisions. We use the notation $H \leq_h G$ to indicate that G contains a subgraph homeomorphic to H. An edge $e = (u, v)$ is said to be *contracted* in G if e and each of its parallel edges are deleted and its end vertices are identified; the resulting graph is denoted by $G \cdot e$. A graph G' is a *contraction* of G if G' can be obtained from G by a possibly empty sequence of edge contractions. Moreover, a graph H is a *minor* of G, denoted $H \leq_m G$, if H is isomorphic to a contraction of some subgraph of G. Thus, a minor H is specified by a subset of edges to be delted and a subset to be contracted. Call a set of vertices of G that collapse into a vertex of H a *pseudo* vertex.

Let G and G^* be two dual planar graphs where e_i in $E(G)$ corresponds to e_i^* in $E(G^*)$. Note that $G - e$ is a planar dual of $G^* \cdot e^*$ for every edge $e \in E(G)$, and vice versa. Thus,

Proposition 1. *Let G and G^* be two dual planar graphs. Then, $H \leq_m G$ if and only if $H^* \leq_m G^*$, for the planar dual graph H^* of H.*

2.1. *Connectivity*

A graph G is k-connected $|V| > k$, if at least k vertices must be removed to obtain a disconnected graph or a single vertex. Thus, every n-clique is $(n-1)$-connected. By Menger's theorem G is k-connected if and only if there exists at least k disjoint paths between any two vertices of G. G is *minimally* k-connected if for every edge (u, v) in G the subgraph $G - (u, v)$ is $(k-1)$-connected but not k-connected. We need the following result:

Lemma 1 [12]. *Let G be a minimally k-connected graph having a complete subgraph H on k vertices. Then at most one vertex of H has degree $\geq k+1$ in G.*

In particular, we need the following special case.

Corollary 1. *Let G be a minimally 3-connected graph having a triangle H. Then, at least two of the vertices in H have degree 3 in G.*

A subset S, $S \subset V$, is a *separating* set if $G\setminus S$ has two or more components. Further, such a separating set is called *elementary* if at most one of the components of $G\setminus S$ has more than one vertex. Let H_0 be some component in $G\setminus S$, for some separating set S. The subgraph H_1 of G induced by $V(H_0) \cup S$ is called an *S-attached* subgraph of G. The graph H_2 obtained from H_1 by adding any missing edges between pairs of vertices in S is called an *S-split* of G. The following lemma is then immediate.

Lemma 2. *Let G be a 3-connected graph, S be one of its separating triplets ($|S| = 3$) and G_1 be an S-split of G. Then G_1 is a 3-connected graph.*

As a consequence we have the following corollary:

Corollary 2. *Let G be a 3-connected Y-Δ graph having a vertex x of degree 3. Then, the graph obtained by applying Y-Δ reduction on x and its incident edges is either 3-connected or a triangle.*

2.2. *k-trees*

The class of *k-trees* has been introduced in Section 1. Note that $(k-i)$-trees, $k > i \geq 1$, are partial k-trees. k-trees are all *chordal* graphs. A graph is *chordal* if it has no induced cycle having four or more vertices (see, for instance [11]). For convenience we call a chordal graph having a maximum clique of size $k+1$ a *k-chordal* graph. A *k-leaf* of a k-tree G is a vertex whose neighbours induce a k-clique. A *leaf* of a partial k-tree G is a vertex x, $\deg_G(x) \leq k$, which is a k-leaf in a particular embedding of G in a k-tree G', $G \subseteq G'$.

A *complete elimination* of a vertex v from G is the elimination of v and its incident edges and the addition of the necessary edges to complete a clique induced by $N(v)$; if $\deg_G(v) \leq k$ then the graph obtained by eliminating v in this way is denoted $\Psi_k(G, v)$. The composition of two complete eliminations $\Psi_k(\Psi_k(G, v_1), v_2)$ is denoted $\Psi_k(G, \langle v_1, v_2\rangle)$. A *k-complete elimination sequence* (k-CES) of a graph G is an ordering of $V(G)$ such that $\deg_G(v_1) \leq k$ and for any i, $2 \leq i < n$, the degree of v_i in $\Psi_k(G, \langle v_1, \ldots, v_{i-1}\rangle)$ is at most k. Thus, a graph is a partial k-tree if and only if it has a k-CES. Moreover, G is a k-chordal graph if the complete eliminations of $V(G)$, according to any given k-CES, do not introduce a new edge to the graph and G contains a $(k+1)$-clique.

The following results can be easily derived from the definition of k-trees; for completeness we sketch a proof to Lemma 4.

Lemma 3. *Every k-tree (partial k-tree) that is not a clique has at least two nonadjacent k-leaves (leaves).*

Lemma 4.
(i) *Given a k-tree G and a complete subgraph H of G, $H \cong K_k$, there exists an ordering S of $V \setminus V(H)$ such that $\Psi_k(G, S) = H$.*
(ii) *Let G_1 and G_2 be two k-trees. Let H_1 and H_2 be two k-cliques in G_1 and G_2, respectively. Then the graph G obtained from G_1 and G_2 by identifying $V(H_1)$ and $V(H_2)$ pairwise is a k-tree.*

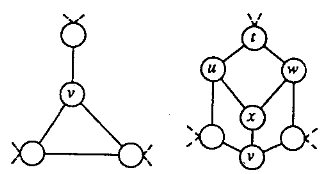
Proof. To show (i) observe that if $G \not\cong H$ then, by Lemma 3, G has at least one k-leaf x such that $\delta(x) \cap E(H) = \emptyset$. Hence, the required sequence S can always be constructed. To show (ii) let S_i, $i = 1, 2$, be an ordering of $V(G_i) \setminus V(H_i)$ such that $\Psi_k(G_i, S_i) = H_i$. Such sequences exist by part (i). Then, $S = \langle S_1, S_2 \rangle$ is a k-complete elimination sequence that reduces G to H_1 (or H_2). Hence, G is a k-tree. □

Corollary 3. *Let H, H_1 and H_2 be complete subgraphs on at most k vertices in a partial k-tree. Then, Lemma 4 holds for partial k-trees.*

Theorem 2 [4]. *Let G be a connected partial 3-tree having no vertex of degree 1 or 2. Then G contains a vertex v of degree 3 satisfying one of the following conditions.*
(i) *All the neighbours of v are also neighbours of another vertex (u) of degree 3.*
(ii) *Two of the neighbours of v are adjacent (Fig. 3a).*
(iii) *The neighbours of v are adjacent to two other vertices u and w of degree 3 which are also adjacent to a fourth vertex (t) (Fig. 3b).*

Observe that the existence of two vertices satisfying condition (i) in a 3-connected planar graph implies that both of the two vertices satisfy condition (ii) also. Consequently, we have the following result.

Corollary 4. *Every 3-connected planar partial 3-tree can be reduced to a triangle by a 3-CES consisting of leaves satisfying condition (ii) or (iii) in Theorem 2.*



(a) (b)

Fig. 3. Two configurations of vertices of degree 3 in a partial 3-tree [4].

Call a vertex v, satisfying condition (ii) in Theorem 2, a *leaf of type* 1. Moreover, if e is an edge joining two neighbours of v then call the connected subgraph H in which $E(H) = \delta(v) \cup \{e\}$ a *configuration of type* 1. Similarly, call a vertex v (and by symmetry w or u), satisfying condition (iii) in Theorem 2, a *leaf of type* 2. Likewise, call the connected subgraph H of G in which $E(H) = \delta(\{u, v, w\})$ a *configuration of type* 2.

We conclude this subsection with the following observation.

Lemma 5. *A simple graph is Y-Δ reducible if and only if it is a planar partial 3-tree.*

Proof. That planar partial 3-trees are all Y-Δ graphs is immediate from the definitions. To see the other direction, note that any sequence Q of degree-1, S, P and Y-Δ reductions that reduces a *simple* Y-Δ graph G can be reordered such that in the new sequence Q' any possible P reduction precedes any S or Y-Δ reduction. The order of S and Y-Δ reductions coincides in Q and Q'. The existence of such a sequence Q' follows since no S or Y-Δ reduction can involve a vertex x incident to a set E_{xy} of two or more parallel edges, sharing a common end vertex y, before reducing E_{xy} to a single edge (x, y).

Now, Q' can be viewed as a sequence of degree-1 reductions and the two extended operations; S_p and $(Y\text{-}\Delta)_p$, each of which produces a simple graph after performing an S or a Y-Δ reduction, respectively. However, degree-1, S_p and $(Y\text{-}\Delta)_p$ reductions are equivalent to complete eliminations of vertices of degrees 1, 2 and 3, respectively. This completes the proof. □

We henceforth use the two terms Y-Δ graphs and planar partial 3-trees interchangeably.

3. More results on connectivity and planar 3-trees

3.1. *Connectivity and minors*

To proof the sufficiency part of Theorem 1 we first prove the following two lemmas.

Lemma 6. *Let G be a 4-connected planar graph having a vertex x of degree k. Then, $W_{2,k} \leqslant_m G$.*

Proof. Denote by C_x the cycle defining the unique face of $G \setminus x$ which contains x in G. Let W be the subgraph of G whose set of edges is $C_x \cup \delta_G(x)$. W is homeomorphic to a wheel on $k + 1$ vertices. Moreover, W satisfies the following:
(i) the subgraph induced by $V' = V(G) \setminus V(W)$ is connected and
(ii) every edge in the subgraph induced by $V(W)$ belongs to $E(W)$.

The first claim follows since at most 3 disjoint paths between any two vertices in V' use vertices of W. To see (ii), assume to the contrary that $E(G)$ contains an edge (a, b), $\{a, b\} \in V(W)$, and $(a, b) \notin E(W)$. Then, a, x and b form a separating triplet in G, contradicting the connectivity of G. Hence, every edge (a, b), $(a, b) \notin E(W)$, which is incident with some vertex a, $a \in C_x$, has its other end vertex $b \in V'$. Moreover, there exists at least one such edge associated with every vertex in C_x.

The first claim allows us to contract V' to a single pseudo vertex by repeatedly contracting edges in $E(G)\setminus E(W)$. Call the resulting minor H. The second claim ensures that H contains a subgraph homeomorphic to $W_{2,k}$. Therefore, $W_{2,k} \leq_m G$. □

Lemma 7. *Let G be a 3-connected triangle-free planar graph, $G \not\equiv C_{4,4}$, in which every separating triplet is elementary. Then, either $W_{2,4} \leq_m G$ or $C_{5,5} \leq_h G$.*

Proof. To derive a contradiction, let G be the smallest graph violating the lemma. Let f be a face of G with the largest possible number of vertices. Denote by C_f the simple cycle of G defining f. Obtain an embedding of G whose exterior face is f. Denote by f' the exterior face of $G' = G\setminus V(f)$ ($V(f)$ denotes $V(C_f)$ for short). We note the following:

(i) Since G is a 3-connected graph having no triangles then G contains a matching M, $M \subset E(G)$, between $V(f)$ and $V(f')$ incident to every vertex in $V(f)$.

(ii) $G' = G\setminus V(f)$ is a block.

To show (ii), assume to the contrary that G' has a cut vertex x. Let G_1' be an $\{x\}$-split of G'. $V(G_1')$ contains x and at least one other vertex, say x'. Denote by Z_1' the subset of $V(f)$ incident to the edges $\delta(V(G_1'), V(f))$. All the vertices of Z_1' are contained in a path (section) of the simple cycle C_f whose origin is denoted a and whose terminus is denoted b.

Now, if $|Z_1'| > 2$ or $|V(G_1')| > 2$ then (a, b, x) forms a nonelementary separating triplet in G, contradicting the assumption. Otherwise, $|Z_1'| = 2$, (i.e. $Z_1' = \{a, b\}$) and $|V(G_1')| = 2$ (i.e. $V(G_1') = \{x, x'\}$). However, this latter statement together with the connectivity of G imply that (x', a, b) is a triangle of G, a contradiction. Thus, G' is a block.

We distinguish the following cases.

Case 1. $|V(f)| \geq 5$.

Denote by $C_{f'}$ the cycle defining the exterior face of G'. The subgraph of G formed by $C_f \cup M \cup C_{f'}$ contains a subgraph homeomorphic to $C_{5,5}$.

Case 2. $|V(f)| = 4$.

By remark (i) above, $|V(f')| \leq |V(f)|$. However, if $|V(f')| > |V(f)|$ then G has a face containing more than 4 vertices which contradicts the assumption that f is a face having the largest possible number of vertices. Hence, $|V(f')| = 4$ and $V(f) \cup V(f')$ induces a subgraph isomorphic to $C_{4,4}$ in G. By assumption, $G \not\equiv C_{4,4}$ and hence $V'' = V(G)\setminus(V(f) \cup V(f')) \neq \emptyset$. In addition, the assumptions ensure that no three vertices of f' form a nonelementary separating triplet in G. Furthermore, no such triplet forms an elementary separating triplet, otherwise, G has a triangle. However, by contracting V'' and $V(f)$ into two pseudo vertices one obtains a minor isomorphic to $W_{2,4}$ which is a contradiction.

Hence the lemma holds. □

3.2. *k-trees*

In this section we prove two results related to k-trees. First, we show that partial k-trees are closed under the operation of forming a minor. This result has also been obtained in [5]. Second, we show that every 3-connected planar partial 3-tree, that is not $C_{4,4}$, contains two edge-disjoint configurations, each is of type 1 or 2.

Lemma 8. *Every minor of a partial k-tree is a partial k-tree.*

Proof. Let G be a partial k-tree, $\bar{G}$ be a k-tree such that $G \subseteq \bar{G}$, and $e = (x, y)$ be an edge of G. By definition $G - e$ is a partial k-tree. To show that $G \bullet e$ is a partial k-tree it suffices to show that $\bar{G} \bullet e$ is partial k-tree. Let H be a k-clique containing e in $\bar{G}$. Such a clique exists by the definition of a k-tree. Further, let S be a prefix of a k-complete elimination sequence that reduces $\bar{G}$ to H. Such a sequence exists by Lemma 4(i). One may verify that $\Psi_k(\bar{G} \bullet e, S) = H \bullet e$. Hence, $\bar{G} \bullet e$ is a partial k-tree. The lemma then follows by considering the two sets of edges to be deleted and contracted to obtain a particular minor. □

Lemma 9. *Every 3-connected Y-Δ graph, G, $|V| \geq 5$, $G \not\equiv C_{4,4}$, has at least two edge-disjoint subgraphs; each of which is a configuration of type* 1 *or* 2.

Proof. The proof is by induction on $|V|$. One may verify that every 3-connected Y-Δ graph on 5 vertices has two edge-disjoint configurations of type 1. Assume that the lemma holds inductively for all 3-connected Y-Δ graphs having at most $n-1$ vertices, $n > 5$. Let G be such a graph on n vertices. We distinguish the following two cases.

Case 1. *Every separating triplet of G is elementary.*

Note that, by the recursive structure of planar partial 3-trees, every such graph is a subgraph of a planar 3-tree. Denote by $\bar{G}$ a planar 3-tree containing G as a spanning subgraph. We claim that $|V(\bar{G})| = |V(G)| \leq 8$. To see this, note that any 3-tree on 5 or more vertices has an induced subgraph H isomorphic to $K_5 - e$. Let $V(H) = \{a, b, c, d_1, d_2\}$ and let (d_1, d_2) be the missing edge from this copy of $K_5 - e$ in $\bar{G}$. Denote by F_i, $i = 1$ or 2, the set of 3 faces of H that share the vertex

d_i. $\bar{G}$ is obtained from H by inserting the remaining vertices $V' = V(G)\setminus V(H)$ in $\boldsymbol{F}_1$ or $\boldsymbol{F}_2$ subject to the following.

(i) At most one of the two sets $\boldsymbol{F}_1$ or $\boldsymbol{F}_2$ contains vertices in any of its 3 faces, and

(ii) Each face in the set of faces $\boldsymbol{F}_i$, $i = 1$ or 2, selected in (i) contains at most one vertex from V'.

The violation of (i) or (ii) results in a nonelementary separating triplet in $\bar{G}$ and hence a contradiction. The claim that $|V(G)| \leqslant 8$ then follows since (i) and (ii) imply that at most 3 vertices can be added to H. Now, G is obtained by removing some edges from $\bar{G}$ while maintaining the 3-connectivity. By inspection, G satisfies the lemma.

Case 2. *G has at least one nonelementary separating triplet, say* (a, b, c).

Denote by G_1, G_2 the two $\{a, b, c\}$-splits of G where $|V(G_1)|, |V(G_2)| \geqslant 5$. We show that each G_i, $i = 1, 2$, contains a configuration H_i which is edge-disjoint from the triangle (a, b, c). The lemma then follows by considering H_1 and H_2 in G. Let $L(G_i)$, $i = 1$ or 2, be a maximal set of independent (pairwise nonadjacent) leaves of G_i that does not include any vertex from the set $\{a, b, c\}$.

Now, each graph $G_i' = \Psi_3(G_i, L(G_i))$ (cf. Section 2.2) contains at least 4 vertices. Otherwise, there would be two non-adjacent vertices in $V(G_i)\setminus\{a, b, c\}$ adjacent to (a, b, c), contradicting the planarity of G. Thus, G_i' has a leaf x of degree 3 which is distinct from a, b and c. We proceed from this point exactly as in the proof of Theorem 2 [4].

Namely, denote by L_x the subset of $L(G_i)$ adjacent to x in G_i. Note that x's neighbours in G_i' consist of two sets $N_G(x) - L_x$ (the original neighbours) and $N_G(L_x) - \{x\}$ (the neighbours attached to x by complete eliminations of L_x). If these two sets are not disjoint then $\delta(L_x) \cup \delta(x)$ contains a configuration of type 1 which is edge-disjoint from the triangle (a, b, c) as required. Otherwise, the two sets are disjoint and we solve Arnborg and Proskurowski's inequality at x:

$$|N_G(x) - L_x| + |N_G(L_x) - \{x\}| = |N_{G_i'}(x)| \leqslant 3$$

one obtains a configuration H_i of type 1 or 2 having

(i) $V(H_i) \subseteq \{x\} \cup L_x \cup N_G(L_x)$ and

(ii) each edge in H_i is incident to at least one vertex in $\{x\} \cup L_x$.

Thus, $E(H_i)$ is edge-disjoint from the triangle (a, b, c) and the lemma follows. □

4. Characterizing Y-Δ graphs

We are now ready to prove the necessity and the sufficiency of Theorem 1. For convenience, we call a graph G that does not have a minor isomorphic to another graph H an *H-free* graph.

Lemma 10. *Let G be a planar graph that contains a minor isomorphic to* $W_{2,4}$ *or* $C_{5,5}$. *Then G is not a partial* 3*-tree* (Y-Δ *graph*).

Proof. One may verify that $W_{2,4}$ is not a partial 3-tree. By Lemma 8, if $W_{2,4} \leqslant_m G$ then G is not a partial 3-tree. In addition, a complete elimination of any vertex in a graph isomorphic to $C_{5,5}$ results in another graph that contains a minor isomorphic to $W_{2,4}$. Hence, $C_{5,5}$ is not a partial 3-tree. Again, by Lemma 8, if $C_{5,5} \leqslant_m G$ then G is not a partial 3-tree. □

To show the sufficiency part we hypothesize the existence of a counterexample to the theorem. Then we show that such an assumption leads to contradictions. In particular, such a hypothetical graph must satisfy the properties listed in the following lemma.

Lemma 11. *Let G be a smallest* $(W_{2,4}, C_{5,5})$*-free planar graph that is not a partial* 3*-tree* (Y-Δ *graph*). *Then*

(i) *G does not have a separating vertex, a pair of separating vertices or a nonelementary separating triplet,*

(ii) *G is* 3*-connected but not* 4*-connected,*

(iii) *G has no triangle with a vertex of degree three* (*i.e. a configuration of type* 1) *and*

(iv) *G has at least one triangle.*

Proof.

(i) To derive a contradiction, assume that such a separating set S exists. Let G_1 be an S-split of G. If $|S| = 1$ or 2 then it is easy to verify that $G_1 \leqslant_m G$. Now, G_1 is a $(W_{2,4}, C_{5,5})$-free planar graph and is smaller than G. Thus, G_1 is a partial 3-tree. By Corollary 3, there exists a 3-complete elimination sequence Q that reduces G_1 to an $|S|$-clique whose set of vertices is S. Again, the graph $G_2 = \Psi_3(G, Q)$ is a $(W_{2,4}, C_{5,5})$-free planar graph and is smaller than G. Hence, G_2 is a partial 3-tree. Thus G can be reduced to an $|S|$-clique on the set S. It follows that G is a partial 3-tree, a contradiction. Thus, if such a graph exists it must be 3-connected. The latter case when S forms a nonelementary separating triplet is handled similarly. In particular, since G is 3-connected and $|V(G)\setminus V(G_1)| \geqslant 2$ it follows that G has a minor isomorphic to the bipartite graph $K_{2,3}$, where two vertices of one partition correspond to pseudo vertices in $V(G)\setminus V(G_1)$ and the three vertices of the other partition are in S. Hence, $G_1 \leqslant_m G$.

(ii) By (i), G is 3-connected. If G is 4-connected then $W_{2,4} \leqslant_m G$, by Lemma 6, contradicting the assumption.

(iii) Suppose G has a 3-leaf of type 1, say x. Then, $G' = \Psi_3(G, x) \leqslant_m G$. Thus, $\Psi_3(G, x)$ is a smaller $(W_{2,4}, C_{5,5})$-free planar graph than G. Since G is a smallest possible graph violating the lemma, it follows that G' is a partial

3-tree. However, this implies that G is a partial 3-tree, contradicting the assumption.

(iv) By assumption G is $(W_{2,4}, C_{5,5})$-free graph that is not a partial 3-tree (hence, $G \not\cong C_{4,4}$). By conditions (i) and (ii) above G is a 3-connected graph in which every separating triplet is elementary. The statement then follows using Lemma 7. □

Lemma 12. *Let G be a $(W_{2,4}, C_{5,5})$-free planar graph. Then G is a partial 3-tree (Y-Δ graph).*

Proof. The proof is by contradiction. Let G be as in Lemma 11. Then G is 3-connected but not 4-connected. We distinguish two cases.

Case 1. *G is not minimally 3-connected.*

Then $|V(G)| \geq 5$ and there exists an edge (u, v) such that $G' = G - (u, v)$ is 3-connected. G' is a 3-connected $(W_{2,4}, C_{5,5})$-free graph which is smaller than G. Hence, G' is a partial 3-tree. We prove the following claims in order.

(A1) u and v are leaves of G' of type 1 or 2.

(A2) $G'' = G' \setminus \{u, v\}$ is a block.

(A3) $W_{2,4} \leq_m G$; contradicting the assumptions.

Proof of A1. By Lemma 9, G' has at least two edge-disjoint configurations, each of which is of type 1 or 2. The removal of an edge can possibly result in these two leaves by decreasing their degrees from 4 in G to 3 in G'. This proves A1 since there is no other way in which these two leaves could arise.

Proof of A2. To derive a contradiction assume G'' has 2 or more blocks. Denote by B_1 and B_2 some two end blocks of G''. One of the following two cases arises.

*Case A*2.1. *B_1 (or B_2) is isomorphic to K_2, say $B_1(x, c)$, with a cut vertex x in G''.*

Now, c is adjacent in G' to u, v and x and no other vertex. That is, $\deg_{G'}(c) = \deg_G(c) = 3$. However, two of c's neighbours (u and v) are adjacent in G. Hence, c is a type 1 vertex in G, contradicting condition (iii) of Lemma 11.

*Case A*2.2. $|V(B_1)|, |V(B_2)| \geq 3$.

Let x be a cut vertex of G'', $x \in V(B_1)$. Then, u, v and x from a nonelementary separating triplet in G, contradicting condition (i) in Lemma 11.

Hence, G'' is a block.

Proof of A3. Consider a planar embedding of G in which u and v lie on the exterior face f of G. Denote by f'' the exterior face of G'', according to that particular embedding, and let $C_{f''}$ denote its defining cycle. Since both u and v have degree 3 in G' and they can share at most two vertices of f'' it follows that $|V(f'')| \geq 4$. Let $N_{G'}(u) = \{a, b, c\}$ and $N_{G'}(v) = \{a', b', c'\}$. Let $\boldsymbol{P} = \{P_{aa'}, P_{bb'}, P_{cc'}\}$ be a set of 3 vertex disjoint paths in G' joining $N_{G'}(u)$ to $N_{G'}(v)$. By the planarity of G we may assume that a, b, c, c', b', a' occur in that order when traversing $C_{f''}$ in one direction. The subgraph induced by $V(f'') \cup V(\boldsymbol{P}) \cup \{u, v\}$ in G has a minor isomorphic to $W_{2,4}$ whose pseudo vertices are: $\{u\}$, $\{v\}$, $V(P_{aa'})$, $\{b\}$, $V(P_{bb'}) \setminus \{b\}$, $V(P_{cc'})$.

Hence, if G exists it must be minimally 3-connected.

Case 2. *G is minimally 3-connected.*

G can not possibly have a triangle; otherwise G would contain a 3-leaf of type 1 by Corollary 1 to Halin's theorem (Lemma 1). However, this contradicts Lemma 11(iii). On the other hand, Lemma 11 (i) requires G to have at least one triangle. Hence, G can not be minimally 3-connected.

Thus, no such G exists and the sufficiency part of Theorem 1 follows. □

5. Characterizing Δ-Y graphs

We first prove the following result.

Lemma 13. *Every minor of a Δ-Y graph is a Δ-Y graph.*

Proof. It suffices to prove the lemma for Δ-Y graphs having no cut-vertices. The proof is by induction on the minimum number r of series, parallel and Δ-Y operations required to reduce such a Δ-Y graph to an edge. The lemma holds for the graph on two parallel edges where $r = 1$. Assume it holds inductively for any Δ-Y graph that has no cut-vertex and which can be reduced to an edge using fewer than r operations. Let G be a Δ-Y graph without a cut-vertex that can be reduced by a sequence S having at most r operations and let $e \in E(G)$. Furthermore, let G' be the graph obtained from G after applying the first operation OP in S.

If OP does not involve e then applying this operation to $G - e$ or $G \bullet e$ results in a graph G'', $G'' \leq_m G'$. Otherwise, OP involves e and one of the following holds.

(1) OP is a parallel reduction or a Δ-Y reduction; then $G - e$ and the underlying simple graph of $G \bullet e$ are minors of G'.

(2) OP is a series reduction involving two edges e and e^*. Then e^* is attached to the rest of the graph $G - e$ by a single vertex and $G - e - e^*$ is a minor of G'. In addition, $G \bullet e \cong G'$.

In each of the above cases the lemma follows by the induction hypothesis. □

One may verify that K_5 and $K_{3,3}$ are not Δ-Y graphs. Using the above lemma, one can see that Δ-Y graphs form a subset of planar graphs. In addition, reducing a Δ that is not a face of a 3-connected planar graph creates a minor isomorphic to $K_{3,3}$. Thus, a Δ-Y reduction that appears in a reduction of a 3-connected Δ-Y graph to an edge involves some triangular face of that graph. The two facts mentioned above have been previously obtained in [16]. We also need the following result.

Lemma 14. *A 2-connected planar graph is Δ-Y graph if and only if its planar dual is a Y-Δ graph.*

Proof. Let G and G^* be two dual planar graphs. Note that if G admits a S, P or Y-Δ reduction then G^* admits a P, S or Δ-Y reduction, respectively. In addition, if G admits a Δ-Y reduction of one of its triangular faces then G^* admits a Y-Δ reduction of the vertex corresponding to that face. In each case, the graph G_1 obtained by applying one of the above reductions to G is the dual of the graph G_1^* obtained by applying a dual reduction to G^*. Hence, if G is a Y-Δ graph (or a 3-connected Δ-Y graph) then G^* is a Δ-Y graph (respectively, a 3-connected Y-Δ graph).

To prove the remaining case where G is a 2-connected Δ-Y graph with at least one pair of separating vertices we need the following observation. First, recall that by Corollary 3 every Y-Δ graph H can be reduced to any one of its edges e with a sequence of reductions that does not contain e. Using this latter fact and the duality between 3-connected Δ-Y and Y-Δ graphs it follows that any 3-connected Δ-Y graph H^* can be reduced to any one of its edges e^* with a sequence of reductions that does not involve e^*.

We now use the above observation to show that G^* is a Y-Δ graph. Here, it suffices to show a reduction sequence Q of G in which every Δ-Y reduction involves some triangular face of the current reduced graph. Call a subgraph H of G that is attached to the rest of the graph by exactly two vertices an *end 2-attached* subgraph of G. Let H be such a subgraph of G and x and y be its two vertices of attachments. Now, $H+(x,y)$ is a 3-connected minor of G and hence it is a Δ-Y graph. The first subsequence of Q reduces H to the edge (x,y). Consequently, Δ-Y reductions appearing in this first part involve only triangular faces. Call the resulting graph G_1. Subsequently, the ith subsequence of Q, $i>1$, reduces the current graph G_{i-1} to an edge if G_{i-1} is a 3-connected graph or else it reduces one of its end 2-attached subgraphs to an edge between its vertices of attachments. □

Theorem 1 and Lemma 14 then imply the following result:

Theorem 1 (dual). *A planar graph is a Δ-Y reducible graph if and only if it has no minor isomorphic to $C_{4,4}$ or $W_{2,5}$.*

6. Recognizing Δ-Y graphs and Y-Δ graphs

6.1. *Recognizing Δ-Y graphs*

We start by outlining an $O(n)$ algorithm for recognizing a Δ-Y graph on n vertices and m edges. By Corollary 4 and the planar duality between Δ-Y and Y-Δ graphs (Lemma 14) we have the following result, stated also in [16].

Lemma 15 [16]. *Every 2-connected Δ-Y graph can be reduced to an edge by a sequence of series, parallel and Δ-Y reductions in which each delta satisfies one of the following conditions.*

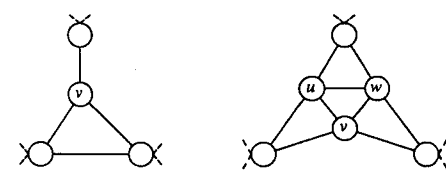
(i) *The delta has at least one vertex of degree 3 (Fig. 4a).*

(ii) *The delta is one of three edge-disjoint delta's; each pair of them share a distinct common vertex of degree 4 (Fig. 4b).*

The algorithm employs degree-1 reductions, series reductions and Δ-Y reductions of delta's satisfying Lemma 15. Parallel reductions are implicit in the implementation. We henceforth call a vertex of degree 1 or 2 a *Δ-Y leaf*. In addition, we call a vertex similar to v in Fig. (4a) a Δ-Y leaf of *type* 1. Similarly, any vertex similar to u, w or v in Fig. 4b is called a Δ-Y leaf of *type* 2.

A description of the recognition algorithm now follows. We use a queue to hold the leaves identified so for in the graph. The algorithm is organized into two phases. The *first phase* starts by *unmarking* all the vertices. Subsequently, it scans each vertex in the graph sequentially. At each step of the scanning, it checks whether the current vertex is an *unmarked* Δ-Y leaf having no *marked* neighbours. If v satisfies these conditions then it *marks* v and inserts it in the queue.

The *second phase* iterates until the queue becomes empty. At each iteration the algorithm removes a vertex v from the queue, performs a corresponding reduction on the current graph. Here, we avoid adding new vertices to the graph, after performing a Δ-Y reduction, by combining Δ-Y reductions and series reductions. Subsequently, the algorithm inspects each of the *unmarked* neighbours of v; if one of those vertices becomes a Δ-Y leaf and none of its



(a) (b)

Fig. 4. Two configurations of delta's in a Δ-Y graph [16].

neighbours has been previously *marked* then it is *marked* and inserted in the queue. Finally, to verify that G is a Δ-Y graph we verify that the number of vertices reduced at the end of the second phase equals $n-1$.

Vertices of degree 1 or 2 are identified by simply checking their degrees. A vertex v is of type 1 if any of its neighbours are adjacent. Finally, a vertex v is of type 2 if (i) it has two adjacent neighbours, say u and w, and (ii) each pair of (u, v, w) share a common neighbour distinct from (u, v, w) and distinct from one another. In each case the associated reduction follows as in Fig. 1. The correctness of the above algorithm is given by the next lemma.

Lemma 16. *The above algorithm recognizes Δ-Y graphs.*

Proof. At the end of the first phase the contents of the queue include a maximal set of independent Δ-Y leaves of G. This property remains true following each iteration of the second phase with respect to the current reduced graph. This ensures that the state of a Δ-Y leaf at queuing time remains intact until the time of removal from the queue. By the end of the second phase the reduced graph G' has no Δ-Y leaves. If G' is not a single vertex then by Lemma 15 G' is not a Δ-Y graph. Hence, G is not a Δ-Y graph. □

6.2. *Time analysis*

We now describe a data structure that yields a linear time implementation of the above algorithm. Note that a general iteration step in phase 1 or 2 performs a number of testing and reduction operations. Each such operation can be expressed as a constant number of the following more primitive operations: (i) deleting an edge, (ii) adding an edge, (iii) finding a degree of a vertex and (iv) listing the neighbours of a vertex having constant degree. Therefore, the problem calls for devising a data structure that supports each of the above four primitive operations in constant time.

We use a data structure based on the standard *adjacency* matrix representation of a graph. We henceforth associate a graph G with its adjacency matrix $G[1..n, 1..n]$. We encode each edge (x, y) of G twice in the two entries $G[x, y]$ and $G[y, x]$. This allows encoding further information in each entry. In addition, we maintain the degree of each vertex in the current graph G in an additional array DEG$[1..n]$. Thus, each of the first three primitive operations can be done in a constant time. We further modify the adjacency matrix to avoid initializing the matrix G in $O(n^2)$ time and to support listing the neighbours of a vertex of a constant degree in $O(1)$ time.

The first modification is based on a technique mentioned in [1]. Briefly, the scheme uses cross pointers between entries in a stack $g[1..2m]$ and entries in the matrix G to distinguish the active entries in the matrix from the random ones. The second modification allows for listing the neighbours of any vertex having a constant degree in a constant time in the current graph.

Namely, in each row x of the matrix G we maintain $N(x)$ in a doubly linked cyclic list. Therefore, we require each entry $G[x, y]$ to hold three pieces of information: a pointer to the stack g and two pointers to the two neighbours of x which lie before and after y in the cyclic list. Moreover, for each row x we maintain an entry pointer, ADJ$[x]$, to some neighbour of x in its cyclic list. Insertions and deletions of edges are done using standard linked list procedures. One may verify that the modified data structure supports each of the intended operations in $O(1)$ time. We then establish timing for the above algorithm.

Lemma 17. *An n-vertex Δ-Y graph G can be recognized in $O(n)$ time.*

Proof. There is a constant number of operations done in each iteration in phase 1 and 2. Each of these operations require $O(1)$ primitive operations supported by the data structure. Hence, each iteration in phase 1 or 2 requires an $O(1)$ time. In addition, each phase requires at most n iterations. Thus, the overall running time is $O(n)$. □

6.3. *Recognizing Y-Δ graphs*

The algorithm presented above requires minor modifications to recognize planar partial 3-trees. Here, we search for leaves of degrees 1 and 2 and leaves of type 1 and 2, illustrated in Fig. 3. The associated reduction operations are the corresponding complete eliminations. The resulting algorithm recognizes Y-Δ graphs and other nonplanar partial 3-trees (e.g. $K_{3,3}+e$, where e is any edge not in $K_{3,3}$). Thus, combining the above algorithm with a linear time algorithm for detecting planar graphs (e.g. [14]) yields a linear time algorithm for recognizing Y-Δ graphs.

7. Embedding Δ-Y graphs into 4-trees

Δ-Y graphs have been shown to be all partial 4-trees [7]. In this section we extend the algorithm described for recognizing Δ-Y graphs to embed such graphs in k-trees, $1 \leq k \leq 4$, in $O(n)$ time. Interest in such an algorithm arises since many hard network problems possess linear time algorithms on partial k-trees for a fixed k, given their embedding as part of the problem (see, for instance [3]). The list of such problems includes: Hamilton circuit and K-terminal network reliability. Briefly, the algorithm first obtains an embedding of a Δ-Y graph in a k-chordal graph, $1 \leq k \leq 4$, then it embeds the k-chordal graph in a k-tree.

To achieve the first goal we use the two phases of the algorithm described in Section 6 with the following modifications to the second phase. At the beginning of phase 2 we initialize an array S that will include a 4-complete elimination sequence of a given Δ-Y graph G. In addition, we maintain two copies of G; one copy is reduced at each iteration step while the second copy receives new edges to complete the graph to a 4-chordal graph G'.

The reduction operations associated with a Δ-Y leave of type 1 is the Y-Δ reduction (rather than the Δ-Y reduction of the original algorithm). Subsequently, the reduced vertex is appended to S. To reduce a Δ-Y leaf v of type 2, having two neighbours u and w, similar to those illustrated in Fig. 4b, we completely eliminate u, v and w from G. Subsequently, u, v and w are appended consecutively in S. Note that each complete elimination operation may add new edges to the second copy of the graph. If the number of vertices removed at the end of phase 2 equals $n-1$ the algorithm terminates *successfully*. The correctness of the above part is described below.

Lemma 18. *The above algorithm embeds a Δ-Y graph G in a k-chordal graph, $1 \leq k \leq 4$, G'.*

Proof. One may verify that the graph G' obtained at the end of a *successful* completion of phase 2 is a k-chordal graph, $1 \leq k \leq 4$, having a k-complete elimination sequence S. It remains to show that if G is a Δ-Y graph then the algorithm completes *successfully*. This is proven inductively by showing that at the end of the ith iteration of phase 2 the reduced graph G_i is a Δ-Y graph and the queue includes a maximal set of independent Δ-Y leaves.

The claim holds prior to the first iteration step of the second phase; here $G_0 = G$ is assumed to be a Δ-Y graph. Assume the hypothesis holds inductively prior to the ith iteration for the reduced graph G_{i-1}. Denote by v_i the first vertex removed from the queue in the ith iteration. We show that it holds at the end of the ith iteration.

If $\deg(v_i) \leq 2$, in G_{i-1} or if v_i is a Δ-Y leaf of type 1 then $G_i \leq_m G_{i-1}$. Similarly, if v_i, v_{i+1} and v_{i+2} are three adjacent Δ-Y leaves, each is of type 2, then the graph G_i obtained from G_{i-1} by complete eliminations of v_i, v_{i+1} and v_{i+2} is again a minor of G_{i-1}. By Lemma 13, G_i is a Δ-Y graph. Finally, the second part of the inductive hypothesis follows since the algorithm inspects the neighbourhood of the configuration reduced so far and updates the queue accordingly. □

We add that the class of graphs that can be recognized and embedded by the above algorithm alone is larger than Δ-Y graphs. For example, it produces an embedding of $K_{3,3}+e$. Yet, $K_{3,3}+e$ is not a Δ-Y graph. Integrating the above algorithm with the data structure used in the recognition algorithm yield the following result.

Lemma 19. *An n-vertex Δ-Y graph can be embedded in a k-chordal graph, $1 \leq k \leq 4$, in linear time.*

The second part of the algorithm obtains an embedding of a k-chordal graph G

in a k-tree G' given a k-CES, say $S = \langle v_1, \ldots, v_n \rangle$, of G. To simplify the description we denote by G_i, $i \geq 2$, the graph $\Psi_4(G, \langle v_1, \ldots, v_{i-1} \rangle)$ and let $G_1 = G$. In addition, let $k_i = \deg_{G_i}(v_i)$ and $N_i = N_{G_i}(v_i)$. Moreover, define the *successor* of a vertex v_i, denoted succ(v_i), to be a vertex v_j such that j is the smallest integer greater than i in S and $v_j \in N_G(v_i)$.

For our purpose, we record the values of k_i and N_i, $1 \leq i \leq n-1$, during the execution of the first part of the algorithm. Clearly, recording such information does not affect the $O(n)$ time required to complete the first phase. Note that G is a k-chordal graph for $k = \max_{1 \leq i \leq n-1}(k_i)$. The second part of the algorithm starts by computing succ(v_i), $1 \leq i \leq n-k-1$. Second, we add any missing edges between pairs of vertices in the set $\{v_{n-k}, \ldots, v_n\}$. Next, for each vertex v_i, $i = n-k-1, \ldots, 1$ (in this order), with $j = \text{succ}(v_i)$, we add $k - Ki$ edges on the form (v_i, v_p) where $v_p \in N_j \setminus N_i$. Subsequently, we update the list N_i to include the subset of $N_j \setminus N_i$ that are now adjacent to v_i in the new augmented graph. The correctness and timing of the above algorithm is given in the following lemma.

Lemma 20. *The above algorithm embeds a k-chordal graph on n vertices in a k-tree in $O(n)$ time, for any fixed k.*

Proof. The correctness of the algorithm follows since the subgraph induced by the vertices $\{v_{n-k-i}, \ldots, v_n\}$, $1 \leq i \leq n-k-1$, following the ith iteration is a k-tree. To verify the timing, note that computing succ(v_i), $1 \leq i \leq n-1$, can be done in $O(n)$ time since $|N_i| \leq k$. Using the data structure presented in Section 6, one may verify that the second phase requires $O(kn)$ time. Thus, the overall algorithm runs in $O(n)$ time for any fixed k. □

8. Conclusion

In this paper we obtained a characterization of Δ-Y and Y-Δ graphs in terms of forbidden minors. The proof uses results of [12] and [4]; by exploiting duality, it unifies some partial results of [16] and [5]. In addition, we devise three linear time algorithms to recognize such graphs and to embed Δ-Y graphs in k-trees, $1 \leq k \leq 4$. We suggest the problem of characterizing the recognizing partial 4-trees as an interesting problem for future research.

Acknowledgements

This research has been done as a part of the first author's Ph.D. dissertation at the Department of Computer Science, University of Waterloo. Research of the second author is supported by NSERC Canada grant A0579.

References

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

[2] S. Akers, The use of wye-delta transformations in network simplification, Operations Research, 8 (1960) 311–323.

[3] S. Arnborg and A. Proskurowski, Linear Time Algorithms for NP-Hard Problems on Graphs Embedded in k-trees, TRITA-NA-8404, Royal Institute of Technology, S-100 44 Stockholm, Sweden (1984).

[4] S. Arnborg and A. Proskurowski, Characterization and recognition of partial 3-trees, SIAM J. Algebraic and Discrete Methods, 7, 2 (1986) 305–314.

[5] S. Arnborg, A. Proskurowski and D. Corniel, Forbidden Minors Characterization of Partial 3-trees, Technical Report CIS-TR-86-07, Dept. of Computer and Information Science, University of Oregon.

[6] J.A. Bondy and U.S.R. Murty, Graph Theory with Applications (McMillan Press, New York, 1976).

[7] E.S. El-Mallah and C.J. Colbourn, Reliability of Δ-Y networks, Proceedings of the Southeastern Conference on Combinatorics, Graph Theory and Computing, 48 (1985) 49–54.

[8] G.V. Epifanov, Reduction of a plane graph to an edge by a star-triangle transformation, Doklady, Tom 166, 1 (1966) 13–17.

[9] T. Feo, Efficient reduction of planar networks for solving certain combinatorial problems, University of California, Berkeley, preprint.

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

[11] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs (Academic Press, Inc., New York, 1980).

[12] R. Halin, Untersuchungen uber minimale n-fach zusammenhangende graphen, Math. Ann., 182 (1969) 175–188.

[13] F. Harary, Graph Theory (Addison-Wesley, Reading, MA, 1970).

[14] J. Hopcroft and R. Tarjan, Efficient planarity testing, J. Assoc. Comp. Mach., 21 (1974) 549–568.

[15] A. Lehman, Wye-delta transformations in probabilistic networks, J. SIAM, 11, 3 (1963) 773–805.

[16] T. Politof, A characterization and efficient reliability computation of Δ-Y reducible networks, Ph.D. Thesis, University of California, Berkeley (1983).

[17] N. Robertson and P.D. Seymour, Graph Minors—A Survey, in Surveys in Combinatorics 1985, ed. I. Anderson, (Cambridge University Press, 1985).

[18] T. Takamizawa, T. Nishizeki and N. Saito, Linear time computability of combinatorial problems on series parallel graphs, JACM, 29, 3 (1982) 623–641.

[19] J.A. Wald and C.J. Colbourn, Steiner trees, partial 2-trees and minimum IFI networks, Networks, 13 (1983) 159–167.