1

of/de

1

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Service des thèses canadiennes

Ottawa, Canada
K1A 0N4

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, tests publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.

MATCHING AND STABLE CUTSETS IN GRAPHS

by

Augustine M. Moshi, B.Sc., Dip. Q.Sc., M.Sc.

A thesis submitted to

the Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Mathematics and Statistics

Carleton University

Ottawa, Canada

June 20, 1987

3

Permission Has been granted to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film.

The author (copyright owner) has reserved other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without his/her written permission.

L'autorisation a été accordée à la Bibliothèque nationale du Canada de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur (titulaire du droit d'auteur) se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation écrite.

ISBN 0-315-39365-3

The undersigned recommend to the Faculty of Graduate Studies and Research acceptance
of the thesis

Matching and Stable Cutsets in Graphs

submitted by Augustine Masumbuko Moshi, B Sc., D.C Sc., M Sc., in partial fulfillment
of the requirements for the degree of Doctor of Philosophy.

Brian Mortimer
Dr. B.C. Mortimer, Thesis Supervisor

Sheila Whitesides
Dr. S.H. Whitesides, External Examiner

Cyril W. L. Garner
Dr. C. W. L. Garner, Chairman
Department of Mathematics and Statistics

August 24, 1987

ABSTRACT

Let $G = (V, E)$ be an undirected graph. A subset F of E is a **matching cutset** of G if F is a matching and $G - F$ has more components than G . Similarly, $S \subset V$ is a **stable cutset** of G if S is an independent set and $G - S$ has more components than G . Recognizing graphs with a matching cutset is an NP-Complete problem for graphs with maximum degree 4. We prove:

- Every connected graph with maximum degree ≤ 3 and with more than 7 points has a matching cutset. (In particular, $K_{3,3}$ and K_4 are the only cubic graphs without a matching cutset).
- Recognizing graphs with a matching cutset can be done in polynomial time when the input is restricted to either line graphs or to graphs without a chordless circuit of length ≥ 5 .

We point out that recognizing graphs with a stable cutset is an NP-Complete problem even if the input is restricted to line graphs with maximum degree 6, and prove:

- The following classes of graphs must have stable cutsets: (i) Connected graphs with maximum degree ≤ 3 and with more than 6 points. (ii) Critically 2-connected graphs with more than 3 points. (iii) Connected K_4 -free line graphs whose root graphs have more than 7 points.
- Recognizing graphs with a stable cutset can be done in polynomial time if the input is restricted to either line graphs or to planar line graphs.
- Both problems of recognizing graphs with a matching cutset and recognizing graphs with a stable cutset are NP-Complete for perfect graphs.

ACKNOWLEDGEMENT

I am most thankful to my Thesis Supervisor, Dr. B.C. Mortimer. He has patiently guided and encouraged me throughout my period of study at Carleton University. I thank him also for his financial support which enabled me to attend the 12th International Symposium on Mathematical Programming (Boston, Mas. 1985), The 3rd SIAM Conference on Discrete Mathematics (Clemson, S.C. 1986), and the CMS Annual Summer Meeting (Kingston, Ont. 1987).

The enormous cost of my study in Canada was met by the CIDA-sponsored Link Agreement Between the Universities of Dar es Salaam and Carleton. I wish to thank CIDA and the administrators of this very useful program. Special thanks go to Mrs. M. Sheek of the Paterson Center for International Programs.

Finally, I wish to thank Faiz Bannani, Janice Jeffs and Professors W.H. Cunningham, C.W.L. Garner, J.D. Dixon and P.C. Tan, all from the Department of Mathematics and Statistics, Carleton University.

TABLE OF CONTENTS.

Abstract	ii
Acknowledgements	iii
Chapter 1: INTRODUCTION AND PRELIMINARY RESULTS.	
1.1 Introduction	1
1.2 Equivalent Formulations	4
1.3 Matching Cut and SATISFIABILITY	10
1.4 Locally Connected Graphs	13
Chapter 2: NP-COMPLETENESS RESULTS.	
2.1 Some Special NP-Complete Problems	17
2.2 The NP-Completeness of MATCHING CUT and STABLE CUT	20
2.3 Related NP-Completeness Results	27
Chapter 3: MATCHING AND STABLE CUTSETS IN PLANAR GRAPHS.	
3.1 Planarity and Tractability	33
3.2 Matching Cutsets in Planar Graphs	36
3.3 Stable Cutsets in Planar Graphs	49
Chapter 4: MATCHING AND STABLE CUTSETS IN GRAPHS WITH MAXIMUM DEGREE 3 AND IN QUADRANGULATED GRAPHS.	
4.1 Preliminary Remarks	46
4.2 Stable Cutsets in Graphs with Maximum Degree 3 and 4	47
4.3 Matching Cutsets in Graphs with Maximum Degree 3	50
4.4 Quadrangulated Graphs	53

Chapter 5: MATCHING CUTSETS IN LINE GRAPHS.

5.1 Preliminaries	61
5.2 MATCHING CUT for Line Graphs	62
5.3 Some Open Questions	67
References:	70

CHAPTER 1

INTRODUCTION AND PRELIMINARY RESULTS.

1.1 Introduction.

Let $G = (V, E)$ be an undirected graph. A subset F of G is a *matching cutset* of G if no two edges of F are coincident and $G - F$ has more components than G . (Two edges are coincident if they have a common endpoint).

R.L.Graham [11] asked whether it is NP-Complete to recognize graphs with a matching cutset. The answer to this question was given by Chvátal [4]. He 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.

Let us abbreviate by MATCHING CUT the problem of recognizing graphs with a matching cutset. We shall study this problem together with the *stable cutset* problem. A *stable cutset* of a graph $G = (V, E)$ is a subset S of V such that no two points of S are adjacent and $G - S$ has more components than G . The problem of recognizing graphs with a stable cutset, which we shall abbreviate by STABLE CUT, is at least as hard as MATCHING CUT. This claim will be proved in Chapter 2.

Each of STABLE CUT and MATCHING CUT can be formulated as certain types of vertex and edge colouring problems. We make this assertion precise in Section 2 of this chapter. In Section 3 of this chapter, an explicit polynomial reduction of MATCHING CUT into SATISFIABILITY is given. We end Chapter 1 by giving a sufficient condition

for a graph to have neither a matching cutset nor a stable cutset. The condition which we give is sufficiently general to imply that some well known classes of graphs can have neither a matching cutset nor a stable cutset.

In Chapter 2, we prove that both MATCHING CUT and STABLE CUT have 'degree jumps' in their complexity. We show also that STABLE CUT has a 'subgraph jump' in its complexity. A few more complexity results are given in Chapter 2. These include a proof that both MATCHING CUT and STABLE CUT are NP-Complete even if the input is restricted to *perfect graphs*, and that it is NP-Complete to recognize graphs whose vertex set can be partitioned into 3 subsets V_1 , V_2 and V_3 such that V_1 induces a clique and every point of V_2 is adjacent to every point of V_3 .

In Chapter 3, we consider MATCHING CUT and STABLE CUT for planar graphs. We introduce the following graph problem:

Problem 1: Given a planar graph $G = (V, E)$ and a pair of points $x, y \in V$, does there exist a path P in G from x to y such that P does not contain two edges from the same face of G ?

We prove that Problem 1 is at least as hard as MATCHING CUT for planar graphs. We also give some partial results concerning the recognition of planar graphs with a matching cutset or a stable cutset. Chapter 4 addresses MATCHING CUT and STABLE CUT when the input is restricted to graphs with maximum degree ≤ 3 . We prove that every connected graph with maximum degree ≤ 3 and on more than 7 points must have *both* a matching cutset and a stable cutset. We also prove in that chapter that MATCHING CUT is solvable in polynomial time if the input is restricted to graphs without a chordless cycle of length 5 or more.

The complexity of MATCHING CUT for line graphs is analyzed in Chapter 5. (STABLE

CUT is shown to be NP-Complete for line graphs in Chapter 2). We combine ideas about stable cutsets and matching cutsets to give an $O(|E|)$ algorithm to solve MATCHING CUT when the input is restricted to line graphs. As a corollary, we get a proof that STABLE CUT is solvable in polynomial time if the input is restricted to *line graphs of line graphs*.

It is useful to note that a multigraph $G = (V, E)$ has a stable cutset if and only if its underlying simple graph has a stable cutset. Matching cutsets do not have this nice property. Figure 1.1.1(a) shows a multigraph which does not have a matching cutset. Its underlying simple graph is shown in Figure 1.1.1(b), and it has a matching cutset.

Figure 1.1.1(a)-(c)

Let G be a multigraph. We can obviously delete any loops without affecting the existence or otherwise of matching cutsets in G . So, we shall assume that G does not have loops. Let G' be obtained from G by subdividing an edge e which has both its endpoints in common with an edge f , as shown in Figure 1.1.1(c). Let u and v be the common endpoints of e and f , and let x be the new point which is obtained when e is subdivided. Suppose that F is a matching cutset of G . Then u and v must be in the same component of $G - F$ since e and f cannot both be in F . Thus, F is also a matching cutset of G' . Similarly, suppose

that F' is a matching cutset of G' . Then, since the 3 points u, v and x induce a triangle in G' , they must be in the same component of $G' - F'$. Hence, F' is also a matching cutset of G .

By repeatedly subdividing one of the edges in a pair of parallel edges as described above, we obtain a simple graph $\bar{G}$ from the multigraph G such that G has a matching cutset if and only if $\bar{G}$ has a matching cutset. Each time an edge is subdivided the number of multi-edges decreases by one, and the number of points increases by one. Thus, $\bar{G}$ can be constructed from G in polynomial time.

The preceding argument shows that MATCHING CUT for multigraphs is no harder than MATCHING CUT for simple graphs. In view of the preceding comments, when we discuss the complexity of MATCHING CUT we will mostly restrict attention to simple graphs.

1.2 Equivalent Formulations.

Both MATCHING CUT and STABLE CUT can be formulated as certain types of graph colouring problems. We make this claim precise in Propositions 1.2.1, 1.2.3 and 1.2.4.

Definition: Given $v \in V$, the *neighbourset* of v is the set of all points that are adjacent to v . The neighbourset of v will be denoted by $N(v)$. Thus, $N(v) = \{u \in V : uv \in E\}$.

If $U \subseteq V$, we shall let $G[U]$ denote the subgraph of G that is induced by U . The notations $\delta(U)$, $\gamma(U)$ and $[U_1, U_2]$ will denote, respectively, the set of edges with precisely one endpoint in U , the set of edges with both endpoints in U , and the set of edges with an endpoint in each of U_1 and U_2 . All the graphs that we consider are undirected, so by 'graph', we shall always mean an undirected graph.

Proposition 1.2.1: Let $G = (V, E)$ be a simple 2-edge-connected graph. Then the following are equivalent:

- (i) G has a matching cutset.
- (ii) The vertices of G can be coloured red and blue so that both colours are assigned, and for each vertex v , there is at most one vertex $u \in N(v)$ such that u is coloured differently from v .
- (iii) The edges of G can be coloured red and blue so that both colours are assigned, each red edge is coincident with at most 2 blue edges, for each blue edge $e = xy$ at most one of x and y is incident with a red edge, and the set of blue edges which are coincident with red edges is a matching.

Proof: (i) $\Rightarrow$ (ii) : Let F be a matching cutset of G , and G_1 be a component of $G - F$. Colour each vertex of G_1 red and all other vertices blue. Now, $V(G_1) \neq \emptyset$, $V - V(G_1) \neq \emptyset$, so both colours are assigned. Suppose that for some $v \in V$, there are $u_1 \in N(v)$ and $u_2 \in N(v)$ such that both u_1 and u_2 are coloured differently from v . Then u_1 and u_2 are not in the same component as v . Hence, $vu_1, vu_2 \in F$. This contradicts the fact that F is a matching. Thus, this colouring has the property defined in (ii).

(ii) $\Rightarrow$ (iii): Let V_1 be the set of red vertices. Colour each edge of $G[V_1]$ red and all other edges blue. Let v be a red point. Then v has at least 2 neighbours since G is simple and 2-edge connected. By hypothesis, at most one of the neighbours of v is blue. So there is a red neighbour u of v . Hence the edge uv is a red edge. A similar argument shows that there is a blue edge, and so both colours are assigned.

Suppose a red edge $e = xy$ is coincident with more than 2 blue edges. Then one of the points x and y , say x , is incident with at least two edges xz and xw where z and w are blue vertices. Now, we have $x \in V$, $z, w \in N(x)$, and yet z and w are both coloured

differently from x , which is a contradiction. Thus, no red edge is coincident with more than two blue edges.

Let F be the set of blue edges which are coincident with red edges. Then F is precisely the set $[V_1, V - V_1]$. Thus, no blue edge $e = xy$ is coincident with red edges at both x and y . F must be a matching, for otherwise either a red point has two blue neighbours, or a blue point has two red neighbours, and this would contradict the hypothesis.

(iii) $\Rightarrow$ (i): Let F be the set of blue edges which are coincident with red edges. Then F is a matching. Let $f = xy \in F$. We show that there is no path in $G - F$ from x to y . Since $f \in F$, we may assume that there is a red edge g which is coincident with f at x . Suppose there is also a blue edge h which is coincident with f at x . Now, h is a blue edge which is coincident with the red edge g , and so $h \in F$. But also $f \in F$ and h and f are coincident. This contradicts the fact that F is a matching. Thus, all edges in $\delta(x) - \{f\}$ are red. Moreover, all edges in $\delta(y) - \{f\}$ are blue. Hence any path in $G - f$ from x to y must start with a red edge and end with a blue edge. Such a path must therefore use a blue edge which is coincident with a red edge, i.e. an edge of F . Hence $G - F$ is disconnected. $\square$

In Chapter 2, we shall prove that MATCHING CUT remains NP-Complete when the input is restricted to bipartite graphs. In this section, we shall solve in an essentially negative sense, the matching cutset problem for complete bipartite graphs.

Proposition 1.2.2: Let $G = (V, U, E)$ be a complete bipartite graph with bipartition (V, U) . Suppose $|V| \geq 3$ and $|U| \geq 3$. Then for every matching F of G the subgraph $G - F$ is connected.

Proof: Suppose that $G - F$ is a disconnected graph. Then F is a matching cutset of G . So, the vertices of G can be coloured red and blue as in part (ii) of the statement of

Proposition 1.2.1.

Let $u \in U$, and suppose u is coloured red. Then at most one point of V is coloured blue, since $N(u) = V$.

Suppose all the points of V are coloured red. Since both colours are assigned, then there is a point $x \in U$ which is coloured blue. But $N(x) = V$, and now x has more than one neighbour with a colour which is different from the colour of x . This is a contradiction.

Thus, there is a point $y \in V$ such that y is coloured blue. Since $y \in N(u)$, u is coloured red and $N(u) = V$, then all points of $V - y$ are coloured red.

Let $t \in (U - u)$. Since $|V - y| \geq 2$ and $V - y \subseteq N(t)$ and all points of $V - y$ are red, then t is a red point. Now y has two neighbours u and t which are coloured red while y is blue.

This is a contradiction. □

A similar argument shows that a complete graph on more than 2 points cannot have a matching cutset.

Formulations of STABLE CUT as edge and vertex colouring problems are given in Propositions 1.2.3 and 1.2.4, respectively. These formulations will be used to prove some results in Section 1.3 of this chapter and in Chapters 2 and 4.

Proposition 1.2.3: Let $G = (V, E)$ be a simple connected graph. Then the following are equivalent:

(i) G has a stable cutset.

(ii) The edges of G can be coloured red and blue so that both colours are assigned and no edge is coincident with edges of a colour which is different from its own colour at more than one of its endpoints.

Proof: (i) $\Rightarrow$ (ii) : Let S be a minimal stable cutset of G and G_1 be a component of $G - S$. Colour all edges of $E(G_1) \cup \delta(V(G_1))$ red and all the remaining edges blue. (See

Figure 1.2.1). Since G is connected and S is a vertex cutset then $\delta(V(G_1)) \neq \emptyset$, and $\delta(V - V(G_1)) \neq \emptyset$. Hence both colours are assigned, and S is precisely the set of points which are incident with edges of both colours.

Figure 1.2.1

(ii) $\Rightarrow$ (i) : Let $S = \{v : v \text{ is incident with edges of both colours}\}$. Suppose there are points u and v in S such that $uv \in E$. Then the edge uv is incident with edges of a colour which is different from its own at both of its end points, which is a contradiction. Thus, S is an independent set.

It remains to be shown that $G - S$ is disconnected.

Let $e = uv$ and $f = xy$ be a red and a blue edge, respectively (we can find two such edges since both colours are assigned). We may assume that neither u nor x is a point of S , because every edge has at least one endpoint outside S . Also, $u \neq x$, otherwise we get $u \in S$. We will show that there is no path in $G - S$ from u to x .

Let P be any path in G from u to x . Since $u \notin S$ and e is a red edge, then the edge of P which is incident with u is red. Similarly, the edge of P which is incident with x is blue. Hence, there is a point of P which is incident with both a red edge and a blue edge, and

any such point must be in S . So, every path in G from u to x uses a point of S . Thus, $G - S$ is a disconnected graph. $\square$

We have formulated MATCHING CUT as certain types of vertex and edge colouring problems. This was done in Proposition 1.2.1. In Proposition 1.2.3, STABLE CUT is formulated as a type of an edge colouring problem. One may wonder then as to whether it is possible to formulate STABLE CUT simply as a type of a vertex colouring problem. We answer this in the affirmative.

Proposition 1.2.4: *A simple connected graph $G = (V, E)$ has a stable cutset if and only if its points can be coloured red and blue such that there is a red vertex which is adjacent to both a blue and a red vertex, and the set of red vertices which have blue neighbours is independent.*

Proof: Let S be a minimal stable cutset of G . Let G_1 and G_2 be any components of $G - S$. Colour all points of $G_1 \cup S$ red and colour all the remaining points blue.

Since S is a minimal vertex cutset, then every point of S has neighbours in both G_1 and G_2 . Let $x \in S$. Such a point x exists since $S \neq \emptyset$. Then x is a red point and it has both a red neighbour and a blue neighbour.

Now suppose a colouring of the vertices of G with the said property exists. Let S be the set of red vertices which are adjacent with blue vertices. So, there is $x \in S$ such that x has both a blue neighbour v and a red neighbour u . By hypothesis, S is a stable set. We show that every path in G from u to v uses a point of S .

The point u is not in S since $x \in S$, $xu \in E$ and S is independent. Thus any path P in G from u to v starts with an edge ut , where t is a red point. Following along P from u towards v , let z be the first blue point. Such a point z exists since v itself is blue. Thus there is an edge wz of P , where w is a red point. Now w is a red point with a blue

neighbour, so $w \in S$. Hence $G - S$ is disconnected. □

1.3 MATCHING CUT and SATISFIABILITY

SATISFIABILITY is the following problem: Given m clauses $C_1, \dots, C_m$, is the formula $C_1 \vee C_2 \vee \dots \vee C_m$ satisfiable? SATISFIABILITY was the first problem which was shown to be NP-Complete. This was done by S.A. Cook [5] in 1971. Since then, a standard way to show that a problem Q in the class NP is NP-Complete has been to give a polynomial transformation of SATISFIABILITY into Q . We have pointed out that MATCHING CUT has been shown by Chvátal to be an NP-Complete problem. It follows from this that there must be a polynomial transformation of MATCHING CUT to SATISFIABILITY. We shall explicitly give such a transformation in this section.

Restating the first two parts of Proposition 1.2.1, we see that a 2-edge connected graph $G = (V, E)$ has a matching cutset if and only if there is a (colouring) function $\lambda : V \rightarrow \{0, 1\}$ such that λ is onto and for each $v \in V$ with a neighbour set $N(v)$, there exists at most one $u \in N(v)$ such that $\lambda(u) \neq \lambda(v)$. We shall use the term **decomposition colouring** for such a map λ .

Given any connected graph $G = (V, E)$, we shall construct a Boolean function F in polynomial time such that F is satisfiable if and only if G has a decomposition colouring. For the remainder of this section, all arithmetic will be carried out in mod 2. We shall denote by R^V the set of all Boolean vectors indexed by V .

Let $V = \{1, 2, \dots, n\}$. For $i = 1, 2, \dots, n$, associate a Boolean variable x_i with the point i .

Let $A: R^V \rightarrow \{0, 1\}$ be given by

$$A(X) = \prod_{i=1}^n x_i + \prod_{i=1}^n \bar{x}_i \quad (1)$$

For $i = 1, \dots, n$, let B_i , C_i and D_i be maps $R^V \rightarrow \{0, 1\}$ which are defined by

$$B_i(X) = \prod_{j \in N(i)} x_i x_j \quad (2)$$

$$C_i(X) = \prod_{j \in N(i)} \bar{x}_i \bar{x}_j \quad (3)$$

$$D_i(X) = \sum_{j \in N(i)} \left(\prod_{k \in N(i), k \neq j} x_i x_k \bar{x}_j + \prod_{k \in N(i), k \neq j} \bar{x}_i \bar{x}_k x_j \right) \quad (4)$$

Finally, let F and F_i be maps $R^V \rightarrow \{0, 1\}$ defined by

$$F_i(X) = B_i(X) + C_i(X) + D_i(X) \quad (5)$$

$$F(X) = (1 + A(X)) \prod_{i=1}^n F_i(X) \quad (6)$$

We first observe that for any Boolean vector $X^0 \in R^V$, we have $A(X^0) = 0$ if and only if there are points s and t in V such that $x_s^0 \neq x_t^0$. This is so since if for every s and t in V we have $x_s^0 = x_t^0$, then either

$$\prod_{i=1}^n x_i^0 = 0 \text{ and } \prod_{i=1}^n \bar{x}_i^0 = 1$$

or

$$\prod_{i=1}^n x_i^0 = 1 \text{ and } \prod_{i=1}^n \bar{x}_i^0 = 0$$

depending on whether $x_s^0 = 0$ or $x_s^0 = 1$. Therefore, the factor $1 + A(X)$ which appears in (6) takes the value 1 if and only if there are s and t such that $x_s^0 \neq x_t^0$.

Lemma 1.3.1: Let $i \in V$ and $X^0 \in R^V$ be a Boolean vector. Then $F_i(X^0) = 1 \iff$ there is at most one neighbour t of i such that $x_i^0 \neq x_t^0$.

Proof: First suppose that for every $t \in N(i)$, we have $x_t^0 = x_i^0$. Then precisely one of $B_i(X^0)$ and $C_i(X^0)$ is 1 and the other is 0, depending on whether $x_i^0 = 1$ or $x_i^0 = 0$. Moreover, $D_i(X^0) = 0$ since each of its terms has a factor of the form $x_i^0 \bar{x}_j^0$ or $\bar{x}_i^0 x_j^0$, where $x_i^0 = x_j^0$. Hence, $F_i(X^0) = 1$ in this case.

Now suppose there are two points s and t in $N(i)$ such that $x_i^0 \neq x_s^0 = x_t^0$. Then $B_i(X^0) = 0$ and $C_i(X^0) = 0$ because these terms have x_i^0 multiplied by x_s^0 or $\bar{x}_i^0$ multiplied by $\bar{x}_s^0$.

$D_i(X^0) = 0$ also since each of its products has x_i^0 multiplied by x_s^0 or x_t^0 or $\bar{x}_i^0$ multiplied by $\bar{x}_s^0$ or $\bar{x}_t^0$. Hence $F_i(X^0) = 0$ in this case.

It remains to consider the case when there is precisely one neighbour t of i such that $x_i^0 \neq x_t^0$.

As before, $B_i(X^0) = 0$, and $C_i(X^0) = 0$. Consider formula (4). When $j = t$, we get the term

$$\prod_{k \in N(i), k \neq t} x_i^0 x_k^0 \bar{x}_t^0 + \prod_{k \in N(i), k \neq t} \bar{x}_i^0 \bar{x}_k^0 x_t^0$$

If $x_i^0 = 1$ then $\bar{x}_t^0 = 0$ and so the first product has value 1 while the second product has value 0. If $x_i^0 = 0$ then $x_t^0 = 1$ and so the first product has value 0 while the second product has value 1.

When $j \neq t$, then each product of $D_i(X^0)$ contains the factor $x_i^0 x_j^0$ or $\bar{x}_i^0 \bar{x}_j^0$ and is thus 0.

Hence $F_i(X^0) = 1$ in this case. $\square$

We have obtained the following explicit complexity result relating MATCHING CUT to SATISFIABILITY.

Theorem 1.3.2: *MATCHING CUT is an instance of SATISFIABILITY.*

Proof: Given any instance of MATCHING CUT on a graph $G = (V, E)$ we reduce it in polynomial time into an instance of SATISFIABILITY, as follows. Construct the function F from G as given in formula (6) above. By the preceding lemma and the observation made in the paragraph preceding formula (6), F is satisfiable if and only if there is a Boolean vector $X^0 \in R^V$ such that the map $\lambda : V \rightarrow \{0, 1\}$ given by $\lambda(i) = x_i^0$ is a decomposition colouring of G . Thus F is satisfiable if and only if G has a matching cutset. It only remains to be shown that that F can be constructed in a number of steps which is bounded by a polynomial in $|V|$.

For each $i = 1, 2, \dots, n$, F_i has only $2|N(i)| + 2$ terms, and the number of variables in each term is at most equal to the maximum degree of a vertex of G plus one. $A(X)$ has two terms, each in n variables. Thus, F can be constructed from G in polynomial time. $\square$

4 Locally Connected Graphs

In this section, we shall give a sufficient condition for a graph to have neither a matching cutset nor a stable cutset. The condition will be sufficiently general so that the class of graphs which satisfy it will include several well known types of graphs.

Definition: An undirected graph $G = (V, E)$ is locally connected if and only if for each $v \in V$, the subgraph $G[N(v)]$ is connected.

An example of a locally connected graph is shown in Figure 1.4.1(a). Figure 1.4.1(b) shows a graph which is not locally connected ($N(u)$ induces a disconnected graph).

We shall prove the following facts about locally connected graphs:

- (i) The class of locally connected graphs contains all 2-connected chordal graphs and all plane triangulations.

- (ii) Connected, locally connected graphs on more than 3 points can have neither a matching cutset nor a stable cutset.

Figure 1.4.1(a)-(c)

Proposition 1.4.1 : Let $G = (V, E)$ be a 2-connected chordal graph. Then G is locally connected.

Proof: We shall in fact show that 2-connected chordal graphs have the following (stronger) property:

- (a) For each $u \in V$ every point of $N(u)$ is adjacent to a point of $N(u)$.
- (b) Let $u \in V$ and $x, y \in N(u)$ such that $xy \notin E$. Then every shortest path P from x to y with $u \notin V(P)$ satisfies $V(P) \subset N(u)$.

Let v be a neighbour of u . Since G is 2-connected, any two points lie on a circuit. Let C be a shortest circuit containing u and v . Since G is chordal, this circuit has length 3 and the third point is in $N(u)$ and is adjacent to v .

Suppose that some points $x, y \in N(u)$ are not adjacent but that some point z is adjacent to both of them. Then the 4-cycle u, x, z, y, u has a chord and so $z \in N(u)$.

Consider now a shortest path P from x to y such that $u \notin V(P)$. Let k be the number of points of P strictly between x and y . We show by induction on k that $V(P) \subset N(u)$.

The last paragraph dealt with the case $k = 1$. Suppose that $k \geq 2$. Consider the cycle C through u defined as (u, x, P, y, u) . There must be a chord, and by minimality of P the chord has the form ut where $t \in V(P)$ and $t \neq x, y$. At least one of x and y is not joined to t otherwise we have $k = 1$. If $xt \notin E$, then by induction all points of P between x and t are in $N(u)$. If $ty \in E$ then we are done and if $ty \notin E$ then we again apply induction to the points of P between t and y . This demonstrates property (b) and completes the proof of the Proposition. [1]

It is quite easy to show that a 2-connected plane triangulation has the properties (a) and (b) listed in the proof of the last proposition. Let $G = (V, E)$ be a plane triangulation. Let $u \in V$, and $x \in N(u)$. Since G is a plane triangulation, every edge is in a triangle. So, there is a $y \in V$ such that (u, x, y) induces a triangle. Hence, $N(x) \cap N(u) \neq \emptyset$. To demonstrate that plane triangulations satisfy the property (b) listed in the proof of Proposition 1.4.1, let $x, y \in N(u)$ such that $xy \notin E$. Let P be a shortest path in G from x to y which does not pass through u . Every face in the region bounded by P and the edges ux and uy has degree 3, and so $V(P) \subseteq N(u)$.

Figure 1.4.1(c) shows an example of a locally connected graph which is not a chordal graph.

We have shown that the class of locally connected graphs contains all 2-connected plane triangulations as well.

We shall use the notation P_k for a path of length k .

Theorem 1.4.2: *Let $G = (V, E)$ be a connected locally connected simple graph other than P_1 , and U and F be vertex and edge cutsets of G , respectively. Then there exists $u, v \in U$ and $e, f \in F$ such that uv is an edge and e, f have a common endpoint.*

Proof: Let $S \subset U$ be a minimal vertex cutset of G . Let G_1, G_2 be any two components

of $G - S$ and $u \in S$. Minimality of S implies that for $i = 1, 2$, $N(u) \cap V(G_i) \neq \emptyset$. For $i = 1, 2$, let $y_i \in N(u) \cap V(G_i)$. Since G is locally connected and $y_1 y_2 \notin E$ then there is a path in G from y_1 to y_2 such that $V(P) \subset N(u)$. Thus there is a point $v \in S \cap (V(P))$ such that $uv \in E$. Hence U is not a stable set.

To complete the proof, we show that in a connected, locally connected graph other than P_1 , every edge is contained in a triangle.

Let $g = xy$ be an edge of G . Since G is connected and has more than 2 points, we can assume x has a neighbour $z \neq y$. Let P be a path from z to y such that $V(P) \subset N(x)$.

Let t be the neighbour of y on P . Then $\{xy, xt, yt\}$ is the edge set of a triangle.

Now, every edge lies in a triangle implies that G does not have a matching cutset, and this implies that for any edge cutset F of G there are $e, f \in F$ such that e and f have a common end point. □

CHAPTER 2

NP-COMPLETENESS RESULTS

2.1 Some Special NP-Complete Problems

Detailed information about the subject of NP-Completeness can be found in [6]. We wish to introduce the following concept:

Definition: Let P be the problem of recognizing graphs with a property Q . We shall say that the complexity of P has a degree jump at (k, n_0) if there are integers k and n_0 and a class C of graphs such that P is NP-Complete for graphs in C with maximum degree k , but every graph in C with maximum degree $\leq k - 1$ has property Q , provided it has at least n_0 points.

One of the interesting results which we shall prove in this chapter is that the complexity of MATCHING CUT has a degree jump at $(4, 7)$, when we take C to be the class of all connected graphs. In the literature, there seems to be only a few NP-Complete problems which have been shown to have degree jumps in their complexity. We give 3 illustrative examples:

1. **Graph k -edge colourability:** A graph is k -edge colourable if its edges can be coloured with no more than k colours such that no edge is coincident with an edge of the same colour. It is NP-Complete to recognize k -edge colourable graphs even for cubic graphs

and $k = 3$ [15]. However, every simple graph with maximum degree $\leq k - 1$ is k -edge colourable (by Vizing's Theorem, which states that a simple graph with maximum degree k has chromatic index k or $k + 1$. (See [1])). Hence, for every $k \geq 3$, the complexity of graph k -edge colourability has a degree jump at $(k, 0)$ when C is the class of all simple graphs.

2. Hamilton Circuit: A Hamilton circuit in a graph $G = (V, E)$ is a simple cycle which passes through all points of G . Recognizing graphs with a Hamilton circuit is an NP-Complete problem for 2-connected graphs with maximum degree 3 [7]. However, every 2-connected graph with maximum degree 2 and on more than two points is a Hamilton circuit. Thus the complexity of the Hamilton circuit problem has a degree jump at $(3, 3)$, when C is the class of all 2-connected graphs.

3. 3-Vertex Colourability: A graph is *3-vertex colourable* if its vertices can be coloured with no more than 3 colours such that adjacent vertices have different colours. It is mentioned in the first example above that it is an NP-Complete problem to recognize graphs which are 3-edge colourable even if the input is restricted to 3-regular graphs. Now, a graph G is 3-edge colourable if and only if the line graph of G is 3-vertex colourable. (A definition of line graphs can be found in Section 2 of this chapter). Moreover, the line graph of a 3-regular graph is a 4-regular graph. It follows from this that recognizing 3-vertex colourable graphs is an NP-Complete problem for 4-regular graphs. However, R.L. Brooks [3] proved that if G is a connected graph which is not a complete graph or an odd circuit and k is the maximum degree of G , then G is k vertex colourable. Now the only complete graphs with maximum degree ≤ 3 are K_1 , K_2 , K_3 and K_4 , and these have fewer than 5 points. Moreover, every odd circuit is 3-vertex colourable. Hence every connected graph with maximum degree 3 and on more than 4 points is 3-vertex colourable.

Thus 3-vertex colourability has a degree jump at $(4, 5)$, when the input is restricted to connected graphs.

Figure 2.1.1(a)-(b)

Definition: Let P be the problem of recognizing graphs with a property Q . We shall say that the complexity of P has a subgraph jump at (H, n_0) if there is a class C of graphs, a fixed graph H and a positive integer n_0 such that P is NP-Complete for graphs in C in general, but every graph in C which does not contain an induced subgraph which is isomorphic to H has property Q , provided it has at least n_0 points.

One of our results in this chapter will be that the complexity of STABLE CUT has a subgraph jump at $(K_4, 10)$ when C is the class of simple line graphs and K_4 is the complete graph on 4 points. Two other examples of graph recognition problems whose complexities have subgraph jumps are given next:

1. **Hamilton circuit:** Let C be the class of 2-connected graphs, and P be the problem of recognizing graphs in C with a Hamilton circuit. Then P is NP-Complete for C [17]. However, Goodman and Hedetniemi [10] proved that any 2-connected graph which does not contain an induced subgraph isomorphic to $K_{1,3}$ or Z_1 must have a Hamilton circuit. ($K_{1,3}$ and Z_1 are the graphs shown in Figure 2.1.1(a) and (b), respectively). Hence the

complexity of the Hamilton circuit problem has a subgraph jump at $(H, 1)$ where C is the class of 2-connected graphs and H consists of both of the graphs of Figure 2.1.1(a) and (b).

2. **3-Set Partition:** We shall say that a graph $G = (V, E)$ has a *3-set partition* if there is a partition of V into non empty disjoint sets V_1, V_2 and V_3 such that V_1 is a clique and every point of V_2 is adjacent to every point of V_3 .

In Section 2.3 of this chapter, we shall prove that the problem P of recognizing graphs with a 3-set partition is NP-Complete. We shall prove also that every complement of a 2-connected line graph which does not contain a subgraph which is isomorphic to $\bar{K}_4$, (the complement of a complete graph on 4 points), must have a 3-set partition, provided it has more than 9 points. Hence the complexity of P has a subgraph jump at $(\bar{K}_4, 10)$.

2. The NP-Completeness of MATCHING CUT and STABLE CUT.

A *hypergraph* H is a finite set $V(H)$ of points and a finite set $E(H)$ of edges together with an assignment of subsets of $V(H)$ to each edge E . The subset of $V(H)$ which is assigned to an edge E is called the set of endpoints of E . We write $H = (V(H), E(H))$ and say an edge E is *incident* with a point v if v is an endpoint of E .

A hypergraph H is *r-uniform* if every edge of H has r endpoints. Thus a *2-uniform* hypergraph is a graph without loops. Detailed information about hypergraphs can be found in [1].

A hypergraph is *bicolourable* if its points can be coloured red and blue so that no edge is monochromatic.

L. Lovász [19] considered the problem of recognizing bicolourable hypergraphs. He proved that this problem is NP-Complete even if the input is restricted to 3 - uniform hyper

graphs.

Chvátal [4] gave a polynomial-time reduction of any instance of the hypergraph bicolouration problem for 3-uniform hypergraphs into an instance of MATCHING CUT. Indeed, given any 3-uniform hypergraph, he shows how to construct a graph G with maximum degree 4 so that the given hypergraph is bicolourable if and only if G has a matching cutset. Thus, MATCHING CUT is NP-Complete even if the input is restricted to graphs with maximum degree 4. In the paper cited above, Chvátal also gave an algorithm to solve MATCHING CUT when the input is restricted to graphs with maximum degree at most 3. (We shall in fact show, in Chapter 3, that every connected graph with maximum degree at most 3 has a matching cutset provided it has at least 8 points). We show in this section that the degree limit of solvability of MATCHING CUT, as obtained by Chvátal, can be refined. We do this in Theorems 2.2.1 and 2.2.3.

Figure 2.2.2(a)-(d)

Theorem 2.2.1: *It is NP-Complete to recognize graphs with a matching cutset even if the input is restricted to graphs in which every point has degree 3 or 4, and no two points of degree 3 are adjacent.*

Proof: We only need to show how to transform any instance of MATCHING CUT for a graph with maximum degree 4 into an instance of the same problem in a graph in which each point has degree 3 or 4, and no two points of degree 3 are adjacent.

If u is any point of degree 2 in G , replace u by the subgraph with 5 points as shown in Figure 2.2.2(a)-(b). In case G has a point of degree 1, introduce a K_3 and adjoin u to all the points of the K_3 . Now let u and v be adjacent points of degree 3. Substitute for u and v and the edge e joining them as shown in Figure 2.2.2(c)-(d). All unlabelled points in Figure 2.2.2(a)-(d) are new points. (They are not points of G). It is easy to verify that the substitutions described preserve matching cutsets, and they do not create any new ones. This completes the proof of Theorem 2.2.1. $\square$

We shall now analyze the case when the input is restricted to graphs with maximum degree 4 but without any points of degree 3. In this analysis, we shall restrict attention to 2-edge connected graphs since any graph with a cut-edge has an obvious matching cutset. The main result of this analysis is stated in Theorem 2.2.3, where we assume that G is 2-connected. We show first that this is a reasonable assumption. For this purpose, we shall find it convenient to use the following notation: if H is a subgraph of G and $v \in V(G)$, we shall use $H + v$ to denote the subgraph induced by $V(H) \cup \{v\}$.

Proposition 2.2.2: *Let G be a connected graph with a cut-point v . Then G has a matching cutset if and only if there is a component H of $G - v$ such that $H + v$ has a matching cutset.*

Proof : $\Rightarrow$: Let F be a minimal matching cutset of G . Let $e = xy \in F$ so that there is no path in $G - F$ from x to y . We may assume that $x \neq v$ since at least one of x and y is different from v . Let H be the component of $G - v$ which contains x . Suppose $y \notin V(H + v)$. Then there is a component H' of $G - v$ such that $y \in V(H')$, and

$H' \neq H$. But now $e = xy$ is a path from H to H' which does not use v , contradiction. So, $y \in V(H + v)$.

Suppose there is $f \in F$ such that f has an endpoint in a component K of $G - v$ where $K \neq H$. Then, using the argument which showed that $e \in E(H + v)$, we have $f \in E(K + v)$. Since F is a minimal edge cutset of G , then $G - F + f$ is a connected graph. So, there is a path P in $G - F + f$ from x to y . Now, the portions of P which are in $H + v$ give a path in $G - F$ from x to y , and this is a contradiction.

$\Leftarrow$: Let F be a matching cutset of $H + v$, where H is a component of $G - v$. Let H_1 be the component of $H + v - F$ which contains v , and H_2 be any other component of $H + v - F$. Every path in G from a component of $G - v$, other than H , to a point of H_2 uses a point of H_1 (in fact such a path goes through v). Every path from a point of H_1 to a point of H_2 uses an edge of F . Hence, $G - F$ is a disconnected graph. []

Theorem 2.2.3 *Every simple 2-connected graph with maximum degree 4 such that no point of degree 4 is adjacent to a point of degree 3 or 4 has a matching cutset, provided it has at least 12 points.*

Proof The proof is an algorithm which finds a matching cutset of G after examining at most $|V| + 11$ points and the edges incident with them.

Let v be a point of degree 4, with neighbours u_i for $i = 1, 2, 3, 4$. Denote by H the graph $G[\{u_1, u_2, u_3, u_4\}]$. The graph H can have at most one edge, for if $|E(H)| = 2$ then $G[\{v, u_1, u_2, u_3, u_4\}]$ would be a component of G . If $|E(H)| = 1$ then let $u_1 u_2$ be an edge of H . Let x_3 and x_4 be neighbours of u_3 and u_4 other than v . (See Figure 2.2.3(a)). But $x_3 \neq x_4$, otherwise x_3 would be a cut-point of G . Therefore $\{u_3 x_3, u_4 x_4\}$ is a matching cutset of G . Now suppose $|E(H)| = 0$. Let x_i be the neighbour of u_i different from v for $i = 1, 2, 3, 4$. If all the points x_i are distinct then $\{u_1 x_1, u_2 x_2, u_3 x_3, u_4 x_4\}$ is a matching

cutset of G . If precisely two of the x_i are the same, say $x_1 = x_2$ and $x_3 \neq x_4$, then $\{vu_1, u_2x_2, u_3x_3, u_4x_4\}$ is a matching cutset. Suppose that three of the x_i are the same point, say $x_1 = x_2 = x_3 = x \neq x_4$. But x is not a point of degree 3, for in that case x_4 would be a cut-point of G . So let y be the neighbour of x different from u_1, u_2, u_3 and u_4 . Now the point y is not x_4 otherwise we get a component with 6 points. Now $\{xy, u_4x_4\}$ is a matching cutset of G . We cannot have $x_1 = x_2 = x_3 = x_4$ because we would then get a component with 6 points. Finally it remains to consider the case $x_1 = x_2 \neq x_3 = x_4$. The points x_1 and x_3 cannot both have degree 2 otherwise $G[\{v, u_1, u_2, u_3, u_4, x_1, x_3\}]$ would be a component of G .

Figure 2.2.3(a)-(b)

Case 1: at least one of x_1 and x_3 is a point of degree 3. Let x_1 have degree 3 (The other case is symmetric). Let y_1 be the neighbour of x_1 which is different from u_1 and u_2 . If $y_1 \neq x_3$ then $\{vu_4, u_3x_3, x_1y_1\}$ is a matching cutset of G . Suppose on the other hand, that $y_1 = x_3$. Then at least one of x_1 and x_3 has degree 4, otherwise $G[\{v, u_1, u_2, u_3, u_4, x_1, x_3\}]$ would be a component of G . But now we have a point of degree 4 with a neighbour of degree more than 2 which is a contradiction.

Case 2: Neither x_1 nor x_3 has degree 3. Then both points x_1 and x_3 have degree 4, for if

one had degree 2 and the other degree 4 then the one with degree 4 would be a cut-point. Now x_1x_3 is not an edge of G since by hypothesis points of degree 4 are not allowed to have neighbours of degree more than 2. Let y_1 and y_2 be neighbours of x_1 different from u_1 and u_2 . Let y_3 and y_4 be similar neighbours of x_3 (see Figure 2.2.3(b)). Note that each y_i is a point of degree 2 in G . Let $K = G[\{y_1, y_2, y_3, y_4\}]$. If K has 2 edges then $G[\{v, x_1, x_3, u_i, y_i : i = 1, 2, 3, 4\}]$ is a component of G with 11 points. Suppose that $|E(K)| = 1$. If y_3y_4 is an edge of K then $\{vu_1, u_2x_1\}$ is a matching cutset. Similarly, if y_1y_2 is an edge then $\{vu_4, u_3x_3\}$ is a matching cutset. So we can assume that the edge of K is y_2y_3 . Then $\{x_1y_2, x_3y_4\}$ is a matching cutset of G . Finally suppose that K has no edges. For $i = 1, 2, 3, 4$, let t_i be a neighbour of y_i with $t_i \neq x_1$ and $t_i \neq x_3$. Then $\{vu_1, u_2x_1, x_3y_3, y_4t_4\}$ is a matching cutset of G . $\square$

We will now show that STABLE CUT is an NP-Complete problem even if the input is restricted to line graphs with maximum degree 6.

Given a graph $G = (V, E)$, the line graph of G , which we denote by $L(G)$, is the graph whose points are the edges of G , and two points of $L(G)$ are adjacent if and only if the corresponding edges in G are coincident. Line graphs are also called *edge graphs* by some authors. For more information about line graphs the reader should consult any standard book on graph theory (for example [9], page 101).

If F is a set of edges of G then $L(F)$ will denote the points of $L(G)$ arising from F . More generally, if G_1 is any subgraph of G then $L(G_1)$ is the subgraph of $L(G)$ arising from G_1 . If H is a subgraph of $L(G)$ then we will denote the pre-image of H in G by $L^{-1}(H)$.

Lemma 2.2.4: *Let $G = (V, E)$ be a simple, 2-connected graph. Then a subset F of E is a matching cutset of G if and only if $L(F)$ is a stable cutset of $L(G)$.*

Proof: $\Rightarrow$ By construction of the line graph $L(G)$ the set of points $L(F)$ is a stable set

whenever F is a matching in G . Let G_1 and G_2 be any two components of $G - F$. Let P be any path in $L(G)$ from $L(G_1)$ to $L(G_2)$. Then $L^{-1}(P)$ is a path in G from G_1 to G_2 , and so it uses an edge of F . Consequently, P uses a point of $L(F)$. Thus $L(G) - L(F)$ is a disconnected graph.

⇐ Let Y be any stable cutset of $L(G)$. Let e_1 and e_2 be any two edges of $L^{-1}(Y)$. For $i = 1, 2$, let $v_i = L(\{e_i\})$. Then v_1 and v_2 are points of Y and so they are non adjacent vertices of the line graph. Hence e_1 and e_2 do not have a common end point in G . So, $L^{-1}(Y)$ is a matching in G . Let G'_1 and G'_2 be any two components of $L(G) - Y$. Let $L^{-1}(G'_1)$ and $L^{-1}(G'_2)$ be the pre-images of G'_1 and G'_2 in G . Let P be any path in G from $L^{-1}(G'_1)$ to $L^{-1}(G'_2)$. Then $L(P)$ is a path in $L(G)$ from G'_1 to G'_2 and so it uses a point of Y . Hence P uses an edge of $L^{-1}(Y)$. Thus $G - L^{-1}(Y)$ is a disconnected graph.

□

We are now in a position to give some limits on the solvability of STABLE CUT.

Theorem 2.2.5: *STABLE CUT is an NP-Complete problem even if the input is restricted to line graphs with maximum degree 6 and for which every point is contained in a K_3 or a K_4 . However, every connected K_4 -free line graph has a stable cutset provided its root graph is a 2-connected graph with more than 7 points.*

Proof: By Lemma 2.2.4 and the NP-Completeness of MATCHING CUT it follows that STABLE CUT is an NP-Complete problem. Moreover, we have shown in Theorem 2.2.1 that MATCHING CUT remains NP-Complete if the input is restricted to graphs in which every point has degree 3 or 4. If G is such a graph then every point of the line graph of G will be in a K_3 or a K_4 , and the maximum degree will be at most 6. Combining this result with the Lemma, we get that STABLE CUT remains NP-complete for line graphs with maximum degree 6 such that every point is in a K_3 or a K_4 .

The proof of the second part of the theorem uses a result which we will prove in Chapter 4. There, we will prove that every connected graph with maximum degree at most 3 and on more than 7 points has a matching cutset. Now if G is a connected K_4 -free line graph with a root graph H , then H is connected, and has maximum degree at most 3. If in addition $|V(H)| \geq 8$ then H has a matching cutset F and the image of F in G is a stable cutset of G . $\square$

2.3 Related NP-Completeness Results

We begin this section by giving a polynomial equivalence between MATCHING CUT and a closely related problem.

Proposition 2.3.1: *MATCHING CUT is polynomially equivalent to the following problem P*

P : Given an undirected graph $G = (V, E)$ and an edge e , does G have a matching cutset which contains e ?

Proof: Let G be given. We shall first construct a graph G' containing a specified edge e so that G has a matching cutset if and only if G' has a matching cutset containing e .

Figure 2.3.1(a)-(b)

The construction of G' from G is illustrated in Figure 2.3.1(a). There, u and v are any

points of G and x, y, z, t are new points. It is easy to verify that if G has a matching cutset F then $F + e$ is a matching cutset of G' which contains e . Moreover, any matching cutset F' of G' containing e is such that $F' - e$ is a matching cutset of G .

Now let a graph G and an edge e be specified. We shall construct a graph $\bar{G}$ such that $\bar{G}$ has a matching cutset if and only if G has a matching cutset containing e .

For each edge f of G which is coincident with e we introduce a new point x and we join x to the end points of f . (See Figure 2.3.1(b)). It is easy to see that if F is any matching cutset of G containing e then F is a matching cutset of $\bar{G}$. Furthermore, any matching cutset $\bar{F}$ of $\bar{G}$ does not contain an edge which is coincident with e (since all edges which are coincident with e are in a triangle). So, $\bar{F} + e$ is a matching cutset of G . $\square$

We turn now to the complexity of solving MATCHING CUT and STABLE CUT on *perfect* graphs.

A graph G is called *perfect* if for every induced subgraph G' of G , the chromatic number of G' is equal to the clique number of G' . Perfect graphs have stimulated a lot of research in recent years. A detailed discussion of perfect graphs can be found in [9].

Some famous hard problems are solvable in polynomial time when the input is restricted to perfect graphs. These include the *independent set* problem, *clique number*, *partition into cliques* and *chromatic number* [12].

We should observe however that the hard problems which are solvable when the input is restricted to perfect graphs seem to be only those problems of computing the defining parameters of perfect graphs and their complements. It is not surprising then that perfection does not buy tractability when it comes to MATCHING CUT and STABLE CUT. We prove this now.

Theorem 2.3.2: *Both MATCHING CUT and STABLE CUT remain NP-Complete when*

the input is restricted to perfect graphs.

Proof: We shall prove the following *stronger* result: MATCHING CUT is NP-Complete for bipartite graphs with maximum degree at most 8. The validity of the Theorem will then follow from the fact that both bipartite graphs and the line graphs of bipartite graphs are perfect graphs.

Given a simple graph $G = (V, E)$ let G' be the graph obtained from G by replacing each edge e by a 4-cycle α_e as shown in Figure 2.3.2.

The graph G' has two types of points: *old points* (points of G), and *new points* (points of G' which are not points of G). Every edge of G' joins an old point to a new point, so *old* and *new* is a proper 2-colouration of G' . Thus G' is a bipartite graph.

Figure 2.3.2

The next step is to show that G has a matching cutset if and only if G' has a matching cutset. If G has a matching cutset then it has a colouring of its vertex set by the colours red and blue so that both colours are assigned, and for each $v \in V$, there is at most one $u \in N(v)$ such that v and u are coloured differently. (See Proposition 1.2.1). Beginning with such a decomposition colouring of G , we colour G' by using the following algorithm:

Step (i): Every *old* point retains the colour it has in G .

Step (ii): Let $\{x, y\}$ be a pair of new points which have the same neighbours u and v . (This is reasonable because every new point belongs to precisely one such pair). Now, u and v are old points, so they have been coloured in Step (i). If u and v have the same colour, then colour x and y by this colour of u and v . If u and v have different colours, then colour x red and y blue.

In G' every new point x has precisely two neighbours, and the colour of x is chosen in Step (ii) in such a way that at most one of the neighbours of x has a colour which is different from the colour of x .

Let x be an old point of G' . If x does not have an adjacent old point in G which is coloured differently from x , then Step (ii) coloured all new neighbours of x by the colour of x , and so x does not have any neighbours which have a colour different from that of x . If x is adjacent to an old point y where y is coloured differently from x , then precisely one new point out of the unique pair of points which have $\{x, y\}$ as its neighbourset, say t , is coloured differently from x . Since every neighbour of x different from y has the colour of x , then Step (ii) guarantees that in G' no neighbour of x other than t has a colour which is different from the colour of x .

Conversely, suppose G' has a decomposition colouring. Consider the colouring of G which is inherited from the colouring of G' . We show this colouring of G is a decomposition colouring. If not, let a red point x have two blue neighbours y and z . (The case when x is blue and y and z are red is entirely symmetric). Let y_1 and y_2 be the unique pair of new points of G' which have x and y as common neighbours. Let z_1 and z_2 be the similar pair of new points of G' which have x and z as common neighbours.

In G' , at most one of y_1 and y_2 is coloured red. Both y_1 and y_2 are not coloured blue, otherwise we get the contradiction that the red point x has two blue neighbours. The

same argument applies to the colours of z_1 and z_2 in G' , so we may assume that y_1 and z_1 are red while y_2 and z_2 are blue. (See Figure 2.3.3). But now we get the contradiction that in G' , the red point x has two blue neighbours y_2 and z_2 .

It is not possible that in G' all *old* points have the same colour because every *new* point has two *old* neighbours and this would force all points of G' to have the same colour which is a contradiction. So the colouring that G inherits from G' has both a red point and a blue point.

We have thus shown that G has a matching cutset if and only if G' has a matching cutset. This completes the proof that MATCHING CUT remains NP-Complete for bipartite graphs. Hence STABLE CUT remains NP-Complete when the input is restricted to the line graphs of bipartite graphs. $\square$

Figure 2.3.3

The 3-SET PARTITION Problem: We shall end this chapter by proving that the 3-SET PARTITION problem defined in Section 2.1 (of this chapter) is NP-Complete for perfect graphs.

Theorem 2.3.3: 3-SET PARTITION is NP-Complete even if the input is restricted to perfect graphs.

Proof: Let $G = (V, E)$ be any connected graph and $\bar{G}$ be the complement of G . (That is the graph on the same vertex set as G , but with $xy \in E(\bar{G})$ if and only if $xy \notin E(G)$).

Suppose S is a stable cutset of G . Let G_1 be any component of $G - S$. Put $V_1 = S$, $V_2 = V(G_1)$, and $V_3 = V - (V_1 \cup V_2)$. There is no edge in G from a point of V_2 to a point of V_3 , and so in $\bar{G}$ every point of V_2 is adjacent to every point of V_3 .

V_1 is a clique in $\bar{G}$ because S is a stable set in G . Moreover, none of V_1 , V_2 , and V_3 is the empty set. So, $\bar{G}$ has a 3-set partition.

Conversely, suppose $\bar{G}$ has a 3-set partition. Let V_1 be the clique in this partition and V_2 and V_3 be the other sets. Then in G , no point of V_2 has a neighbour in V_3 , so every path in G from a point of V_2 to a point of V_3 must use a point of V_1 . Moreover, V_1 is a stable set in G , and so G has a stable cutset.

Since we have reduced STABLE CUT into 3-SET PARTITION, and since in addition 3-SET PARTITION is clearly in NP, then 3-SET PARTITION is NP-Complete. In fact, since STABLE CUT remains NP-Complete for the line graphs of bipartite graphs and since the complements of line graphs of bipartite graphs are perfect graphs, we have shown that 3-SET PARTITION is NP-Complete for perfect graphs. $\square$

We shall prove in chapter 4 that every connected simple line graph on more than 9 points which does not have a subgraph which is isomorphic to K_4 has a stable cutset. It follows that if G is a graph with independence number at most 3 such that $\bar{G}$ is a simple, connected line graph then G must have a 3-set partition. Hence the complexity of 3-SET PARTITION has a subgraph jump at $(\bar{K}_4, 10)$.

CHAPTER 3

MATCHING AND STABLE CUTSETS IN PLANAR GRAPHS

3.1 Planarity and Tractability.

The complexities of STABLE CUT and MATCHING CUT for planar graphs seem to be quite difficult to determine. However, certain observations lead us to conjecture that these problems remain NP-Complete when the input is restricted to planar graphs. In fact, practically all NP-hard problems remain NP-hard when restricted to planar graphs. We describe below 4 NP-hard problems which are known to be solvable in polynomial time when restricted to planar graphs.

In Section 3.2, we give some partial results concerning the solvability of MATCHING CUT for planar graphs. We prove that planar graphs which do not have faces of degree more than four have disjoint matching cutsets, and we can find all their matching cutsets in $O(|E|^2)$ time. We also give a polynomial time reduction of MATCHING CUT for planar graphs into the following problem (which we find to be of independent interest):

Input: A plane graph $G = (V, E)$ and a pair of points $x, y \in V$.

Output: Either a path P in G from x to y such that P does not contain more than one edge from any face of G , or the message that no such path exists.

Partial results concerning STABLE CUT for planar graphs are given in Section 3.3. We prove there that planar line graphs which have a stable cutset can be recognized in $O(|V|^2)$

time.

We now describe four NP-hard problems which are known to be solvable in polynomial time when the input is restricted to planar graphs.

MAX-CUT: Given $G = (V, E)$, a weight function $(w(e) : e \in E)$ and an integer k , does there exist a partition $\{V_1, V_2\}$ of V such that if $E' = [V_1, V_2]$, then

$$\sum_{e \in E'} w(e) \geq k?$$

This problem is NP-Complete even if the maximum degree of G is 3 and $w(e) = 1$ for each $e \in E$ [30]. However, it is shown in [21] that MAX-CUT can be solved in polynomial time when the input is restricted to planar graphs.

CLIQUE: The problem of recognizing graphs with a set of k or more points which induce a complete subgraph is NP-Complete [6]. This problem becomes trivially solvable when the input is restricted to planar graphs. This is due to the fact that the maximum possible clique size of a planar graph is 4, and hence exhaustive search requires only $O(n^4)$ time. Actually, a linear time algorithm to solve CLIQUE has been given in [22].

COUNTING PERFECT MATCHINGS: The problem of counting the number of perfect matchings of a graph has been shown to be $\#P$ -Complete [27] for general graphs but this problem is solvable in polynomial time for planar graphs [6].

BIPARTITE SUBGRAPH: Given $G = (V, E)$ and a positive integer k , is there a subset E' of E such that $|E'| \geq k$ and the graph $G' = (V, E')$ is bipartite? This problem is NP-Complete for general graphs [6]. However, a polynomial time algorithm to solve it when the input is restricted to planar graphs is given in [13].

We shall make use of the following terminology:

By a *plane graph* we shall mean an embedding, in the plane, of a planar graph. Given a plane graph $G = (V, E)$ we shall let $G^* = (\Phi, E^*)$ be the plane dual of G . Here, Φ is the set of faces of G . In constructing G^* from G in the plane, we denote by e^* the edge of G^* which crosses the edge e of G . More generally, for $T \subseteq E$, we will let $T^* = \{e^* : e \in T\}$. We shall similarly associate each vertex f^* of G^* with the face f of G in which f^* is drawn. If an edge e is on the boundary of a face f , we shall say f and e are *incident*. Note that an edge e is incident with a face f , if and only if the edge e^* is incident with the vertex f^* .

We shall use the following property of planar graphs:

Proposition 3.1.1: *T is an edge cutset of a plane graph G if and only if T^* contains a circuit of G^* .*

Proof: Let T be an edge cutset of G . We show by induction on $|T|$ that T^* contains a circuit of G^* . If $|T| = 1$ then T^* is a loop of G^* . Suppose the claim is true for cutsets with fewer than k edges and let T be a cutset with $|T| = k \geq 2$. Delete an edge e from T . Then $T - e$ is an edge cutset of $G - e$. By the inductive hypothesis, $(T - e)^*$ contains a circuit C^* of $(G - e)^*$. Hence $C^* + e^*$ contains a circuit of G^* .

Conversely, Let T^* contain a circuit C^* of G^* . Let u and v be faces of G^* which are contained in the interior and in the exterior of the region bounded by C^* , respectively. Now, u and v are vertices of G , and every path in G from u to v uses an edge of $C \subseteq T$. Hence there is no path in $G - T$ from u to v . Thus, T is an edge cutset of G . $\square$

3.2 Matching Cutsets in Planar Graphs

In this section, we give two partial results on the solvability of MATCHING CUT for planar graphs, and introduce a new problem which is at least as hard as MATCHING CUT for planar graphs, and may be of independent interest.

Definition: Let G be a plane graph. A path P in G is *independent* if no two edges of P are incident with the same face of G . The *Independent Path Problem* (which we shall abbreviate as INDEPENDENT PATH), is: Given a plane graph $G = (V, E)$ and a pair of points $x, y \in V$, does G have an independent path from x to y ? We shall prove that this problem is at least as hard as recognizing planar graphs with a matching cutset.

Definition: Given a plane graph G a circuit C of G will be called a *Proper Jordan Separator* (PJS) if no two edges of C are incident with the same face of G .

Figure 3.2.1(a)-(b)

Lemma 3.2.1: T is a minimal matching cutset of a plane graph G if and only if T^* is a PJS of G^* .

Proof: T is a minimal matching cutset of G if and only if T is an edge cutset of G , for each $e \in T$, $T - e$ is not an edge cutset of G and no two edges of T are incident with the same point, if and only if T^* contains a circuit of G^* , for each $e^* \in T^*$, $T^* - e^*$ does not contain a circuit of G^* and no two edges of T^* are incident with the same face of G^* , if

and only if T^* is a circuit of G^* and no two edges of T^* are incident with the same face of G^* if and only if T^* is a PJS of G^* . $\square$

Figure 3.2.1(a) shows a plane graph G with a matching cutset F shown as wiggled edges.

The corresponding PJS of G^* is shown in Figure 3.2.1(b).

Figure 3.2.2

Proposition 3.2.2 : Let $G = (V, E)$ be a plane graph without a face of degree 5 or more, and $e \in E$. Suppose G has a minimal matching cutset F which contains e . Then F is unique.

Proof : Suppose F' is a matching cutset of G which contains e and $F' \neq F$. Consider G^* . By Lemma 3.2.1, both F^* and F'^* are proper Jordan separators of G^* . We note that since G does not have a face of degree more than 4, then G^* has maximum degree ≤ 4 . Let P be a longest path in G^* such that $e^* \in E(P) \subseteq F^* \cap F'^*$. Since $F^* \neq F'^*$, we can find an edge $h^* = xy \in F^* - E(P)$ such that $x \in V(P)$ but $y \notin V(P)$ and an edge $t^* = xz \in F'^* - E(P)$ such that $z \notin V(P)$ (see Figure 3.2.2). Let r^* be the edge of P that has x as an end point. Now, neither of the edges h^* and t^* is incident with the same face of G^* as r^* , because F^* and F'^* are PJS's. This contradicts the fact that if v is a point of degree ≤ 4 in a plane graph and $f \in \delta(v)$, then there is at most one edge $g \in \delta(v) - f$

such that f and g are not incident with the same face. Hence $F' = F$. $\square$

Theorem 3.3.3 : *If $G = (V, E)$ is a planar graph without a face of degree 5 or more, then G has at most $|E|$ minimal matching cutsets. Moreover, we can construct all the matching cutsets of G in $O(|E|^2)$ steps.*

Proof : By Proposition 3.2.2, each $e \in E$ is contained in at most one minimal matching cutset of G . Hence G has at most $|E|$ minimal matching cutsets. Construct the plane dual G^* of G . This can be done in $O(|E|)$ steps. For each $e \in E$, we shall determine whether G has a matching cutset containing e by determining whether G^* has a PJS containing e^* .

We start by putting $T = \{e^* = xy\}$. We consider y to be the last point added to T , at this stage. The general step consists of looking for an edge $f^* = zt$ where z was the last point added to T , such that f^* is not incident with the same face of G^* as any edge in T . If such an edge f^* exists and $t = x$ then we have constructed a PJS of G^* which contains e^* . If no such edge f^* exists, then we conclude that G does not have a matching cutset containing e . If f^* exists and $t \neq x$, then we add f^* to T and consider t to be the last point to be added to T . We repeat the general step.

The validity of this procedure follows from the fact that at each step there is at most one edge of $G^* - T$ which may be added to T . This observation also implies that a PJS of G^* containing e can be found, or its non-existence established, in $O(|E|)$ time. Hence, all the matching cutsets of G can be found in $O(|E|^2)$ steps. $\square$

Theorem 3.2.4 : *INDEPENDENT PATH is at least as hard as MATCHING CUT for planar graphs.*

Proof : By Proposition 2.3.1, it is enough to show how to transform, in polynomial time, an arbitrary instance of the problem of finding a matching cutset of a planar graph

containing an edge e , into an instance of INDEPENDENT PATH.

Let $G = (V, E)$ be a plane graph and let $e \in E$. Let $G^* = (\Phi, E^*)$ be the plane dual of G , and x, y be the end points of the edge e^* . Let $D^* = \{t^* \in E^* : \text{there is a face } f \in \Phi^* \text{ such that } t^* \text{ and } e^* \text{ are both on the boundary of } f\}$. For each $g \in D^* - e^*$, subdivide g (replace g with a path of length 2). Let G' be the resulting graph. We claim that G has a matching cutset containing e if and only if G' has an independent path from x to y .

First, let F be a minimal matching cutset of G such that $e \in F$. By Lemma 3.2.1, F^* is a PJS of G^* , and $e \in F^*$. We claim that $P := F^* - e^*$ is an independent path in G' from x to y . Since F^* is a PJS of G^* , and $e^* \in F^*$ then $F^* \cap (D^* - e^*) = \emptyset$. Hence no edge of F^* is subdivided when G' is constructed from G^* . Thus $E(P) \subseteq E(G')$. Moreover, if two edges of G^* are not incident with the same face, then they do not become incident with the same face when an edge is subdivided. Thus P is an independent path in G' from x to y .

Now let P be any independent path in G' from x to y . Suppose there is an edge $t \in E(P) - E^*$. Then there is an edge t' of G' such that t and t' have a common end-point a in G' , where $d_{G'}(a) = 2$. (In other words, there is an edge in D^* whose image in G' is a path of length 2, consisting of the edges t and t'). The point a is not an end-point of P , since $a \notin V(G^*)$. Hence $t' \in E(P)$. But t and t' are incident with the same face of G' . This contradicts the fact that P is an independent path in G' . Hence $E(P) \subseteq E^*$.

We claim further that $P + e^*$ is a PJS of G^* . It is clear that $P + e^*$ is a circuit of G^* . There is no edge $t \in E(P)$ such that $t \in D^*$, so no edge of P is incident with the same face of G^* as e^* . Suppose there are edges t_1 and t_2 of P such that these two edges are incident with the same face f of G^* . Then f is not a face of G' , since P is an independent path of G' . But every face of G' is obtained from a face of G^* by a sequence of subdivisions of edges

of D^* . These edge subdivisions do not include subdividing either of t_1 or t_2 , and they do not change the fact that t_1 and t_2 are incident with the same face. This contradicts the fact that P is an independent path in G' . Hence $P + e^*$ is a PJS of G^* as claimed, and by Lemma 3.2.1, G has a matching cutset containing e . $\square$

3.3 Stable Cutsets in Planar Graphs

We begin this section with a proof that STABLE CUT is solvable in $O(|V|^2)$ time if the input is restricted to planar line graphs.

Theorem 3.3.1: *Let $G = (V, E)$ be a connected planar line graph with a 2-connected root graph K , and with maximum degree ≥ 5 . Let W be the set of points of degree 4 in K . Then G has a stable cutset if and only if there exists $U \subseteq W$ and a component H of $K - W$ such that the subgraph induced by $V(H) \cup U$ has a matching cutset.*

Proof: Suppose G has a stable cutset S . Let $L^{-1}(S)$ be the pre-image of S in K . By Lemma 2.2.4, $L^{-1}(S)$ is a matching cutset of K . Since G has points of degree ≥ 5 , then K has points of degree ≥ 4 . By a theorem of J. Sedlacek [25], K does not have points of degree more than 4, and all points of degree 4 in K are cut-points. Hence $W \neq \emptyset$, and each point in W is a cut-point of K . By Proposition 2.2.2 $L^{-1}(S)$ is contained in the edge-set of the subgraph induced by $V(H) \cup U$ where U is the set of points of W which have neighbours in the component H of $K - W$.

The remaining part of the proof follows immediately from Lemma 2.2.4. $\square$

We give an algorithm to recognize planar line graphs with a stable cutset. The input will be restricted to planar line graphs with maximum degree more than four (see the proof of Corollary 3.3.2 for a justification).

Algorithm B

Input: A connected planar line graph $G = (V, E)$ with maximum degree ≥ 5 .

Output: Either a stable cutset S of G or the message that G does not have a stable cutset.

Step 1: Construct the root graph K of G .

Step 2: Let $W = \{u \in K : d_K(u) = 4\}$, and Γ be the collection of components of $K - W$.

Step 3: Let $H \in \Gamma$ and $U = \{u \in W : N_G(u) \cap V(H) \neq \emptyset\}$. Now $V(H) \cup U$ induces a subgraph with maximum degree ≤ 3 . Use the algorithm in the proof of Theorem 4.2.1 to determine, in linear time, whether the subgraph induced by $V(H) \cup U$ has a matching cutset.

Step 4: If the subgraph induced by $V(H) \cup U$ has a matching cutset F , then the image of F in G is a stable cutset of G . Otherwise, replace Γ by $\Gamma - H$. If Γ is now empty, then report that G does not have a stable cutset. If $\Gamma \neq \emptyset$, then go back to Step 3.

Corollary 3.3.2: Planar line graphs with a stable cutset can be recognized in $O(|V|^2)$ time.

Proof: Let $G = (V, E)$ be a planar line graph. First suppose $d_G(u) \leq 4$ for each $u \in V$. Construct the root graph K of G . This can be done in $O(|E|)$ time [18]. Now K has maximum degree ≤ 3 , and so we can use Theorem 4.3.3 and the remarks following it, to determine in linear time whether K has a matching cutset. We then use Lemma 2.2.4.

Assume now that G has maximum degree ≥ 5 . Apply Theorem 3.3.1. We can use Algorithm B to determine whether the hypotheses of Theorem 3.3.1 are satisfied by G . Steps 1 and 2 of the algorithm take $O(|E|)$ and $O(|V|)$, respectively and are executed only once. Steps 3 and 4 are executed fewer than $|V|$ times and each execution takes $O(|V|)$ time. Hence, the whole procedure takes $O(|V|^2)$ time. $\square$

We shall introduce now a graph which is associated with a plane graph in a very natural way.

Definition: Given a plane graph $G = (V, E)$, associate with G a graph $G' = (V, E')$ where $f_0 = uv$ is an edge of G' if and only if in the graph G the points u and v are not adjacent and there is a face f of G which is incident with both u and v . We shall think of V as a fixed set of points in the plane and draw the edge f of G' in the face f of G . We shall refer to G' as the auxiliary graph of G .

An example of a plane graph and its auxiliary graph are shown in Figure 3.3.1

Figure 3.3.1

Suppose that G is a plane graph with a set of faces Φ and a stable cutset S . Then we shall denote by Γ_S the set of faces of G which have points from more than one component of $G - S$. In other words, $\Gamma_S = \{f \in \Phi : \text{there are components } G_1 \text{ and } G_2 \text{ of } G - S \text{ such that for } i = 1, 2, \text{ we have } V(f) \cap V(G_i) \neq \emptyset\}$. If $f \in \Phi$ then we shall denote by $\bar{f}$ the set of edges of G' which are drawn in the face f .

Lemma 3.3.3: Let G be a 2-connected plane graph. Suppose S is a minimal stable cutset of G . Then for each $v \in S$, there are two distinct faces f and g of G such that each of f

and g is incident with v and a point of $S - v$. Moreover, each of f and g is in Γ_S .

Proof: Let G_1 and G_2 be components of $G - S$. Since S is a minimal vertex cutset, each point of S has neighbours in G_1 and in G_2 . Let v be any point of S . We consider two cases: *Case 1:* v is incident with the external face f' . For $i=1,2$ let $v_i \in N(v) \cap V(G_i)$. (Here, $N(v)$ is the neighbourset of v). Let P be a shortest path from v_1 to v_2 such that $v \notin V(P)$. Such a path exists since G is 2-connected. Note that $V(P) \cap (S - v) \neq \emptyset$. Now, P together with the edges vv_1 and vv_2 is a circuit. So, we can pick a shortest circuit C through v and containing neighbours of v in both G_1 and G_2 . Then the region bounded by C contains a face h of G such $v \in V(h)$ and h is not f' , and $h \in \Gamma_S$. Furthermore, $V(h) \cap (S - v) \neq \emptyset$, since $h \in \Gamma_S$ and G is 2-connected. Thus take $f = f'$ and $g = h$.

Case 2: v is not incident with the external face. Then there is a shortest circuit C such that v is in the interior of the region bounded by C , and for $i=1,2$ $V(C) \cap V(G_i) \neq \emptyset$. Moreover, C is incident with at least 2 points v_1 and v_2 of $S - v$. It is now easy to construct a shortest circuit C_1 from v to a point of G_1 then through v_1 and a point of G_2 and back to v . A similar shortest circuit C_2 can be constructed so that it passes through v and v_2 . Now the regions bounded by C_1 and C_2 contain faces f and g respectively, where $v \in V(f) \cap V(g)$. Moreover, each of f and g contains a point of $S - v$, since G is 2-connected and $f, g \in \Gamma_S$. □

Remark : In the proof of Lemma 3.3.3 we only needed that the minimal stable cutset S should contain at least 2 points. We did not need the stronger condition that G be 2-connected. We shall use this fact in the proof of the next Lemma.

Lemma 3.3.4: Let G be a plane graph with an auxiliary graph G' . Suppose S is a minimal stable cutset of G such that $|S| \geq 2$. Let H' be the subgraph induced by S in G' . Then H' contains a spanning circuit C such that for each $f \in \Gamma_S$, we have $E(C) \cap f \neq \emptyset$.

Proof: We shall use induction on $|S|$. Consider the case when $|S| = 2$. Let $S = \{x, y\}$. By Lemma 3.3.3 and the remark above, every face in Γ_S contains at least two points from S , and $|\Gamma_S| \geq 2$. Therefore, for each $f \in \Gamma_S$, $xy \in \bar{f}$. Thus in this case, H' consists of two points joined by at least two edges. Hence H' has a spanning circuit C , in this case, such that $E(C) \cap \bar{f} \neq \emptyset$ for each $f \in \Gamma_S$.

Now let $|S| \geq 3$, and $v \in S$. By Lemma 3.3.3 and the remark above, there are two faces $h, g \in \Gamma_S$ such that $v \in V(h) \cap V(g)$; and there are points x and y in $V(h) \cap (S - v)$ and $V(g) \cap (S - v)$, respectively. Let l be the face of $G - v$ which results from h and g when v is deleted. $G - v$ is a plane graph and $S - v$ is a minimal stable cutset of $G - v$. Moreover, $|S - v| \geq 2$. Thus by induction, the subgraph $(H - v)'$ induced by $S - v$ in $(G - v)'$ has a spanning circuit $C - v$ such that $E(C - v) \cap \bar{f} \neq \emptyset$ whenever $f \in \Gamma_{S-v}$. Moreover $xy \in E(C - v) \cap \bar{l}$. Deleting the edge xy from $C - v$ and adjoining the point v to $C - v$ by the edges $vx \in \bar{h}$ and $vy \in \bar{g}$, we get a spanning circuit C of H with the required properties. $\square$

In the next Theorem, we shall use the following concept:

Definition: Let H be a plane graph and H' be its auxiliary graph. A circuit C of H' is H -independent if the vertex set of C is an independent set in H . If in addition, not all edges of C are drawn in the same face of H then C will be said to be proper.

Theorem 3.3.5: A 2-connected plane graph G has a stable cutset if and only if the associated auxiliary graph G' has a G -independent circuit C which is proper.

Proof: Let S be a minimal stable cutset of G and consider the subgraph H' of G' which is induced by the set S . By Lemmas 3.3.4, H' contains a spanning circuit C with edges from each face in Γ_S . Since G is 2-connected then $|\Gamma_S| \geq 2$, and so C is a proper circuit of G' . Moreover, S is a stable set in G , so C is a proper G -independent circuit of G' .

Now let C be a proper G - independent circuit of G' ; and put $S = V(C)$. We shall show that $G - V(C)$ is a disconnected graph.

Since not all edges of C are drawn in the same face of G , there is an edge uv of C and a face f of G such that uv is drawn in f and the second neighbour of u on C is in the exterior of f . (The other neighbour of u on C is v). There are points on the boundary of f which are not points of C , (an example of this are the neighbours of u on the boundary of f). Suppose all the points on the boundary of f which are not on C are in the interior of C . Then all the points on the two paths in G , from u to v along the boundary of f are in the interior of f . But this contradicts the fact that the edge uv is drawn in the face f . A similar contradiction is obtained if we assume that all the points on the boundary of f which are not points of C are in the exterior of C . Hence there are points y and z of G such that y is in the interior of C while z is in the exterior of C . Now every path in G from y to z uses a point of C , and so $G - V(C)$ is a disconnected graph. □

CHAPTER 4

MATCHING AND STABLE CUTSETS IN GRAPHS WITH MAXIMUM DEGREE 3 AND IN QUADRANGULATED GRAPHS

4.1 We observed in Chapter 2 that MATCHING CUT is NP-Complete even if the input is restricted to graphs with maximum degree 4, and STABLE CUT remains NP-Complete for line graphs with maximum degree 5. In this chapter we investigate the solvability of MATCHING and STABLE CUT when the input is restricted to graphs with maximum degree ≤ 3 .

Several of the better known NP-Complete problems remain NP-Complete when the input is restricted to graphs with maximum degree 3. We give examples:

MAX-CUT: This problem was defined in Section 3.1 where we also pointed out that MAX-CUT remains NP-Complete for graphs with maximum degree 3.

CHROMATIC INDEX: Given $G = (V, E)$ and a positive integer k , can the edges of G be coloured with k or fewer colours so that no point is incident with two edges with the same colour? Holyer [15] proved that it is NP-Complete to recognize graphs with chromatic index 3, even if the input is restricted to graphs with maximum degree 3.

CONNECTED LINE SUBGRAPH: Given $G = (V, E)$ and a positive integer k , does there exist a subset E' of E such that $|E'| \geq k$ and $G' = (V, E')$ is a connected line graph? It is shown in [30] that this problem is NP-Complete for graphs with maximum degree 3.

BIPARTITE SUBGRAPH: This problem was defined in Section 3.1. It remains NP-Complete when the input is restricted to graphs with maximum degree 3 [6].

COMPARABILITY SUBGRAPH: Given $G = (V, E)$ and a positive integer k , does there exist a subset E' of E such that $|E'| \geq k$ and $G' = (V, E')$ is a comparability graph? (See [9] for a discussion about comparability graphs). It is shown in [30] that this problem remains NP-Complete for graphs with maximum degree 3.

DOMINATING SET: Given $G = (V, E)$ and a positive integer k , does there exist a subset D of V such that $|D| \leq k$ and every point of G is either in D or is adjacent to a point of D ? This problem is known to be NP-Complete even when the input is restricted to planar graphs with maximum degree 3 [6].

The complexities of MATCHING CUT and STABLE CUT are quite different from those of the problems given above, when the input is restricted to graphs with a small maximum degree. We investigate this claim in detail in Sections 4.2 and 4.3. In Section 4.4, we introduce a condition on the polynomial solvability of MATCHING CUT and STABLE CUT in terms of the size of the longest chordless cycle allowed in the input graph.

4.2 Stable Cutsets in Graphs with Maximum Degree 3 and 4.

We prove in this section that any connected graph with maximum degree ≤ 3 and with more than 6 points must have a stable cutset. We show that this result is in some sense the best possible, by giving a construction for an infinite class of graphs with maximum degree 4 which do not have a stable cutset. In contrast, we prove in the next section that every line graph with maximum degree 4 has a stable cutset, provided that its root graph has a 2-connected subgraph with at least 8 points. We end this section with a sufficient connectivity condition for a graph to have a stable cutset.

Theorem 4.2.1 : Let $G = (V, E)$ be a simple connected graph with maximum degree at most 3 and $|V| \geq 7$. Then G has a stable cutset.

Proof : The proof is a fast algorithm for finding a stable cutset S in G . (It examines at most $|V| + 6$ points and the edges which are incident with them).

Step 1 : Search for a point v of degree 3. If none exists then let S be the neighbourset of any point and halt.

Step 2 : Let the neighbourset of v be $N(v) = \{x_1, x_2, x_3\}$. Let k be the number of edges in $G[N(v)]$. (i) If $k = 0$ then put $S = N(v)$ and halt. (ii) If $k = 1$ then go to Step 3. (iii) If $k = 2$ then let $G_1 \cong G[\{v, x_1, x_2, x_3\}]$. Put $S = \{x_i, x_j\}$ where $i, j \in \{1, 2, 3\}$ and x_i and x_j have degree 2 in G_1 then halt.

Step 3 : Relabeling if necessary, we have $x_1 x_2 \in E$ and x_3 the third neighbour of v ; point x_3 is not joined to either of x_1 or x_2 . Let y_1 and y_2 be neighbours of x_1 and x_2 respectively such that y_1 and y_2 are not in $\{v, x_1, x_2, x_3\}$. If such a point y_1 does not exist then put $S = \{x_2, x_3\}$ and halt. If y_2 does not exist then put $S = \{x_1, x_3\}$ and halt. If $y_1 y_2 \in E$ then go to Step 4. Otherwise put $S = \{y_1, y_2, v\}$ and halt.

Step 4 : If $y_1 x_3 \notin E$ then put $S = \{y_1, x_2, x_3\}$ and halt. If $y_2 x_3 \notin E$ then put $S = \{y_2, x_1, x_3\}$ and halt. □

The correctness of this algorithm can easily be verified by the reader.

We now give a construction to prove that the condition 'maximum degree ≤ 3 ' of the preceding theorem cannot be replaced with 'maximum degree ≤ 4 '.

Let $2n$ be an even integer with $n \geq 3$. Let $G = (V, E)$ where $V = \{1, 2, \dots, 2n\}$. The construction of G starts with two n cycles one inside the other (see Figure 4.2.1). Label the points in the inner circle from 1 up to n . Similarly label the points in the outer cycle from

$n+1$ to $2n$. For $i = 1, 2, \dots, n-1$ join point i to points $i+n$ and $i+n+1$. Join point n to points $2n$ and $n+1$. The cases $n = 4$ and $n = 6$ are illustrated in Figure 4.2.1. (These are just the line graphs of the three dimensional Archimedean polyhedra called anti-prisms).

Figure 4.2.1

It is easy to check that the graphs we have constructed are locally connected. It therefore follows from Theorem 1.4.2 of Chapter 1 that these graphs have no stable cutsets.

Theorem 4.2.2 : *Every critically 2-connected graph on more than 3 points must have a stable cutset.*

Proof : A critically k -connected graph must have a point of degree k (see [14]). Let v be a point of degree 2 in G with neighbours u_1 and u_2 . Let P be any path in $G - v$ connecting u_1 to u_2 . Point u_1 is not adjacent to u_2 otherwise P together with the edges u_1v and u_2v would be a cycle with a chord u_1u_2 ; this is contrary to the fact that a graph is critically 2-connected if and only if no cycle has a chord [24]. Hence v has a stable neighbourhood (which is a stable cutset of G). $\square$

4.3 Matching Cutsets in 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 [2]). In view of this, the following propositions seem to be interesting:

Proposition 4.3.1 *Let G be a 3-regular, 2-edge connected bipartite graph. Then G is 1-factorable.*

Proposition 4.3.2 *Let G be any 3-regular graph which does not have a matching cutset. Then G is 1-factorable.*

Proof : First let us suppose that 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 [2]) 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 . F together with the two perfect matchings of $G - F$ give a 1-factorization of G . This completes the proof of Proposition 4.3.2.

To complete the proof of Proposition 4.3.1, we note that the hypothesis does not forbid the set F of the preceding argument from disconnecting G . However, $G - F$ consists of a collection of cycles and since G is bipartite, every cycle in the collection is even. Hence we can get two disjoint perfect matchings of G by taking perfect matchings of each cycle in the collection. These together with F give a 1-factorization of G . □

The usefulness of Proposition 4.3.2 is unfortunately ruined by the following result:

Theorem 4.3.3 *Let G be a connected 3-regular graph 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 $\delta(C)$ be the set of edges with precisely one endpoint in C . If $|V(G)| \geq 5$ then $\delta(C)$ is a matching

cutset, otherwise we can find a circuit which is shorter than C . Consider the cases that $|V(C)| \leq 4$. Case 1: $|V(C)| = 3$: If $\delta(C)$ is not a matching then let $x \in V(G) - V(C)$ such that $N(x) \cap V(C) = \{z, y\}$. (Here, $N(x)$ is the neighbourset 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 $\delta(C)$ 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 neighbours 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 neighbour of y cannot be on C , by minimality of C). Since $G \neq K_{3,3}$, 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 $\{x'_1t, yy'\}$ is a matching cutset of G . □

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

In [4], Chvátal points out an algorithm for recognizing graphs with a matching cutset when the input is restricted to graphs with maximum degree at most 3. In view of the result in Theorem 4.3.3, the usefulness of the algorithm given by Chvátal is ruined. We should note also that the proof of Theorem 4.3.3 is a good algorithm. The first step of this

algorithm is the same as the first phase of Chvátal's algorithm, and it consists of finding a shortest cycle in the input graph G . One starts with any cycle C . If $|V(C)|=3$ or 4 , then proceed to Case 1 or Case 2, (of the proof of 4.3.3) accordingly. Otherwise search for a point of $V - V(C)$ which has two neighbours on C . If no such point exists, then $\delta(C)$ is a matching cutset of G . If such a point exists, then we can find a shorter cycle T , and the search for a shortest cycle resumes with $C = T$.

In Chapter 3, we introduced the concept of a circuit C in a plane graph such that C does not contain two edges from any face of the graph. Recognizing planar graphs with such a circuit (we called such a circuit a PJS), is equivalent to determining whether the plane dual of the given graph has a matching cutset. We do not know the complexity of this problem at this time. We give now a result guaranteeing the existence of a PJS for a class of planar graphs.

Corollary 4.3.4 : *Every connected maximal plane graph with more than 6 faces has a PJS.*

Proof : The dual G^* of G is a 3-regular plane graph with more than 6 points, and so it has a matching cutset. Thus by Lemma 3.2.1, G has a PJS. $\square$

Corollary 4.3.5 : *For every even $n \geq 24$ such that 3 divides n , there is a 4-regular graph on $n/2$ points which has a stable cutset.*

Proof : Construct first a 3-regular graph G on $|V|$ points where $3|V| = n$. We have $|V| \neq 4$, $|V| \neq 6$. Then by Theorem 4.3.3 G has a matching cutset. By Lemma 2.2.3, the line graph $L(G)$ of G has a stable cutset. The number of points of $L(G)$ is the number of edges of G , which is $3|V|/2$. $\square$

We note that the reasoning used at the beginning of the proof of Theorem 4.3.3 also proves that any 3-regular graph with girth $k \geq 5$ has a matching cutset of size at least k .

4.4 Quadrangulated Graphs

Recall that recognizing connected graphs with a matching cutset is equivalent to recognizing graphs with a *decomposition colouring*. We shall refer to a colouring of some of the points of G by two colours as a *partial decomposition colouring* of G if each vertex v has at most one neighbour with a colour which is different from the colour of v .

Definition : A graph $G = (V, E)$ is *quadrangulated* if G does not have a chordless cycle of length 5 or more.

We see that every triangulated graph is quadrangulated, so *quadrangulated graphs* is a proper generalization of *triangulated graphs*. (Figure 4.4.1 shows a quadrangulated graph which is not triangulated).

Figure 4.4.1

No 2-connected triangulated graph can have a matching cutset, since these graphs are locally connected. (We proved in the last section of Chapter 1 that a locally connected graph cannot have a matching cutset). In contrast, a quadrangulated graph can have both a matching cutset and a stable cutset. An example of such a graph is the 4-cycle.

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 $G = (V, E)$ and an edge e of G , we can determine in polynomial time whether G has a matching cutset which contains the edge e . If $e = uv$ the algorithm that we shall give colours u red and v blue. It then colours a point t red (respectively blue) only if red (respectively blue) is the only colour which may be assigned to t by every partial decomposition colouring of G which colours u red and v blue. If the algorithm finds a point which must be re-coloured, it halts with the message that G does not have a matching cutset which contains the edge e . The algorithm may (correctly) arrive at the conclusion that G has a matching cutset containing e before it colours all the points of G .

Algorithm D

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

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

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

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

Step 1.1 : If there is no point a with neighbours b and c such that $\alpha(a) \neq \alpha(b)$ and $\alpha(b) = \alpha(c) \neq 0$, then go to Step 2.1.

Step 1.2 : If $\alpha(a) \neq 0$, then go to Step 4. Otherwise put $\alpha(a) = \alpha(b)$ and go to Step 1.1

Step 2.1 : If there is no point x with a neighbour y such that $\alpha(y) \neq 0$, $\alpha(x) \neq \alpha(y)$ and y has a neighbour z with $0 \neq \alpha(z) \neq \alpha(y)$ then go to Step 3.1

Step 2.2 : If $\alpha(x) \neq 0$ then go to Step 4. Otherwise put $\alpha(x) = \alpha(y)$ and go to Step 1.1

Step 3.1 : If there is no pair of adjacent points i and j such that i and j have a common neighbour k with $\alpha(k) \neq 0$ and either $\alpha(i) \neq \alpha(k)$ or $\alpha(j) \neq \alpha(k)$, then go to Step 5.

Step 3.2 : If either $\alpha(i) \neq 0$ and $\alpha(i) \neq \alpha(k)$, or $\alpha(j) \neq 0$ and $\alpha(j) \neq \alpha(k)$ then go to Step 4.

Otherwise put $\alpha(i) = \alpha(j) = \alpha(k)$ and go to Step 1.1

Step 4 : G does not have a matching cutset containing e , halt.

Step 5 : 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.

Figure 4.4.2(a) shows an input to Algorithm D, consisting of a graph G and an edge e of G . The partial decomposition colouring which Algorithm D produces when given this input is shown as labels 1 and 2 on the points of G .

Figure 4.4.2(a)-(b)

An example of an input to Algorithm D for which the algorithm fails is shown in Figure 4.4.2(b). The graph G of this example is of course, not quadrangulated.

Lemma 4.4.1 : If Algorithm D halts in Step 5, then the set F produced by the algorithm is a matching.

Proof : If not, let xy and xz be edges of F , where $y \neq z$. By definition of F , neither of $\alpha(x)$, $\alpha(y)$ and $\alpha(z)$ is zero. We can assume $\alpha(x) = 1$. (The case when $\alpha(x) = 2$ is completely similar). We thus have $\alpha(y) = \alpha(z) = 2$. The last 3 steps which control went through before entering Step 5 were Steps 1.1, 2.1 and 3.1, in this order. Using $a = x$,

$b = y$ and $c = z$ in Step 1.1 we see that control should have passed from Step 1.1 to Step 1.2 instead of 2.1. $\square$

Lemma 4.4.2 : *If Algorithm D halts in Step 4, then G does not have a matching cutset F such that $e = uv \in F$.*

Proof : Suppose the algorithm halts in Step 4 and yet G has a 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 $v \in V(G_2)$. Let the vertex labels 0, 1, 2 stand for vertices which are not coloured, vertices coloured red and vertices coloured blue, respectively.

Define $\gamma : \rightarrow \{0, 1, 2\}$ by

$$\gamma(t) = \begin{cases} 0, & \text{if } t \notin V(G_1) \cup V(G_2); \\ 1, & \text{if } t \in V(G_1); \\ 2 & \text{when } t \in V(G_2). \end{cases}$$

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

Suppose the claim is true when the algorithm has coloured $k - 1$ points, and let x be the k^{th} point which is coloured by the algorithm. Suppose x is coloured red. (The case when x is coloured blue is similar). Then we must have one of the situations depicted in Figure 4.4.3(a)-(c), which correspond to x being coloured red in Steps 1.2, 2.2 and 3.2, respectively. By induction, the colours assigned to the points y, z and j by γ are as shown in Figure 4.4.3(a)-(c). Since γ is a decomposition colouring of G , it must then colour x red. Hence the claim is true.

Now, if control passes to Step 4 from Step 1.2, then this happens because the algorithm finds a point a which is coloured blue (respectively red) such that two neighbours b and c of a are coloured red (respectively blue). Now γ agrees with the colouring produced

by the algorithm on the points a , b and c . This contradicts the assumption that γ is a decomposition colouring of G .

A similar contradiction is obtained if control passes to Step 4 from Steps 2.2 or 3.2. $\square$

Figure 4.4.3(a)-(c).

Theorem 4.4.3 : Recognizing decomposable graphs can be done in $O(|V||E|^2)$ steps if the input is restricted to graphs without a chordless circuit of length 5 or more.

Proof : Apply Algorithm D using different choices of e until a matching cutset is found or until all edges have been tried.

If the algorithm halts in Step 4, then by Lemma 4.4.2 no matching cutset containing e exists. Suppose the algorithm halts in Step 5. Then by Lemma 4.4.1, 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 coloured red and blue, respectively. If $G - F$ is connected, then let P be a shortest path from a red point to a blue point such that P contains a point which is not coloured. Let $x \in V(P) \cap V_1$ and $y \in V(P) \cap V_2$. 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 . Note that C is a chordless circuit.

Let z be the non coloured point of C which is adjacent to a red point, and let this red

point be t_1 . By Step 2.1 of Algorithm D, t_1 does not have a blue neighbour. Let t_2 be the neighbour of t_1 on C . Then, t_2 is a red point. Let z_1 be the neighbour of z on C , other than t_1 . Note that z_1 is not a red point. If z_1 is not a coloured point, then C has at least two red points (t_1 and t_2), two non-coloured 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 2.1 of Algorithm D, z_1 does not have a red neighbour. Hence the neighbour of z_1 on C is a blue point. Thus, C has at least 2 blue points, 2 red points and at least one non-coloured 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.

Steps 1 and 2 take $O(|V|^2)$ work each. Step 3 takes $O(|E|)$ work. So each run of the algorithm may require $O(|V|^2|E|)$ work. The algorithm needs to be run at most $|E|$ times and this gives a total of $O(|V|^2|E|^2)$ work. $\square$

Corollary 4.4.4 *STABLE CUT is solvable in polynomial time if the input is restricted to line graphs of quadrangulated graphs.*

Proof : Let G be a line graph with root graph H where H is a quadrangulated graph. Then by Lemma 2.2.3 the graph G has a stable cutset if and only if H has a matching cutset. Determining whether H has a matching cutset can be done in polynomial time by Algorithm D. $\square$

We show now that the class of quadrangulated graphs contains and is related to some well known classes of graphs.

We shall say that a graph $G = (V, E)$ has a *closed neighbourhood* if for every $u \in V$ the following holds:

(a) : For each $x \in N(u)$, there is a $y \in N(u)$ such that $xy \in E$.

(b) : If $x, y \in N(u)$ satisfy $xy \notin E$, then every shortest path P in G from x to y which avoids u must satisfy $V(P) \subseteq N(u)$.

Note that these two conditions are precisely the conditions (a) and (b) given in the proof of Proposition 1.4.1 of Chapter 1. It was shown there that the 2-connected chordal graphs have properties (a) and (b), and so all 2-connected chordal graphs have a closed neighbourhood.

The proof of Theorem 1.4.2 also shows that the class of 2-connected graphs with a closed neighbourhood is a subclass of the class of graphs with a connected neighbourhood.

Proposition 4.4.5 : A 2-connected graph $G = (V, E)$ is triangulated if and only if it is quadrangulated and has a closed neighbourhood.

Proof : Triangulated graphs are obviously quadrangulated, and the proof of Proposition 1.4.1 shows that triangulated graphs have a closed neighbourhood.

Now let G be a quadrangulated graph which has a closed neighbourhood. We show that G is triangulated. G cannot have a chordless cycle of length more than 4 since it is quadrangulated. Suppose G has a chordless 4-cycle $C = (u_1, u_2, u_3, u_4, u_1)$. (See Figure 4.4.5(a)). We have $u_2, u_3 \in N(u_1)$, $u_1 u_3 \notin E(G)$ and $P = \{u_1, u_4, u_3\}$ is a shortest path in G from u_1 to u_3 (it has length 2, and no path from u_1 to u_3 of length 1 exists) and yet $V(P)$ is not contained in $N(u_1)$. This contradicts the fact that G has a closed neighbourhood.

□

Another class of graphs which is closely related to quadrangulated graphs is the class of Meyniel graphs. A *Meyniel graph* is an undirected graph with the property that every odd circuit of length greater than 3 has at least two chords. Meyniel proved that such graphs are perfect (see [9]). Figure 4.4.5(b) shows a quadrangulated graph which is not perfect. (Its complement is a cycle of length 7, which is not perfect, and so by the Perfect Graph

Theorem (see [9]), the graph of Figure 4.4.4(b) cannot be perfect)).

We observe that a quadrangulated graph G is Meyniel if and only if every circuit of G of length 5 has at least 2 chords. Moreover, a Meyniel graph is quadrangulated if and only if it contains no even chordless circuit of length more than 4.

Figure 4.4.4(a)-(b)

CHAPTER 5

MATCHING CUTSETS IN LINE GRAPHS

5.1 Preliminaries

We have proved that both MATCHING CUT and STABLE CUT are NP-Complete even if the input is restricted to perfect graphs and to graphs with maximum degree 6. We did this in Chapter 2. We proved in Chapters 3 and 4 that STABLE CUT is solvable in polynomial time for planar graphs, and both problems are solvable in polynomial time when the input is restricted to graphs without a chordless circuit of length 5 or more. The similarity between the complexities of the two problems was shown to be greater for graphs with a small maximum degree, by our proof that every connected graph with maximum degree ≤ 3 and on more than 6 points has both a matching cutset and a stable cutset. In Chapter 2, we gave a class of graphs for which MATCHING CUT is NP-Complete while STABLE CUT is polynomially solvable. It is natural to wonder at this stage whether there is a class of graphs for which MATCHING CUT is polynomially solvable while STABLE CUT is NP-Complete. We settle this question in this chapter. More specifically, we prove that MATCHING CUT is solvable in polynomial time when the input is restricted to line graphs, whereas STABLE CUT is NP-Complete for line graphs.

A proof of the fact that STABLE CUT is NP-Complete for line graphs was given in Chapter 2. Thus, to complete the proof of the claim above, it is enough for us to now

give a polynomial algorithm to solve MATCHING CUT when the input is restricted to line graphs. We do this in Section 5.2. But, before that, we state a very simple sufficient condition for a line graph not to have a matching cutset.

Proposition 5.1.1 : *Let G be a line graph with a root graph H . Suppose that H has minimum degree at least 3. Then G does not have a matching cutset.*

Proof : Let e be any edge of G . Then there are edges f and g of H such that f and g are incident with the same point v of H and, in G , the edge e is given as $e = fg$. (This follows from the way G is constructed from H). Since $d_H(v) \geq 3$, then v is incident with an edge h where $h \notin \{f, g\}$. Now e is an edge of the triangle of G which is induced by the points f, g and h . Thus, every edge of G lies in a triangle, and so G cannot have a matching cutset. $\square$

The statement of the proposition above does not give a characterization of line graphs with a matching cutset, as can be seen by taking G to be K_3 . Then the root graph of G is either K_3 or $K_{1,3}$. Each of these graphs has minimum degree ≤ 2 , and yet G does not have a matching cutset.

5.2 MATCHING CUT for Line Graphs.

We prove in this section that we can recognize line graphs with a matching cutset in polynomial time. We give an algorithm which, when given a 2-connected 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. If we relax the condition that G is 2-connected, then the complexity increases to $O(|E|^2)$.

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

Observe that a 2-connected graph with 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 non adjacent 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 which is the basis of our algorithm for solving MATCHING CUT when the input is restricted to line graphs.

Lemma 5.2.1 : Suppose that $L(G)$ is a connected line graph 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 $E' = \{ef : e, f \in E \text{ and } e \text{ and } f \text{ have a common endpoint in } G\}$. Let F be a minimal matching cutset of $L(G)$. Put $S = \{x \in V : \text{there are } e \text{ and } f \text{ in } E \text{ such that } x \text{ is incident with both } e \text{ and } f \text{ and } ef \in F\}$. 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 $e, f \in \delta(x)$. Let $g \in \delta(x) - \{e, f\}$. 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 $\{e\} = \delta(x) - \{f\}$ and $\{g\} = \delta(y) - \{f\}$. Suppose that $e = g$. Then we get that x and y induce a component of $L(G)$ with just 2 points, and this is a contradiction. Hence $e \neq g$. Now, ef and fg are distinct edges of $L(G)$ and since $x, y \in S$, we 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 $L(G)$, we will prove 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 degree 2-stable cutset S and show that $L(G)$ has a matching cutset. Define $F = \{ef \in E' : \text{there is } x \in S \text{ with } e, f \in \delta(x)\}$.

Suppose that two edges in F have a common endpoint. Then the two edges are of the form ef and gf where e, f and g are edges of G . Let x and y be points of S such that $e, f \in \delta_G(x)$ and $f, g \in \delta_G(y)$. Now, $g \notin \delta_G(x)$ since $d_G(x) = 2$ and $e, f \in \delta_G(x)$. 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 as follows: Let $ef \in F$. Then there is $x \in S$ such that $e, f \in \delta(x)$. Let u and v be the other endpoints of e and f (different from x). Now, there is no path in $L(G) - F$ from the point e to the point f . $\square$

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

If e is an edge of G , we shall use G/e to denote the graph obtained from G by contracting the edge e .

Lemma 5.2.2 : *Let G be a 2-connected graph which 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: $\Rightarrow$ 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 $S' = (S - \{u, v\}) \cup \{v'\}$. 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 which 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 .

$\Leftarrow$ Let S' be a degree 2 stable cutset of G/e and v' be the point obtained from G by identifying u and v . If $v' \notin S'$ then $S = S'$ is a degree 2 stable cutset of G . If $v' \in S'$, we place u in the component of $(G/e) - S'$ which contains the neighbour of u other than v (in G) to get a degree 2 stable cutset of G . □

Lemma 5.2.3 : Let G be a connected graph which does not have a special 3-set. Suppose 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 which is contained in S , and let $x \in S'$. Since S' is a minimal vertex cutset, then x has neighbours in every component of $G - S'$. So, $G - S'$ has two components G_1 and G_2 . For $i = 1, 2$, let $x_i \in N(x) \cap V(G_i)$. Suppose $d_G(y) = 2$ for each $y \in V(G_1)$. This together with the fact that G_1 is connected implies that there is a point $z \in V(G_1) \cup S'$ such that $\{z, x, x_1\}$ 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 trivial. □

The algorithm that we give below accepts as input only 2-connected line graphs. The restriction of 2-connectedness is sensible since a 2-edge 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 K

Input : A 2-connected line graph $G = (V, E)$.

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 $\delta(\{u, v\})$ 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 $K_{1,3}$ then it is well known that G has a unique line 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 S be the pre-image of S' in H and F be the pre-image of S in G ; stop. If $H' - S'$ is connected then G does not have a matching cutset, stop.

Theorem 5.2.4 : Let $G = (V, E)$ be a line graph. Then : (a) If G is 2-connected we can determine in $O(|E|)$ time whether G has a matching cutset. (b) If G is not 2-connected, we can determine in $O(|E|^2)$ time whether G has a matching cutset.

Proof : If G has a special 2-set then the algorithm stops in Step 1. Otherwise, control proceeds to Step 2, and by Lemma 5.2.1, it is enough to decide whether the graph H which is constructed in Step 2 has a degree 2 stable cutset. Since G does not have a special 2-set then H does not have a special 3-set. Hence, by Lemma 5.2.2, we can contract all edges $e = xy$ where $\{x, y\}$ is a special 2-set in H . Lemmas 5.2.3 and 5.2.1 establish the correctness of Step 4.

We consider now a bound on the time complexity of the algorithm. If G is 2-connected, then Algorithm K will accept G directly as input. Step 1 takes $O(|E|)$ work. An algorithm of Philippe G.H. Lehot [18] can be used to execute Step 2 in $O(|E|)$ work. Step 3 takes

$O(|E|)$ work. Information stored while executing Step 3 can be used to find the set S' of Step 4 in $O(|E|)$ work. Thus, when G is 2-connected, the whole algorithm takes $O(|E|)$ work.

Suppose now that G is not 2-connected. We have commented that it is enough to apply the algorithm to the 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 $O(|E|)$ time by using a depth first search algorithm of J.E.Hopcroft. Algorithm K will then be applied at most once to each resulting subgraph. This gives a complexity of $O(|E|^2)$. $\square$

Corollary 5.2.5: *STABLE CUT is solvable in $O(|E|)$ time if the input is restricted to line graphs of line graphs.*

Proof: Let $G = (V, E)$ be a line graph of a line graph. We can construct the root graph H of G in $O(|E|)$ time using an algorithm of P.Lehot [18]. By Lemma 2.2.4, G has a stable cutset if and only if H has a matching cutset. Since H is a line graph, we can use Algorithm K to determine in $O(|E|)$ time whether H has a matching cutset. $\square$

5.3 Some Open Questions.

We have considered the limits of polynomial solvability of MATCHING CUT and STABLE CUT in terms of restrictions on the types of graphs allowed as input. We proved that if the input is restricted to graphs without a chordless cycle of length 5 or more then MATCHING CUT IS solvable in polynomial time. In contrast, we also proved that every graph without a chordless cycle of length 4 or more can have neither a stable cutset nor a matching cutset. It would be nice to answer the following question:

Are MATCHING CUT and STABLE CUT NP-Complete problems even if the input

is restricted to graphs without a chordless cycle of length 6 or more?

We have shown that MATCHING CUT is solvable in polynomial-time if the input is restricted to quadrangulated graphs. Consequently, MATCHING CUT is solvable in polynomial time for bipartite graphs without a chordless cycle of length 6 or more. STABLE CUT is obviously solvable in polynomial time for bipartite graphs. Hence, we have answered the question posed above in the case of bipartite graphs.

We have shown that MATCHING CUT has a degree jump in its complexity at $(4,6)$. We did not settle the question:

Is there a class of graphs for which the complexity of MATCHING CUT has a subgraph jump?

We were able to prove that STABLE CUT has both a subgraph and a degree jump in its complexity. We did this by restricting the input to line graphs. We showed that STABLE CUT is NP-Complete for line graphs with maximum degree 6, but every connected line graph with maximum degree ≤ 4 has a stable cutset provided that its root graph has at least 12 points. We did not settle the following question:

Is STABLE CUT solvable in polynomial time for all graphs with maximum degree ≤ 5 ?

If STABLE CUT is NP-Complete for graphs with maximum degree 4, then it will follow that STABLE CUT has a degree jump at $(4,6)$. The validity of this claim follows from our proof that every connected graph with maximum degree ≤ 3 must have a stable cutset, provided it has at least 7 points.

A proof of the fact that MATCHING CUT is NP-Complete for bipartite graphs was given in Chapter 2. We in fact showed that MATCHING CUT is NP-Complete even if the input is restricted to graphs such that every edge joins a point of degree 8 or 6 to a point of

degree 2. We have not settled the following question:

What is the degree bound on the polynomial solvability of MATCHING CUT for bipartite graphs?

In Chapter 4, we introduced the class of quadrangulated graphs. We proved that this class properly contains the class of triangulated graphs. R.E.Tarjan and M.Yannakakis [26] have given a linear time algorithm for recognizing triangulated graphs. We ask:

Can quadrangulated graphs be recognized in linear time?

In Chapter 1, we gave a rather general sufficient condition for a graph to have neither a matching nor a stable cutset. The condition was that G be locally connected. We showed that some large, well known classes of graphs were locally connected, and so the condition was in a sense rather general. In Chapter 4, we gave a sufficient connectivity condition for a graph to have a stable cutset. The condition was simply that the graph be critically 2-connected. We did not give a similar sufficient connectivity condition for a graph to have a matching cutset. We ask:

Is there a simple but significantly general connectivity condition for a graph to have a matching cutset?

What we mean by *significantly general* in the last question is that the condition should apply to some large classes of graphs.

Perhaps the most interesting question about MATCHING CUT and STABLE CUT which we have not answered is:

Are MATCHING CUT and STABLE CUT NP-Complete even if the input is restricted to planar graphs?

REFERENCES

1. C.Berge, *Graphs and Hypergraphs*, North Holland American Elsevier, 1973
2. J.A.Bondy and U.S.R.Murty, *Graph Theory with Applications*, North Holland, N.Y. 1976
3. R.L.Brooks, *On Colouring the Nodes of a Network*, Proc. Camb. Phil. Soc. 37 (1941) 194-197.
4. V.Chvátal, *Recognizing Decomposable Graphs*, J. Graph Th. 8 (1984) 51-53
5. S.A.Cook, *The Complexity of Theorem Proving Procedures*, Proc. 3rd ACM Sympo. on Theory of Computing, ACM (1971) 151-158
6. M.R.Garey and D.S.Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.
7. M.R.Garey, D.S.Johnson and R.E.Tarjan, *The Planar Hamiltonian Problem is NP-Complete*, S.I.A.M Journ. Comput. 5 (1976) 704-714
8. M.R.Garey, D.S.Johnson and L.J.Stockmeyer, *Some Simplified NP-Complete Graph Problems*, Theor. Comp. Sc. 1 (1976) 237-267
9. M.C.Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, 1980.
10. S.Goodman and S. Hedetniemi, *Sufficient Conditions for a Graph to be Hamiltonian*, J.Comb. Theor. Series B, 16 (1974) 175-180,
11. R.L.Graham, *On Primitive Graphs and Optimal Vertex Assignments*, Ann. N.Y. Acad. Sci. 175 (1970) 170-186.
12. M.Grötschel, L.Lovász, and A.Schrijver, *The Ellipsoid Method and Its Consequences*

- in *Combinatorial Optimization*, *Combinatorica* 1 (1981) 169-197.
13. F.Hadlock, *Finding a Maximum Cut of a Planar Graph in Polynomial Time*, S.I.A.M. Journ. Comput. 4 (1975) 221-225.
 14. R.Halin, *A Theorem on n-Connected Graphs*, J. Comb. Theor. 7 (1969) 150-154.
 15. I.Holyer, *The NP-Completeness of Edge Colouring*, S.I.A.M Journ. Comput. 12 (1981) 718-720.
 16. J.E.Hopcroft and R.E.Tarjan, *Efficient Planarity Testing*, J. Assoc. Comput. Mach. 21 (1974) 549-468.
 17. R.M.Karp, *Reducibility Among Combinatorial Problems*, in *Complexity of Computer Computations*, R.E.Miller and J.W.Thatcher, eds., Plenum Press, (1972) 85-108.
 18. Philippe G.Lehot, *An Optimal Algorithm to Detect a Line Graph and Output Its Root Graph*, J. Assoc. Comput. Mach. 21 No 4 (1974) 569-575.
 19. L.Lovász, *Coverings and Colourings of Hypergraphs*, in Proc., Fourth South Eastern Conf. on Comb., Graph Theor. and Comput., Utilitas Mathematical, Winnipeg (1973) 3-12.
 20. W.McCuag, *A Simple Proof of Menger's Theorem*, J. Gr. Theor. 8 (1984) 427-429.
 21. G.I.Orlova and Y.G.Dorfmann, *Finding the Maximum Cut in a Graph*, Eng. Cybernetics 10 (1972) 5.2-506.
 22. C.H.Papadimitriou and M.Yanakakis, *The Clique Problem for Planar Graphs*, Info. Proc. Letts. 13 (1981) 131-133.
 23. J.Plesnik, *A Note on the Complexity of Finding Regular Subgraphs*, Discr. Math. 49 (1984)161-167.
 24. M.D.Plummer, *On Minimal Blocks*, Trans. Amer. Math. Soc. 134 (1968) 85-94
 25. J.Sedlacek, *Some Properties of Interchange Graphs*, in *The Theory of Graphs and Its*

- Applications*, Academic Press, N.Y. (1963) 145-450.
26. R.E.Tarjan and M.Yanakakis, *Simple Linear-time Algorithms to Test Chordality of Graphs, Test Acyclicity of Hypergraphs and Selectively Reduce Acyclicity of Hypergraphs*, S.I.A.M Journ. Comput. 13 (1984) 566-579.
 27. L.G.Yaliant, *The Complexity of Computing the Permanent*, Theor. Comput. Sc. 8 (1979) 189-201.
 28. A.C.M. van Rooij and H.S.Wilf, *The Interchange Graph of a Finite Graph*, Acta Math. Acad. Sci. Hungar. 16 (1974) 263-269.
 29. S.H.Whitesides, *An Algorithm for Finding Clique Cut-sets*, Info. Proc. Letts. 12 (1981) 31-32.
 30. M.Yanakakis, *Edge and Node Deletion NP-Complete Problems*, Proc. Tenth Annual ACM Symp. on Theory of Computing, Assoc. for Comput. Mach., N.Y. (1978) 253-264.

END

20.06.88

FIN