# On graph-transverse matching problems

by

Ross William Churchley
B.Sc., University of Victoria, 2010

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Mathematics and Statistics

On graph-transverse matching problems

by

Ross William Churchley
B.Sc., University of Victoria, 2010

Supervisory Committee

Dr. Jing Huang, Supervisor
(Department of Mathematics and Statistics)

Dr. Kieka Mynhardt, Departmental Member
(Department of Mathematics and Statistics)

**Supervisory Committee**

Dr. Jing Huang, Supervisor
(Department of Mathematics and Statistics)

Dr. Kieka Mynhardt, Departmental Member
(Department of Mathematics and Statistics)

## ABSTRACT

Given graphs $G, H$, is it possible to find a matching which, when deleted from $G$, destroys all copies of $H$? The answer is obvious for some inputs—notably, when $G$ is a large complete graph the answer is "no"—but in general this can be a very difficult question. In this thesis, we study this decision problem when $H$ is a fixed tree or cycle; our aim is to identify those $H$ for which it can be solved efficiently.

The *$H$-transverse matching problem*, TM($H$) for short, asks whether an input graph admits a matching $M$ such that no subgraph of $G - M$ is isomorphic to $H$. The main goal of this thesis is the following dichotomy. When $H$ is a triangle or one of a few small-diameter trees, there is a polynomial-time algorithm to find an $H$-transverse matching if one exists. However, TM($H$) is NP-complete when $H$ is any longer cycle or a tree of diameter $\geq 4$. In addition, we study the restriction of these problems to structured graph classes.

# Contents

# List of Figures

## ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr. Jing Huang, for introducing me to the world of transversal problems. I am grateful for the guidance, patience, support, and advice he has given me as an undergraduate and as his master's student.

Many other professors and graduate students at the University of Victoria made my time as a master's student productive, rewarding, and enjoyable. I would like to recognize the Department of Mathematics and Statistics for providing a great environment for my studies.

This thesis would not have been possible without the encouragement and support from my family: my parents, my brother, and especially my fiancée, Leah Pile. Thanks to you all.

Finally, I would like to acknowledge the financial support given to me by the University of Victoria and by NSERC through the Alexander Graham Bell Canadian Graduate Scholarship program.

*Solutions to problems are easy to find:*
*the problem's a great contribution.*
*What is truly an art is to wring from your mind*
*a problem to fit a solution.*
Piet Hein

# DEDICATION

To the memories of

Dr. Jim Totten
(1947–2008)

and

Dr. Robb Fry
(1961–2011).

# Chapter 1

# Introduction

One of the best-known problems of graph theory is the *perfect matching problem*, which asks whether a given graph has a matching which covers every vertex. Many other interesting problems have a similar form: for a fixed collection of structures in a graph, can we find a matching that covers each structure? Although the perfect matching problem has an efficient solution [8], some of these general transversal problems can be difficult to solve. This thesis studies problems of this type; our goal is to develop algorithms to solve them, and to identify those which are unlikely to admit efficient solutions.

Fix a graph $H$. A matching $M$ is called *$H$-transverse* in a graph $G$ if $G - M$ does not contain $H$ as a subgraph. The *$H$-transverse matching problem*, written $\mathrm{TM}(H)$ for short, asks whether an input graph admits such a matching. In this thesis, we focus on the $H$-transverse matching problems where $H$ is a tree or a cycle. Our results make progress towards a general dichotomy for graph-transverse matching problems: they are easy when $H$ is "small", and NP-complete otherwise.

Given this apparent dichotomy, it is natural to ask how assumptions on the input graph affect the computational complexity of these transversal problems. We therefore consider the restriction of $\mathrm{TM}(H)$ to some structured graph classes, including bipartite graphs, triangle-free graphs, planar graphs, and chordal bipartite graphs.

## 1.1 Motivation

$H$-transverse matching problems come up in a variety of contexts. One application is to *tatami tilings*, which use $1 \times 1$ and $1 \times 2$ rectangular tiles such that no four meet at the same point. These were named by Kotani in [23] after the patterns in which Japanese rice mats (or *tatami*) are traditionally arranged. Tatami tilings have been the subject of several papers since Knuth mentioned them in [22]. For instance, Erickson et al. enumerated the tatami tilings of a given rectangular region [14].

As Frank Ruskey pointed out to us [27], a tatami tiling can be viewed as a $C_4$-transverse matching. Consider a grid made from $1 \times 1$ square tiles. We can associate this configuration with a grid graph (the planar dual) whose vertices correspond to the square tiles. If a new tiling (using $1 \times 1$ and $1 \times 2$ tiles) is obtained from the grid by merging adjacent pairs of square tiles, the merged pairs correspond to a matching in the planar dual. For example, the tiling in Figure 1.1 (left) was obtained from a $3 \times 3$ grid of square tiles by merging the pairs represented by the matching in the planar dual (right).

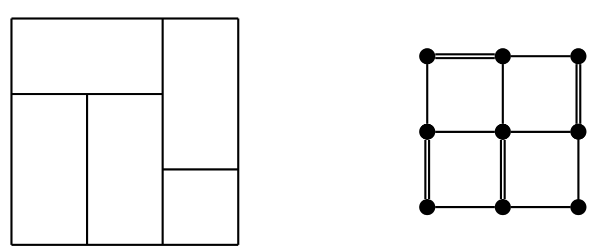


Figure 1.1: A tatami tiling and its corresponding $C_4$-transverse matching

It turns out that the tatami condition—that no four tiles meet at a vertex—coincides with the restriction that the corresponding matching is $C_4$-transverse. Thus graph-transverse matchings (particularly those of planar graphs) generalize tatami tilings.

Graph-transverse matchings can in turn be generalized to more general vertex partitions which have attracted recent attention. A *stable $F$-transversal* of a graph $G$

is a set of vertices $S$, no two of which are adjacent, such that $G - S$ does not contain $F$ as an induced subgraph. For example, a stable $P_2$-transversal is a stable set whose deletion leaves another stable set; i.e., it is one of the partite sets in a bipartition of $G$. Similarly, the deletion of a stable $\overline{P_2}$-transversal leaves a clique; a graph whose vertices can be divided into a stable set and a clique in this way is called a *split graph* [16].

An $H$-transverse matching in a graph $G$ corresponds to a stable $L(H)$-transversal in the line graph $L(G)$. Therefore, the study of graph-transverse matchings can be viewed as the study of stable graph-transversals in line graphs. A number of papers have studied stable $P_3$-transversals in structured graph classes [4,10,11,13,24] including line graphs [3,12,21]. In particular, it is shown in [3] that the $P_4$-transverse matching problem can be solved in polynomial time. However, detecting stable graph-transversals is generally NP-complete [15].

Graph transverse matchings have also been studied in the context of graph Ramsey theory. Consider the more general problem of whether the edges of an input graph $G$ can be coloured red and blue such that there is no red $H_1$ or blue $H_2$. (When $H_1 = P_3$, the red edges form an $H_2$-transverse matching.) Ramsey's theorem, a classic result of graph theory, states that only finitely many complete graphs $G$ admit such an edge colouring.

The complexity of the general Ramsey problem—where $G$ is unrestricted—has been studied by a few researchers. In particular, Burr showed in [2] that it is NP-complete when $H_1$ and $H_2$ are fixed 3-connected graphs. A more recent paper by Schaefer [28] studied the computational complexity of the problem when $H_1$ and $H_2$ are not fixed.

## 1.2 Terminology

In general, we follow the standard terminology in [30]. A *graph* $G$ consists of a vertex set $V(G)$ and an edge set $E(G)$ in turn consisting of unordered pairs of distinct vertices. In other words, we do not consider graphs with loops or parallel edges. A *matching* is a set of edges, no two of which share a vertex. If we say that a graph $G$ *contains* $H$, we mean that $H$ is a (not necessarily induced) subgraph of $G$.

An *$H$-transverse matching of* $G$ is a matching $M$ such that $G-M$ does not contain $H$. The *$H$-transverse matching problem*, written TM($H$) for short, asks whether an input graph admits such a matching. We also consider the stronger problem of deciding whether a graph admits an $H$-transverse matching containing a given set of edges. This is called the *$H$-transverse matching extension problem*, or TMEXT($H$).

We use $P_k$ and $C_\ell$ to denote the *path* and *cycle* on $k$ and $\ell$ vertices, respectively. A *clique* $K_n$ consists of $n$ vertices, each adjacent with every other; a *biclique* $K_{k,\ell}$ consists of $k+\ell$ vertices, $k$ of which are adjacent to each of the remaining $\ell$ vertices.

A *pendant vertex* is a vertex of degree one. The unique edge incident with a pendant vertex is called a *pendant edge*. A vertex which is adjacent to a pendant vertex is called a *support vertex*. The *diameter* of a graph is the longest length among shortest paths between pairs of vertices; in particular, the diameter of a tree is just the length of a longest path.

Figure 1.2: The $\pi_{4,3}$ and the $Y$ graph

A *tree* is a connected graph which does not contain a cycle. We find it convenient to introduce the following notation for small trees: the $\pi_{k,\ell}$ is the diameter-3 tree with a *center edge* $uv$ whose vertices are of degree $k$ and $\ell$, respectively. The $Y$ *graph* refers to the special case $\pi_{3,2}$.

Several structured graph classes play an important role in this thesis. A graph is *bipartite* if it contains no cycles of odd length; equivalently, its vertices can be partitioned into two sets so that no edge has both endvertices in the same set. A *triangle-free* graph is one which does not contain the triangle $C_3$. A graph is *planar* if it can be drawn in the plane without crossing edges.

A *chordal bipartite* graph is a bipartite graph which contains no induced cycles of length longer than four. These graphs are the bipartite analogues of *chordal graphs* (which contain no induced cycles of length longer than three) and are related to perfect elimination schemes of non-symmetric matrices. They enjoy many useful properties, surveyed in [19].

A problem is *NP-complete* if it is possible to reduce the following problem to it in polynomial time [7].

**Problem.** BOOLEAN SATISFIABILITY

> *Instance:* A propositional formula $\xi$ consisting of clauses $c_1, c_2, \ldots, c_q$ over the variable set $a_1, a_2, \ldots, a_p, \overline{a_1}, \overline{a_2}, \ldots, \overline{a_p}$.
>
> *Question:* is there a function $f : \{a_1, a_2, \ldots, a_p, \overline{a_1}, \overline{a_2}, \ldots, \overline{a_p}\} \to \{0, 1\}$ which sends at least one variable from each clause to 1, and so that $f(a_i) \neq f(\overline{a_i})$?

The *incidence graph* of the propositional formula $\xi$ is a bipartite graph with vertex set $\{a_1, \ldots, a_p\} \cup \{c_1, \ldots, c_q\}$ and edges between $a_i$ and $c_j$ whenever $a_i$ or $\overline{a_i}$ appears in $c_j$. There are many NP-complete variants of BOOLEAN SATISFIABILITY which can be obtained by restricting the form of the incidence graph of the input formula. For example, SUBCUBIC SAT [17], PLANAR SAT [25], and SUBCUBIC PLANAR SAT [26]—in which the input's incidence graph is restricted (respectively) to have maximum degree three, to be planar, or both—are all NP-complete.

The NP-completeness proofs in this thesis typically reduce from variants of SUBCUBIC SAT. The subcubicness of the incidence graph means that each clause contains at most three variables and each variable appears in at most three clauses. In fact, a standard reduction [26] allows us to further assume that each variable appears in two clauses positively (as $a_i$) and in one clause negated (as $\overline{a_i}$).

## 1.3 Outline

The structure of this thesis is as follows. First, in Chapter 2, we consider the $H$-transverse matchings where $H$ is a cycle. Churchley, Huang, and Zhu showed in [5] that $\mathrm{TM}(C_\ell)$ is solvable in polynomial time when $\ell = 3$ and is NP-complete otherwise. For completeness, we reproduce their algorithm for the $C_3$-transverse matching problem. Then, we refine their NP-completeness results: we reduce SUBCUBIC PLANAR SAT to $\mathrm{TM}(C_\ell)$ for each $\ell \geq 4$, showing that these problems remain NP-complete when the input is restricted to planar graphs. In fact, the proof shows that $\mathrm{TM}(C_\ell)$ is still NP-complete for planar triangle-free graphs when $\ell$ is odd and for planar bipartite graphs when $\ell$ is even.

These results are an interesting contrast to those of Chapter 3, where we show that $\mathrm{TM}(C_4)$ can be solved in polynomial time among chordal bipartite graphs. The algorithm relies on an important new separator theorem: every separable chordal bipartite graph has a biclique which divides the graph roughly in half. This structure allows us to use a "nondeterministic divide-and-conquer" strategy to solve $\mathrm{TMEXT}(C_4)$ for chordal bipartite graphs in (deterministic) polynomial time.

In Chapter 4 we move on to tree-transverse matching problems and find efficient solutions only for a few "small" trees. We first review a known algorithm for star-transverse matching problems. Next, we provide a polynomial-time solution—based on a reduction to the perfect matching problem—to TM($H$) when $H$ is a diameter-3 tree and the input graph is triangle-free. In two cases, we can remove the restriction that the input graph is triangle-free: we give a new algorithm for TM($P_4$) and a more complicated solution for TM($Y$). Finally, when $H$ is a tree of diameter 4 or greater, we show that TM($H$) is NP-complete.

We conclude this thesis in Chapter 5 with a discussion about future research directions, and suggest some ways our results could be strengthened.

# Chapter 2

# Cycle-transverse matchings

In [5], Churchley, Huang, and Zhu gave the following dichotomy for cycle-transverse matching problems:

**Theorem 2.1** (Churchley, Huang, and Zhu [5]). *The $C_\ell$-transverse matching problem is solvable in polynomial time when $\ell = 3$, and is NP-complete when $\ell \geq 4$.*

This chapter refines their results by showing $\mathrm{TM}(C_\ell)$ is NP-complete even when the input is restricted to be a planar triangle-free graph (Theorem 2.14). For completeness, we first reproduce their algorithm for the $C_3$-transverse matching problem.

## 2.1 The polynomial case: $H = C_3$

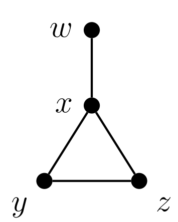



Figure 2.1: The paw

Consider the graph—sometimes called a *paw*—consisting of a triangle $xyz$ and an edge $wx$ (see Figure 2.1). It is easy to see that any $C_3$-transverse matching containing $wx$ also contains $yz$. We call an edge set $M$ *pawssible* if it has this property: for every paw in the graph, $xw \in M$ implies $yz \in M$ (with vertices labelled as above). The above can be restated as:

**Proposition 2.2.** *Every $C_3$-transverse matching is pawssible.* □

The converse is not true; a pawssible matching need not be $C_3$-transverse. The union of pawssible sets is pawssible, although it might not be a matching. We use $F_e$ to denote the minimal pawssible edge set containing the edge $e$; it can be constructed by repeatedly adding edges which are "forced by" paws.

**Lemma 2.3.** *Suppose that $M$ is a pawssible matching, $e$ is an edge in $G$, and let $F_e$ be the smallest pawssible edge set containing $e$. If both $M \cup \{e\}$ and $F_e$ are matchings, then $M \cup F_e$ is a pawssible matching.*

**Proof:** We claim that $M \cup F_e$ is a matching. Indeed, suppose that a paw consists of the triangle $xyz$ and edge $xw$, and that $M \cup \{xw\}$ is a matching. Since $M$ is pawssible, $M$ does not contain any edge external to $xyz$ and incident with $y$ or $z$; otherwise, $M$ would contain $xz$ or $xy$ and $M \cup \{xw\}$ would not be a matching. Hence $M \cup \{xw, yz\}$ is also a matching.

Repeated application of this argument (starting with the assumption that $M \cup \{e\}$ is a matching) shows that no edge of $M$ is adjacent to any edge of $F_e$. Since $M$ and $F_e$ are both matchings, this implies that $M \cup F_e$ is a matching. □

**Theorem 2.4.** *If $G$ admits a $C_3$-transverse matching, then every maximal pawssible matching is $C_3$-transverse.*

**Proof:** Let $M$ be a maximal pawssible matching. Suppose to the contrary that $G - M$ contains a triangle $xyz$. Without loss of generality, $xy$ is contained in some $C_3$-transverse matching of $G$. Any such matching contains $F_{xy}$, so $F_{xy}$ is a matching. Since $M$ is pawssible, it does not contain any edge incident with $x$, $y$, or $z$; in particular, $M \cup \{xy\}$ is a matching. Lemma 2.3 says that $M \cup F_{xy}$ should be a pawssible matching, contradicting the maximality of $M$. Hence $G - M$ does not contain a triangle and $M$ is $C_3$-transverse. □

In fact, the above proof shows that every maximal pawssible matching breaks the same triangles: a triangle $xyz$ is left uncovered if and only if $F_{xy}$, $F_{xz}$, $F_{yz}$ are not matchings.

**Theorem 2.5.** *TM($C_3$) can be solved in polynomial time.*

**Proof:** Compute $F_e$ for each edge $e$. This can be done by letting $F_e = \{e\}$ initially, and then adding edges one by one which are forced by paws. If there is a triangle $xyz$ for which $F_{xy}$, $F_{yz}$, $F_{xz}$ are not matchings, then no $C_3$-transverse matching exists. Otherwise, any maximal pawssible matching is $C_3$-transverse. □

It follows from Theorem 2.4 that an edge set can be extended to a $C_3$-transverse matching if and only if it can be extended to a pawssible matching and $G$ admits at least one $C_3$-transverse matching. This can be easily checked by constructing the union of the appropriate $F_e$ before applying the above algorithm.

**Theorem 2.6.** *TMEXT($C_3$) can be solved in polynomial time.* □

## 2.2 NP-complete cases: $H = C_\ell$, $\ell \geq 4$

### 2.2.1 $H = C_4$

It is helpful to establish some properties of $C_4$-transverse matchings before we show that TM($C_4$) is NP-complete. Consider the $P$ graph in Figure 2.2: any $C_4$-transverse matching containing $vw$ clearly contains one of $xy, yz$. This observation is analogous to the one we made about the $C_3$-transverse matchings of paws in the previous subsection.

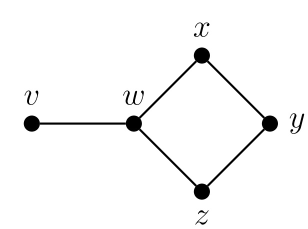



Figure 2.2: The $P$ graph

We can extend the above property to a structure we call a *chain of $C_4$'s* (see Figure 2.3). This structure is a sequence of edge-disjoint $C_4$'s such that each edge shares a vertex with the cycle preceding or succeeding its own.

**Lemma 2.7.** *If $e, e'$ are joined by a chain of $C_4$'s as in Figure 2.3, then no $C_4$-transverse matching contains both $e, e'$.* □

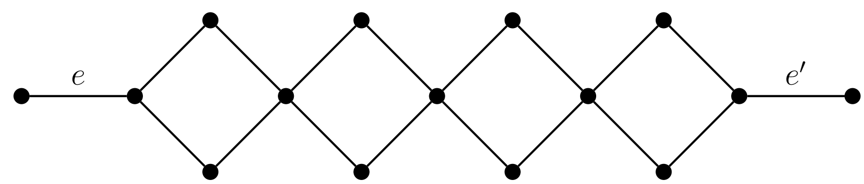



Figure 2.3: A chain of $C_4$'s.

The biclique $K_{2,3}$ is another important small example. For convenience, we call the two degree-3 vertices the *tips* of a $K_{2,3}$. Any $C_4$-transverse matching contains exactly two edges from a $K_{2,3}$; in particular, it contains an edge incident with each tip. This has the following implication on edges surrounding a $K_{2,3}$.

**Lemma 2.8.** *If $e$ is an edge external to a $K_{2,3}$ but incident with one of the tips (as in Figure 2.4, left), then it is not contained in any $C_4$-transverse matching.* □

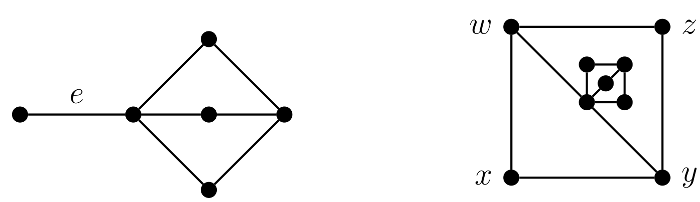



Figure 2.4: Edges forced and forbidden by $K_{2,3}$'s

This observation is useful in cataloguing the $C_4$-transverse matchings of more complicated structures. For instance, Figure 2.4 (right) depicts an intersection of two $K_{2,3}$'s which plays an important role in our proof.

**Lemma 2.9.** *A $C_4$-transverse matching of the graph shown in Figure 2.4 (right) contains two edges from the outer cycle $wxyz$.* □

We are now ready to describe our reduction. Recall that Subcubic Planar SAT is the NP-complete problem which asks whether there is a truth assignment satisfying a given planar propositional formula with at most three variables per clause and where each variable appears in three clauses (twice positive and once negated). Consider an instance $\xi$ of this problem, with variables $a_1, \ldots, a_p$, $\overline{a_1}, \ldots, \overline{a_p}$ and clauses $c_1, \ldots, c_q$. We construct a graph $G$ which admits a $C_4$-transverse matching if and only if $\xi$ is satisfiable.

The graph $G$ is obtained from the incidence graph of $\xi$ as follows. Each vertex representing a variable of $\xi$ is replaced with a planar bipartite graph called a *variable gadget*, and each vertex representing a clause is replaced with a *clause gadget*. The variable gadgets each have three distinguished vertices $x, x', \bar{x}$ corresponding to the clauses the variable appears in; similarly, the clause gadgets have vertices $y_1, y_2, y_3$ corresponding to variables. Both types of gadgets have a planar embedding in which the distinguished vertices appear in the outer face.

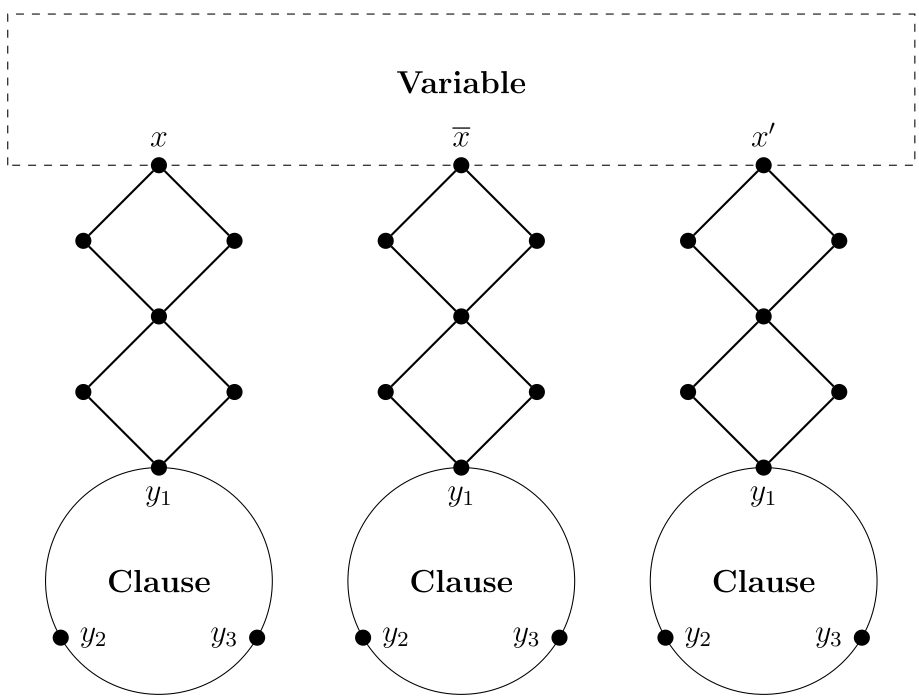



Figure 2.5: Connection of variable and clause gadgets

The edges of the incidence graph are replaced by chains of $C_4$'s between distinguished vertices as in Figure 2.5. A chain of $C_4$'s connects $\bar{x}$ to one of the distinguished vertices in the gadget corresponding to the unique clause where the variable appears negated. The vertices $x, x'$ are connected to the gadgets representing the clauses containing the two positive instances of the variable.

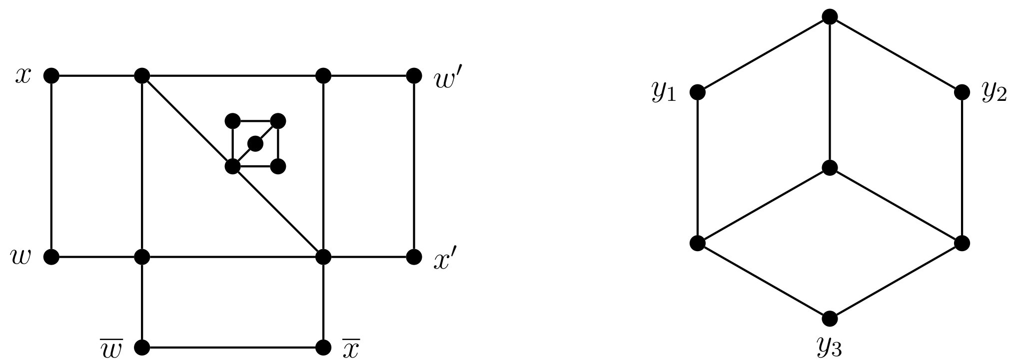



Figure 2.6: Variable and clause gadgets

The variable gadget is shown in Figure 2.6 (left). Incident with its distinguished vertices $x, x', \bar{x}$, it has three special edges $wx, w'x', \overline{w}\overline{x}$ from which we read the truth value of the corresponding variables $a_i, \overline{a_i}$. As we show later, any $C_4$-transverse matching of this gadget contains $\overline{w}\overline{x}$ or both of $wx, w'x'$—the former case corresponds to a truth assignment in which $\overline{a_i}$ is false (and $a_i$ is true); in the latter case, $a_i$ is false.

The clause gadget is illustrated in Figure 2.6 (right). It consists of three $C_4$'s pairwise intersecting in an edge. Any $C_4$-transverse matching of this gadget covers at least one of the distinguished vertices $y_1, y_2, y_3$; we will show that this condition forces at least one of the clause's variables to be true.

We should make note of an important subtlety here: the above gadget represents clauses of size three, but $\xi$ may have smaller clauses. Therefore, we construct (simple) gadgets to represent smaller clauses. A clause of size two corresponds to a $C_4$ whose distinguished vertices $y_1, y_2$ are nonadjacent; a $C_4$-transverse matching clearly covers one of $y_1, y_2$ (again, this property forces one of the clause's two variables to be true).

For a clause of size one, we use a $K_{2,3}$ whose distinguished vertex $y_1$ is one of the tips; we have already observed that a $C_4$-transverse matching covers this vertex.

**Claim 1.** *$G$ is a planar bipartite graph.*

**Proof:** We have constructed $G$ from the planar incidence graph of $\xi$ by replacing vertices and edges by planar subgraphs, so that any two intersect in a vertex in the outer face of both. Clearly, this graph is planar.

The fact that $G$ is bipartite can be inferred from the observation that a path between any pair of $x, x', \bar{x}$ or $y_1, y_2, y_3$ has even length. □

**Claim 2.** *Any $C_4$-transverse matching of the variable gadget contains either $\overline{w}\bar{x}$ or both of $wx, w'x'$. There are $C_4$-transverse matchings containing $\overline{w}\bar{x}$ but not $wx, w'x'$, and vice versa.*

**Proof:** The central feature of the variable gadget is two $K_{2,3}$'s which intersect in a vertex. We have already observed in Lemma 2.9 that any $C_4$-transverse matching $M$ of this structure contains two edges from the outer $C_4$. This cycle—call it $rstu$—shares an edge of three other cycles $wxrs$, $tuw'x'$, and $st\bar{x}\overline{w}$. If $rs, tu \in M$, then $\bar{x}t, st, s\overline{w} \notin M$ and hence $\bar{x}\overline{w} \in M$. A similar argument shows $wx, w'x' \in M$ if $ru, st \in M$.

Conversely, the matchings $\{wx, w'x', ru, st\}$ and $\{\overline{w}\bar{x}, rs, tu\}$ can both be extended to $C_4$-transverse matchings by adding two edges from the center $K_{2,3}$. □

**Claim 3.** *Any $C_4$-transverse matching of the clause gadget covers at least one of $y_1, y_2, y_3$. There are $C_4$-transverse matchings covering each nonempty subset of $y_1, y_2, y_3$.*

**Proof:** We prove the statement for clauses of size three. Any matching $M$ contains at most one edge incident with the middle vertex. If $M$ fails to cover $y_1, y_2, y_3$, then the remaining edges incident with the middle vertex form an (unbroken) $C_4$ with two edges of the outer cycle.

Conversely, let $y_1z_1y_2z_2y_3z_3$ be the outer $C_6$ and $o$ the middle vertex of the clause gadget. The following matchings break all three $C_4$'s in the gadget:

$$\{y_1z_1, oz_2\} \qquad \{y_2z_2, oz_3\} \qquad \{y_3z_3, oz_1\}$$

$$\{y_1z_1, y_3z_3, oz_2\} \quad \{y_1z_1, y_2z_2, oz_3\} \quad \{y_2z_2, y_3z_3, oz_1\} \quad \{y_1z_1, y_2z_2, y_3z_3\}.$$

Similar statements hold for gadgets representing clauses of size one and two. □

**Claim 4.** *Suppose the distinguished vertex $x$ of a variable gadget is connected to a distinguished vertex $y_1$ of a clause gadget. If a $C_4$-transverse matching of $G$ contains a clause gadget edge incident with $y_1$, then it does not contain $wx$.*

**Proof:** The distinguished edge $wx$ in the variable gadget is joined by a chain of $C_4$'s to each clause gadget edge incident with $y_1$. We observed in Lemma 2.7 that a $C_4$-transverse matching does not contain a such a pair of edges. (Similar statements hold when $wx$ is replaced with $w'x'$ or $\overline{w}\bar{x}$ and when $y_1$ is replaced with $y_2$ or $y_3$.) □

Now, we show that $G$ has a $C_4$-transverse matching if and only if the propositional formula $\xi$ is satisfiable. Suppose first that $M$ is a $C_4$-transverse matching of $G$. Let $w_i$, $x_i, \ldots,$ denote the copies of $w$, $x, \ldots$ in the gadget corresponding to the variables $a_i, \overline{a_i}$. We define a truth assignment $f$ for $\xi$ by taking

$$f(a_i) = \begin{cases} 1 & \overline{w}_i\bar{x}_i \in M \\ 0 & \text{otherwise} \end{cases}$$

and letting $f(\overline{a_i}) \neq f(a_i)$. That is, $f(\overline{a_i}) = 1$ when $f(a_i) = 0$ and vice versa.

Consider a clause $c_j$ of $\xi$. Claim 3 tells us that $M$ covers at least one of $y_1, y_2, y_3$

by an edge from the corresponding clause gadget. Suppose first that this vertex is connected to the copy of $x$ in the gadget representing the variables $a_i, \overline{a_i}$. Claim 4 implies that $w_i x_i \notin M$ and hence by Claim 2, $\overline{w}_i \bar{x}_i \in M$. So $f(a_i) = 1$ and $f(\overline{a_i}) = 0$; in particular, $f$ sends $a_i$—one of the variables of $c_j$—to 1. Similarly, $w_i x_i \notin M$ (and so $f(\overline{a_i}) = 1$) when $\overline{a_i}$ is contained in $c_j$. Therefore $f$ is a satisfiability function for $\xi$: it sends at least one variable from each clause to 1.

Conversely, suppose $f$ is a truth assignment satisfying $\xi$. Claims 2 and 3 give us a $C_4$-transverse matching of the gadgets which contains $wx$, $w'x'$ from the variable gadgets where $a_i$ is false; $\overline{w}\bar{x}$ from the variable gadgets where $a_i$ is true; and edges from each clause gadget covering the nonempty subset of $y_1, y_2, y_3$ connected to true variables. This matching can be extended to break the chains of $C_4$'s connecting variable gadgets with clause gadgets. There are no other copies of $C_4$ in $G$, as the chains of $C_4$'s connecting variable and clause gadgets are too long for a $C_4$ to span more than one gadget. Thus the resulting matching $M$ is $C_4$-transverse.

**Theorem 2.10.** *TM($C_4$) remains NP-complete in the class of planar bipartite graphs.*

□

### 2.2.2 $H = C_\ell$, $\ell \geq 5$

The proof of Theorem 2.10 can be adapted to show that TM($C_\ell$) is NP-complete for $\ell \geq 4$; all we need are gadgets analogous to those in Figure 2.6 and a way to connect them together. In fact, the constructions in this subsection are only slight alterations to TM($C_4$)'s gadgets, using chains of $K_{2,3}$'s to pad out cycle lengths.

A *chain of* $K_{2,3}$ is a sequence of edge-disjoint $K_{2,3}$'s, each sharing a tip with the next (see Figure 2.7). Unlike a $C_4$-transverse matching, a $C_\ell$-transverse matching in general does not have to contain any edges of a $K_{2,3}$. However, since it is a matching, it is still limited to at most two edges from each, so we have the following.

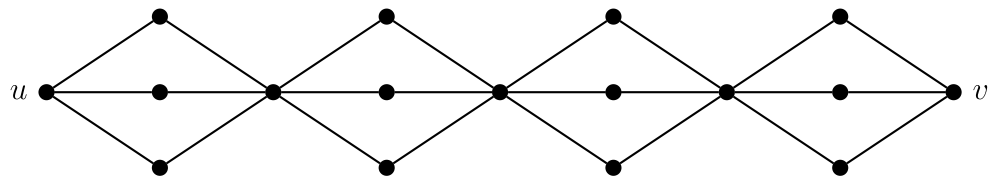



Figure 2.7: A chain of $K_{2,3}$'s

**Lemma 2.11.** *If $u$, $v$ are joined by a chain of $k$ copies of $K_{2,3}$ as in Figure 2.7, then any matching fails to cover some $u$-$v$ path of length $2k$.* □

When another path $P$ of length $\ell - 2k$ is added from $u$ to $v$, this places an interesting constraint on the possible $C_\ell$-transverse matchings on the graph (Lemma 2.12). We give the name *hinged cycle* to the resulting structure consisting of the path $P$ and a string of $K_{2,3}$'s. Figure 2.8 depicts a hinged cycle made from a $P_3$ and two copies of $K_{2,3}$.

**Lemma 2.12.** *Consider a hinged cycle made from a path $P$ of length $\ell - 2k$ and a string of $k$ copies of $K_{2,3}$. Any $C_\ell$-transverse matching contains at least one edge from $P$.* □

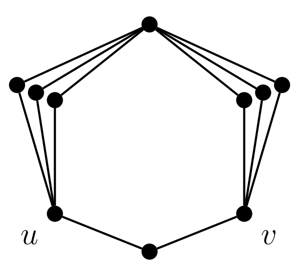



Figure 2.8: A hinged cycle

For example, any $C_6$-transverse matching of hinged cycle in Figure 2.8 contains one of the two bottom edges; in particular, it must cover one of $u$, $v$. This fact allows us to use chains of hinged cycles in much the same way that we used chains of $C_4$'s in Theorem 2.10.

**Lemma 2.13.** *Suppose $e, e'$ are joined by a chain of hinged cycles each consisting of a $P_3$ and $\frac{\ell-2}{2}$ copies of $K_{2,3}$ (when $\ell$ is even) or a $P_4$ and $\frac{\ell-3}{2}$ copies of $K_{2,3}$ (when $\ell$ is odd). Then no $C_\ell$ transverse matching contains both $e, e'$.* □

In particular, we use a chain of at least $\ell/2$ hinged cycles to connect a distinguished vertex from a variable gadget to a distinguished vertex from a clause gadget.

Figures 2.9 and 2.10 illustrate this connection for when $\ell = 6$ and $\ell = 5$, respectively. When $\ell$ is even, each hinged cycle is made from a $P_3$ and $\frac{\ell-2}{2}$ copies of $K_{2,3}$; when $\ell$ is odd, each is made from a $P_4$ and $\frac{\ell-3}{2}$ copies of $K_{2,3}$.

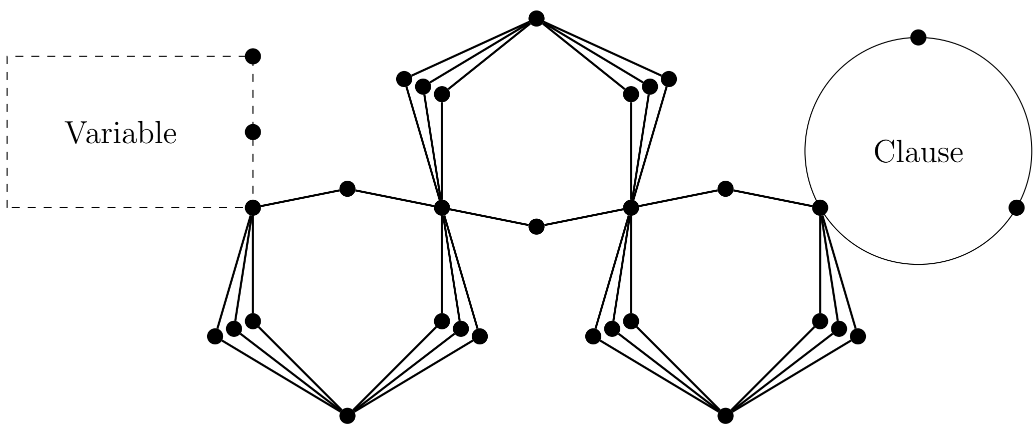



Figure 2.9: Connection of a variable gadget with a clause gadget

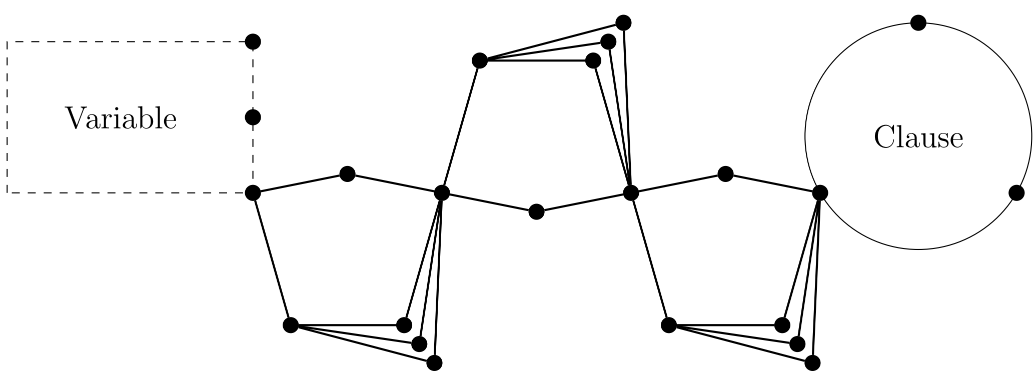



Figure 2.10: Connection of a variable gadget with a clause gadget

Chains of $K_{2,3}$'s also play an important role in the construction of the variable and clause gadgets. Recall that the clause gadget for $\mathrm{TM}(C_4)$ consists of three $C_4$'s, each pair intersecting in an edge. The general clause gadget (see Figure 2.11) consists of three hinged cycles, pairwise intersecting in an edge. When $\ell$ is even, the hinged cycles are each made from a $P_5$ and $\frac{\ell-4}{2}$ copies of $K_{2,3}$; when $\ell$ is odd, they are each made from a $P_4$ and $\frac{\ell-3}{2}$ copies of $K_{2,3}$. Any $C_\ell$-transverse matching of the clause gadget covers at least one of the distinguished vertices $y_1, y_2, y_3$ indicated in the figure.

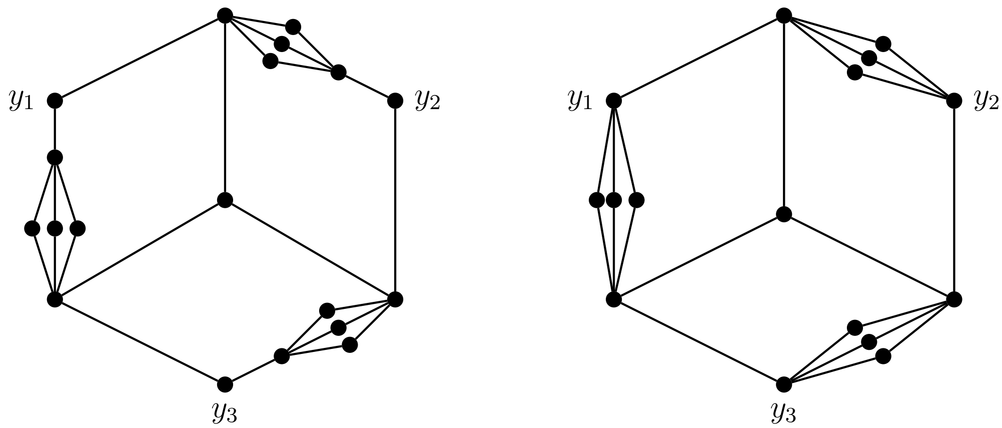



Figure 2.11: Clause gadgets

Finally, we construct the variable gadget. To fit in the proof of Theorem 2.10, this should have three edges $wx$, $w'x'$, and $\overline{w}\overline{x}$ such that any $C_\ell$-transverse matching contains either $\overline{w}\overline{x}$ or both of $wx, w'x'$. Figure 2.12 gives an example of such a gadget when $\ell$ is even ($= 6$, top) and when $\ell$ is odd ($= 5$, bottom).

Like the gadget we originally used in Theorem 2.10, TM($C_\ell$)'s variable gadget is organized around a $C_4$ from which any $C_\ell$-transverse matching must take two edges. When $\ell$ is even, this constraint is the consequence of a chain of $\frac{\ell-2}{2}$ copies of $K_{2,3}$'s between the opposite vertices of the $C_4$. When $\ell$ is odd, placing this constraint is a little more complicated. Name the $C_4$ $rstu$; we add an edge $rv$ and a chain of $\frac{\ell-3}{2}$ copies of $K_{2,3}$ between $v$ and $t$. (This forms two hinged cycles: one containing the path $vrst$ and the other containing $vrut$.) We then forbid $rv$ from being in any $C_\ell$-transverse matching by adding a hinged cycle consisting of a single edge and $\frac{\ell-1}{2}$ copies of $K_{2,3}$.

Surrounding the central $C_4$ are three hinged cycles which contain $wx, w'x'$, and $\overline{w}\bar{x}$. When $\ell$ is even, each such cycle is made from a $P_5$ and $\frac{\ell-4}{2}$ copies of $K_{2,3}$. When $\ell$ is odd, each is made from a $P_4$ and $\frac{\ell-3}{2}$ copies of $K_{2,3}$. Figure 2.12 illustrates the cases $\ell = 6$ and $\ell = 5$, respectively.

The above gadgets are obviously planar, triangle-free, and have the distinguished vertices $x, x', \bar{x}$ and $y_1, y_2, y_3$ on the outer face as desired. Therefore, our construction produces a planar triangle-free graph $G$ which admits a $C_\ell$-transverse matching if and only if an input propositional formula $\xi$ is satisfiable. In fact, when $\ell$ is even, the resulting $G$ is planar bipartite. We have thus proved the following dichotomy.

**Theorem 2.14.** *TM($C_\ell$) is solvable in polynomial time when $\ell = 3$, and NP-complete otherwise. When $\ell \geq 4$ is even, the problem remains NP-complete for the class of planar bipartite graphs. When $\ell \geq 5$ is odd, it remains NP-complete for the class of planar triangle-free graphs.* □

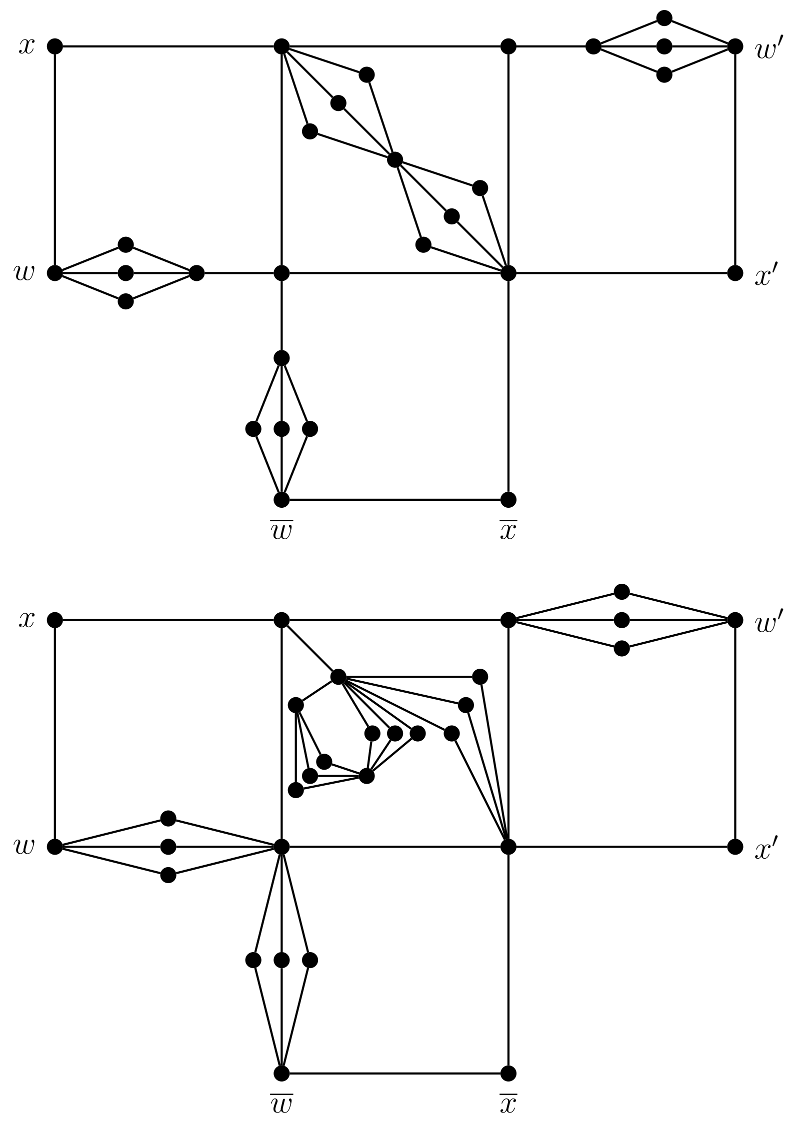



Figure 2.12: Variable gadgets

# Chapter 3

# $C_4$-transverse matchings in chordal bipartite graphs

This chapter continues our investigation into cycle-transverse matching problems in restricted graph classes. We have already seen classes—planar and triangle-free graphs—for which TM($C_\ell$) remains NP-complete. In this chapter, we exhibit a graph class for which TM($C_4$) is solvable in polynomial time.

A *chordal bipartite graph* is a graph having no odd cycles and no induced cycles of length greater than four. Like chordal graphs, chordal bipartite graphs enjoy many useful properties, which are surveyed in [19]. For instance, every chordal bipartite graph has a *bisimplicial edge*; that is, an edge $uv$ for which $N(u) \cup N(v)$ induces a biclique.

Another useful structure in chordal bipartite graphs is a *biclique separator*: every chordal bipartite graph with an induced $2K_2$ has a biclique $B$ such that $G - V(B)$ has more than one component. The main result of Section 3.1 is an important variation on this fact. We show that we can choose such a $B$ which is connected and such that each component of $G - V(B)$ has fewer than half as many vertices as $G$. This structure should be very useful in the study of chordal bipartite graphs; in particular, it helps us devise a divide-and-conquer algorithm for TM($C_4$) in Section 3.2.

## 3.1 A separator theorem for chordal bipartite graphs

**Lemma 3.1.** *Any connected chordal bipartite graph $G$ which contains an induced $2K_2$ has a connected biclique $B$ such that $G - V(B)$ has more than one component.*

**Proof:** Consider a minimal vertex set $S$ such that $G-S$ has more than one nontrivial component. Golumbic and Goss proved in [20] that in a chordal bipartite graph, such a set $S$ induces a biclique. If the biclique induced by $S$ is connected, then we are done; otherwise, it is a stable set with at least two vertices.

Let $u, v \in S$. The minimality of $S$ implies that $u, v$ both have neighbours in each component of $G - S$. Since there are at least two components of $G - S$ and $G$ has no induced cycle of length greater than four, every induced $u$-$v$ path is of length two. In other words, $N(u) = N(v)$. We can therefore find a vertex $s$ in a nontrivial component which is adjacent to every vertex in $S$, and let $B$ be the star induced by $S \cup \{s\}$. □

A *central biclique separator* of a graph $G$ is a biclique $B$ such that $G - V(B)$ has no component of size larger than $\frac{1}{2}|V(G)|$. Theorem 3.2 shows that every chordal bipartite graph with an induced $2K_2$ has a connected central biclique separator.

This result is not surprising, given that the analogous theorem for non-bipartite graphs holds. Every chordal graph has a central clique separator, which can be found by an algorithm by Gilbert, Rose, and Edenbrandt [18]. They describe their approach as making a clique "ooze around the graph like an amoeba". Although the details of our proof are different, we can use the same basic idea to find a central biclique separator in a chordal bipartite graph.

**Theorem 3.2.** *If $G$ is a connected chordal bipartite graph which has an induced $2K_2$, then it contains a connected biclique $B$ whose deletion leaves no component with more than $\frac{1}{2}|V(G)|$ vertices.*

**Proof:** The idea of the proof is as follows. Given a connected biclique separator $B$ and a component $A$ of $G-V(B)$, we move a vertex $a$ from $A$ to $B$ and move some vertices from $B$ to the other components. The result is a new connected biclique which separates $A-a$ from the rest of $G$. In this way, we can make $B$ "ooze towards" the largest component until $B$ becomes a central biclique separator.

Let $B$ be a connected biclique with bipartition $B_1 \cup B_2$ and let $A$ be a (nonempty) component of $G-V(B)$. If $B$ has one or two vertices, we can add a vertex from $A$ to $B$ and maintain a biclique. If $B$ contains a vertex $b$ which is not adjacent to any vertex of $A$ and such that $B-\{b\}$ is still connected, then removing $b$ from $B$ does not increase the size of $A$.

Now suppose, without loss of generality, that $|B_1| \geq 2$, $|B_2| \geq 1$, and every vertex in $B_1$ has a neighbour in $A$. If $A$ has a vertex which is adjacent to every vertex in $B_1$, then we can add it to $B$. Otherwise, there are distinct vertices $a, a' \in A$ and $b, b' \in B_1$ such that $a$ is adjacent to $b$ and $a'$ is adjacent to $b'$, but not vice versa. Let $d \in B_2$. Since $A$ is a component of $G-V(B)$, it contains a path $ac\ldots a'$. Hence $ac\ldots a'b'dba$ is a cycle in $G$; it is not induced because $G$ is chordal bipartite. In particular, at least one of $a'b$, $ab'$, $cd$ is a chord; since we already assumed that $ab'$ and $a'b$ are not edges, $c$ is adjacent to $d$. Repeating this argument shows that every vertex in $B_2$ is adjacent to $c$, so we can add $c$ to $B$.

The above arguments show that we can make $B$ ooze towards any component $A$. We can obtain a central biclique separator using the following algorithm. Start with an arbitrary connected biclique separator $B$, whose existence is guaranteed by Lemma 3.1. As long as $G-B$ contains a component $A$ with more than $\frac{1}{2}|V(G)|$ vertices, let $B$ ooze towards $A$. One of the vertices of $A$ is engulfed by $B$, while the vertices removed from $B$ join components of size $< \frac{1}{2}|V(G)|$. This reduces the size of the largest component of $G-V(B)$; we repeat this process until no component has more than $\frac{1}{2}|V(G)|$ vertices. □

## 3.2 A divide-and-conquer algorithm

In this section, we use Theorem 3.2 (and the "amoeba algorithm" used in its proof) to solve TM($C_4$) for chordal bipartite inputs. In fact, our algorithm solves the $C_4$-*transverse matching extension problem*; it decides whether an input chordal bipartite graph admits a $C_4$-transverse matching which contains a given edge set.

Our algorithm—which we call `TMEXT`$(G, M)$—could be described as using a non-deterministic divide-and-conquer approach. It splits the input graph into smaller subgraphs, recursively calls itself several times for each such subgraph, and decides whether some family of matchings detected in the subgraphs can be merged to form a desired matching of the whole graph.

We now explain the steps of the algorithm in detail. A summary is given below.

---

`TMEXT(`$G, M$`)`

---

1: Check if $G$ contains a $K_{2,4}$ or two $K_{2,3}$'s sharing a tip. If so, $G$ has no desired matching.
2: **if** $G$ has no induced $2K_2$ **then** decide whether it has a desired matching
3: Otherwise, find a central biclique separator $B$. Write $G$ as the union of smaller subgraphs $G_1, G_2 \ldots, G_k$ intersecting in $B$.
4: Run `TMEXT` to find the vertex sets of $B$ covered by desired $C_4$-transverse matchings of each $G_i$.
5: Determine whether Step 4 has recorded a family of disjoint sets where the corresponding matchings can be merged to form a desired matching of $G$.

---

**Step 1.** *Check if $G$ contains a $K_{2,4}$ or two edge-disjoint $K_{2,3}$'s which share a tip. If so, report that G has no desired matching.*

**Step 2.** *If $G$ has no induced $2K_2$, decide whether it has a desired matching.*

First, delete the pendant edges of $G$ until every vertex is of degree $\geq 2$; this does not affect the existence of a $C_4$-transverse matching because no pendant edge is contained in a cycle. The resulting graph is chordal bipartite, contains no induced $2K_2$ and no $K_{2,4}$, and has minimum degree $\geq 2$. We claim that such a graph is of bounded size.

To see this, consider a bisimplicial edge $uv$ of $G$. Then $N(u) \cup N(v)$ induces a biclique $K_{d(u),d(v)}$. Since $G'$ has no $K_{2,4}$, the degrees of $u$ and $v$ in $G$ are at most three. Because $G$ has no induced $2K_2$, each vertex of $G'$ is adjacent to one of the six vertices in $N(u) \cup N(v)$. As each pair of vertices has at most three common neighbours, $G$ has at most $3\binom{6}{2} + 6$ vertices. It can thus be decided whether $G$ has a desired matching in $O(1)$ time.

**Step 3.** *Find a connected central biclique separator $B$. Write $G$ as the union of connected subgraphs $G_1, G_2, \ldots, G_k$, each having at most $\frac{1}{2}|V(G)| + 6$ vertices and intersecting $B$ in at most six vertices, and whose pairwise intersection is contained in $B$.*

The central biclique separator $B$ can be found in one of two ways. Since $G$ contains no $K_{2,4}$, it has at most $|V(G)| + \binom{|V(G)|}{6}$ maximal connected bicliques. It is possible to check them all in polynomial to find a central separator. However, it is more efficient to find connected biclique separator as in Lemma 3.1 and then apply the amoeba algorithm from Theorem 3.2.

Let $A_1, A_2, \ldots, A_k$ be the components of $G - V(B)$. If $B$ is not a star, then (since $G$ contains no $K_{2,4}$) it has at most six vertices. We can then let each $G_i$ be the subgraph induced by the vertices $V(B) \cup V(A_i)$.

On the other hand, if $B$ is a maximal star with center vertex $b$, then each $A_i$ contains neighbours of at most three vertices $a_i, a_i', a_i''$ in $B$. Take $G_i$ to be the subgraph induced by $V(A_i) \cup \{a_i, a_i', a_i'', b\}$.

**Step 4.** *For each vertex set $S$ and matching $M'$ of $B$, and for each $i = 1, 2, \ldots, k$, recursively call* `TMEXT` *to decide whether $G_i$ has a $C_4$-transverse matching $M' \cup M_{i,S}$ such that $M_{i,S}$ does not cover any vertices of $V(B) \setminus S$.*

Specifically, modify $G_i$ by adding a pendant edge to each vertex of $B$ not in $S$ nor covered by $M'$. Run `TMEXT` on this modified $G_i$ as the input graph, and with input matching consisting of $M \cap E(G_i)$, $M'$, and these new pendant edges.

**Step 5.** *Determine whether there is a collection of matchings $M' \cup M_{i,S_i}$ as above such that $S_1, S_2, \ldots, S_k$ are disjoint. If so, their union is a desired $C_4$-transverse matching of $G$; if not, report that $G$ has no such matching.*

Run the following for each matching $M'$ of $B$. First, check that for each $i$, Step 4 has found at least one matching $M' \cup M_{i,S_i}$. If not, there is no desired $C_4$-transverse matching of $G$ containing the edges of $M'$.

Next, identify the indices $i$ for which we have recorded a matching of the form $M' \cup M_{i,\emptyset}$. We can safely ignore these indices, as $M_{i,\emptyset}$ can be merged with any matching of the other $G_i$. For each of the remaining indices, a desired $C_4$-transverse matching of $G_i$ must cover at least one vertex of $B$ with an edge not in $M'$.

If $B$ is not a star, then it has at most six vertices. If more than six indices remain, then no family of associated matchings can be merged. Otherwise, we need only decide whether we can merge six of a bounded number of matchings. This can be done in constant time.

Suppose, then, that $B$ is a maximal star with center $b$ and several points. The remaining indices $i$ fall into three categories:

- those for which there is only one minimal set $S_i$ such that we have recorded some $M' \cup M_{i,S_i}$;
- those for which $G_i$ contains three points of the star; and

- those for which $G_i$ contains two points $u_i, v_i$ of the star and we have recorded matchings $M' \cup M_{i,\{u_i\}}$ and $M' \cup M_{i,\{v_i\}}$.

We may without loss of generality associate each index $i$ of the first type with $S_i$. There are a bounded number of vertices of the first type, since $G$ does not contain two $K_{2,3}$'s sharing a tip; thus there are a bounded number of ways of assigning these indices $i$ to vertex sets $S_i$ which are disjoint from each other and from the above sets. For each such assignment, we decide whether there is a system of distinct representatives of the family $\{u_i, v_i\}$ for each index $i$ of the third type. If so, we have succeeded in finding a disjoint family of sets corresponding to a mergable family of matchings $M' \cup M_{i,S_i}$.

If none of the above results in a desired $C_4$-transverse matching, report that no such matching exists.

**Lemma 3.3.** *The above algorithm is correct and runs in polynomial time.*

**Proof:** It is easy to see that Step 1 is correct, and Steps 2 and 5 have already been discussed. The correctness of Step 3 follows from Theorem 3.2. Finally, Step 4 is correct by induction; it only relies on the assumption that its recursive calls of `TMEXT` work as claimed.

The only steps which are not clearly executable in polynomial time are the recursive calls of `TMEXT`. Observe that `TMEXT` is called a constant number of times $\lambda$ on each subgraph $G_i$ in Step 4; this follows from the fact that $G_i$ intersects the central biclique $B$ in a bounded number of vertices. Hence the runtime $T(n)$ of `TMEXT` satisfies the recurrence

$$\begin{aligned} T(n) &\leq \lambda \cdot T\left(\frac{n}{2}\right) + p_1(n) \\ &\leq \lambda^{\log n} \cdot p_2(n) \end{aligned}$$

which is polynomial in $n = |V(G)|$. $\square$

**Theorem 3.4.** *TMEXT($C_4$) is solvable in polynomial time when the input is restricted to be a chordal bipartite graph.* $\square$

# Chapter 4

# Tree-transverse matchings

In this chapter, we move on to tree-transverse matching problems. First, we review the known polynomial cases: TM($K_{1,k}$) and TM($P_4$). We provide a novel solution to the latter problem and extend it to solve TM($Y$). In contrast, we show that TM($H$) is NP-complete whenever $H$ is a tree of diameter $\geq 4$.

## 4.1 Polynomial-time solvable cases

### 4.1.1 $H$ is a star

It was shown in [2] that each star-transverse matching problem TM($K_{1,k}$) can be solved in polynomial time. For completeness, we reproduce a solution here.

**Proposition 4.1.** *Let $M$ be a matching in a graph $G$. The following are equivalent:*

1. *$M$ is $K_{1,k}$-transverse in $G$*
2. *every vertex in $G - M$ has degree $< k$*
3. *$G$ has no vertex of degree $> k$ and $M$ covers every degree-$k$ vertex in $G$.* □

Thus to solve TM($K_{1,k}$), it is sufficient to decide whether the given graph has a matching covering a given set of vertices (namely, those of degree $k$).

**Lemma 4.2.** *Given a graph $G$ and vertex set $S$, it can be determined in polynomial time whether $G$ has a matching covering every vertex of $S$.*

**Proof:** We reduce the problem to that of finding a perfect matching, which is solvable in polynomial time [8]. Let $G'$ be a copy of $G$ (disjoint from $G$) with $f : G \to G'$ an isomorphism. Let $G^*$ be obtained from $G \cup G'$ by adding the edge $vf(v)$ if $v \in V(G) - S$. If $M$ is a matching of $G$ covering every vertex of $S$, then $M \cup f(M) \cup \{vf(v) : v \text{ is not covered by } M\}$ is a perfect matching of $G^*$. Conversely, any perfect matching of $G^*$ restricts to a desired matching of $G$. Thus $G$ has a matching covering every vertex of $S$ if and only if $G^*$ has a perfect matching. □

**Theorem 4.3.** *TM($K_{1,k}$) is solvable in polynomial time for each $k$.* □

In fact, a slight addition to the above algorithm allows us to specify edges which should be contained or avoided by the matching. This allows us to solve the star-transverse extension problems.

**Lemma 4.4.** *Given a graph $G$, vertex set $S$, a matching $M$, and an edge set $N$, it can be determined in polynomial time whether $G$ has a matching covering every vertex of $S$, containing every edge of $M$, and containing no edge in $N$.*

**Proof:** Apply Lemma 4.2 to $G$ and $S$, after deleting the edges $N$ from $G$ and deleting the vertices covered by $M$ from both $G$ and $S$. □

**Theorem 4.5.** *TMEXT($K_{1,k}$) can be solved in polynomial time for each $k$.* □

### 4.1.2 $H$ is a tree of diameter three and the input is triangle-free

The $P_4$-transverse matching problem was first solved for bipartite graphs in [12, 21]. This subsection presents a simple solution to TM($\pi_{k,\ell}$) when the input is restricted to be triangle-free. When $k = \ell = 2$, this restricts to a new, conceptually simpler solution to TM($P_4$) for bipartite graphs.

Recall that $\pi_{k,\ell}$ denotes the diameter-three tree with center vertices of degree $k$ and $\ell$.

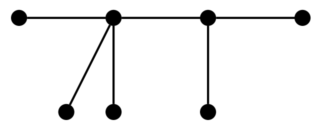


Figure 4.1: A $\pi_{4,3}$

Our algorithm constructs a list of edges contained in, and a list of vertices covered by, every $\pi_{k,\ell}$-transverse matching. It then checks whether the graph admits a matching containing those edges and covering those vertices. If not, then the graph has no $\pi_{k,\ell}$-transverse matching. Otherwise, we show that such a matching can be extended to one which is $\pi_{k,\ell}$-transverse.

Our algorithm constructs the lists by considering each $\pi_{k,\ell}$ in the input graph. If $d(u) > k$ and $d(v) > \ell$, it adds the center edge $uv$ to the list of edges to be contained in the matching. If $d(u) > k$ and $d(v) = \ell$, it adds $v$ to the list of vertices to be covered. Likewise, if $d(u) = k$ and $d(v) > \ell$, it adds $u$ to the list of vertices to be covered. Finally, if $d(u) = k$ and $d(v) = \ell$, our algorithm does nothing to that copy.

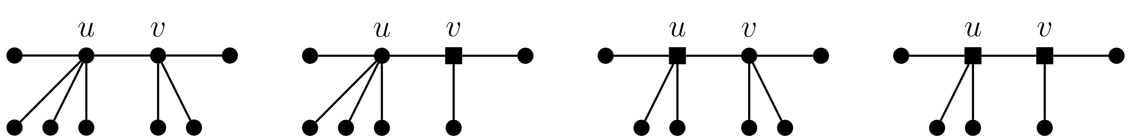



Figure 4.2: The four cases of $\pi_{4,3}$ in a triangle-free graph

It is not hard to see that any $\pi_{k,\ell}$-transverse matching $M$ contains the center edge from every $\pi_{k,\ell}$ which falls into the first case. Indeed, $u$ has at least $k$ neighbours and $v$ has at least $\ell$ neighbours in $G - M$. Since $G$ is triangle-free, these vertices are distinct and would form a $\pi_{k,\ell}$ in $G - M$ if $M$ did not contain $uv$. Similarly, any

$\pi_{k,\ell}$-transverse matching of $G$ covers $v$ in any $\pi_{k,\ell}$ of the second type, and $u$ in any of the third type.

Suppose that $G$ admits a matching which contains the edges and covers the vertices in the respective lists. Such a matching contains the center edge of each $\pi_{k,\ell}$ of the first type and at least one edge of each $\pi_{k,\ell}$ of the second and third types. We can extend it to a matching $M$ which contains an edge from every $\pi_{k,\ell}$ of the fourth type: simply add the center edge $uv$ for each $\pi_{k,\ell}$ whose center vertices $u, v$ are both uncovered. This matching $M$ is $\pi_{k,\ell}$-transverse.

The above procedure clearly takes polynomial time and reduces TM($\pi_{k,\ell}$) to the problem of finding a matching containing a given set of edges and covering a given set of vertices. This was shown to be polynomial in Lemma 4.4. We therefore have the following.

**Theorem 4.6.** *In the class of triangle-free graphs, TM($\pi_{k,\ell}$) is solvable in polynomial time for all $k$, $\ell$.* □

To see if a given matching can be extended to a $\pi_{k,\ell}$-transverse one, we can simply add its edges to the list before executing the rest of the algorithm.

**Theorem 4.7.** *In the class of triangle-free graphs, TMEXT($\pi_{k,\ell}$) is solvable in polynomial time for all $k, \ell$.* □

### 4.1.3 $H$ is a $P_4$

The $P_4$-transverse matching problem for general graphs was solved indirectly by Ekim in [9] before Churchley and Huang gave a structural characterization and linear time algorithm in [3]. In this subsection, we present a new solution to TM($P_4$).

The algorithm is along the same lines as the one in the previous subsection. It considers each $P_4$ of the input graph and constructs a list of edges that a desired matching should contain and a list of vertices that it should cover. We call on Lemma 4.4 to find such a matching if one exists, and extend it to a $P_4$-transverse matching.

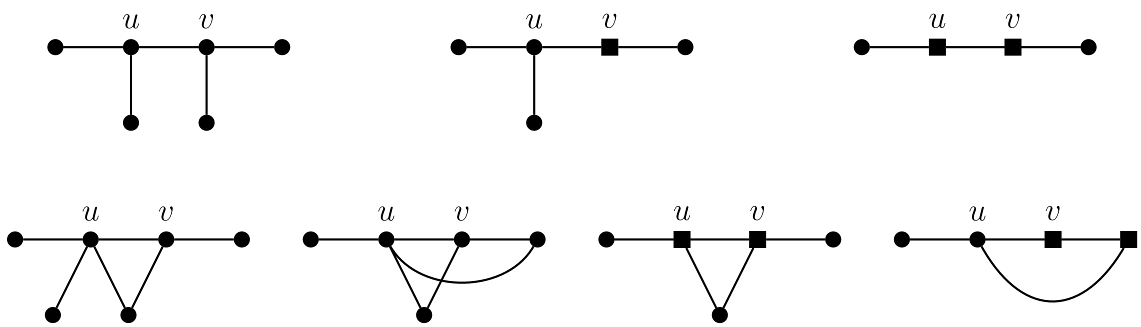


Figure 4.3: The several cases of $P_4$

Consider a $P_4 : u'uvv'$. If $u, v$ do not have a common neighbour, the algorithm proceeds as before: it adds $uv$ to the list of edges to contain if $d(u), d(v) > 2$; adds $u$ of the list of vertices if $d(v) > 2 = d(u)$; adds $v$ to the list of vertices to cover if $d(u) = 2 < d(v)$; and does nothing if $d(u) = d(v) = 2$.

There are a few more cases to consider when $u, v$ do have a common neighbour. If $d(u) \geq 4$ and $d(v) \geq 3$, then $G$ does not admit a $P_4$-transverse matching. If $d(u) = d(v) = 3$, the algorithm adds $u'u$ and $vv'$ to the list of edges to be contained in the matching. Finally, if $d(u) \geq 3$ and $d(v) = 2$, then the algorithm adds $vv'$ to the list.

The analysis of last subsection applies to the cases where $u, v$ have no common neighbour. Suppose, then, that $u, v$ have a common neighbour $w$. If $d(u), d(v) \geq 3$, then none of $uv, uw, vw$ is contained in any $\pi_{k,\ell}$-transverse matching because each is adjacent to the three edges of a $P_4$. This implies that $u'u$ and $vv'$ are contained in every $\pi_{k,\ell}$-transverse matching when $d(u) = d(v) = 3$ and that $G$ has no $\pi_{k,\ell}$-transverse matching when $d(u) \geq 4$. Finally, when $d(u) \geq 3$ and $d(v) = d(v') = 2$— i.e., the $P_4 : u'uvv'$ is contained in a paw—neither $uv$ nor $uv'$ is contained in any $\pi_{k,\ell}$-transverse matching. We may therefore assume that $vv'$ is included in such a matching.

We have reduced TM($P_4$) to the problem of finding a matching which contains a given edge set and covers a given set of vertices. By appealing to Lemma 4.4 as we did in the previous subsection, we have the following results:

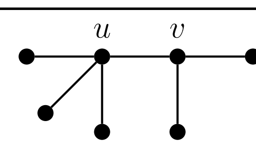
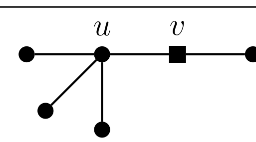
**Theorem 4.8.** *TM($P_4$) is solvable in polynomial time.*

**Theorem 4.9.** *TMEXT($P_4$) is solvable in polynomial time.*

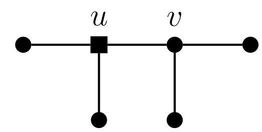
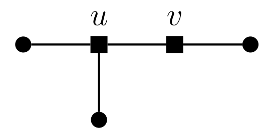
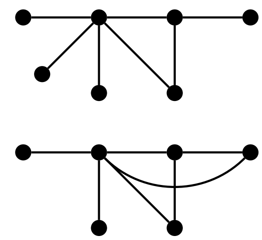
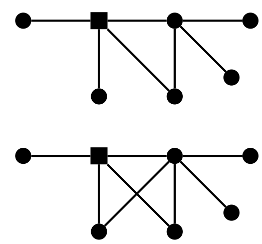
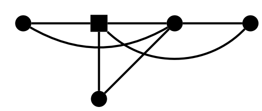
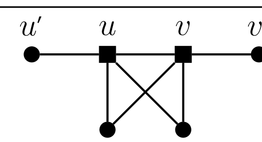
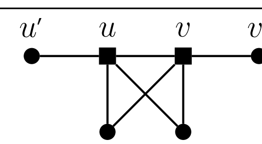
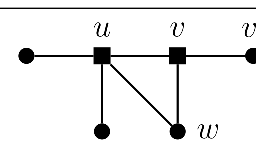
### 4.1.4 $H$ is the $Y$ graph

The $Y$ graph is the name given to $\pi_{3,2}$, which consists of a $P_4$ and an additional pendant edge. This subsection solves TM($Y$) using a similar strategy as the previous subsections, albeit with a more complicated case analysis.

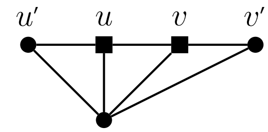
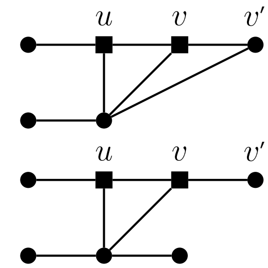
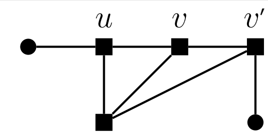
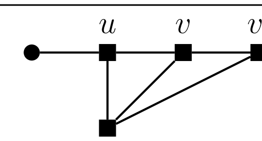
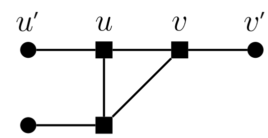
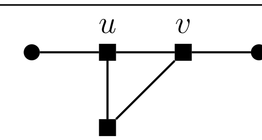
Our algorithm reduces the problem to that of deciding whether a graph has a matching containing a given set of edges, avoiding another set of edges, and covering a given set of vertices. It considers each copy of $Y$ to construct these lists according to a number of cases. In a few special cases, the algorithm makes alterations to the graph. The cases are summarized and illustrated in the following table. In each figure, squares indicate vertices of fixed degree; i.e. each square vertex is incident only with the edges shown.

| | Case | Action | Illustration |
|---|---|---|---|
| 1: | $d(u) > 3$, $d(v) > 2$, no common neighbours | contain $uv$ |  |
| 2: | $d(u) > 3$, $d(v) = 2$, no common neighbours | cover $v$ |  |

↪ continued on next page

| | Case | Action | Illustration |
|---|---|---|---|
| 3: | $d(u) = 3$, $d(v) > 2$, no common neighbours | cover $u$ |  |
| 4: | $d(u) = 3$, $d(v) = 2$ | do nothing |  |
| 5: | $d(u) > 4$, $d(v) > 2$, one or more common neighbour(s) | $G$ has no $Y$-transverse matching |  |
| 6: | $d(u) = 4$, $d(v) > 3$, one to $d(v) - 3$ common neighbours | $G$ has no $Y$-transverse matching |  |
| 7: | $d(u) = 4$, $d(v) > 3$, three common neighbours | $G$ has no $Y$-transverse matching |  |
| 8: | $d(u) = d(v) = 4$, two common neighbours | contain $u'u$ and $vv'$ |  |
| 9: | $d(u) = 4$, $d(v) = 3$, two common neighbours | contain $u'u$ |  |
| 10: | $d(u) = 4$, $d(v) = 3$, one common neighbour | avoid $uv, uw$; cover $u$; contain $vv'$ |  |

↪ continued on next page

| | Case | Action | Illustration |
|---|---|---|---|
| 11: | $d(u) = d(v) = 3$, one common neighbour adjacent to $u', v'$ | cover $u, v$ |  |
| 12: | $d(u) = d(v) = 3$, one common neighbour of degree $> 3$ adjacent to $v'$ | included in Case 10 and/or 11 of another $P_4$ |  |
| 13: | $d(u) = d(v) = 3$, $d(v') = 3$, one common neighbour of degree 3 adjacent to $v'$ | alter graph as described below |  |
| 14: | $d(u) = d(v) = 3$, $d(v') = 2$, one common neighbour of degree 3 adjacent to $v'$ | cover $u, v$ |  |
| 15: | $d(u) = d(v) = 3$, one common neighbour adjacent to $u'$ | same as cases 12–14 | see above |
| 16: | $d(u) = d(v) = 3$, one common neighbour of degree 3 not adjacent to $u', v'$ | alter graph as described below |  |
| 17: | $d(u) = d(v) = 3$, one common neighbour of degree 2 | cover $u, v$ |  |

Two cases—13 and 16—require some alteration of the graph (see Figure 4.4). In Case 13, the algorithm deletes the vertices $v, w$, adds an edge $uv'$, and adds $u, v'$ to the list of vertices to be covered. In Case 16, the algorithm deletes the edges of the triangle $uvw$ and replaces them with a new vertex $x$ and $P_3$'s between $x$ and each of $u, v, w$. It then adds $x$, $u$, $v$, $w$ to the list of vertices to cover. All alterations are done after the algorithm has finished processing every copy of $Y$ in the graph.

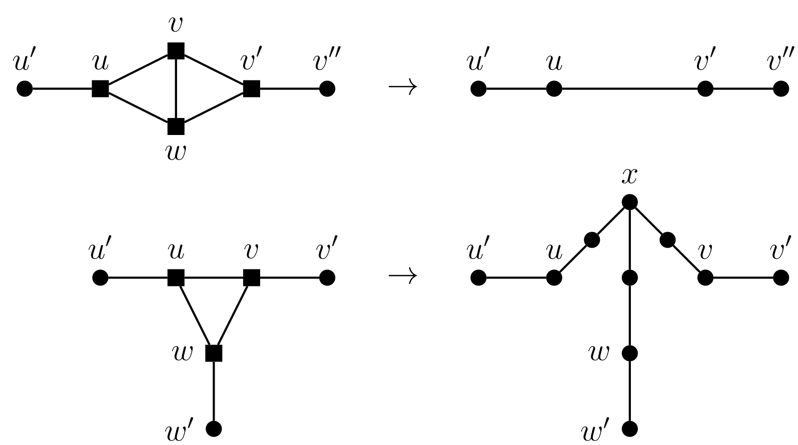



Figure 4.4: Alterations made to the graph in cases 13 and 16

We now discuss the correctness of the algorithm. The first four cases are identical to those in Subsection 4.1.2 and have already been justified. The following is a typical argument for the algorithm's response in cases 5 through 11. No $Y$-transverse matching contains an edge which is adjacent to four other edges forming a $Y$. Therefore, in Case 10, the algorithm should add $uv, uw$ to the list of edges to avoid. Any two edges incident with $u$ form a $Y$ with $uvv'$, so the algorithm should add $vv'$ to the list of edges to be contained in a desired matching. Finally, it is clear from inspection that a desired matching must take one of the two remaining edges incident with $u$, so the algorithm should add $u$ to the list of vertices to cover.

In Case 12, there are other copies of $Y$, centered at the common neighbour of $u$ and $v$, which fall into Cases 10 or 11. When the algorithm processes these $Y$'s, it adds $u, v$ to the list of vertices to cover; no further action is required.

The algorithm adds $u, v$ to the list of vertices to cover in Cases 14 or 17, although the input graph may have a $Y$-transverse matching which does not cover both. However, it is always possible to add an edge—$uv$, $uw$, or $vw$—to such a matching to ensure that $u, v$ are both covered.

A $Y$-transverse matching of the structure in Case 13 cannot contain exactly one of $u'u, v'v''$. Similarly, a $Y$-transverse matching of the structure in Case 16 contains at least one of $u'u, vv', ww'$. (See Figure 4.4). The algorithm alters the graph so that these constraints are placed on the desired matching.

The above discussion shows that the resulting graph has a desired matching if the input graph has a $Y$-transverse matching. Conversely, when the algorithm processes a $Y$, it adds a vertex or edge to one of the lists so that a desired matching must break (or can be extended to a matching which breaks) that copy of $Y$. Thus if the algorithm detects a desired matching, the input graph has a $Y$-transverse matching.

**Theorem 4.10.** *TM($Y$) can be solved in polynomial time.* □

The $Y$-transverse matching extension problem can be solved by adding edges to the appropriate list before executing the rest of the algorithm.

**Theorem 4.11.** *TMEXT($Y$) can be solved in polynomial time.* □

## 4.2 NP-complete cases

This section shows that TM($H$) is NP-complete whenever $H$ is a tree of diameter $\geq 4$. We begin with the case $H = P_5$, which remains NP-complete when the input is restricted to be a planar bipartite graph. Subsections 4.2.2 and 4.2.3 show how to adapt the proof when $H$ is one of several "minimal" trees of diameter four or five. The remaining tree-transverse matching problems can be reduced from these minimal problems; we do this in Subsection 4.2.4.

### 4.2.1 $H = P_5$

Given an instance $\xi$ of Subcubic Planar SAT, we construct a planar bipartite graph $G$ which admits a $P_5$-transverse matching if and only if $\xi$ is satisfiable. Let

$\xi$ consist of variables $a_1, \ldots, a_p, \overline{a_1}, \ldots, \overline{a_p}$ and clauses $c_1, \ldots, c_q$. As $\xi$ is an instance of Subcubic Planar SAT, each $a_i$ appears in two clauses, each $\overline{a_i}$ appears in one, and each clause $c_j$ has at most three variables.

We construct $G$ from the incidence graph of $\xi$ by replacing each vertex with one of two planar bipartite graphs as follows. Each vertex which represents variables $a_i, \overline{a_i}$ is replaced by a copy of the variable gadget shown in Figure 4.5 (left). This subgraph has three distinguished edges $e, e', \bar{e}$ from which we read the truth value of the corresponding variables $a_i, \overline{a_i}$. Any $P_5$-transverse matching contains either $\bar{e}$ or both $e, e'$: in the first case, $a_i$ is false; in the second, $a_i$ is true.

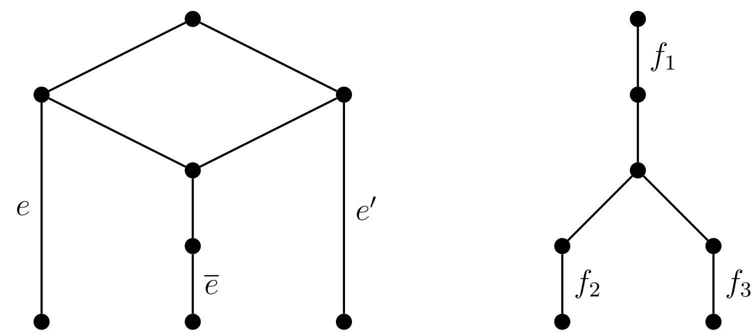



Figure 4.5: Variable and clause gadgets

Each vertex representing a clause of $\xi$ is replaced by a copy of the clause gadget in Figure 4.5 (right). Any $P_5$-transverse matching contains at least one of its distinguished edges $f_1, f_2, f_3$. This condition forces at least one of the clause's variables to be true.

The variable and clause gadgets are connected by a series of vertex identifications as in Figure 4.6. Consider a variable gadget corresponding to $a_i, \overline{a_i}$. We identify the pendant vertices incident with $e$ and $e'$ with those from the two gadgets representing clauses containing $a_i$. Likewise, we identify $\bar{e}$'s pendant vertex with one from the gadget representing the clause containing $\overline{a_i}$.

**Claim 1.** *$G$ is planar bipartite.*

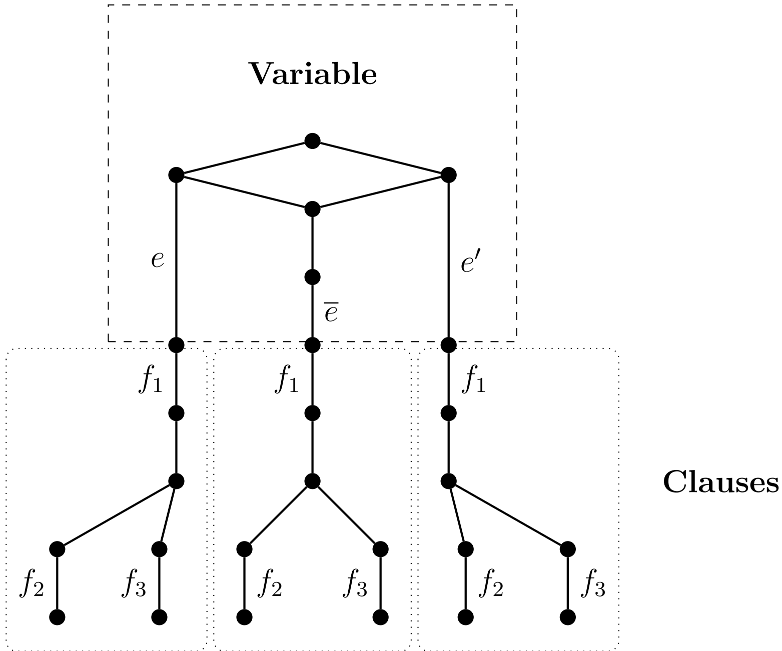



Figure 4.6: Connection of variable and clause gadgets

**Proof:** We have constructed $G$ from the planar incidence graph of $\xi$ by replacing vertices by planar subgraphs and identifying pairs of vertices according to the edges of the incidence graph. Such a graph is clearly planar.

Moreover, $G$ is bipartite, as it can be obtained from a bipartite graph (namely, the disjoint union of the variable and clause gadgets) by identifying vertices in the same partite set of a bipartition. □

**Claim 2.** *Any $P_5$-transverse matching of the variable gadget contains $\bar{e}$ or both $e, e'$. There are $P_5$-transverse matchings containing $\bar{e}$ and not $e, e'$, and vice versa.*

**Proof:** Let the variable gadget consist of the $C_4 : uv\bar{u}v'$, the edge $\bar{u}\bar{v}$, and the distinguished edges $e = vw$, $e' = v'w'$, and $\bar{e} = \bar{v}\bar{w}$. Let $M$ be a $P_5$-transverse matching which does not contain $\bar{e}$. Since $M$ contains at most one edge incident with $\bar{u}$ and each of $v, v'$, $\bar{u}\bar{v} \in M$. Thus $v\bar{u}$ and $v'\bar{u} \notin M$. As $M$ misses at least one of $v'u, v'w'$, this implies that $e \in M$. Similarly, $e' \in M$.

For the second statement, observe that $\{\bar{e}, uv, \bar{u}v'\}$ and $\{e, e', \bar{u}\bar{v}\}$ are both $P_5$-transverse matchings. □

**Claim 3.** *Any $P_5$-transverse matching of the clause gadget contains one of $f_1, f_2, f_3$. There are $P_5$-transverse matchings containing each nonempty subset of $f_1, f_2, f_3$.*

**Proof:** The first statement is clearly true as any matching misses at least two edges incident with the middle vertex. For the second statement, observe that

$$\{f_1, f_2'\}, \quad \{f_2, f_3'\}, \quad \{f_3, f_1'\}, \quad \{f_1, f_2\}, \quad \{f_2, f_3\}, \quad \{f_1, f_3\}, \quad \{f_1, f_2, f_3\},$$

(where $f_1', f_2', f_3'$ are the center edges adjacent to $f_1, f_2, f_3$, respectively) are all $P_5$-transverse matchings. □

Suppose $\xi$ is satisfiable; that is, there is a function $f : \{a_1, \ldots, a_p, \overline{a_1}, \ldots, \overline{a_p}\} \to \{0, 1\}$ which sends at least one variable from each clause to 1. Consider the matching

$$M = \left\{ \begin{array}{rl} e, e' & \text{from } a_i\text{'s copy of the variable gadget whenever } f(a_i) = 0 \\ \bar{e} & \text{from } a_i\text{'s copy of the variable gadget whenever} f(a_i) = 1 \\ f_1, f_2, f_3 & \text{vertex-disjoint from the above} \end{array} \right\}.$$

For instance, if $f(a_i) = 1$ then $M$ contains the corresponding $\bar{e}$ and the two edges adjacent to $e, e'$ from the gadgets representing clauses containing the (true) variable $a_i$. Because $f$ is a solution to $\xi$, $M$ contains at least one of $f_1, f_2, f_3$ from each clause gadget. By Claims 2 and 3 above, we can extend $M$ to a matching which is $P_5$-transverse when restricted to any single variable or clause gadget. Since the components of $G - M$ are subgraphs of these gadgets, the resulting matching is $P_5$-transverse in $G$.

Conversely, suppose $M$ is an $H$-transverse matching of $G$. A solution to $\xi$ can be recovered in much the same way: let

$$f(a_i) = \begin{cases} 0 & e, e' \in M \text{ in } a_i\text{'s copy of the variable gadget} \\ 1 & \text{otherwise} \end{cases}$$

and $f(\overline{a_i}) \neq f(a_i)$. By Claim 3, $M$ contains at least one of $f_1, f_2, f_3$ from each clause gadget. If that edge is adjacent to (say) $e$ in some variable gadget, then the corresponding variable $a_i$ is true as $e \notin M$. Otherwise, if that edge is adjacent to $\bar{e} \notin M$, then by Claim 2 $e, e' \in M$, $a_i$ is false, and hence the corresponding variable $\overline{a_i}$ is true. Hence $f$ satisfies all clauses of $\xi$.

We have therefore constructed a planar bipartite graph $G$ which admits a $P_5$-transverse matching if and only if the input instance $\xi$ of Subcubic Planar SAT is satisfiable.

**Theorem 4.12.** *TM($P_5$) is NP-complete, and remains so in the class of planar bipartite graphs.* □

### 4.2.2 $H$ is a diameter-four tree with few high-degree vertices

In this subsection, we adapt the above proof to show that TM($H$) is NP-complete whenever $H$ is a tree of diameter four and has at most two vertices of degree $\geq 3$. Two examples of such graphs are given below.

Figure 4.7: Diameter four trees with at most two high-degree vertices

One incredibly useful structure in the general proof is the graph $K_n + e$ consisting of a clique and a pendant edge. When $n = |V(H)| - 1$, we call this graph a *forcing* *gadget* for $H$ because every $H$-transverse matching contains the pendant edge.

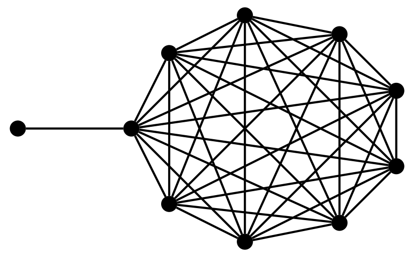


Figure 4.8: Forcing gadget

**Lemma 4.13.** *Let $H$ be a tree which is not a star or a $P_4$. Consider a $K_n + e$ with $n = |V(H)| - 1$ and $e = uv$ (with $v$ belonging to the $K_n$). Any $H$-transverse matching of this graph contains $e$; moreover, $\{e\}$ is $H$-transverse.*

**Proof:** Clearly, $\{e\}$ is an $H$-transverse matching: deleting it leaves a graph with too few vertices to contain a copy of $H$. On the other hand, let $M$ be any matching of the clique. Since $H$ is neither a star nor a $P_4$, it has a leaf $h$ such that $H' = H - h$ is not a star. In other words, $H$ can be obtained from $H'$ by adding an edge $oh$.

A result of Chvátal and Harary [6] implies that if $H'$ is a tree which is not a star, then the complete graph on $|V(H')|$ vertices has no $H'$-transverse matching. Hence $K - M$ contains a copy of $H'$. In fact, by symmetry and the fact that $H'$ is not a star, there is a copy of $H'$ in $K - M$ where $o$ corresponds to $v$. Adding the edge $e = uv$ to the copy of $H'$ would result in a copy of $H$, so any $H$-transverse matching must contain $e$. □

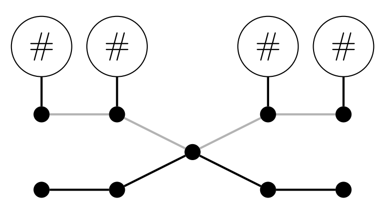



Figure 4.9: Using the forcing gadget

The importance of the forcing gadget can be seen in Figure 4.9. By hanging a clique and an edge off a vertex, we can ensure that no other edge incident with that vertex is contained in an $H$-transverse matching. In Figure 4.9, no grey edge is contained in an $H$-transverse matching.

Using this fact, we can easily alter the variable and clause gadgets from the previous subsection to show that TM($H$) is NP-complete when $H$ is a diameter-4 tree whose only high-degree vertex is its center.

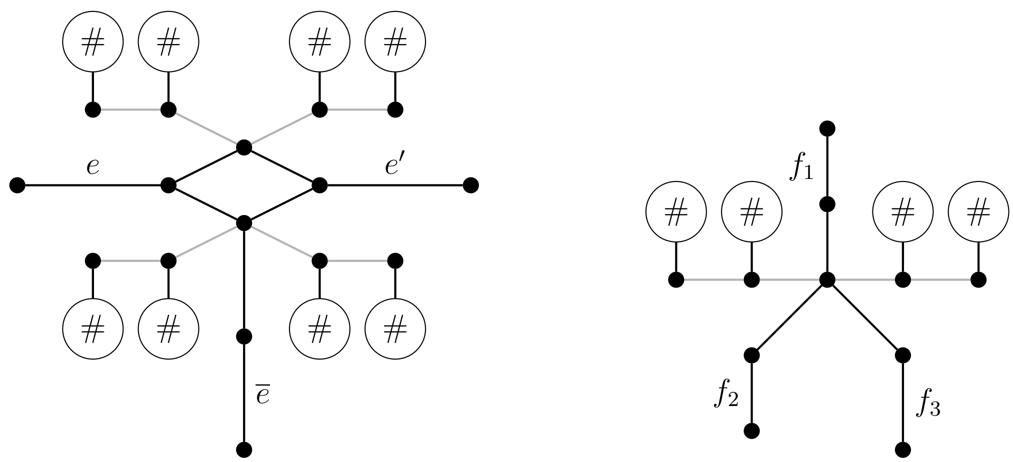



Figure 4.10: Variable and clause gadgets

To be more specific, the variable and clause gadgets are obtained by adding new vertices, edges, and forcing gadgets to the original gadget to turn every relevant copy of $P_5$ into a copy of $H$. See Figure 4.10 for an illustration when $H$ is the tree in Figure 4.7.

The above gadgets satisfy the essential properties used by the proof of the previous subsection: since the edges of the original gadget are the only ones available to an $H$-transverse matching, Claims 2 and 3 in Subsection 4.2.1 still apply. Although the resulting graph is neither planar nor bipartite—due to the large cliques in the forcing gadget—it is still enough to show that TM($H$) is NP-complete for general graphs.

**Lemma 4.14.** *TM(H) is NP-complete when H is a tree of diameter* 4 *whose only* *vertex of degree* $\geq 3$ *is its center.* □

When $H$ is a diameter four tree with a non-center vertex of degree $\geq 3$, the construction of the clause gadget is much the same: vertices, edges, and forcing gadgets are added to $P_5$'s clause gadget to form two intersecting copies of $H$ (see Figure 4.10, right). However, we use a different style of variable gadget, shown in Figure 4.11 (left). White vertices indicate where forcing vertices are attached, such that no grey edge is contained in any $H$-transverse matching.

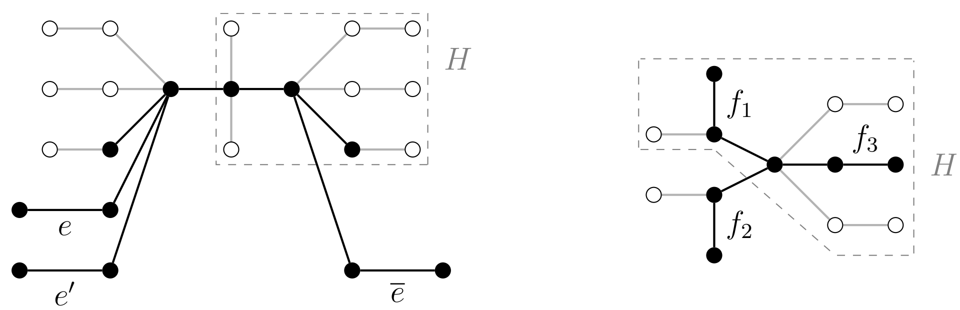



Figure 4.11: More variable and clause gadgets

The variable gadget is based on two copies of $H$ which share their respective non-center vertices of degree $\geq 3$. Forcing gadgets leave only two edges from each copy available for $H$-transverse matchings: one incident with the shared vertex, and one other incident with the center vertex of each copy. Finally, two $P_3$'s (containing $e, e'$) are attached to one of the copies of $H$ and one (containing $\bar{e}$) is attached to the other.

It is easy to to verify that the variable gadget satisfies Claim 2 of the previous section; that is, any $P_5$-transverse matching contains $\bar{e}$ or both $e, e'$. Similarly, the clause gadget satisfies Claim 3. Applying the same proof as in Subsection 4.2.1 shows that TM($H$) is NP-complete.

**Lemma 4.15.** *TM(H) is NP-complete when H is a tree of diameter* 4 *with at most two vertices of degree* $\geq 3$. □

### 4.2.3 $H$ is a diameter-five tree with few high-degree vertices

We now show that TM($H$) is NP-complete when $H$ is a tree of diameter five with at most two vertices of degree $\geq 3$. In fact, we only need consider trees whose high-degree vertices are the two center vertices of the graph.

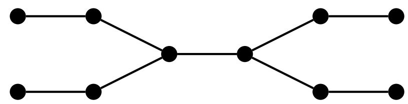


Figure 4.12: A diameter five tree with two high-degree vertices

The gadgets for these cases are essentially the same as the gadgets for the second class of diameter four trees. The primary difference is in the clause gadget, in which an extra vertex (attached to a forcing gadget) is added to $P_5$'s clause gadget before more vertices, edges, and forcing gadgets are added to create two intersecting copies of $H$. Both gadgets are shown in Figure 4.13, with white vertices indicating the attachment of forcing gadgets.

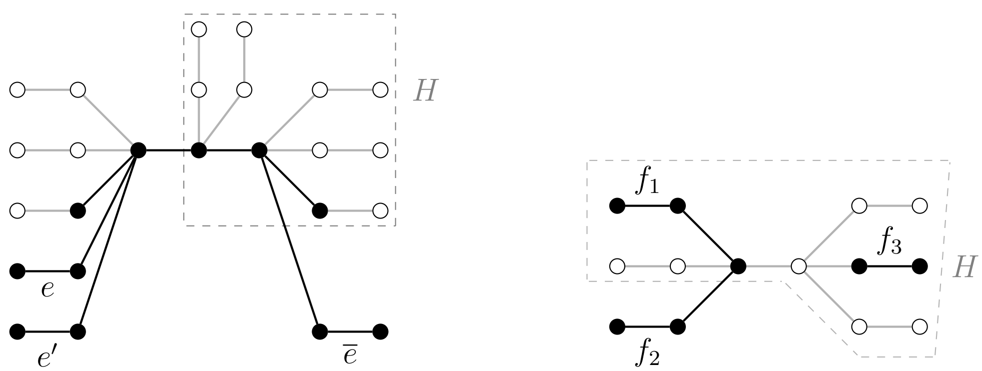



Figure 4.13: Variable and clause gadgets

**Lemma 4.16.** *TM(H) is NP-complete when H is a tree of diameter* 5 *with at most two vertices of degree* $\geq 3$. □

### 4.2.4 $H$ is any other large tree

The previous subsections directly reduce an NP-complete satisfiability problem to various TM($H$). It turns out that we can reduce these "minimal" transverse matching problems to larger ones using forcing gadgets.

**Lemma 4.17.** *Suppose $H$ is a tree which is not a star or a $P_4$. Let $H'$ be obtained from $H$ by deleting one leaf adjacent to each support vertex. If TM($H'$) is NP-complete, then TM($H$) is as well.*

**Proof:** Consider a graph $G'$, and let $G$ be obtained from $G'$ as follows: for each $v \in V(G')$, add a new vertex $v'$, an edge from $v$ to $v'$, and attach a copy of the forcing gadget to $v$. Suppose $M$ is an $H$-transverse matching of $G$. By Lemma 4.13, each extra $vv'$ is not contained in any $H$-transverse matching of $G$. If $G' - M \subseteq G$ contains a copy of $H'$, then adding the extra edges would result in a copy of $H$ in $G - M$. Consequently, any $H$-transverse matching of $G$ restricts to an $H'$-transverse matching of $G'$.

Conversely, any copy of $H$ in $G' \cup \{vv' : v \in V(G')\}$ restricts to a copy of $H'$ in $G'$, as $H'$ is obtained by deleting at most one pendant edge incident with each vertex. Therefore, any $H'$-transverse matching $M'$ in $G'$ can be extended to an $H$-transverse matching in $G$ by adding the pendant edges of the added copies of the forcing gadget.

Since $G$ can be obtained from $G'$ in polynomial (indeed, linear) time, TM($H'$) reduces to TM($H$). If the former is NP-complete, so is the latter. □

We claim that any tree whose diameter is at least four can be "pruned" (by repeatedly deleting one leaf from each support vertex) to obtain a tree considered in one of the above subsections. Indeed, deleting a leaf from each vertex lowers the diameter of the tree by at most two. We therefore need only consider trees of diameter four and five. Pruning a tree of diameter five only decreases the diameter by two when there are at most two vertices of degree $\geq 3$, and these vertices are in the center of the tree; in other words, the tree is as in Subsection 4.2.3. Pruning a tree of diameter four only decreases the diameter when there are at most two vertices of degree $\geq 3$. These cases were treated in Subsections 4.2.1 and 4.2.2. Since all minimal cases have been shown to be NP-complete, Lemma 4.17 implies the following theorem.

**Theorem 4.18.** *TM(H) is NP-complete when $H$ is a tree of diameter $\geq 4$.* □

# Chapter 5

# Concluding remarks

The results of this thesis indicate that graph-transverse matching problems are generally NP-complete. We have shown as much for cycle- and tree-transverse matching problems: TM($H$) is NP-complete when $H$ is a cycle of length $\geq 4$ or a tree of diameter $\geq 4$. However, we have found polynomial-time algorithms for a few graph transverse matching problems, particularly where $H$ is small or the input is restricted.

In this chapter, we suggest some ways our work could be further refined and extended. We present some conjectures about graph-transverse matching problems and point out topics that are worth further study.

## 5.1 Further NP-complete problems

Our proofs can be adapted to show that other TM($H$) are NP-complete. In particular, the proof of Theorem 2.10 is valid for any 2-connected $H$ for which the appropriate variable and clause gadgets can be constructed. (As one would expect, these gadgets do not exist when $H = C_3$). We have a sketch of such a construction for some highly-connected graphs, and believe that they should be possible for sufficiently large 2-connected graphs.

**Conjecture.** *TM(H) is NP-complete if H is a sufficiently large 2-connected graph.*

A monotonicity theorem would be an even nicer result; perhaps TM($H'$) is NP-complete whenever TM($H$) is NP-complete for a subgraph $H$ of $H'$. We do not have enough evidence to confidently present this as a conjecture, but have shown that it holds in some restricted cases (see Lemma 4.17, for example).

In Chapter 2, we showed that TM($C_\ell$) remains NP-complete when restricted to planar graphs for each $\ell \geq 4$. The same is likely true for tree-transverse matching problems.

**Conjecture.** *TM(H) remains NP-complete for the class of planar graphs when H is a tree of diameter* $\geq 4$*.*

In fact, the proofs in Chapter 4—when adapted to reduce from SUBCUBIC PLANAR SAT rather than SUBCUBIC SAT—are almost sufficient to prove this conjecture. The only obstacles to planarity are the forcing gadgets, which contain large cliques. We would be surprised if these gadgets could not be replaced by planar versions.

## 5.2 Further structured graph classes

As we mentioned in Chapter 2, chains of $K_{2,3}$ are an essential part of our reductions for long cycle-transverse matching problems. We would be very interested to see how the complexity of TM($C_\ell$) is affected by restricting the input to graphs of large girth, which do not contain $K_{2,3}$'s.

We showed in Chapter 3 that TM($C_4$) can be solved in polynomial time for chordal bipartite graphs. We believe that different techniques—for example, those in [3, 4]—may be applicable to TM($C_4$) in other restricted graph classes, including the class of graphs with no cycles of length six.

## 5.3 Open problems

In Chapter 4, we found polynomial-time solutions for TM($P_4$) and TM($Y$). However, we have not been able to extend our techniques to solve TM($H$) when $H$ is a diameter 3 tree in general, even though they have simple solutions for triangle-free inputs. These open problems would be very interesting subjects for further study.

## 5.4 Other directions

This thesis has only briefly touched on forbidden subgraph characterizations of graphs which admit a $H$-transverse matching. In Chapter 2, we gave an implicit characterization of the graphs which admit a $C_3$-transverse matching; this could be made more explicit by defining a structure analogous to the "well-formed walks" in [3].

We are aware of an infinite family of minimal chordal bipartite graphs which do not admit $C_4$-transverse matchings. They all have the same tree-like shape, with up to four branches coming off of each point, and each leaf consisting of a $K_{2,3}$. This structure is similar to the minimal non-monopolar chordal graphs, which Stacho constructed in [29] by a graph grammar. We believe that it may be possible to find a forbidden subgraph characterization of this type for chordal bipartite graphs which admit $C_4$-transverse matchings.

Interestingly, a chordal bipartite graph which contains no $K_{2,3}$ always admits a $C_4$-transverse matching.

**Proposition 5.1.** *If $G$ is a chordal bipartite graph which does not contain $K_{2,3}$ as a subgraph, then $G$ has a $C_4$-transverse matching.*

**Proof:** Several authors have pointed out [1,19] that the edges of any chordal bipartite graph $G$ can be ordered $e_1, e_2, \ldots, e_m$ in such a way that each $e_i$ is bisimplicial in the graph $G_i$ induced by the edges $\{e_1, e_2, \ldots, e_i\}$. We claim that a maximal matching $M$ constructed greedily according to this ordering is a $C_4$-transverse matching.

Let $uvwx$ be a $C_4$ in $G$ and let $i$ be the largest index of $uv, vw, wx, xu$ in the above order. Since $e_i$ is bisimplicial in $G_i$, there is a biclique in $G_i$ containing all of the edges adjacent to and ordered before $e_i$. As $ux, vw$ are ordered before $e_i$, but $G$ has no $K_{2,3}$, this biclique is exactly the $C_4 : uvwx$. In particular, $vw, ux$ are the only edges adjacent to $e_i = uv$ which come before it in the ordering. Since $M$ is constructed greedily, $uv \in M$ unless $vw \in M$ or $ux \in M$. In any case, $M$ covers at least one of the edges of $uvwx$. $\square$

This proposition opens up new questions about counting and listing $C_4$-transverse matchings in such graphs, similar to those asked about grid graphs in the study of tatami tilings.

# Bibliography

[1] Andreas Brandstädt, Van Bang Le, and Jeremy P. Spinrad, *Graph Classes: A Survey*, vol. 3 of SIAM Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia, 1999.

[2] Stefan A. Burr, *On the computational complexity of Ramsey-type problems*, in Mathematics of Ramsey Theory, Jaroslav Nešetřil and Vojtěch Rödl, eds., Springer-Verlag, 1990.

[3] Ross Churchley and Jing Huang, *Line-polar graphs: characterization and recognition*, SIAM Journal on Discrete Mathematics, 25 (2011), pp. 1269–1284.

[4] ——, *Partitions via colour-bipartitions*. 2011.

[5] Ross Churchley, Jing Huang, and Xuding Zhu, *Complexity of cycle transverse matching problems*, Lecture Notes in Computer Science, 7056 (2011), pp. 135–143.

[6] Václav Chvátal and Frank Harary, *Generalized Ramsey theory for graphs. III. Small off-diagonal numbers*, Pacific Journal of Mathematics, 41 (1972), pp. 335–346.

[7] Stephen A. Cook, *The complexity of theorem-proving procedures*, in STOC '71 Proceedings of the third annual ACM symposium on Theory of computing, 1971, pp. 151–158.

[8] Jack Edmonds, *Paths, trees, and flowers*, Canadian Journal of Mathematics, 17 (1963), pp. 449–467.

[9] Tinaz Ekim, *Polarity of claw-free graphs*. 2009.

[10] Tinaz Ekim, Pinar Heggernes, and Daniel Meister, *Polar permutation graphs*, Lecture Notes in Computer Science, 5879 (2009), pp. 218–229.

[11] Tinaz Ekim, Pavol Hell, Juraj Stacho, and Dominique de Werra, *Polarity of chordal graphs*, Discrete Applied Mathematics, 156 (2008), pp. 2469–2479.

[12] Tinaz Ekim and Jing Huang, *Recognizing line-polar bipartite graphs in time O(n)*, Discrete Applied Mathematics, 158 (2010), pp. 1593–1598.

[13] Tinaz Ekim, N.V.R. Mahadev, and Dominique de Werra, *Polar cographs*, Discrete Applied Mathematics, 156 (2008), pp. 1652–1660.

[14] Alejandro Erickson, Frank Ruskey, Mark Schurch, and Jennifer Woodcock, *Auspicious tatami mat arrangements*, Lecture Notes in Computer Science, 6196 (2010), pp. 288–297.

[15] Alastair Farrugia, *Vertex-partitioning into fixed additive induced-hereditary properties is NP-hard*, Electronic Journal of Combinatorics, 11 (2004), pp. 1–9.

[16] Stéphane Foldes and Peter L. Hammer, *Split graphs*, in Proceedings of the Eighth Southeastern Conference on Combinatorics, Graph Theory and Com-

puting (Louisiana State Univ., Baton Rouge, La., 1977), Utilitas Mathematica, 1977, pp. 311–315.

[17] Michael R. Garey and David S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, vol. 44 of Books in the Mathematical Sciences, W. H. Freeman and Company, New York, NY, 1979.

[18] John R. Gilbert, Donald J. Rose, and Anders Edenbrandt, *A separator theorem for chordal graphs*, Society for Industrial and Applied Mathematics. Journal on Algebraic and Discrete Methods, 5 (1984), pp. 306–313.

[19] Martin Charles Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, vol. 57 of Annals of Discrete Mathematics, Elsevier, Amsterdam, 2 ed., 2004.

[20] Martin Charles Golumbic and Clinton F. Goss, *Perfect elimination and chordal bipartite graphs*, Journal of Graph Theory, 2 (1978), pp. 155–163.

[21] Jing Huang and Baogang Xu, *A forbidden subgraph characterization of line-polar bipartite graphs*, Discrete Applied Mathematics, 158 (2010), pp. 666–680.

[22] Donald E. Knuth, *The Art of Computer Programming, Volume 4A*, vol. 4A of Art of Computer Programming, Addison-Wesley, 2006.

[23] Yoshiyuki Kotani, *Y2K problem of dominoes and tatami carpeting*, in Puzzlers' Tribute: A Feast for the Mind, David Wolfe and Tom Rodgers, eds., A K Peters, 2002, pp. 413–420.

[24] Van Bang Le and Ragnar Nevries, *Recognizing polar planar graphs using new results for monopolarity*, Lecture Notes in Computer Science, 7074 (2011), pp. 120–129.

[25] David Lichtenstein, *Planar formulae and their uses*, SIAM Journal on Computing, 11 (1982), pp. 329–343.

[26] Matthias Middendorf and Frank Pfeiffer, *On the complexity of the disjoint paths problem*, Combinatorica, 13 (1993), pp. 97–107.

[27] Frank Ruskey. personal communication, 2011.

[28] Marcus Schaefer, *Graph Ramsey theory and the polynomial hierarchy*, Journal of Computer and System Sciences, 8 (2000), pp. 592–601.

[29] Juraj Stacho, *Complexity of generalized colourings of chordal graphs*, PhD thesis, Simon Fraser University, 2008.

[30] Douglas West, *Introduction to Graph Theory*, Prentice Hall, Englewood Cliffs, NJ, 2 ed., 2001.