Graphs and Combinatorics (2001) 17:745–757

Graphs and Combinatorics

Recognizing Planar Strict Quasi-Parity Graphs*

Cláudia Linhares Sales[1†], Frédéric Maffray[2], and Bruce Reed[3]

[1] Universidade Federal do Ceará, Rua Deputado Moreira da Rocha 380, 601, Meireles, Fortaleza-CE, CEP 60160-060, Brazil. e-mail: linhares@lia.ufc.br
[2] CNRS, Laboratoire Leibniz-IMAG, 46 avenue Félix Viallet, 38031 Grenoble Cedex, France
[3] CNRS, Equipe Combinatoire, Université Pierre et Marie Curie, 4 place Jussieu, 75005 Paris, France

Abstract. A graph is a strict-quasi parity (SQP) graph if every induced subgraph that is not a clique contains a pair of vertices with no odd chordless path between them (an "even pair"). We present an $O(n^3)$ algorithm for recognizing planar strict quasi-parity graphs, based on Wen-Lian Hsu's decomposition of planar (perfect) graphs and on the (non-algorithmic) characterization of planar minimal non-SQP graphs given in [9].

Key words. Perfect graphs, Even pairs, Planar graphs, Strict quasi-parity graphs, Recognition

1. Introduction

A graph G is *perfect* if for each induced subgraph H of G, the chromatic number of H is equal to the maximum size of its cliques. Perfect graphs were defined by Claude Berge in the early 1960s (see [1]) and since then they have become an exciting subject of research in Graph Theory. One of their most attractive properties is that some problems which are *NP-complete* in general become *polynomially solvable* for perfect graphs [4], namely the minimum coloring problem and the maximum clique problem (actually, for perfect graphs these two problems are equivalent). However, a good characterization of perfect graphs remains an open question. Let a *hole* be a chordless cycle with at least five vertices and an *antihole* be the complement of a hole. A hole or an antihole is *even* (*odd*) if it has an even (odd) number of vertices. Clearly, neither odd holes nor odd antitholes are perfect. Berge made the following famous conjecture:

* Research partially supported by CAPES-COFECUB, grant number 213/97
† Partially supported by CNPq-Brazil grant number 301315/96-5

Conjecture I (Strong Perfect Graph Conjecture [1]). *A graph G is perfect if and only if it does not contain as an induced subgraph an odd hole nor an odd antihole.*

This conjecture remains open despite much effort spent trying to prove it. In order to prove or disprove it, several other related conjectures have been proposed. In parallel, a study of the *minimally imperfect graphs* was developed as another way to prove the Strong Perfect Graph Conjecture. A *minimally imperfect graph* is a graph which is not perfect while every proper subgraph is. According to the Strong Perfect Graph Conjecture, a graph is minimally imperfect if and only if it is an odd hole or an odd antihole.

Following Meyniel let us call *even pair* any pair of non-adjacent vertices such that every chordless path between them has even length.

Theorem A (Meyniel [10]). *No minimally imperfect graph has an even pair.*

Meyniel defined the *strict-quasi parity* (SQP) graphs as the graphs where every induced subgraph either is a clique or contains an even pair. As a consequence of Theorem A, every SQP graph is perfect. This class seems to be very large since it contains many classical families of perfect graphs (see [3]). We will call here *obstruction* any graph that is not a clique, contains no even pair, and is minimal with these two properties. It follows immediately that a graph is SQP if and only if it contains no obstruction as induced sugraph. In order to understand the structure of SQP graphs, S. Hougardy proposed the following two conjectures [5]:

Conjecture II. *If a graph G is an obstruction, one of the following holds*:

- *G is an antihole*;
- *G is an odd hole*;
- *G is a line-graph of a bipartite graph.*

Conjecture III. *Every minimally imperfect graph is an obstruction.*

The second conjecture suggests that the only obstructions which are not perfect are the odd holes and odd antiholes. Hence if Conjectures II and III are true, the only minimally imperfect graphs are the odd holes and the odd antiholes, and the validity of the Strong Perfect Graph Conjecture is implied. Conjecture II was proved for planar graphs [9] and claw-free graphs [8].

Note that antiholes of length at least seven are not planar and that the antihole with six vertices is the line graph of $K_{2,3}$. For planar graphs we proved in [9] that:

Theorem B. *If G is a planar obstruction, then one of the following holds*:

- *G is an odd hole*;
- *G is a line graph of a bipartite graph.*

The original proof of Theorem B, however, does not lead to a polynomial-time algorithm that would enable us to recognize planar SQP graphs, since this theorem does not give a characterization of planar obstructions. In this article, we solve this recognition problem. We should recall that finding an even pair can be done in polynomial time in the case of a planar perfect graph, as claimed by Hsu and Porto [7] (the general case is co-NP-complete [2]; Reed conjectures that it is polynomial for all perfect graphs [11]). We could use this algorithm to decide if a given planar graph G is SQP, but this would take exponential time because we would have to apply it to every induced subgraph of G. Instead, we propose here a polynomial algorithm to recognize planar strict quasi-parity graphs that, as Hsu and Porto's algorithm, uses the algorithm due to Hsu [6] for recognizing planar perfect graphs. Roughly speaking, Hsu decomposes every planar graph G along several types of cutsets, thus making a decomposition tree for G; he then shows that the initial graph G is perfect if and only if the leaves of the tree (the undecomposable graphs) belong to several well-defined and easy to recognize classes of perfect graphs. We will follow this idea, and show that a planar graph G is SQP if and only if the undecomposable graphs are SQP and satisfy certain additional technical properties that are easy to test. The complete description of Hsu's algorithm is given in Section 3. First, in Section 2, we need to introduce some properties of obstructions from [9]. Finally, in Section 4, we will present our algorithm.

We finish this section by introducing some notation. Let $G = (V, E)$ be a graph. Whenever convenient, $V(G)$ and $E(G)$ denote respectively the set of vertices and the set of edges of G. If $X \subseteq V$, then $G[X]$ means the subgraph of G induced by X.

2. Some Properties of Obstructions

In this section, we recall from [9] some definitions and properties of obstructions.

Analogously to the definition of even pairs, an *odd pair* is a pair of non-adjacent vertices such that every chordless path between them has odd length. We then call *even-pair cutset* and odd-pair cutset respectively any two vertices a, b that form a cutset of G and such that $\{a, b\}$ is an even (resp. odd) pair.

Remark. If G is any 2-connected perfect graph and x,y are two non-adjacent vertices that form a cutset of G, then $\{x, y\}$ is either an even pair or an odd pair of G. Indeed, in the opposite case there would exist two chordless x, y-paths of different parity; we could then choose two such paths P and Q such that $P - \{x, y\}$ and $Q - \{x, y\}$ lie in distinct components of $G - \{x, y\}$; but then $P + Q$ would induce an odd hole.

Lemma A (The Cutset Lemma for Obstructions). *Let G be a planar obstruction different from an odd hole. Then,*

(i) *G has no clique-cutset (in particular G is 2-connected);*
(ii) *Every 2-cutset of G is an odd pair;*
(iii) *If G is 3-connected then every 3-cutset contains exactly one edge and is the neighbourhood of a vertex.*

Sketch of proof. Suppose that G has a clique-cutset C and let B be any component of $G - C$. If $G[V(B) \cup C]$ is not a clique it contains an even pair $\{x, y\}$, by the minimality of G. It is easy to see that all chordless x, y-paths of G are in $G[V(B) \cup C]$, because C is a clique; hence x, y is also an even pair of G, a contradiction. If $G[V(B) \cup C]$ is a clique for every component B of $G - C$ then it is easy to see that any two non-adjacent vertices of G form an even pair, again a contradiction. This proves (i). Item (ii) is an immediate consequence of (i), of the Remark, and of the fact that G has no even pair. Item (iii) is more difficult and we do not repeat here the proof from [9]. □

Lemma B (The Neighbourhood Lemma for Obstructions). *Let G be a planar obstruction different from an odd hole. For each vertex x of G, the neighbourhood of x induces either $K_2 + K_1$ or $2K_2$ (see Figure 1).*

Proof. Since G is the line-graph of a bipartite graph, the neighbourhood $N(x)$ of any vertex x is either one clique or two non-adjacent cliques. If such a clique has size at least three, then G contains a K_4; hence either we have $G = K_4$ (a contradiction), or, by planarity, one triangle of the K_4 is a cutset of G, a contradiction to the preceding lemma. So the clique size in $N(x)$ is at most two. If $N(x)$ consists of just two non-adjacent vertices u, v then $\{u, v\}$ is an even pair, for otherwise we could obtain an odd hole by adding x to any odd chordless u, v-path; but that is a contradiction. If $N(x)$ consists of two adjacent vertices then these two form a clique-cutset contradicting (i). The remaining possibilities for $N(x)$ lead to the desired conclusion. □

The neighbourhood of a vertex v is called *correct* if $N(v) + v$ induces a graph as in Figure 1, and *incorrect* otherwise.

Definition 1 (Candidate). *We call candidate any planar graph G satisfying the following properties*:

(c1) *G is the line-graph of a bipartite graph*;
(c2) *Every neighbourhood is correct*;
(c3) *G is 2-connected*;
(c4) *Every 2-cutset is an odd pair.*

The following two properties of candidates are consequence of their definition:

(c5) *A candidate has no clique-cutset.*
(c6) *In a candidate, every minimal 3-cutset C contains at most one edge, and if it contains one edge then C is the neighbourhood of a vertex.*

Fig. 1. The allowed neighbourhood of a vertex

To see that (c5) holds, consider a minimal clique-cutset C of a candidate G. If $|C| \leq 2$ then (c3) or (c4) is contradicted. If $|C| \geq 3$ then, recalling that every vertex of a minimal cutset C must have a neighbour in every component of $G - C$, we observe that the vertices of C cannot have a correct neighbourhood. To see that (c6) holds, first consider a minimal cutset $C = \{a, b, c\}$ with edges ab, bc, where ac is not an edge by (c5). Since each of a, b, c must have a neighbour in each component of $G - C$, neighbourhood correctness implies that one component of $G - C$ contains vertices u, v with edges ua, vb, vc and another component contains vertices x, y with edges xa, xb, yc; but then a, v is a cutset contradicting (c4). The second part of (c6) is easily established using similar arguments. □

There is a certain difference between candidates and obstructions, since not every candidate is an obstruction: an example is given by the line-graph of $K_{3,3}$. The reason for introducing candidates is that they will be easier to handle during the algorithm than obstructions. The main aspect is the following theorem:

Theorem C. *Every candidate is even pair-free.*

This theorem can be obtained as a corollary of a result due to Hougardy [5] and its proof is in [9]. The following consequences are obvious.

Corollary 1. *Every obstruction is a candidate. Every candidate contains an obstruction.*

Corollary 2. *A planar graph is a strict-quasi parity graph if and only if it does not contain a candidate.*

One more definition concerning the structure of candidates will be useful to the understanding of our algorithm. An edge is called *flat* if it does not lie in a triangle. In a graph G that contains no odd hole, it is easy to see that an edge ab is flat if and only if $\{a, b\}$ is an odd pair in the graph $G - ab$.

Definition 2 (Glueing). *Let $M_1 = (V_1, E_1), \ldots, M_k = (V_k, E_k)$ be k graphs. Assume that there exist two vertices a, b such that $V_i \cap V_j = \{a, b\}$ for all i, j $(1 \leq i < j \leq k)$, and ab is a flat edge of each M_i. We will say that the graphs are glueable (along ab).*

Given glueable graphs, we can build a graph with vertex-set $V_1 \cup \cdots \cup V_k$ and edge-set $E_1 \cup \cdots \cup E_k - \{ab\}$. This construction will be called glueing the graphs $M_1, \ldots, M_k$ along the edge ab.

Definition 3 (Unglueing). *Let G be a graph with an odd-pair cutset $\{a, b\}$, and let $B_1, \ldots, B_k$ be the components of $G - \{a, b\}$. For each i, build a graph with vertex-set $V(B_i) \cup \{a, b\}$ and with all the edges of $G[V(B_i) \cup \{a, b\}]$ plus the edge ab. This will be called unglueing the graph G along the pair ab.*

It is easy to see that the operations of glueing and unglueing preserve planarity, and also that they preserve the absence of odd holes. Hence they are perfection-preserving for planar graphs. (The fact that they are perfection-preserving for general graphs is also true and is an easy consequence of a lemma of Tucker

[12].) The proofs of the following lemmas consist in a routine check of properties (c1)–(c4) of candidates and we omit it.

Lemma C (The Glueing Lemma for Candidates). *Let M_1 and M_2 be two graphs that are glueable along a common flat edge ab. If M_1 and M_2 are candidates, then the graph obtained by glueing them along ab is a candidate.*

Lemma D (The Unglueing Lemma for Candidates). *Let M be a candidate which admits an odd-pair cutset. Let M_1, M_2 be the two graphs obtained by unglueing M along this cutset. Then each of M_1, M_2 is a candidate.* □

3. Hsu's Planar Graph Decomposition

Given a planar graph G, Hsu [6] associates with G a decomposition tree defined as follows. The root of the tree is G. If a graph H is a node of the tree, and H admits a cutset Q of one of the four types below, then the children of H in the tree are some specific graphs H_i built from the components of $H - Q$ as explained below. If H has no such cutset, then H is a leaf of the tree. Hsu's main result in [6] is that the root graph is perfect if and only if every leaf is in the union of three special classes $\mathscr{S}$, $\mathscr{L}$, $\mathscr{C}$. We now define precisely the types of cutsets. Let Q be a cutset of H, and $B_1, \dots, B_k$ the components of $H - Q$.

- *Type I*: *Q is a clique-cutset.*
  In this case the children are the graphs $H[V(B_i) \cup Q]$ for each $i = 1, \dots, k$.
- *Type II*: *$Q = \{a, b\}$ with a, b not adjacent.*
  Then the i-th child is defined by taking $H[V(B_i) \cup Q]$ and adding between a and b a chordless path of length two if Q is an even-pair cutset and three if Q is an odd-pair cutset. This path has one or two interior vertices which we call *artificial*. (In fact we want to build these children only if none of them is isomorphic to H itself, i.e., only when either $k > 2$, or $k = 2$ and each of $H[V(B_1) \cup Q]$ and $H[V(B_2) \cup Q]$ is different from a path P_3 or P_4.)
- *Type III*: *$Q = \{a, b, c\}$, where ab is an edge and ac is not, and there is a chordless even (a, c)-path in each $H[V(B_i) \cup \{a, c\}]$, and a chordless odd (b, c)-path in each $H[V(B_i) \cup \{b, c\}]$.*
  (In fact all (a, c)-paths in $H - b$ must have the same parity or else we find an odd hole; likewise all (a, b)-paths in $H - c$ have the same parity. Also $k = 2$ or else H would contain a subdivision of $K_{3,3}$.)
  Then the i-th child is obtained by taking $H[V(B_i) \cup Q]$ and adding an artificial vertex a' with edges $a'a$ and $a'c$ and, if bc is not an edge, a second artificial vertex b' with edges $b'b$ and $a'b'$. (Again we do this decomposition only if none of the children is isomorphic to H itself.)
- *Type IV*: *$Q = \{a, b, c, d\}$ induces a C_4 in H with edges ab, bc, cd, da, there is a chordless even (a, c)-path in each $H[V(B_i) \cup \{a, c\}]$, and there is a chordless odd (b, d)-path in each $H[V(B_i) \cup \{b, d\}]$.*

Then the i-th child is obtained by taking $H[V(B_i) \cup Q]$ and adding two artificial vertices a', b' with edges $a'a$, $a'b$, $a'c$, $a'b'$, $b'a$, $b'c$ and $b'd$. (Again this decomposition step is applied only if none of the resulting children is isomorphic to H.)

One will apply the decomposition along a cutset of a given type only if there is no possible decomposition along cutsets of lower types. Then let us define the classes $\mathscr{S}, \mathscr{L}, \mathscr{C}$.

- Class $\mathscr{C}$ consists of (some) planar comparability graphs.
- Class $\mathscr{L}$ consists of planar line-graphs of bipartite graphs, such that the degree of each vertex is two, three or four. The neighbourhood of such a vertex must be either two non-adjacent vertices, or three vertices with exactly one edge between them, or four vertices with exactly two non-incident edges.
- Class $\mathscr{S}$ consists of the three graphs S_1, S_2, S_3 in Figure 2, plus any of the five graphs obtained from S_1 by replacing one or more of the edges e, f, g by a chordless path of length three.

Hsu's decomposition tree can be built in time $O(n^3)$ for every planar graph with n vertices [6].

4. Recognition Algorithm for Planar Strict Quasi-Parity Graphs

In this section we propose an algorithm which, given a planar perfect graph G, determines in polynomial time whether G is a strict quasi-parity graph or not. If it is not, this algorithm produces an induced subgraph of G that is an obstruction. It is based on Hsu's algorithm [6] for the decomposition of planar perfect graphs, presented in the last section. Our method tells if a given planar perfect graph is a strict quasi-parity graph by looking at the leaves of the decomposition tree of G. In the next four lemmas, H is any non-leaf node of the tree and Q is the corresponding cutset along which H is decomposed.

Lemma 1. *If Q is a cutset of Type I, then H is a strict quasi-parity graph if and only if all its children are.*

Proof. The lemma is an immediate consequence of the fact that obstructions have no clique-cutset. □

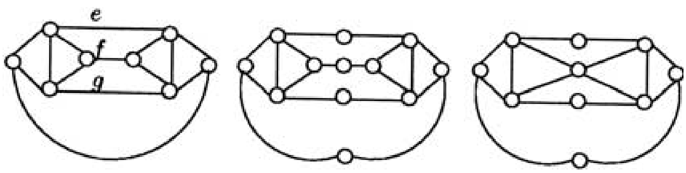

Fig. 2. Graphs S_1, S_2 and S_3, respectively, of $\mathscr{S}$

We must be more careful with cutsets of higher types, as the example in Figure 3 shows: this planar perfect graph, which has a Type II cutset, is not strict quasi-parity while the two children with respect to the cutset are. For 2-cutsets, it will be necessary to distinguish between even-pair cutsets and odd-pair cutsets.

Lemma 2. *If Q is an even-pair cutset, then H is strict quasi-parity if and only if each child of H is.*

Proof. The lemma is immediate from the fact that obstructions have no even-pair cutsets. □

Lemma 3. *If Q is an odd-pair cutset, then H contains an obstruction if and only if either some child of H contains an obstruction, or there are at least two children H_1, H_2 such that each of $H_1 + ab$ and $H_2 + ab$ contains an obstruction which contains the edge ab.*

Proof. This lemma is a direct consequence of the Glueing and Unglueing Lemmas. □

Remark. If H_i contains an obstruction M, and M does not contain both a and b, then M is also in $H_i + ab$. On the other hand, if M does contain a, b, then a, b is an odd pair and a cutset of M, and then, by the Glueing Lemma, $M + ab$ contains an obstruction. In either case, $H_i + ab$ contains an obstruction.

This lemma suggests that the pair a, b of H is special. We will not explicitly build the graphs $H_i + ab$, but we will mark yellow the pair a, b in the child H_i.

Now let us consider cutsets of Type III, with the notation of the preceding section.

Lemma 4. *If $Q = \{a, b, c\}$ is a cutset of Type III, then H contains an obstruction if and only if either some child of H contains an obstruction, or bc is not an edge and each of $H_1 + bc$ and $H_2 + bc$ contains an obstruction which contains the edge bc.*

Proof. Suppose that M is an obstruction in H. We want to see when it is possible that Q separates M.

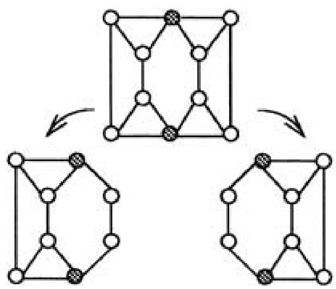

Fig. 3. A decomposition tree whose root is non-SQP while the leaves are SQP

If bc is an edge, then Q is a P_3. By the Cutset Lemma for Obstructions, 3-connected obstructions have no P_3-cutset, so M cannot be separated by Q. Also M cannot be separated by two adjacent vertices of Q. Finally M cannot be separated by two non-adjacent vertices of Q, for these two should be a, c, which form an even pair. In all cases Q does not separate M, i.e., M is an induced subgraph of one child of H.

If bc is not an edge, and $Q \subset M$, then the Cutset Lemma for Obstructions implies that Q must be the neighbourhood of some vertex x in M; but this would contradict the path-parity condition in the definition of a Type-III cutset. So $Q \cap M$ consists of only two vertices of Q. Again, since a, b is an even pair, this is possible only if $Q \cap M = \{b, c\}$, and $\{b, c\}$ is an odd pair. In this case, By the Glueing Lemma, we find an obstruction in $H_1 + bc$ and one in $H_2 + bc$. □

Here the pair b, c will be marked yellow in H_1 and in H_2.

For a cutset of Type IV we can apply the same argument as for the preceding lemma, and we obtain:

Lemma 5. *If Q is a cutset of Type IV, then H contains an obstruction if and only if either some child of H contains a obstruction, or bd is not an edge and each of $H_1 + bd$ and $H_2 + bd$ contains an obstruction which contains the edge bd.* □

Here the pair b, d will be marked yellow in H_1 and in H_2.

Now we look at the leaves of the decomposition tree. Each such indecomposable graph H comes with a (possibly empty) set Y_H of pairs of vertices of H that have been marked yellow at some step of the decomposition. Let $H + Y_H$ denote the graph obtained from H by adding all yellow pairs of Y_H as edges. Lemmas 3–5 imply that if the root graph G contains an obstruction, then at least one $H + Y_H$ contains an obstruction. The converse is not true, as shown in Figure 4. This is because the yellow pair appears in only one obstruction among the leaves of G, and so glueing back along this pair does not preserve obstructions. The next definition and lemma will help us overcome this difficulty.

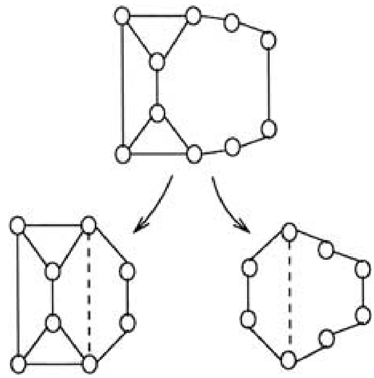

Fig. 4. A decomposition tree whose leaves contain an obstruction while the root is SQP

Definition 4. *A pseudo-candidate is defined recursively as follows*:

- *Any candidate is a pseudo-candidate*;
- *Given k pseudo-candidates that are glueable along a flat edge*, *the graph resulting from their glueing along that edge is a pseudo-candidate*.

Notice that every 2-cutset of a pseudo-candidate G is an odd pair. The only difference between candidates and pseudo-candidates is that the latter admit 2-cutsets that may have many connected components, whereas in a candidate every 2-cutset has exactly two components (because of neighbourhood correctness).

Lemma 6. *Any pseudo-candidate contains a candidate.*

Proof. Assume that G is a pseudo-candidate and that the lemma holds true for all pseudo-candidates with strictly less vertices than G. If, for every 2-cutset Q of G, the subgraph $G - Q$ has exactly two components, then clearly G is already a candidate. Now suppose that there exists a 2-cutset Q of G such that $G - Q$ has $k > 2$ components. Let G' be the subgraph obtained by removing all the vertices of $k - 2$ arbitrary components of $G - Q$. Then G' is a pseudo-candidate. By the induction hypothesis G' contains a candidate. □

Notice that this proof is algorithmically efficient, i.e., given a pseudo-candidate we can easily find an induced candidate. Details are omitted. As a simple consequence of this lemma and of Corollary 1 of section 2, we have:

Corollary 3. *Every pseudo-candidate contains an obstruction.* □

Since, every obstruction is a pseudo-candidate, we also have an other corollary, namely:

Corollary 4. *A planar perfect graph is strict quasi-parity if and only if it contains no pseudo-candidate.* □

Lemma 7. *Suppose that there exists a collection L of candidates $M_1, \ldots, M_p$ such that*:

- *Each M_i is an induced subgraph of $H + Y_H$ for some leaf H*;
- *For $i \neq j$, $M_i \cap M_j$ is either empty or one yellow edge*;
- *Each yellow edge appearing in at least one of the M_i*'s *appears in at least two of them*;

Then the graph $M_1 \cup M_2 \cup \cdots \cup M_p - Y$ is a pseudo-candidate and an induced subgraph of G.

This is a simple consequence of the definition of pseudo-candidates and of Lemmas 3–5. □

Now, by virtue of this lemma, we only have to find an appropriate collection of candidates in the leaves. In fact we will show that there exists essentially one such collection if G is not strict quasi-parity, or none if G is. For a leaf H of G, it will be necessary to determine whether $H + Y_H$ contains a candidate or not. When $H \in \mathscr{S}$ the question is trivial: by exhaustive examination it is a routine matter to find an obstruction (if any) in H, since $\mathscr{S}$ is a finite class. Actually such an obstruction can arise only if H is a subdivision of S_1, and in this case the obstruction is a $\overline{C}_6$. The case when $H \in \mathscr{C}$ can be eliminated, as shown in the next lemma. When $H \in \mathscr{L}$ the situation must be handled with more care.

Lemma 8. *If H is in $\mathscr{C}$, then $H + Y_H$ contains no obstruction.*

Proof. Recall the exact definition of $\mathscr{C}$ from [6]: any graph H in $\mathscr{C}$ contains a stable set S such that $H - S$ is planar bipartite and the neighbourhood $N(s)$ of every vertex $s \in S$ is a C_4 (a square) of $H - S$. Moreover, every element of $\mathscr{C}$ is transitively orientable: indeed, making every vertex of the left side of $H - S$ a source and every vertex of the right side of $H - S$ a sink gives a transitive orientation (note that every edge has at least one endpoint in $H - S$). We now claim that:

$$H + Y_H \text{ is transitively orientable.}$$

Indeed we will show that the above transitive orientation of H can be extended to a transitive orientation of $H + Y_H$. So consider any yellow pair x, y in H. Recall from the definition of the yellow pairs, and from the definition of the decompositions of Type II–IV, that between x and y there exists a chordless path $xuvy$ whose interior vertices u, v are artificial.

In Type II or III, the neighbourhood of each of u, v is a stable set. This immediately entails that x and y are in $H - S$. So we orient the edge xy from left to right, and clearly transitivity is respected.

In Type IV, x, y must be part of a square $xx'yy'$ such that x' and y' both see u and v. Since the neighbourhood of x' contains the P_4 $xuvy$, vertex x' is not in S. Hence one of u, v is in S or else u, v, x' would form a triangle in $H - S$. So we may assume that $u \in S$, and it follows that x, x', v, y' are in $H - S$. Also y is in S or else y, v, y' would form a triangle in $H - S$. We may assume that x is on the left of $H - S$. So we orient the edge xy from x to y. Transitivity is respected because the only new directed paths of length two that arise are xyx' and xyy', and the edges xx' and xy' are already correctly oriented.

Since $H + Y_H$ is a transitively orientable graph, it is a strict quasi-parity graph (see [10]), and it contains no obstruction. □

These two lemmas say that only a $\overline{C}_6$ can appear as an obstruction in the leaves of the decomposition tree of G that are in $\mathscr{S} \cup \mathscr{C}$.

Now we concentrate on the leaves of the decomposition tree that are elements of $\mathscr{L}$. If H is in $\mathscr{L}$, then H contains no artificial vertex coming from the decompositions of Type III or IV, for otherwise some neighbourhood condition in the definition of $\mathscr{L}$ would be violated. So the artificial vertices are all of Type II. These vertices can be matched two by two so that each such pair is the interior of a chordless path of length three, whose extremities must be vertices of degree three that form a yellow pair. It follows that $H + Y_H - A$ is a line-graph of bipartite graph, where A is the set of artificial vertices. It is obvious that $H + Y_H$ contains a candidate if and only if $H + Y_H - A$ does.

Let L be the list of the graphs $H + Y_H - A$ for which H is an $\mathscr{L}$-leaf of the tree. In L we also add the obstruction $\overline{C}_6$ that might arise with a leaf from $\mathscr{S}$, if any. Note that this $\overline{C}_6$ is also a line-graph of a bipartite graph, namely $K_{2,3}$.

Let H' be any element of L. If H' contains a vertex z of degree at most two, obviously z cannot lie in a candidate contained in H; so H' contains a candidate if and only if $H' - z$ does. We replace H' by $H' - z$ in the list, and we repeat this as long as vertices of degree at most two are present. If the final graph is empty, we do not add it to L. This is called the *low-degree vertex reduction* of L.

Now every element of L is a line-graph of bipartite graph that satisfies the requirements to be a candidate, so every element of L is a candidate.

If a yellow edge appears at least twice in L we call it *useful*, else we call it *useless*.

If every yellow edge is useful, we can apply Lemma 7 and obtain a pseudo-candidate in G, certifying that G is not an SQP graph. On the other hand, assume that some yellow edge e is useless and let H' be the unique element of L containing e. Then, by Lemmas 3–5, glueing this H' along e with another graph which might contain this edge (but not as a yellow edge) will not produce a graph containing a candidate. So we can remove this edge, i.e., replace H' by $H' - e$ in L. We call this the *useless-edge reduction* of L. (In fact when a useless yellow edge is removed its endpoints become vertices of degree two, and so they too will be eliminated by low-degree removal.)

We must make an exception for the "central" edge of the $\overline{C}_6$ coming from a subdivision of S_1, i.e., the edge "f" on Figure 2. This edge, if it is yellow, will always be considered useful, for even it if appears only once in L its vertices can still be replaced by two other suitable vertices of this subdivision, namely the endpoints of the curvy edge in Figure 2.

We keep applying vertex reduction and edge reduction, in any order, as long as possible. We obtain a final reduced list L. If L is not empty then the hypotheses of Lemma 7 hold (or else the reduction could be continued), i.e., every element of L is a candidate and all yellow edges appearing in them are useful among the list; so we can find a pseudo-candidate in G. If L is empty then G is a quasi-parity graph.

The two kinds of reduction can be done in time $O(n^2)$ with appropriate counters. So the complexity of this algorithm is the same as Hsu's, $O(n^3)$. In conclusion we have:

Theorem 1. *Given a planar perfect graph, one can decide if this graph is a quasi-parity graph and, if it is not, produce an obstruction, in time* $O(n^3)$. □

References

1. Berge, C., Chvátal, V.: Topics on Perfect Graphs. Ann. Discrete Math. **21**, North-Holland, Amsterdam 1984
2. Bienstock, D.: On the complexity of testing for even holes and induced odd paths. Discrete Math. **90**, 85–92 (1991) (Corrigendum in vol. **102**(1), 1992)

3. Everett, H., de Figueiredo, C.M.H., Linhares-Sales, C., Maffray, F., Porto, O., Reed, B.: Path parity and perfection. Discrete Math. 165–166 (1997), 223–242
4. Grötschel, M., Lovász, L., Schrijver, A.: Polynomial algorithms for perfect graphs. In: C. Berge, V. Chvátal: Topics on Perfect Graphs. (Ann. Discrete Math. **21**, 325–356) North-Holland, Amsterdam 1984
5. Hougardy, S.: Even and odd pairs in linegraphs of bipartite graphs. Eur. J. Comb. **16**, 17–21 (1995)
6. Hsu, H.-L.: Recognizing perfect graphs. J. Assoc. Comput. Machinery **34**, 255–288 (1987)
7. Hsu, H.-L., Porto, O.: Finding even pairs in planar perfect graphs. Private communication (1992)
8. Linhares-Sales, C., Maffray, F.: Even pairs in claw-free perfect graphs. J. Comb. Theory, Ser. B **74**, 169–191 (1998)
9. Linhares-Sales, C., Maffray, F., Reed, B.A.: On planar quasi-parity graphs. (submitted)
10. Meyniel, H.: A new property of critical imperfect graphs and some consequences. Eur. J. Comb. **8**, 313–316 (1987)
11. Reed, B.: Perfection, parity, planarity and packing paths. In R. Kannan, W.R. Pulleyblank: Proc. Int. Prog. and Comb. Opt. Conference, pp 407–419, Waterloo: Univ. of Waterloo Press (1990)
12. Tucker, A.: The validity of the perfect graph conjecture for K_4-free graphs. In: C. Berge, V. Chvátal, Topics on Perfect Graphs, pp 149–157, North Holland 1984

Received: September 21, 1998
Final version received: May 9, 2000