2000

Recognizing Perfect 2-Split Graphs

Chính T. Hoàng
Wilfrid Laurier University, choang@wlu.ca

Van Bang Le
Universität Rostock

SIAM J. DISCRETE MATH.
Vol. 13, No. 1, pp. 48–55

RECOGNIZING PERFECT 2-SPLIT GRAPHS*

CHÍNH T. HOÀNG† AND VAN BANG LE‡

Abstract. A graph is a split graph if its vertices can be partitioned into a clique and a stable set. A graph is a k-split graph if its vertices can be partitioned into k sets, each of which induces a split graph. We show that the strong perfect graph conjecture is true for 2-split graphs and we design a polynomial algorithm to recognize a perfect 2-split graph.

Key words. graph coloring, perfect graph

AMS subject classification. 05C15

PII. S0895480197329089

1. Introduction. A graph is a *split* graph if its vertices can be partitioned into a clique and a stable set. Split graphs were studied in [14] and [11]. Split graphs form a class of perfect graphs and can be recognized in polynomial time. A graph G is *perfect* if for each induced subgraph H of G the chromatic number of H equals the number of vertices in a largest clique of H. The strong perfect graph conjecture (SPGC), proposed by Berge [1], states that a graph is perfect if and if only it does not contain as an induced subgraph the odd chordless cycle with at least five vertices, called *odd hole*, or the complement of such a cycle, called *odd antihole* (for more information on perfect graphs, see [12] and [2].) Nowadays, graphs containing no odd holes and no odd antiholes are called *Berge* graphs. It is not known whether perfect graphs and Berge graphs can be recognized in polynomial time.

The purpose of this paper is to study a generalization of split graphs and its relation to the SPGC. We shall call a graph a k*-split* graph if its vertices can be partitioned into k sets, each of which induces a split graph. The graph C_5 shows that 2-split graphs are not necessarily perfect. We will show that the SPGC holds for 2-split graphs and we will design a polynomial algorithm for recognizing perfect 2-split graphs.

THEOREM 1. *A* 2*-split graph is perfect if and only if it is Berge.*

In the remainder of this section, we shall prove Theorem 1. In the next section, we shall show that perfect 2-split graphs can be recognized in polynomial time.

A graph is *minimal imperfect* if it is not perfect but each of its proper induced subgraphs is. The proof of Theorem 1 relies on several known results on minimal imperfect graphs. First, Lovász [16] proved that

(1) every minimal imperfect graph G has exactly $\omega(G) \cdot \alpha(G) + 1$ vertices,

where $\omega(G)$, respectively $\alpha(G)$, denotes the number of vertices in a largest clique, respectively, stable set, of G. Equation (1) implies that

(2) a graph is perfect if and only if it its complement is.

*Received by the editors October 20, 1997; accepted for publication (in revised form) May 7, 1999; published electronically January 13, 2000. This research was supported by NSERC.
http://www.siam.org/journals/sidma/13-1/32908.html

†School of Mathematical Sciences, Lakehead University, Thunder Bay, Ontario, Canada P7B-5E1 (chinh.hoang@lakeheadu.ca).
‡Fachbereich Informatik, Universität Rostock, Rostock, Germany (le@informatik.uni-rostock.de).

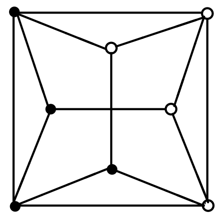

FIG. 1. *A perfect 2-split non-quasi-parity graph.*

Next, a result of Tucker [19] shows that

(3) $\omega(G) \geq 4$ and $\alpha(G) \geq 4$ for every minimal imperfect Berge graph G.

Theorem 1 is a consequence of (1) and (3).

Proof of Theorem 1. The "only if" part is trivial. We shall prove the "if" part by contradiction. Assume that there is an imperfect Berge graph G satisfying the hypothesis of the theorem. Since G contains an induced subgraph that is minimal imperfect, we may, without loss of generality, assume that G is minimal imperfect. Let T be a set of vertices in G such that each of T and $V(G) - T$ induces a split graph. Let T be partitioned into a clique C and a stable set S. Then

$$|T| = |C| + |S| \leq \omega(G) + \alpha(G).$$

Similarly,

$$|G - T| \leq \omega(G) + \alpha(G).$$

Hence by (1),

$$\omega(G)\alpha(G) + 1 = |G| \leq 2(\omega(G) + \alpha(G)).$$

This and (3) give

$$\begin{aligned} 1 &\leq 2\left(\frac{1}{\alpha(G)} + \frac{1}{\omega(G)}\right) - \frac{1}{\omega(G)\alpha(G)} \\ &< 2\left(\frac{1}{\alpha(G)} + \frac{1}{\omega(G)}\right) \\ &\leq 1, \end{aligned}$$

which is a contradiction. $\square$

The class of perfect 2-split graphs contains all split graphs, all bipartite graphs, and their complements. Therefore it is not contained in BIP* [8] and not in the class of strongly perfect graphs [3]. Meyniel [17] established perfectness for a large class of graphs, the class of "quasi-parity" graphs. An *even pair* is a pair of vertices such that all chordless paths joining them have an even number of edges. A graph G is a *quasi-parity graph* if for each induced subgraph H of G either H or its complement contains an even pair. The graph in Figure 1 is a perfect 2-split graph that is not a quasi-parity graph.

2. Recognition algorithm. In this section, we shall show that perfect 2-split graphs can be recognized in polynomial time. As usual, n and, respectively, m denote the number of vertices, respectively edges, of a graph G. A (k, l) partition of a graph G is a partition of its vertices into k stable sets and l cliques. A graph is a (k, l) graph if its vertices admit a (k, l) partition. Thus k-split graphs are (k, k) graphs. (k, l) graphs were first studied by Brandstädt [5], who remarked that deciding whether a graph is a (k, l) graph is NP-complete whenever $k \geq 3$ or $l \geq 3$. However, he showed [4] that finding a $(2, 1)$ partition (respectively, $(2, 2)$ partition) in a graph can be done in $O(n^2m)$ (respectively, $O(n^{10}m)$) time. Brandstädt pointed out that the algorithms he gave in [5] are incorrect; however, his algorithms in [4] are correct. Brandstädt, Le, and Szymczak [6] later designed an $O((n+m)^2)$ algorithm to recognize (2,1) graphs. Recently, Feder et al. [10] gave a new algorithm to find a $(2, 2)$ partition of a graph, if it exists; their algorithm also runs in $O(n^{10}m)$ time.

We shall show that an odd hole in a $(2, 2)$ graph G, if it exists, can be found in $O(n^4m^3)$ time. Since the complement of a $(2, 2)$ graph is again a $(2, 2)$ graph, an odd antihole can also be detected in polynomial time. Thus, to determine if a given graph G is a perfect 2-split graph, we first use the algorithm of Brandstädt to construct a (2,2) partition of G (if such a partition exists) and then use our algorithm to test for the existence of an odd hole or antihole in G. This shows that perfect 2-split graphs can be recognized in polynomial time.

A chordless path, respectively cycle, on k vertices is denoted by P_k, respectively C_k. The *length* of a path or a cycle is the number of its edges. We often write P_k $x_1x_2 \cdots x_k$ for the path on vertices $x_1, x_2, \ldots, x_k$ and edges x_ix_{i+1} $(1 \leq i < k)$; x_1 and x_k are called *end-points* of P. For the path P_4 $x_1x_2x_3x_4$, the vertices x_2, x_3 are called *midpoints*, the edges x_1x_2, x_3x_4 are called *wings* of that P_4, and the edge x_2x_3 is called the *rib* of that P_4. By $N_G(x)$ we denote the set of vertices adjacent to vertex x in G; when there can be no confusion, we shall write $N(x) = N_G(x)$.

Let G be a graph on n vertices and m edges. To analyze our algorithm, we shall need an upper bound on the number of P_k's in G for a certain number k. Since each edge ab can extend into a P_3 in at most n ways, the number of P_3's in G is of order $O(nm)$. Since each edge ab can be the rib of at most n^2 P_4's, the number of P_4 in G is of order $O(n^2m)$. Similarly, the number of P_6's is of order $O(n^4m)$.

The remainder of this section is devoted to showing that an odd hole (if it exists) in a $(2, 2)$ graph can be detected in polynomial time. An (x, y)-path is an induced path whose end-points are vertices x and y. The parity of a path is the parity of its number of edges. We note that the following two problems can obviously be solved in $O(n + m)$ time.

Problem Odd Path

Input:	A $(2, 0)$ graph G, and two nonadjacent vertices x, y in G.
Question:	Is there an odd induced (x, y)-path in G?

Problem Even Path

Input:	A $(2, 0)$ graph G, and two nonadjacent vertices x, y in G.
Question:	Is there an even induced (x, y)-path in G?

Finding an odd hole in (2,1) graphs. We shall describe an algorithm for finding an odd hole in a (2,1) graph G. Let the vertices of G be partitioned into stable sets S_1, S_2, and a clique C_1. If G contains an odd hole H, then H must contain one or two vertices in C_1.

Step 1. *Look for an odd hole H containing one vertex in C_1.*

For each vertex in C_1, list all P_3's containing it as a midpoint. For each P_3 abc with $b \in C_1, a, c \in S_1 \cup S_2$, abc extends into an odd hole if and only if there is an odd induced (a, c)-path in the (2,0) subgraph of G induced by $S_1 \cup S_2 - (N_G(b) - \{a, c\})$.

Since the number of P_3's in G is of order $O(nm)$, the complexity of this step is $O(nm^2)$.

Step 2. *Look for an odd hole H containing two vertices in C_1.*

For each pair of vertices b, c in C_1, list all P_4's containing them as midpoints. For each P_4 $abcd$ with $b, c \in C_1, a, d \in S_1 \cup S_2$, $abcd$ extends into an odd hole if and only if there is an even induced (a, d)-path in the (2,0) subgraph of G induced by $S_1 \cup S_2 - ((N_G(b) \cup N_G(c)) - \{a, d\})$.

Since the number of P_4's in G is of order $O(n^2m)$, the complexity of this step is $O(n^2m^2)$. Thus the complexity of finding an odd hole in a (2,1) graph is $O(n^2m^2)$.

Finding an odd hole in (2,2) graphs. Let the vertices of G be partitioned into stable sets S_1, S_2 and cliques C_1, C_2. If there is an odd hole in $S_1 \cup S_2 \cup C_i$, for $i = 1, 2$, then we can find it using the previous algorithm. So we may assume that there is no odd hole in $S_1 \cup S_2 \cup C_i$, for $i = 1, 2$. Now, any odd hole H in G must contain some vertex in C_1 and some vertex in C_2; we let H_i denote the set of vertices in $H \cap C_i$ for $i = 1, 2$.

First, we can look for a C_5 in G in $O(n^5)$ (actually, we can do slightly better). Thus we may assume that G has no C_5.

Step 1. *Look for an odd hole H such that all the vertices in $H' = H_1 \cup H_2$ appear consecutively in H.*

In this case H' induces a P_k $v_1 \dots v_k$ $(k = 2, 3, 4)$ in G. Thus we need only to test if this P_k extends into an odd hole. For each induced path P $xv_1 \dots v_ky$ with $x, y \in S_1 \cup S_2$, P extends into an odd hole if and only if there is an even (respectively, odd) induced (x, y)-path in the subgraph of G induced by $S_1 \cup S_2 - ((N_G(v_1) \cup N_G(v_2) \dots \cup N_G(v_k)) - \{x, y\})$ if k is even (respectively, odd).

Since the number of P_6's in G is of order $O(n^4m)$, the complexity of this step is $O(n^4m^2)$.

Step 2. *Look for an odd hole H such that not all vertices in $H' = H_1 \cup H_2$ appear consecutively in H.*

We shall test (i) whether there is a P_3 abc with $b \in C_i, a, c \in S_1 \cup S_2$, such that abc extends into an odd hole; and (ii) whether there is a P_4 $abcd$ with $b, c \in C_i, a, d \in S_1 \cup S_2$, such that $abcd$ extends into an odd hole.

A P_3 abc with $b \in C_i, a, c \in S_1 \cup S_2$ is of type 1 if $a, c \in S_j$ for $j = 1, 2$; otherwise the P_3 is of type 2. A P_4 $abcd$ with $b, c \in C_i, a, d \in S_1 \cup S_2$ is of type 1 if $a \in S_i, d \in S_j, i \neq j$; otherwise the P_4 is of type 2.

Substep 2.1. For each P_3 abc of type 1 with $b \in C_i, a, c \in S_j$, we can test if it extends into an odd hole in the following way. If abc does not extend into a P_4, then it does not extend into an odd hole. Now, consider a P_4 of the form $abcd$. We must have $d \in C_{i'} \cup S_{j'}, i \neq i', j \neq j'$. Let F be the subgraph of G induced by $C_{i'} \cup S_1 \cup S_2 - (N_G(b) \cup N_G(c) - \{a, d\})$ and let F' be the graph obtained from F by adding the edge ad. Since G contains no C_5, the P_4 $abcd$ extends into an odd hole in G if and only if there is an odd hole in F' (this odd hole must contain the edge ad by our assumption that $C_{i'} \cup S_1 \cup S_2$ contains no odd hole).

Since F' is a (2,1) graph, this problem can be solved in $O(n^2m^2)$. Thus the complexity of this substep is $O(n^3m^3)$.

Substep 2.2. For each P_4 $abcd$ of type 1 with $b, c \in C_i, a \in S_1, d \in S_2$, let F be the subgraph of G induced by $C_{i'} \cup S_1 \cup S_2 - (N_G(b) \cup N_G(c) - \{a, d\}), i \neq i'$, and let F' be the graph obtained from F by adding the edge ad. Since G contains no C_5, the P_4 $abcd$ extends into an odd hole in G if and only if there is an odd hole in F'.

Since F' is a (2,1) graph, this problem can be solved in $O(n^2m^2)$. Thus the complexity of this substep is $O(n^4m^3)$.

Now, we may assume that no P_3 and no P_4 of type 1 extend into a odd hole of G. We claim that

$$G \text{ contains no odd hole.}$$

Suppose that G contains an odd hole H. Since we are in Step 2, the vertices in $H \cap (C_1 \cup C_2)$ do not appear consecutively on H. Enumerate the vertices of H as $x_1, \ldots, x_t, b_1, \ldots, b_u, y_1, \ldots, y_r, a_1, \ldots, a_s$ in the cyclic order with $x_i \in C_1$ for each i, $b_i, a_i \in (S_1 \cup S_2)$ for each i, and $y_i \in C_2$ for each i. Define Q_1 to be the path $a_s x_1 \ldots x_t b_1$ and Q_2 to be the path $b_u y_1 \ldots y_r a_1$. Q_1 and Q_2 are a P_3 or P_4 of type 2.

Suppose that Q_1 is a P_3 of type 2. If Q_2 is a P_3 of type 2, then the path $b_1 \ldots b_u$ has the same parity as the path $a_1 \ldots a_s$. However, then H has even length, which is a contradiction. Thus Q_2 must be a P_4 of type 2. Now, the path $b_1 \ldots b_u$ has different parity than the path $a_1 \ldots a_s$; then H has even length, which is a contradiction.

Thus we may assume that Q_1, and by symmetry Q_2, is a P_4 of type 2. Now, it is easy to see that the path $b_1 \ldots b_u$ has the same parity as the path $a_1 \ldots a_s$. However, then H has even length, which is a contradiction. Thus, G cannot contain an odd hole as claimed. □

Optimizing perfect and nonperfect 2-split graphs. Grötschel, Lovász, and Schrijver [13] designed a polynomial algorithm to find a largest clique and an optimal coloring of a perfect graph. This important algorithm is a variation of the ellipsoid method for linear programming and is unlike a typical combinatorial algorithm. Thus, one may ask for a more combinatorial algorithm. With this in mind we shall comment on the problem of optimizing 2-split graphs.

Let G be a graph with a (2,2) partition C_1, C_2, S_1, S_2, where the C_i's are cliques and the S_i's are stable sets. Since we can optimally color $C_1 \cup C_2$, we can color G with at most $\chi(G) + 2$ colors ($\chi(G)$ denotes the chromatic number of G). We do not know if there is a combinatorial algorithm to optimally color a perfect 2-split graph.

We shall show that a largest clique of a 2-split graph can be found in polynomial time. It is well known that there is a combinatorial polynomial algorithm for finding a largest stable set and a minimum clique cover of a bipartite graph (for example, see [7, pp. 331–336]). We shall assume that the following function can be computed in polynomial time.

function Find-Omega-Bip(G)
  input: a graph G that is the complement of a bipartite graph.
  output: a number Find-Omega-Bip(G)=$\omega(G)$.

The clique number of a (2,2) graph can be computed in polynomial time by the following function.

function Find-Omega(G)
  input: a (2,2) graph G with partition C_1, C_2, S_1, S_2, where the C_i's are cliques and the S_i's are stable sets.
  output: a number Find-Omega(G)=$\omega(G)$.
begin
max := Find-Omega-Bip($C_1 \cup C_2$);
for each vertex $b \in S_1 \cup S_2$ **do**
  begin

```
        k := Find-Omega-Bip (N(b) ∩ (C_1 ∪ C_2));
        if k + 1 > max then max := k + 1;
    end;
for each edge ab with a, b ∈ S_1 ∪ S_2, do
    begin
        k := Find-Omega-Bip (N(a) ∩ N(b) ∩ (C_1 ∪ C_2));
        if k + 2 > max then max := k + 2;
    end;
return max;
end.
```

We are going to show that function Find-Omega correctly computes the clique number of G. Let $\omega(C_1 \cup C_2) = k$. Obviously, we have $k \leq \omega(G) \leq k+2$. If $\omega(G) = k$, then the value returned by the function is obviously correct. If $\omega(G) = k+1$, then any largest clique of G must contain a vertex in $B = S_1 \cup S_2$; such a clique can be found by finding, for each vertex $b \in B$, a largest clique in $N(b) \cap (C_1 \cup C_2)$. If $\omega(G) = k+2$, then any largest clique of G must contain two adjacent vertices in $B = S_1 \cup S_2$; such a clique can be found by finding, for each pair of adjacent vertices $b_1, b_2 \in B$, a largest clique in $N(b_1) \cap N(b_2) \cap (C_1 \cup C_2)$. Thus function Find-Omega always returns the correct value.

3. Perfect 3-split graphs. We have shown that the SPGC holds for 2-split graphs and we provided a polynomial algorithm for recognizing perfect 2-split graphs. A natural extension of this result would be to show that the SPGC holds for 3-split graphs and, more generally, k-split graphs for any fixed k. We shall show that a minimal imperfect Berge k-split graph cannot have many vertices. When there can be no confusion we shall write $\omega = \omega(G)$ and $\alpha = \alpha(G)$.

LEMMA 2. *For any $k \geq 2, l \geq 2$, any minimal imperfect (k, l) graph has at most $k\alpha + l\omega - r$ vertices where $r = \min\ (k, l)$.*

We shall need a result of Padberg [18], who showed that

(4) if G is a minimal imperfect graph, then any ω-clique of G is disjoint from precisely one α-stable set of G, and vice versa.

Proof of Lemma 2. Suppose that G is a minimal imperfect (k, l) graph. Let G be partitioned into k stable sets $S_1, \ldots, S_k$ and l cliques $K_1, \ldots, K_l$. Then $|V(G)| = \sum |S_i| + \sum |K_i|$.

If no stable set S_i is a maximum stable set and no clique K_i is a maximum clique, then the lemma holds trivially.

Suppose that some two cliques, say, K_1 and K_2, are ω-cliques. Then no stable set S_i can be an α-stable set; otherwise S_i is disjoint from K_1 and K_2, which is a contradiction to (4). Therefore, we have $|V(G)| = \sum |S_i| + \sum |K_i| \leq k(\alpha - 1) + l\omega$. Similarly, if two stable sets S_i and S_j are α-stable sets, then we have $|V(G)| \leq k\alpha + l(\omega - 1)$. Thus the lemma holds in this case.

Now, assume that exactly one clique of $\{K_1, \ldots, K_l\}$ is a ω-clique and exactly one stable set of $\{S_1, \ldots, S_k\}$ is an α-stable set. We have, therefore, $|V(G)| \leq \alpha + (k-1)(\alpha - 1) + \omega + (l-1)(\omega - 1) = k\alpha + l\omega - (k+l) + 2$. Since $k \geq 2$ and $l \geq 2$, the lemma also holds in this case. □

THEOREM 3. *3-split Berge graphs are perfect if the SPGC holds for graphs with at most* 33 *vertices. Furthermore,* $(2, 3)$ *Berge graphs and* $(3, 2)$ *Berge graphs are perfect.*

Proof of Theorem 3. Let G be a 3-split Berge graph and suppose that G is not perfect. Then G contains a minimal imperfect graph and so, without loss of generality, we may assume that G is minimal imperfect. By (1) and Lemma 2, we have

$$|V(G)| = \alpha\omega + 1 \leq 3(\alpha + \omega) - 3. \tag{5}$$

By replacing G with its complement if necessary, we may assume that $\alpha \geq \omega$. Now (5) gives $\alpha\omega + 1 \leq 6\alpha - 3$ and so $\omega \leq 5$. This and (3) imply

$$4 \leq \omega \leq 5. \tag{6}$$

From (5) and (6), it is easy to verify that the largest value that $|V(G)|$ can have is 33 and this occurs when $\alpha = 8$ and $\omega = 4$ (for example, if $\omega = 4$ and $\alpha = 9$, then $\alpha\omega + 1 = 37$ but $3(\alpha + \omega) - 3 = 36$, contradicting (5); similarly, when $\omega = 5$ we must have $\alpha = 5$ and so $|V(G)| = \alpha\omega + 1 = 26$).

Similarly, for the case $k = 2$, $l = 3$, we have $|V(G)| \leq 21$. This occurs when $\omega = 5, \alpha = 4$. (2,3) Berge graphs are perfect because the SPGC holds for graphs with at most 24 vertices. (This unpublished proof is due to V. A. Gurvich and V. M. Udalov, who announced it at the Perfect Graph Conference in Princeton, 1993, and is also reported in [9]; [15] gives a proof that the SPGC holds for graphs with at most 20 vertices.) By (2), (3,2) Berge graphs are also perfect. □

The complexity of recognizing perfect (3,2) graphs. We shall now comment on the difficulty of recognizing Berge (3,2) graphs. Tucker [19] proved that Berge K_4-free graphs are perfect; in other words, they are 3-colorable. The problem of recognizing Tucker's graphs is still open. 3-colorable graphs are (3,0) graphs. However, determining if a graph is a $(3, l)$ graph is at least as hard as determining if a graph is a (3,0) graph, for any $l \geq 1$. To see this, consider a graph G and a graph H that is the union of G and l vertex-disjoint cliques on at least four vertices. Clearly, G is a (3,0) graph if and only if H is a $(3, l)$ graph, and G is Berge if and only if H is. Thus, recognizing Berge (3,2) graphs is at least as hard as recognizing Berge K_4-free graphs. Thus, our algorithm for recognizing Berge (2,2) graphs seems to be the best possible (in the sense that it is polynomial) given the current state of knowledge on perfect graphs.

REFERENCES

[1] C. Berge, *Les problèmes de coloration en théorie des graphes*, Publ. Inst. Stat. Univ. Paris, 9 (1960), pp. 123–160.

[2] C. Berge and V. Chvátal, eds., *Topics on Perfect Graphs*, North-Holland, Amsterdam, 1984.

[3] C. Berge and P. Duchet, *Strongly perfect graphs*, in Topics on Perfect Graphs, C. Berge and V. Chvátal, eds., North-Holland, Amsterdam, 1984, pp. 57–61.

[4] A. Brandstädt, *Partitions of graphs into one or two independent sets and cliques*, Report 105, Informatik Berichte, Fern Universität, Hagen, Germany, 1991.

[5] A. Brandstädt, *Partitions of graphs into one or two independent sets and cliques*, Discrete Math., 152 (1996), pp. 47–54. See also Corrigendum, 186 (1998), p. 295.

[6] A. Brandstädt, V. B. Le, and T. Szymczak, *The complexity of some problems related to graph 3-colorability*, Discrete Appl. Math., 89 (1998), pp. 59–73.

[7] V. Chvátal, *Linear Programming*, W. H. Freeman, New York, 1983.

[8] V. Chvátal, *Star-cutsets and perfect graphs*, J. Combin. Theory Ser. B, 39 (1985), pp. 189–199.

[9] V. Chvátal, *Problems Concerning Perfect Graphs*, Department of Computer Science, Rutgers University, 1993, manuscript.

[10] T. Feder, P. Hell, S. Klein, and R. Motwani, *Complexity of Graph Partition Problems*, manuscript.

[11] S. Foldes and P. L. Hammer, *Split graphs*, in Proceedings 8th Southeastern Conf. on Combinatorics, Graph Theory and Computing, Louisiana State University, Baton Rouge, LA, 1977, Congress. Numer. 19, Utilitas Math., Winnipeg, Manitoba, 1977, pp. 311–315.

[12] M. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.

[13] M. Grötschel, L. Lovász, and A. Schrijver, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica, 1 (1981), pp. 169–197.

[14] P. L. Hammer and B. Simeone, *The splittance of a graph*, Research Report CORR 77-39, Department of Combinatorics and Optimization, University of Waterloo, Ontario, Canada, 1977.

[15] C. W. H. Lam, S. Swiercz, L. Thiel, and E. Regener, *A computer search for* (α, ω)*-graphs*, in Proc. 9th Manitoba Conference on Numerical Mathematics and Computation, Congress. Numer. 27, Utilitas Math., Winnipeg, Manitoba, 1980, pp. 285–289.

[16] L. Lovász, *A characterization of perfect graphs*, J. Combin. Theory Ser. B, 13 (1972), pp. 95–98.

[17] H. Meyniel, *A new property of critical imperfect graphs and some consequences*, European J. Combin., 8 (1987), pp. 313–316.

[18] M. Padberg, *Perfect zero-one matrices*, Math. Programming, 6 (1974), pp. 180–196.

[19] A. Tucker, *Critical perfect graphs and perfect* 3*-chromatic graphs*, J. Combin. Theory Ser. B, 23 (1977), pp. 143–149.