Matching Cutsets in Graphs

Augustine M. Moshi
DEPARTMENT OF MATHEMATICS
UNIVERSITY OF DAR ES SALAAM, TANZANIA

ABSTRACT

Let $G = (V, E)$ be an undirected graph. A subset F of E is a matching cutset of G if no two edges of F are incident with the same point, and G-F has more components than G. Chvátal [2] proved that it is NP-complete to recognize graphs with a matching cutset even if the input is restricted to graphs with maximum degree 4. We prove the following: (a) Every connected graph with maximum degree ≤ 3 and on more than 7 points has a matching cutset. (In particular, there are precisely two connected cubic graphs without a matching cutset). (b) Line graphs with a matching cutset can be recognized in $O(|E|)$ time. (c) Graphs without a chordless circuit of length 5 or more that have a matching cutset can be recognized in $O(|V||E|^3)$ time.

1. INTRODUCTION

Let $G = (V, E)$ be an undirected graph with maximum degree Δ. We shall say that G has a *matching cutset* if there exists $F \subset E$ such that F is a matching and G-F has more components than G. Ronald R. Graham [3] asked whether it is *NP-complete* to recognize graphs with a matching cutset. This was answered by Chvátal [2], who proved that it is *NP-complete* to recognize graphs with a matching cutset even if the input is restricted to graphs with $\Delta = 4$. Chvátal also gave a fast algorithm to recognize those graphs with $\Delta \leq 3$ that have a matching cutset. We will show that, in fact, all connected graphs with $\Delta \leq 3$ and $|V| \geq 8$ have a matching cutset.

For convenience, we shall sometimes abbreviate by *matching cut* the problem of recognizing graphs with a matching cutset. We prove that matching cut is solvable in polynomial time if the input is restricted to line graphs or to graphs without a chordless circuit of length ≥ 5. Line graphs are well explained in [4] and [5], where they are called *interchange graphs*.

Journal of Graph Theory, Vol. 13, No. 5, 527–536 (1989)
 CCC 0364-9024/89/050527-10$04.00

2. GRAPHS WITH MAXIMUM DEGREE 3

One of the most famous problems in mathematics has been to prove or disprove that every 2-edge connected 3-regular planar graph is 1-factorable. (A graph is *1-factorable* if its edge set can be decomposed as the direct sum of perfect matchings.) This problem derives its fame from the fact that it is equivalent to the *Four Colour Problem*. (See for example [1], p. 263.) In view of this, the following proposition does at first look quite interesting.

Proposition 1. Let G be any 3-regular graph that does not have a matching cutset. Then G is 1-factorable.

Proof. Since G does not have a matching cutset, then G cannot have a cut-edge, and so G is 2-edge connected. Now, a well-known corollary to Tutte's Perfect Matching Theorem (see [1], p. 158) states that every 2-edge connected 3-regular graph has a perfect matching. Thus, G has a perfect matching F, and G-F is a connected spanning 2-regular subgraph of G, that is, a hamilton circuit of G. Since G is 3-regular, then $|V(G)|$ is even. Thus, G-F has two disjoint perfect matchings. Now, F together with the two disjoint perfect matchings of G-F give a 1-factorization of G. ∎

The usefullness of Proposition 1 is unfortunately ruined by the following result:

Theorem 1. Let G be a simple connected 3-regular graph that is different from K_4 and $K_{3,3}$. Then G has a matching cutset.

Proof. G is not a tree, so we can choose a shortest circuit C in G. Let $\partial(C)$ be the set of edges with precisely one endpoint in C. If $|V(C)| \geq 5$ then $\partial(C)$ is a matching cutset, otherwise we can find a circuit that is shorter than C. Consider the cases when $|V(C)| \leq 4$.

Case 1. $|V(C)| = 3$: If $\partial(C)$ is not a matching, then let $x \in V(G) - V(C)$ such that $N(x) \cap V(C)$ consists of two points $\{z, y\}$. (Here, $N(x)$ is the neighbor set of x.) Since $G \neq K_4$, then $xt \notin E(G)$, where $t \in V(C) - \{z, y\}$. Let $x' \in N(x) - \{y, z\}$, $t' \in N(t) - \{y, z\}$. If $x' = t'$ then G has a cut-edge (which is incident with x'). If $x' \neq t'$ then $\{xx', tt'\}$ is a matching cutset of G.

Case 2. $|V(C)| = 4$: Let $C = (x_1, x_2, x_3, x_4)$. If $\partial(C)$ is not a matching cutset of G then let $y \in V(G) - V(C)$ such that $|N(y) \cap V(C)| = 2$. Moreover, we know that the neighbors of y on C are not adjacent, by minimality of C. So, let $\{x_2, x_4\} = N(y) \cap V(C)$. Let $\{x_1'\} = N(x_1) - V(C)$, $\{x_3'\} = N(x_3) - V(C)$, $\{y'\} = N(y) - V(C)$. (The third neighbor of y cannot be on C, by minimality of C.) Since G is not $K_{3,3}$, we cannot have $x_1' = x_3' = y'$. If no two of the points x_1', x_3', and y' are the same point, then $\{x_1x_1', x_3x_3', yy'\}$ is a matching cutset of G. We assume now that $x_1' = x_3' \neq y'$ (the cases $x_1' = y' \neq x_3'$ and $x_3' =$

$y' \neq x_1'$ are the same, *mutatis mutandis*). Let $t \in N(x_1') - \{x_1, x_3\}$. If $t = y'$ then the edge $y'z$ where $z \in N(y') - \{x_1', y\}$ is a cut-edge of G. If $t \neq y'$, then $\{x_1't, yy'\}$ is a matching cutset of G. ∎

Using essentially the same argument as in the proof of the preceeding theorem, one can in fact prove the stronger result that every simple connected graph on more than 7 points and with maximum degree at most 3 must have a matching cutset.

3. LINE GRAPHS

We shall prove in this section that we can recognize line graphs with a matching cutset in polynomial time. We give a procedure that, when given a line graph $G = (V, E)$, takes $O|E|)$ steps either to find a matching cutset of G or to arrive at the conclusion that G does not have a matching cutset.

Definition. A special k-set of a graph G is a set T of k points of G such that $d_G(v) = 2$ for each point v in T, and $G[T] = P_{k-1}$ (a path of length $k - 1$).

In the definition above, $G[T]$ refers to the subgraph of G that is induced by T. Observe that a 2-connected graph on more than 3 points that has a special 2-set has an obvious matching cutset.

Definition. A graph $G = (V, E)$ has a *degree* k *stable cutset* if there is a subset S of V, consisting of pairwise nonadjacent points such that $G - S$ has more components than G and $d_G(v) = k$ for each $v \in S$. We can now state the observation that is the basis of our algorithm for solving *matching cut* when the input is restricted to line graphs. We shall denote the line graph of G by $L(G)$. If $U \subseteq V$, we shall write $\partial(U)$ to mean the set of edges of G that have precisely one endpoint in U. For $x \in V$, we shall write $\partial(x)$ instead of $\partial(\{x\})$.

Lemma 1. Suppose that $L(G)$ is a connected line graph with a root graph G and with more than 2 points. Then $L(G)$ has a matching cutset if and only if G has a degree 2 stable cutset.

Proof. Let $G = (V, E)$. Then $L(G) = (E, E')$, where $E' = \{ef\colon e, f \in E$ and e and f have a common endpoint in $G\}$. Let F be a minimal matching cutset of $L(G)$. Put $S = \{x \in V\colon$ there are e and f in E such that x is incident with both e and f and $ef \in F\}$. We claim that $d_G(x) = 2$ for each $x \in S$. If not, let $x \in S$ have degree at least 3 in G. Let $ef \in F$ with $e, f \in \partial(x)$. Let $g \in \partial(x) - \{e, f\}$. Now by construction of $L(G)$, we have that the edges ef, eg, and fg are edges of $L(G)$, and these three edges induce a triangle in $L(G)$. This contradicts the fact that $ef \in F$, since no edge of a triangle can possibly be in a minimal matching cutset. So, $d_G(x) = 2$ for each $x \in S$.

We claim further that S is a stable set. If not, let $x, y \in S$ be coincident with an edge f. Let $\{e\} = \partial(x) - \{f\}$ and $\{g\} = \partial(y) - \{f\}$. Suppose that $e = g$. Then x and y induce a component of $L(G)$ that has just two points (this uses the fact that $d_G(x) = d_G(y) = 2$). This is a contradiction, and so $e \neq g$. Now we have ef and fg are distinct edges of $L(G)$, and since $x, y \in S$, we also have $ef, fg \in F$. This contradicts the fact that F is a matching. Hence S is a stable set, as claimed.

To show that S is a cutset of G, we will show that $G - S$ has as many components as $L(G) - F$. Let G_1 and G_2 be any two components of $L(G) - F$, and choose h_1 and h_2 any points of G_1 and G_2 respectively. Let u and v be endpoints of the edges h_1 and h_2 respectively in G, such that $S \cap \{u, v\} = \emptyset$. Then there is no path in $G - S$ from u to v. Thus, S is a degree 2 stable cutset of G.

We suppose now that G has a minimal degree 2 stable cutset S and show that $L(G)$ has a matching cutset. Define $F = \{ef \in E'$: there is $x \in S$ with $e, f \in \partial(x)\}$. Suppose that two edges in F have a common endpoint. Then the two edges are of the form ef and fg, where e, f, and g are distinct edges of G. Let $x, y \in S$ such that $e, f \in \partial_G(x)$ and $f, g \in \partial_G(y)$. Now $g \notin \partial_G(x)$ since $d_G(x) = 2$ and $e, f \in \partial_G(x)$. So, $x \neq y$. Now we get that the edge f joins two points of S, and this is a contradiction. So, F is a matching. We can see that F is an edge cutset of $L(G)$ as follows: Let $ef \in F$. Then there is an $x \in S$ such that $e, f \in \partial_G(x)$. Now there is no path in $L(G) - F$ from the point e to the point f. ∎

Before we give our algorithm for recognizing graphs with a matching cutset, we need two more lemmas.

If e is an edge, we shall use G/e to denote the graph obtained from G by contracting the edge e (deleting e and identifying u with v. We allow the possibility that G/e may have parallel edges).

Lemma 2. Let G be a 2-connected graph that does not have a special 3-set. Suppose $\{u, v\}$ is a special 2-set in G. Let $e = uv$. Then G has a degree 2 stable cutset if and only if G/e has a degree 2 stable cutset.

Proof. Let S be a degree 2 stable cutset of G and let $t = |S \cap \{u, v\}|$. If $t = 0$ then S is a degree 2 stable cutset of G/e. If $t = 1$, say $v \in S$ and $u \notin S$ then let $w = N(u) - \{v\}$ and v' be the point obtained by identifying u with v (contracting e). The point w is not in S, otherwise $\{u, v, w\}$ would be a special 3-set of G. Let $S' = (S - v) \cup \{v'\}$. Then $S' \subseteq V(G/e)$, and $d_{G/e}(x) = 2$ for each $x \in S'$ and S' is a stable set. Let z be a point of G that is disconnected from u in $G - S$. Now $z \in V(G/e)$ and $w \in V(G/e)$, and there is no path from z to w in $(G/e) - S'$. Thus, S' is a degree 2 stable cutset of G/e.

Let S' be a degree 2 stable cutset of G/e and v' be the point obtained from G by identifying u and v. Note that if u and v have a common neighbor q in G, then $q \notin S'$. If $v' \notin S'$ then $S = S'$ is a degree 2 stable cutset of G. If $v \in S'$ then place u in the component of $(G/e) - S'$, which contains the neighbor of u other than v (in G) to get a degree 2 stable cutset of G. ∎

Lemma 3. Let G be a connected graph that does not have a special 3-set. Suppose that G has a stable set S with $d_G(x) = 2$ for each $x \in S$. Then S contains a degree 2 stable cutset of G if and only if S is a degree 2 stable cutset of G.

Proof. Let S' be a minimal degree 2 stable cutset of G that is contained in S, and let $x \in S'$. Since S' is a minimal vertex cutset, then x has neighbors in every component of $G - S'$. Let G_1 and G_2 be any two components of $G - S'$. For $i = 1, 2$, let $x_i \in N(x) \cap V(G_i)$. Suppose that $d_G(y) = 2$ for each $y \in V(G_1)$. Let $\{z\} = N(x_1) - \{x\}$. Now $z \in V(G_1)$ or $z \in S'$, and in either case $d_G(z) = 2$. Hence $\{z, x, x_1\}$ is a special 3-set. This contradiction proves that there is a point $y_1 \in V(G_1)$ such that $d_g(y_1) \neq 2$. Similarly, there is a $y_2 \in V(G_2)$ such that $d_G(y_2) \neq 2$. Now, in $G - S$ there is no path from y_1 to y_2. Thus, S disconnects G. The other direction of the proof is clearly true. ∎

The algorithm that we give below accepts as input a 2-connected line graph. The restriction of 2-connectedness is sensible since a connected graph with a cutpoint v has a matching cutset if and only if the vertices of at least one of the components of $G - v$ together with v induce a subgraph with a matching cutset.

Algorithm A

Input: A 2-connected line graph $G = (V, E)$, where $|V| \geq 4$.

Output: Either a matching cutset F of G or the message that G does not have a matching cutset.

Step 1: If G has a special 2-set $\{u, v\}$ then $\partial(\{u, v\})$ is a matching cutset of G; stop.

Step 2: Construct the graph H whose line graph is G. (If G is not K_3 or $K_{3,3}$ then it is well known that G has a unique root graph).

Step 3: For each special 2-set $\{x, y\}$ of H, contract the edge xy. Let H' be the resulting graph.

Step 4: Let S' be the set of points of degree 2 in H'. If $H' - S'$ is disconnected, then let T be a set obtained by selecting, for each vertex q of S', a vertex in the preimage of q in H. Let S be any minimal vertex cutset of H that is contained in T, and $F = \{ef \in E(G)$: there exists an $x \in S$ with $e, f \in \partial_H(x)\}$; stop. If $H' - S'$ is connected then G does not have a matching cutset; stop.

Theorem 2. Let $G = (V, E)$ be a line graph. Then we can determine in $O(|E|)$ time whether G has a matching cutset.

Proof. First suppose that G is 2-connected, and apply Algorithm A to G. If G has a special 2-set then the algorithm stops in Step 1. Otherwise, control proceeds to Step 2, and by Lemma 1, it is enough to decide whether the graph H, which is constructed in Step 2, has a degree 2 stable cutset. Since we now have the case that G does not have a special 2-set, then H does not have a special

3-set. Hence, by Lemma 2, we can contract all edges $e = xy$ where $\{x, y\}$ is a special 2-set in H. Lemmas 1 and 3 establish the correctness of Step 4.

We now consider a bound on the complexity of the algorithm. If G is 2-connected, then Algorithm A will accept G directly as input. Step 1 takes $O(|E|)$ work. An algorithm of Philippe G. H. Lehot [4] can be used to execute Step 2 in $O(|E|)$ time. Step 3 takes $O(|E|)$ work. The set S' of Step 4 can be found by examining each point of H' once, that is, $O(|E|)$ work. Thus, when G is 2-connected, the whole algorithm takes $O(|E|)$ work.

Suppose now that G is not 2-connected. We have said that it is enough to apply the algorithm to subgraphs induced by the vertices of the 2-connected components together with the relevant cut-points. Searching for the 2-connected components of G can be done in $O(|E|)$ work by using depth first search. Algorithm A will then be applied at most once to each resulting subgraph. This gives a complexity of $O(|E|)$. ∎

4. QUADRANGULATED GRAPHS

A graph G is quadrangulated if it does not have a chordless circuit of length 5 or more. Thus, every triangulated graph is quadrangulated, but there are quadrangulated graphs that are not triangulated (a 4-cycle is an example). We shall prove in this section that matching cut can be solved in polynomial time when the input is restricted to the class of quadrangulated graphs. In fact, we show that given a quadrangulated graph G and an edge e of G, we can determine in polynomial time whether G has a matching cutset that contains the edge e.

By a *partial decomposition coloring* of G we shall mean a coloring of some of the points of G with two colors so that for each point v, at most one neighbor of v is given a color that is different from the color of v. The motivation for this terminology is that Chvátal [2] calls a graph *decomposable* if its points can be colored red and blue so that both colors are assigned, and for each $v \in V$, at most one neighbor of v is given a color that is different from that of v. If $e = uv$, the algorithm that we shall give colors u red and v blue. It then colors a point t red (respectively blue) only if red (respectively blue) is the color that must be assigned to t by every partial decomposition coloring of G that colors u red and v blue. If the algorithm finds a point that must be recolored, it halts with the message that G does not have a minimal matching cutset containing the edge e. The algorithm may (correctly) arrive at the conclusion that G has a matching cutset containing e before it colors all the points of G.

Algorithm B

Input: A simple 2-edge connected quadrangulated graph $G(V, E)$ and an edge $e = uv$.

Output: Either a minimal matching cutset F containing e or the message that G does not have a minimal matching cutset that contains e.

Step 0: Color u red and v blue and let $\alpha: V \rightarrow \{0, 1, 2\}$ be given by

$$\alpha(t) = \begin{cases} 0, & \text{if } t \text{ is not colored;} \\ 1, & \text{if } t \text{ is a red point;} \\ 2, & \text{if } t \text{ is a blue point.} \end{cases}$$

Step 1: If there is no point x with a neighbor y such that $\alpha(y) \neq 0$, $\alpha(x) \neq \alpha(y)$, and y has a neighbor z with $0 \neq \alpha(z) \neq \alpha(y)$, then go to Step 4.

Step 2: If $\alpha(x) \neq 0$, then go to Step 3. Otherwise, put $\alpha(x) = \alpha(y)$ and go to Step 1.

Step 3: G does not have a matching cutset containing e; halt.

Step 4: The set $F = \{fg \in E: \alpha(f) \neq 0, \alpha(g) \neq 0, \text{ and } \alpha(f) \neq \alpha(g)\}$ is a matching cutset containing e; halt.

Lemma 4. If Algorithm B halts in Step 4, then the set F produced by the algorithm is a matching.

Proof. If not, then let ab and ac be edges of F, where $b \neq c$. By definition of F, neither $\alpha(a)$, $\alpha(b)$, and $\alpha(c)$ is 0. We can assume that $\alpha(a) = 1$. (The case when $\alpha(a) = 2$ is completely similar.) We thus have $\alpha(b) = \alpha(c) = 2$. Now, control can only go to Step 4 from Step 1. But in Step 1, with $x = b$, $y = a$, and $z = c$, we see that control should have passed from Step 1 to Step 2 instead of going to Step 4. ∎

Lemma 5. If Algorithm B halts in Step 3, then G does not have a minimal matching cutset F such that $e = uv \in F$.

Proof. Suppose that the algorithm halts in Step 3 and yet G has a minimal matching cutset F containing e. Let G_1 and G_2 be two components of $G - F$ such that $u \in V(G_1)$ and $v \in V(G_2)$ with $V(G) = V(G_1) \cup V(G_2)$. Let the vertex labels 0, 1, and 2 stand for vertices that are not colored, vertices colored red, and vertices colored blue, respectively. Define a vertex coloring map γ: $V \rightarrow \{1, 2\}$ by

$$\gamma(t) = \begin{cases} 1, & \text{if } t \in V(G_1); \\ 2, & \text{if } t \notin V(G_1). \end{cases}$$

Then γ is a decomposition coloring of G. We claim that $\alpha(t) = \gamma(t)$ for each t such that $\alpha(t) \neq 0$. This is certainly the case when Algorithm B has colored only the points u and v.

Suppose the claim is true when Algorithm B has colored $k - 1$ points, $k \geq 3$, and let x be the kth point that is colored by Algorithm B. Suppose that x is colored red. (The case when x is colored blue is similar.) Then Algorithm B found, in Step 1, a path $P = (x, y, z)$ where $\alpha(y) = 1$, $\alpha(z) = 2$, and

$\alpha(x) \neq \alpha(y)$. By induction, $\gamma(y) = 1$ and $\gamma(z) = 2$. Since γ is a decomposition coloring of G, it must then color x red. Hence the claim is true.

Now, control passes to Step 3 because Algorithm B finds a point a that is colored blue (respectively red) such that a neighbor b of a is colored red (respectively blue), and b has a neighbor c that is colored blue (respectively red). Now γ agrees with the coloring produced by the algorithm on points a, b, and c. This contradicts the fact that γ is a decomposition coloring of G. ∎

Theorem 3. Recognizing decomposable graphs can be done in $O(|V|\ |E|^3)$ work if the input is restricted to simple graphs without a chordless circuit of length 5 or more.

Proof. Apply Algorithm B using different choices of edges e until Algorithm B halts in Step 4 or until all possible choices of edges have been tried.

If the algorithm halts in Step 3, then by Lemma 5 no matching cutset containing e exists in G. Suppose the algorithm halts in Step 4. Then by Lemma 4, the set F produced by the algorithm is a matching. We show that F is an edge cutset.

Let V_1 and V_2 be the sets of points colored red and blue, respectively. If G-F is connected, then let P be a shortest path in G-F from a red point to a blue point. Note that P is incident with at least one noncolored point, and that no edge of P joins a red point with a blue point. Moreover, $|V(P) \cap V_1| = |V(P) \cap V_2| = 1$. Let $\{x\} = V(P) \cap V_1$ and $\{y\} = V(P) \cap V_2$. Observe that $G[V_1 \cup V_2]$ is a connected graph. Let Q be a shortest path in $G[V_1 \cup V_2]$ from x to y. Let C be the circuit formed by P together with Q. From the choices of P and Q, we get that C is a chordless circuit. Let z be the noncolored point of C that is adjacent to x. By Step 1 of Algorithm B, x does not have a blue neighbor. Let x_1 be the neighbor of x on Q. Then x_1 is a red point. Let z_1 be the neighbor of z on P other than x. Note that z_1 is not a red point. If z_1 is not a colored point, then C has at least two red points (x and x_1), two noncolored points (z and z_1), and at least one blue point. Thus $|V(C)| \geq 5$ and C is chordless, which is a contradiction.

We can now assume that z_1 is a blue point. By Step 1 of Algorithm B, z_1 does not have a red neighbor. Hence the neighbor of z_1 on Q is a blue point. Thus, C has at least two blue points, two red points, and at least one noncolored point. Hence $|V(C)| \geq 5$ and C is chordless, which is a contradiction.

In both cases, therefore, we get that G-F is disconnected.

Each execution of Step 1 takes $O(|E|^2)$ work. For each edge, Step 1 may be executed at most $|V|$ times. The algorithm needs to be run at most $|E|$ times, and this gives a total of $O(|V|\ |E|^3)$ work. ∎

Note that the time bound on Algorithm B that is given in Theorem 3 simply shows that Algorithm B runs in polynomial time. We do not claim that $O(|V|\ |E|^3)$ is the best possible bound on the running time of Algorithm B. Indeed, by maintaining a *waiting list* of edges yz with $O \neq \alpha(y) \neq \alpha(z)$ and repeatedly instantiating x to be the neighbors of the *current* edge yz, followed by

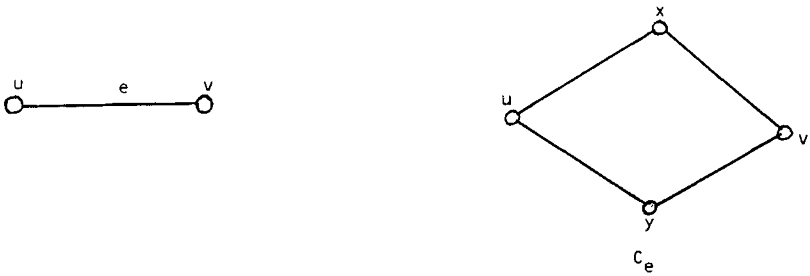

FIGURE 1

checking at each vertex for the exit to Step 3 and for additions to the waiting list, a running time of $O(|V|^3|E|)$ can be achieved.

Remarks. We remark that recognizing graphs with a matching cutset remains an NP-complete problem even if the input is restricted to bipartite graphs with one side of the bipartition consisting only of points of degree 2. This can be seen as follows:

Given $G = (V, E)$ we construct from it a graph $G' = (V', E')$ by replacing each edge e of G by a 4-cycle C_e, as shown in Figure 1. G' has two types of points: *old* points (points of G) and *new* points (points of G' that are not points of G). In Figure 1, u, v are old points and x, y are new points. Every edge of G' joins an old point to a new point, and so the vertices of G' can be properly 2-colored by the colors old and new. Hence G' is a bipartite graph. Moreover, all new points have degree 2. It is easy to verify that G has a matching cutset if and only if G' has a matching cutset.

AKNOWLEDGMENT

I am very thankful to the referee for pointing out errors in earlier versions of the paper. I also wish to thank Dr. B. C. Mortimer of the Department of Mathematics and Statistics, Carleton University, for useful discussions that I had with him on matching cutsets.

References

[1] C. Berge, *Graphs* (2nd rev. ed., Vol. 6, Part 1). North Holland Mathematical Library (1985).
[2] V. Chvátal, Recognizing decomposable graphs. *J. Graph Theory* **8** (1984) 51–53.
[3] R. L. Graham, On primitive graphs and optimal vertex assignments. *NY Acad. Sci.* **175** (1970) 170–186.

[4] Philippe G. H. Lehot, An optimal algorithm to detect a line graph and output its root graph. *J. Assoc. Comput. Mach.* **21** (1974) 569–575.
[5] A. C. M. van Rooij and H. S. Wilf, The interchange graph of a finite graph. *Acta Math. Acad. Sci Hungar.* **16** (1975) 263–269.