The Intersection of Longest Paths in a Graph

Sarah Mark

A thesis

submitted in partial fulfilment

of the requirements for the degree

of

Doctor of Philosophy

in

Mathematics

School of Mathematics and Statistics

University of Canterbury

New Zealand

2022

Abstract

In this thesis we examine the famous conjecture that every three longest paths in a graph intersect, and add to the classes of graphs for which it is known that this conjecture holds. This conjecture arose from a question asked by Gallai in 1966, the question of whether all of the longest paths in a graph intersect (Gallai's question). In 1969, Walther found a graph in which the longest paths do not all intersect, answering Gallai's question. Since then, many other graphs in which the longest paths do not all intersect have been found. However there are also many classes of graphs for which the longest paths all intersect, such as series-parallel graphs and dually chordal graphs. Finding such classes of graphs is an active area of research and in this thesis we add to these classes of graphs.

We begin by investigating Gallai's question for a specific class of graphs. A *theta graph* is a graph consisting of three paths with a pair of common endpoints and no other common vertices. A *generalised theta graph* is a graph with at least one block that consists of at least three paths with a pair of common endpoints and no other common vertices. We show that for a subclass of generalised theta graphs, all of the longest paths intersect.

Next, we consider the conjecture that every three longest paths of a graph intersect. We prove that, for every graph with n vertices and at most $n + 5$ edges, every three longest paths intersect.

Finally, we use computational methods to investigate whether all longest paths intersect, or every three longest paths intersect, for several classes of graphs. Two graphs are *homeomorphic* if each can be obtained from the same graph H by a series of subdivisions. We show that, for every simple connected graph G that is homeomorphic to a simple connected graph with at most 7 vertices, all of the longest paths of G intersect. Additionally, we show that, for every simple connected graph G homeomorphic to a simple connected graph with n vertices, $n + 6$ edges, and minimum vertex degree 3, all of the longest paths of G intersect. We then show that for every graph with n vertices and at most $n + 5$ edges, every three longest paths intersect, independently verifying this result. We also present results for several additional classes of graphs with conditions on the blocks, maximum degree of the vertices, and other properties of the graph, showing that every three longest paths intersect or every six longest paths intersect for these graphs.

Acknowledgements

I owe many thanks to my supervisors Jeanette McLeod, Brendan McKay, and Phillip Wilson. Their guidance and wisdom were essential for completing this PhD, and I could not have done this without their enduring support. The lessons I have learned from them extend far beyond mathematics.

My family has always encouraged me in everything I do, and I am extremely thankful for every one of them. Special thanks to my parents Jayne and David for always being there for me, inspiring me to do my best, and being unerringly supportive and loving. I am grateful to my fiancé Louis for taking care of me, pushing me to be the best version of myself, and always loving me.

I am appreciative of the University of Canterbury for funding my research, and of the School of Mathematics and Statistics for their support. My thanks to the other postgraduate students for bringing me out of my shell; my life as a PhD student has been all the richer for your friendship.

Sarah Mark

Contents

Abstract **1**

Acknowledgements **3**

1 Introduction **8**

1.1 Graphs 8

1.1.1 Subgraphs 9

1.1.2 Walks, paths, and cycles 9

1.1.3 Connectivity 11

1.1.4 Graph classes 11

1.2 Overview of thesis 12

1.3 Intersections of longest paths of a graph 15

1.3.1 Sets of two longest paths 16

1.3.2 Sets of three longest paths 16

1.3.3 All longest paths 17

1.3.4 Related results 20

1.3.4.1 Longest path transversal 22

1.3.4.2 A measure of the distance between longest paths 22

1.3.4.3 Vertices that are not in the vertex set of at least one longest path 23

1.4 Thesis structure 23

2 Generalised theta graphs **25**

2.1 Introduction 25
2.1.1 Trees and the Helly property 27
2.2 Graphs with more than one core 29
2.3 Graphs with a unique core 29
2.3.1 Two types of longest paths 30
2.3.2 Case (1): paths of Type (A) 30
2.3.3 Case (2): paths of Type (A) and Type (B) 31
2.3.4 Case (3): paths of Type (B) 32
2.3.4.1 Forbidden configurations of Type (B) longest paths 34
2.3.4.2 Properties of paths of Type (B) 38
2.3.4.3 Proof of Lemma 2.10 46
2.3.5 Proof of Proposition 2.5 47
2.4 Proof of Theorem 2.1 47
2.5 Concluding remarks 47
3 Graphs with cyclomatic number 6 **50**
3.1 Introduction 50
3.2 Preliminaries 51
3.2.1 The mod 2 sum operation 51
3.2.2 Cycle decomposition of a graph 52
3.2.3 Independent cycles 55
3.2.4 Graph colouring 55
3.3 Properties of a minimal counterexample 57
3.3.1 Forbidden configurations in $\check{H}$ 58
3.3.2 The number of cycles of $\check{H}$ 60
3.4 Preliminary results on bi-coloured cycles 61
3.5 Independence of cycles of $\check{H}$ 63
3.5.1 Sets of cycles of $\check{H}$ 63
3.5.2 The mod 2 sum of two cycles 64

3.5.3 The mod 2 sum of three cycles . . . 68
3.5.4 The mod 2 sum of four cycles . . . 72
3.5.5 The mod 2 sum of five cycles . . . 76
3.5.6 Independent tri-coloured cycle . . . 77
3.5.7 Proof of Theorem 3.1 . . . 81
3.6 Concluding Remarks . . . 82

4 Computational Investigations **84**
4.1 Introduction . . . 84
4.2 Theory . . . 85
4.2.1 Homeomorphic graphs . . . 86
4.2.2 Maximal walks and paths . . . 86
4.2.3 Polyhedra . . . 91
4.2.4 An optimisation . . . 94
4.2.5 Worked example . . . 95
4.3 Method of computation . . . 98
4.3.1 Overview . . . 98
4.3.2 Generating graphs . . . 99
4.3.3 Finding maximal walks . . . 100
4.3.4 Calculating the polyhedron . . . 100
4.3.5 Finding intersections of paths . . . 100
4.4 Results . . . 102
4.5 Linear programming . . . 104
4.5.1 Linear program . . . 104
4.5.2 Properties of a reduced graph . . . 106
4.5.3 Results . . . 110
4.5.3.1 Three longest paths . . . 110
4.5.3.2 Six longest paths . . . 111
4.6 Concluding remarks . . . 112

5 Conclusion **114**

5.1 Results 114

5.2 Methods and extensions 116

Appendices **120**

A Graph classes for which every graph has a Gallai vertex **121**

B Python program maximalwalks.py **124**

B.1 Program code 124

B.2 Sample input 128

B.3 Sample output: file for lrs 129

C Python program postprocessing.py **130**

C.1 Program code 130

C.2 Sample input: file from lrs 133

C.3 Sample output 135

Bibliography **136**

Chapter 1

Introduction

Over 50 years ago, Gallai asked whether the longest paths in a graph intersect [14] and, since then, many related questions have been posed. For example, it has been conjectured that every three longest paths in a graph intersect. While it is easy to show that every two longest paths in a connected graph intersect the problem of three longest paths remains open. Not all graphs have the property that all of the longest paths intersect, but there are classes of graphs for which it has been shown that all of the longest paths intersect. In this thesis, we add to the classes of graphs for which every three longest paths intersect, and to the classes of graphs for which all longest paths intersect.

1.1 Graphs

A *graph* G is a nonempty set $V(G)$ of *vertices* together with a (possibly empty) set $E(G)$ of unordered pairs of vertices of G, called *edges*. For an edge $e = \{u, v\}$ of G, where $u, v \in V(G)$, we write $e = uv$ (or vu). Vertices u and v are the *endpoints* of edge e. These endpoints u and v are said to be *incident* with edge e and e is *incident* with u and v. Two vertices $u, v \in V(G)$ incident with an edge $e \in E(G)$ are *adjacent*, and, similarly, two edges $e, f \in E(G)$ incident with a vertex $v \in V(G)$ are *adjacent*. The *degree* of v is the number of edges with which it is incident, denoted $\deg_G(v)$. A *leaf* of G is a vertex of G with degree one. A *loop* of G is an edge $e = uu$ where $u \in V(G)$. For two vertices $u, v \in V(G)$, if there are two or more edges of G that each have endpoints u and v, then these edges are *parallel edges*.

A *simple graph* is a graph with no loops or parallel edges. Hereafter, all graphs are assumed to be simple unless otherwise stated.

Two graphs G and G' are *isomorphic*, written $G \cong G'$, if there is a bijection $\phi : V(G) \to V(G')$ such that $uv \in E(G)$ if and only if $\phi(u)\phi(v) \in E(G')$.

For two graphs G and G', we denote by $G \cup G'$ the graph with vertex set $V(G) \cup V(G')$ and edge set $E(G) \cup E(G')$. Similarly, we denote by $G \cap G'$ the graph with vertex set $V(G) \cap V(G')$ and edge set $E(G) \cap E(G')$.

A *graph labelling* is the assignment of an identifier, or *label*, to each edge or vertex of a graph. A *labelled graph* is a graph for which each vertex has a unique label. A *partially labelled graph* is a graph for which some, but not necessarily all, vertices are labelled, and these labels are unique.

1.1.1 Subgraphs

A *subgraph* G' of a graph G is a graph such that $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$. If $E(G')$ is the set of edges of G whose endpoints are vertices in $V(G')$, then G' is an *induced subgraph* of G. If $V(G')$ is the set of vertices of G that are endpoints of the edges in $E(G')$ then G' is an *edge-induced subgraph* of G. A *spanning subgraph* is a subgraph G' of G such that $V(G') = V(G)$. For $v \in V(G)$, the subgraph of G obtained by deleting the vertex v and its incident edges is called a *vertex-deleted subgraph* of G, denoted $G - v$.

For a set $\mathcal{S}$ of subgraphs of G, if there exists a vertex $v \in V(G)$ such that $v \in V(S)$ for each subgraph $S \in \mathcal{S}$, then the subgraphs in $\mathcal{S}$ have a *common vertex*.

1.1.2 Walks, paths, and cycles

Walks, paths, and cycles are central to this thesis. A *walk* W of a graph G is a non-empty, finite sequence of vertices $W = v_1v_2 \dots v_k$ $(k \geq 1)$ for which $v_i \in V(G)$ for every i $(1 \leq i \leq k)$ and $v_jv_{j+1} \in E(G)$ for each j $(1 \leq j \leq k-1)$. The *length* of W, $|W|$, is $k-1$. We define $V(W) = \{v_1, v_2, \dots, v_k\}$ and $E(W) = \{v_jv_{j+1} \in E(G) : 1 \leq j \leq k-1\}$. We do not distinguish between walks $v_1v_2 \dots v_{k-1}v_k$ and $v_kv_{k-1} \dots v_2v_1$. However, we may specify the order of a subset of the vertices of a walk, for example, for a walk W we may say that vertices $x, y, z \in V(W)$ are in the order x, y, z in W. This means that, if

$W = v_1v_2 \dots v_{k-1}v_k$, then $x = v_i$, $y = v_j$, and $z = v_\ell$ where $1 \le i \le j \le \ell \le k$. Two walks W_1 and W_2 of a graph are *vertex-disjoint* if $V(W_1) \cap V(W_2) = \emptyset$ and are *edge-disjoint* if $E(W_1) \cap E(W_2) = \emptyset$.

A *path* of G is a walk whose vertices are distinct. For a path $P = v_1v_2 \dots v_k$ of G, vertices v_1 and v_k are the *endpoints* of P, and $v_2, \dots, v_{k-1}$ are *internal vertices*. We call P a v_1v_k*–path*. A *subpath* of a path $P = v_1v_2 \dots v_k$ is a path $v_iv_{i+1} \dots v_j$ where $1 \le i \le j \le k$, denoted v_iPv_j. Two paths P and Q of G are *internally disjoint* if P and Q do not have any common vertices that are internal vertices of P or Q. This definition extends in the natural way to more than two paths — for a set $\mathcal{P}$ of paths of G, these paths are *internally disjoint* if every pair of paths in $\mathcal{P}$ are internally disjoint. A spanning path of G is called a *Hamiltonian path*.

A *longest path* of G is a path of maximum length. Some graphs have only one longest path while others have many. For example, the graph in Figure 1.1(a) has only one longest path, while the graph in Figure 1.1(b) has 60 longest paths. When we have said that a set of longest paths of a graph *intersect*, we mean that these paths have at least one common vertex. In the rest of this thesis, we use the latter terminology. Longest paths are central to this thesis.

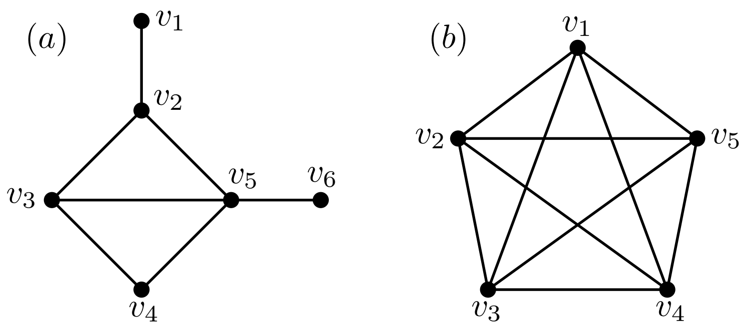

Figure 1.1: (a) A graph with only one longest path, $v_1v_2v_3v_4v_5v_6$. (b) The graph K_5, which has 60 longest paths.

For two paths $P = v_1v_2 \dots v_i$ and $Q = v_iv_{i+1} \dots v_j$ $(1 \le i < j)$ of G that have a common endpoint v_i but have no other common vertices, the *concatenation* of these two paths is the path $PQ = v_1v_2 \dots v_iv_{i+1} \dots v_j$. We use similar notation for the concatenation of two internally disjoint subpaths xRy and ySz of paths R and S (respectively) of G: if $xRy \cup ySz$ is a path, we denote the concatenation of these paths by $xRySz$.

A *cycle* of G is a walk $v_1v_2\dots v_k$ of length at least three where $v_1 = v_k$ but $v_1, \dots, v_{k-1}$ are distinct. If C_1 and C_2 are two cycles of G with $E(C_1) = E(C_2)$, then $C_1 = C_2$. A graph G is *Hamiltonian* if it has a cycle C where $V(C) = V(G)$. A *tree* is a connected graph with no cycles. A *forest* is a graph in which every connected component is a tree. Cycles are central to proving the main result of Chapter 3 of this thesis.

1.1.3 Connectivity

A graph G is *connected* if, for every pair of vertices $u, v \in V(G)$, there exists a uv–path of G. Every maximal connected subgraph of a graph G is a *(connected) component* of G. A graph G that is not connected is *disconnected*. A vertex $v \in V(G)$ is a *cut vertex* if deleting v increases the number of connected components of G. The associated notion for edges is that of a *bridge*, which is an edge $e \in E(G)$ whose deletion increases the number of connected components of G. A *bond* of a connected graph G is a minimal set $E' \subseteq E(G)$ such that deleting the edges in E' from G results in a disconnected graph. A *block* of G is a maximal connected subgraph with no cut vertices. A block consisting of a vertex with degree zero or an edge and its endpoints is a *trivial block*, and all other blocks are *non-trivial*. A graph G is *k-connected* if it has at least $k + 1$ vertices and remains connected when any set of fewer than k vertices is deleted.

1.1.4 Graph classes

In this section, we define a number of classes of graphs which will be used later in this chapter when we discuss for which classes all of the longest paths have a common vertex.

A *planar graph* is a graph that can be drawn on the plane such that no two edges meet at a point other than a common endpoint. A *complete graph* is a graph for which every two vertices are adjacent. A *block graph* is a graph in which every block is a complete graph. A graph is *bipartite* if its vertex set can be partitioned into two sets V_1 and V_2 such that every edge of the graph has one endpoint that is in V_1 and one endpoint that is in V_2. A *cactus* is a graph in which every non-trivial block is a cycle. A *chordal graph* is a graph for which every cycle of length four or more has an edge that is not an edge of the cycle but whose endpoints are vertices of the cycle. A *series-parallel graph* is a (not necessarily simple) graph with two distinguished vertices s and t that can be reduced to the edge st by a sequence of the following two operations: (i) replacement of a pair of parallel edges by a single edge with the same pair of endpoints, and (ii) replacement of a pair of edges xy and yz, where y has degree two, $s \neq y$, and $t \neq y$, by a single edge xz. The graph $2K_2$ is the disconnected graph consisting of two edges that do not have a common vertex, and P_4 is the path with four vertices.

Let $\{G_1, G_2, \ldots\}$ be a set of graphs. A graph is called $(G_1, G_2, \ldots)$-*free* if it has no induced subgraph isomorphic to a graph in $\{G_1, G_2, \ldots\}$. A *cograph* is a P_4-free graph.

A *clique* is a set of vertices of G such that every pair of vertices in the set is adjacent. A *stable set* is the complementary concept, namely, a set of vertices of G such that no pair of vertices in the set is adjacent. A *split graph* is a graph for which the vertices can be partitioned into a clique and a stable set. The *stability number* of a graph is the maximum size of a stable set of the graph.

An *intersection graph* is a graph constructed from a set $\mathcal{S}$ of sets, where each vertex of the graph corresponds to a set in $\mathcal{S}$, and two vertices are adjacent if and only if the corresponding sets have nonempty intersection. A *circular arc graph* is the intersection graph of a set of arcs on a circle. A *dually chordal graph* is the intersection graph of the maximal cliques of a chordal graph.

Further terms will be defined as required.

1.2 Overview of thesis

In 1966, at a graph theory colloquium held in Tihany, Hungary, Gallai [14] asked the following question:

Question 1. *(Gallai's question) Do all of the longest paths of a connected graph have a common vertex?*

Consequently, such a vertex (one that is in all of the longest paths of a graph) is called a *Gallai vertex*. For example, in Figure 1.2, which shows an example of a graph and two of its longest paths, vertices v_2 and v_5 are Gallai vertices. Gallai's question was answered in the negative in 1969 when Walther [38] found a connected graph on 25 vertices, and with 70 longest paths, that does not have a Gallai vertex.

Although Gallai's question has been answered in the negative, there are graphs that do have a Gallai vertex. It is then natural to ask: how can graphs with a Gallai vertex be characterised? This question has been considered by many mathematicians, who have shown that there are several classes of graphs

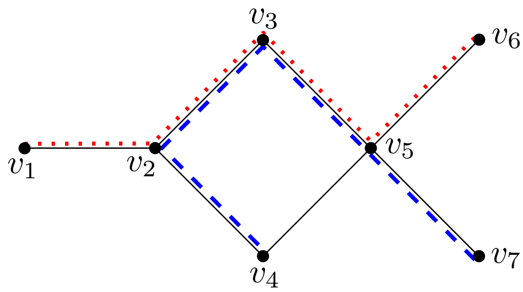

Figure 1.2: A graph and two of its longest paths, $v_1v_2v_3v_5v_6$ (red dotted lines) and $v_4v_2v_3v_5v_7$ (blue dashed lines).

for which each graph has a Gallai vertex, as detailed in Section 1.3.3.

We ask a related question, paraphrased from Kensell [25] and Zamfirescu [44]: for an integer k ($k \geq 2$), do every k longest paths of a connected graph have a common vertex? In 1975, Schmitz [31] found a planar graph on 17 vertices, shown in Figure 1.3, in which there is a set of only seven longest paths that do not have a common vertex, showing that the answer is the negative for $k = 7$. In 1996, Skupien [33] proved that, for every k where $k \geq 7$, there is a graph that answers this question in the negative. On the other hand, we know that the answer is the positive for $k = 2$ (see Lemma 1.1 in Section 1.3.1). However, for k such that $3 \leq k \leq 6$, the answer remains unknown. In particular, the question of whether every three longest paths of a graph have a common vertex, attributed to T. Zamfirescu [17], has occupied the minds of many mathematicians for over 30 years. This question remains unanswered, though the following conjecture has been posed independently by Harris et al. [18, p. 69] and Kensell [25]. This conjecture is known to hold for certain classes of graphs, including those for which every graph has a Gallai vertex.

Conjecture 1. *Every three longest paths of a connected graph have a common vertex.*

This thesis adds to the classes of graphs for which it is known that Conjecture 1 holds and to the classes of graphs for which each graph has a Gallai vertex.

First, we investigate a specific class of graphs and prove that each graph in this class has a Gallai vertex. This class of graphs is described below, and detailed further in Section 2.1. A *theta graph* is a graph consisting of three internally disjoint paths with a common pair of distinct endpoints, an example of which is shown in Figure 1.4(i). A *generalised theta graph* is a graph G with at least one non-trivial

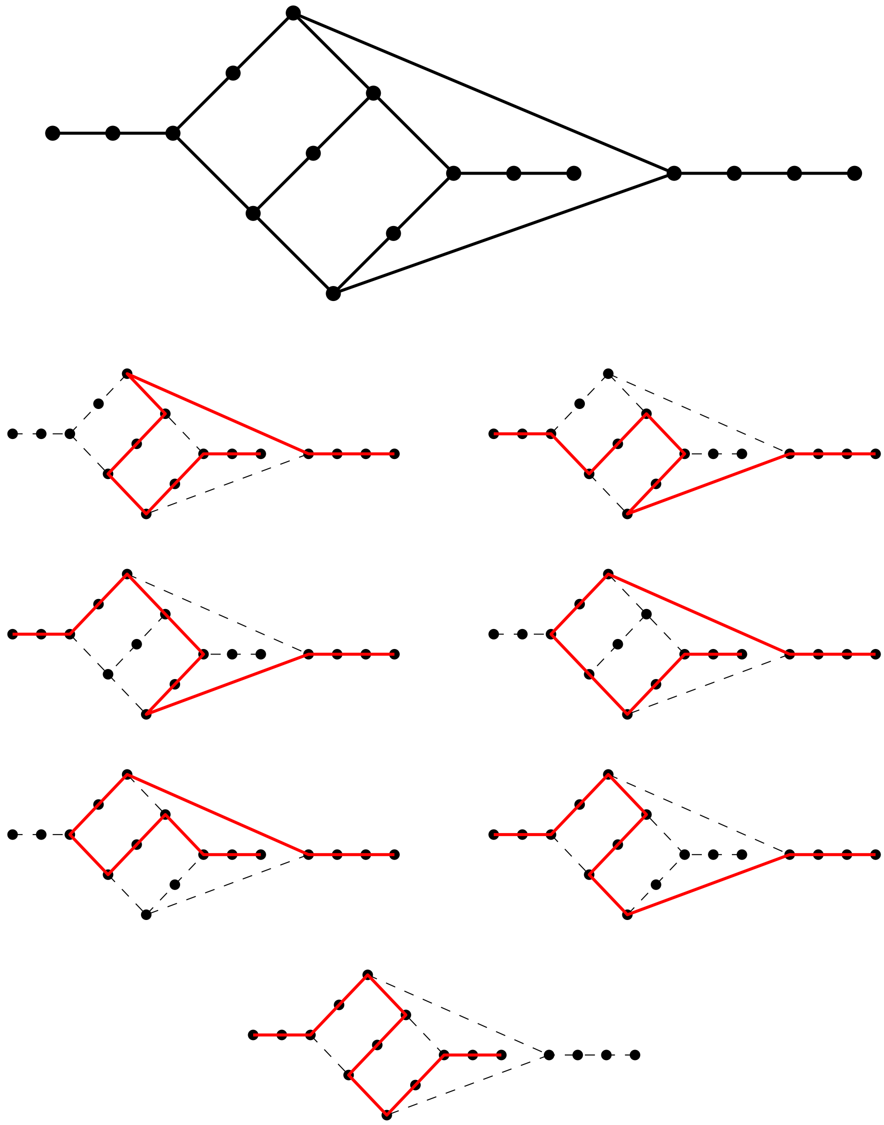

Figure 1.3: A planar graph on 17 vertices that does not have a Gallai vertex, found by Schmitz [31], and a set of seven longest paths of the graph (shown with red solid lines, other graph edges shown by black dashed lines) that do not have a common vertex. Observe that each vertex of the graph is excluded by at least one of these seven longest paths.

block consisting of $k \geq 3$ internally disjoint paths with a common pair of distinct endpoints. An example of a generalised theta graph is shown in Figure 1.4(ii). We investigate a subclass of generalised theta graphs and prove that every graph in this class has a Gallai vertex.

Next, we consider Conjecture 1 for graphs with n vertices and up to $n+k$ edges for small k. We prove that, for every graph with n vertices and at most $n+5$ edges, every three longest paths have a common

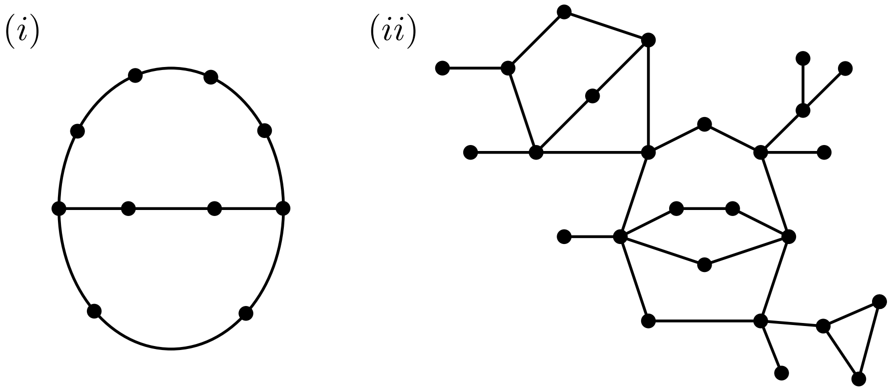

Figure 1.4: (i) A theta graph and (ii) A generalised theta graph.

vertex. This result also shows that a minimal (with respect to edges) counterexample to Conjecture 1 with n vertices has at least $n+6$ edges, adding the the known properties of such a graph.

Finally, we use computational methods to obtain results for Gallai's question and Conjecture 1 for specific classes of graphs. The *subdivision* of an edge uv of a graph G is the operation of deleting uv and inserting a new vertex w and edges uw and wv. A graph obtained by performing a sequence of subdivisions on edges of G is called a *subdivision* of G. Two graphs G and G' are *homeomorphic* if and only if there exists a subdivision of G that is isomorphic to a subdivision of G'. We present a computational method of approaching Gallai's question and obtain the following two results. We show that, for every simple connected graph G that is homeomorphic to a simple connected graph with at most 7 vertices, G has a Gallai vertex. We also show that every simple connected graph G homeomorphic to a graph with n vertices, $n+6$ edges, and minimum vertex degree 3 has a Gallai vertex. We then use an alternative computational method to show that for every graph with n vertices and at most $n+5$ edges, every three longest paths have a common vertex, independently verifying the result in Chapter 3. We also present two additional classes of graphs for which every three longest paths have a common vertex, and two classes of graphs for which every six longest paths have a common vertex.

1.3 Intersections of longest paths of a graph

In this section we examine the literature on Conjecture 1 and Gallai's question, including graphs that do not have a Gallai vertex. We also survey the literature on several closely related questions.

1.3.1 Sets of two longest paths

We state the result that every two longest paths of a connected graph have a common vertex, and provide a proof. The proof given here illustrates a method that will be used frequently in Chapter 2.

Lemma 1.1. *Every two longest paths of a connected graph have a common vertex.*

Proof. Let P and Q be two longest paths of G. Assume that P and Q do not have a common vertex. Then there exist internal vertices $u \in V(P)$ and $v \in V(Q)$, and a uv–path R of G of length at least one such that $V(P) \cap V(R) = \{u\}$ and $V(Q) \cap V(R) = \{v\}$. Let P_1 and P_2 be the two subpaths of P such that u is an endpoint of both P_1 and P_2, and $P = P_1P_2$. Similarly, let Q_1 and Q_2 be the two subpaths of Q such that v is an endpoint of both Q_1 and Q_2, and $Q = Q_1Q_2$. Let $R_1 = P_1RQ_1$ and $R_2 = P_2RQ_2$. Since P is a longest path of G,

$$\begin{aligned} |P| &\geq |R_1| \\ |P_1P_2| &\geq |P_1RQ_1| \\ |P_2| &\geq |R| + |Q_1| \end{aligned} \tag{1.1}$$

Similarly, since Q is a longest path of G,

$$\begin{aligned} |Q| &\geq |R_2| \\ |Q_1Q_2| &\geq |P_2RQ_2| \\ |Q_1| &\geq |P_2| + |R| \end{aligned} \tag{1.2}$$

From 1.1 and 1.2, we have $|R| \leq 0$, a contradiction. Therefore P and Q have a common vertex. $\square$

1.3.2 Sets of three longest paths

Of particular interest to this thesis is Conjecture 1, that every three longest paths of a graph have a common vertex. While it is easy to show that every two longest paths of a graph have a common vertex (Lemma 1.1), determining whether every three longest paths of a graph have a common vertex is far more difficult. As discussed in Section 1.2, this question has not yet been answered.

Conjecture 1 holds for each graph that has a Gallai vertex. Such classes of graphs are discussed in Section 1.3.3 and Appendix A. Additionally, it has been shown that Conjecture 1 holds for the class of graphs for which all nontrivial blocks are Hamiltonian [10], though it is not known whether all of the graphs in this class have a Gallai vertex. Figure 1.5 shows a number of classes of graphs for which Conjecture 1 holds (shown in green) and classes for which it is currently unknown whether Conjecture 1 holds (grey), along with a selection of subclass to superclass relationships between these classes. Note that this diagram does not include all relationships of the classes shown. We refer the reader to [5] for the definitions of classes of graphs in Figure 1.5 that are not defined in this chapter or Appendix A.

Properties of a counterexample

One approach to proving Conjecture 1 is to assume that there exists a minimal counterexample and to deduce results about its structure, aiming to obtain a contradiction. We outline a number of results from Axenovich [2] and Kensell [25] about the structure of such a counterexample.

Let $\check{H}$ be a minimal (with respect to edges) counterexample to Conjecture 1. Then $\check{H}$ is a connected graph in which there exist three longest paths, P_0, P_1, and P_2, that do not have a common vertex. First observe that, since $\check{H}$ is minimal with respect to edges, $\check{H} = P_0 \cup P_1 \cup P_2$. As noted by Kensell [25], it follows that $\deg_{\check{H}}(v) \leq 4$ for each vertex v of $\check{H}$, since each of the three longest paths P_0, P_1, and P_2 has at most two edges incident with v, and v is a vertex of at most two of P_0, P_1, and P_2. Kensell additionally proved that $\check{H}$ does not have a subgraph that is a cycle of length 3, among other more technical results, the details of which can be found in [25]. Axenovich [2] proved that each union of any two of the paths P_0, P_1, and P_2 has at least two subgraphs that are cycles, a key result which will be used to prove the main result in Chapter 3. Axenovich [2] describes two configurations that are forbidden in $\check{H}$, and Kensell [25] describes a third; we refer the reader to the cited papers for details.

1.3.3 All longest paths

We first consider graphs that do not have a Gallai vertex, and then provide an overview of classes of graphs for which each graph has a Gallai vertex.

T. Zamfirescu [43] and Walther and Voss [39] independently found a graph with only 12 vertices, shown in Figure 1.6, that does not have a Gallai vertex. This was confirmed to be the graph with the fewest vertices that does not have a Gallai vertex when Brinkmann and van Cleemput [see 32] exhaustively

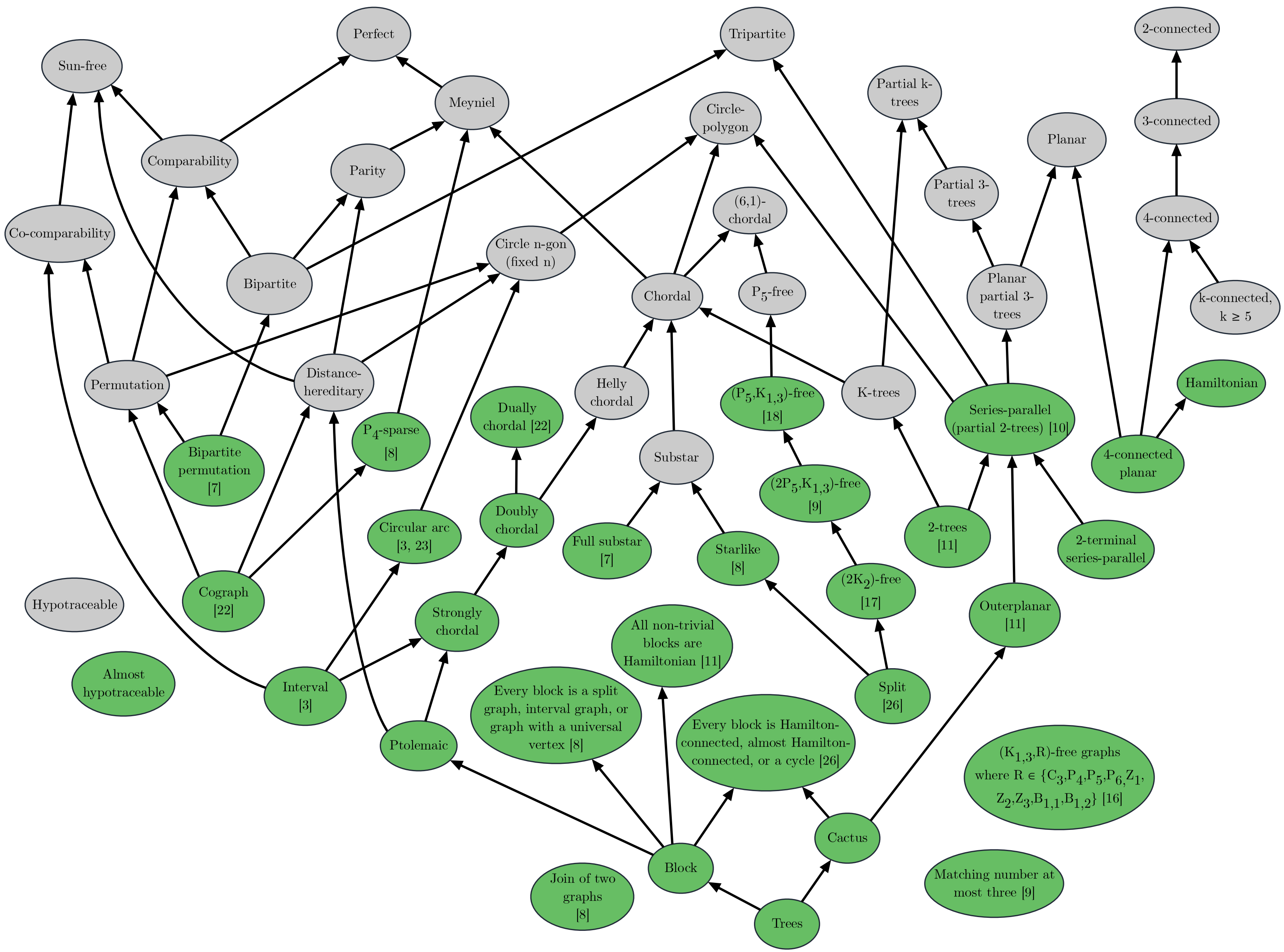

Figure 1.5: A diagram showing a number of graph classes for which Conjecture 1 holds (green) and classes for which it has not yet been determined whether Conjecture 1 holds (grey). Arrows indicate subclass to superclass relationships. This diagram does not include all such relationships of the classes shown.

checked that every graph with at most 11 vertices does not have a Gallai vertex. Additionally, this graph has been confirmed to be unique by McKay [28], who checked all graphs on 12 vertices and found that this is the only graph on 12 vertices with no Gallai vertex.

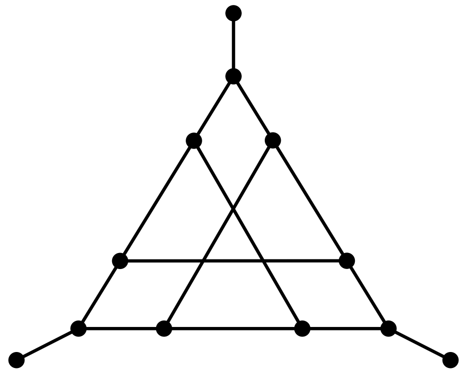

Figure 1.6: A graph with 12 vertices, 15 edges, and 42 longest paths that does not have a Gallai vertex, found independently by T. Zamfirescu [43] and Voss and Walther [39].

The smallest known planar graph that does not have a Gallai vertex, shown in Figure1.3, was found in 1975 by Schmitz [31] — a planar graph on 17 vertices in which there exists a set of seven of the 15 longest paths that do not have a common vertex, and hence the graph does not have a Gallai vertex. This graph is also bipartite, and therefore there are both planar graphs and bipartite graphs that do not have a Gallai vertex.

There exist infinite families of graphs such that each graph in the family does not have a Gallai vertex. One such family of graphs is that of *hypotraceable graphs.* A graph G is hypotraceable if it does not have a Hamiltonian path, but every vertex-deleted subgraph of G has a Hamiltonian path. Hence, for every vertex v of a hypotraceable graph G, there exists a longest path P in G such that $v \notin V(P)$. Therefore, there exists no vertex of G that is in every longest path, and hence every graph in the family of hypotraceable graphs does not have a Gallai vertex. The set of hypotraceable graphs was shown to be an infinite family by Thomassen [35].

A number of classes of graphs have been found for which every graph has a Gallai vertex, including trees ([see 10, 32]), series-parallel graphs [9], and circular arc graphs [3, 23]. Although there are planar graphs that do not have a Gallai vertex [31], 4-connected planar graphs have a Gallai vertex (since such graphs have a Hamiltonian path) [10]. It has also been shown that dually chordal graphs and connected cographs have a Gallai vertex [22], along with $2K_2$-free graphs [16]. A list of these results and others is given in Appendix A.

Figure 1.7 shows a number of classes of graphs for which every graph has a Gallai vertex (shown in green), classes for which it is currently unknown whether every graph has a Gallai vertex (grey), and classes for which there exists a graph that does not have a Gallai vertex (orange), along with a selection of subclass to superclass relationships between these classes. Note that this diagram does not include all relationships of the classes shown. As before, we refer the reader to [5] for the definitions of classes of graphs in Figure 1.5 that are not defined in this chapter or Appendix A.

Note that if a graph has a Gallai vertex, then every subset of the longest paths of the graph also have a common vertex. Hence, for the classes of graphs mentioned above that have a Gallai vertex, and those listed in Appendix A, every k longest paths have a common vertex for every integer $k \geq 1$ (where k is at most the number of longest paths of the graph).

1.3.4 Related results

We provide an overview of several results and questions related to Gallai's question.

For certain classes of graphs for which it is not known whether each graph has a Gallai vertex, results have been proved about the existence of a subgraph that has a nonempty intersection with every longest path of the graph. A *directed graph* D is a nonempty set $V(D)$ of vertices together with a set $A(D)$ of ordered pairs of vertices of D, called *arcs*. Havet [20] proved that, for every directed graph with stability number at most two, there exists a stable set S such that at least one vertex of each longest path of the graph is in S. For chordal graphs, Balister et al. [3] claimed (without proof) that there exists a clique Q such that at least one vertex of each longest path of the graph is in Q. Wei et al. [41] proved that every connected graph has a bond E such that at least one edge of each longest path of the graph is in E. Klavžar and Petkovšek [26] proved that G has a Gallai vertex if and only if, for each block B of G, the longest paths of G with at least one edge that is an edge of B have a common vertex. Later, de Rezende et al. [10] claimed, without proof, that the proof of Klavžar and Petkovšek's result implies the stronger result that, for a subset $\mathcal{P}$ of the longest paths of G, if the paths of $\mathcal{P}$ do not have a common vertex, then there exists a block B of G such that $E(B) \cap E(P) \neq \emptyset$ for each path $P \in \mathcal{P}$.

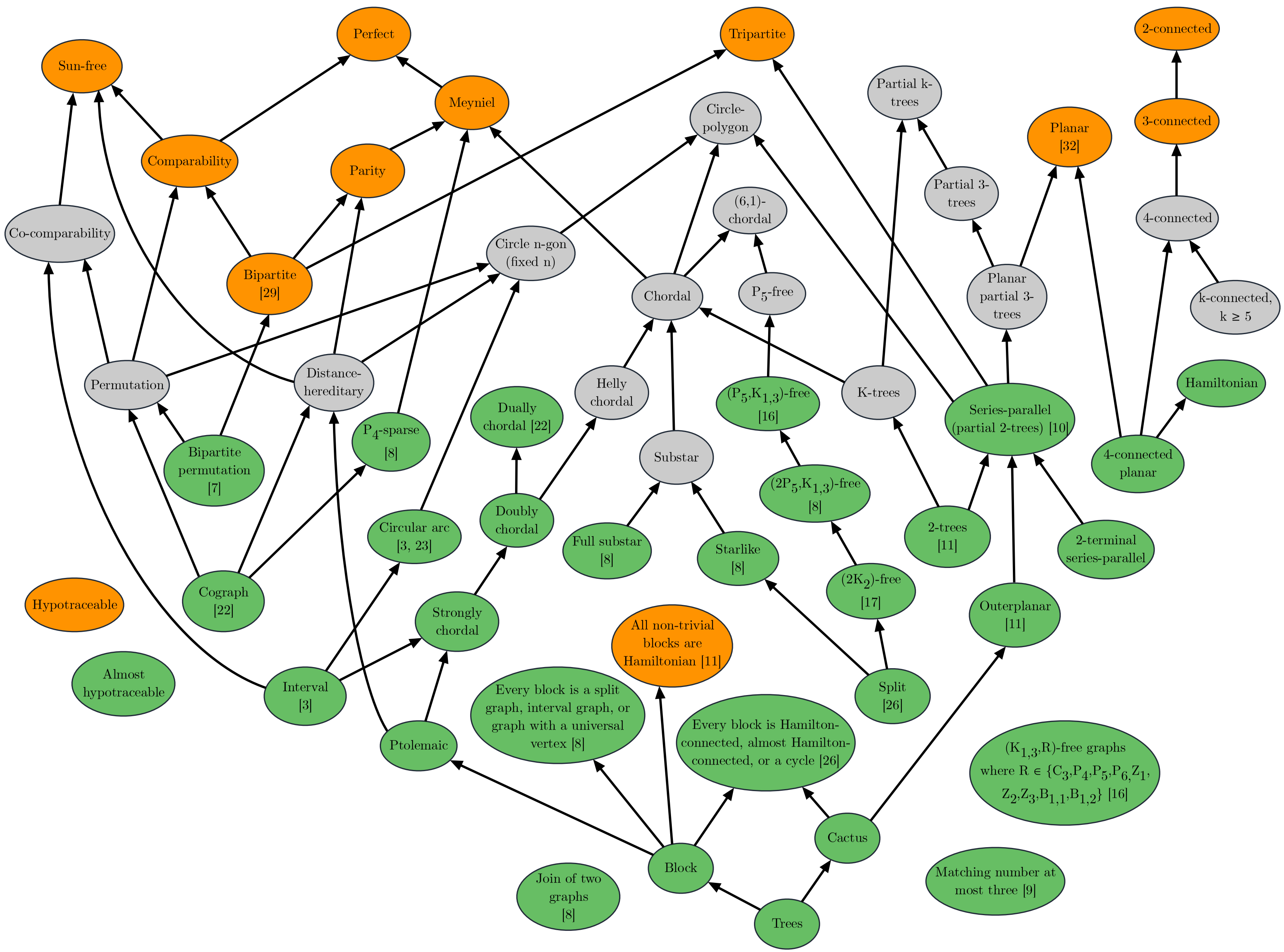

Figure 1.7: A diagram showing a number of graph classes for which every graph has a Gallai vertex (green), classes for which it has not yet been determined whether every graph has a Gallai vertex (grey), and classes for which there exists a graph with no Gallai vertex (orange). Arrows indicate subclass to superclass relationships. This diagram does not include all such relationships of the classes shown.

1.3.4.1 Longest path transversal

A *longest path transversal* of a graph G is a set $S \subseteq V(G)$ such that, for every longest path P of G, $S \cap V(P) \neq \emptyset$. The minimum size of a longest path transversal of a graph G is denoted by $lpt(G)$. Rautenbach and Sereni [30] asked (paraphrased): *What is the smallest integer j ($j \geq 1$) such that, for every connected graph G, $lpt(G) \leq j$?* We know that $j \neq 1$ since there exist graphs that do not have a Gallai vertex, and, additionally, we know that $j \neq 2$ [40]. Rautenbach and Sereni presented an upper bound on j based on the number of vertices of G, proving that if G is a connected graph with n vertices then $\mathrm{lpt}(G) \leq \left\lceil \frac{n}{4} - \frac{n^{2/3}}{90} \right\rceil$. Wei et al. [41] show that, for a connected graph G, $\mathrm{lpt}(G) \leq \max |C|$, where C is a bond of G.

It is natural to ask whether this bound can be reduced when restricted to a specific class of graphs. Rautenbach and Sereni [30] proved, in 2014, that if G is planar and $n \geq 2$, then $\mathrm{lpt}(G) \leq 9\sqrt{n}\log(n)$. In 2020, Cerioli et al. [6] showed that if G is a chordal graph then $\mathrm{lpt}(G) \leq \max\{1, \omega(G) - 2\}$ where $\omega(G)$ is the size of a largest clique in G. Harvey and Payne [19] showed that $\mathrm{lpt}(G) \leq 4\left\lceil \frac{\omega(G)}{5} \right\rceil$ for chordal graphs, which improves the result of Cerioli et al. when $\omega(G) \geq 27$ or $\omega(G) \in \{15, 19, 20, 23, 24, 25\}$. Additionally, for 2-connected chordal graphs, Harvey and Payne proved that $\mathrm{lpt}(G) \leq 2\left\lceil \frac{\omega(G)}{3} \right\rceil$, which is an improvement on the result of Cerioli et al. when $\omega(G) \geq 11$ or $\omega(G) = 9$.

1.3.4.2 A measure of the distance between longest paths

Fujita et al. [12] introduced a measure of the distance between longest paths in a graph, which they call $f(G, \mathcal{P})$ for a graph G and a subset $\mathcal{P}$ of the longest paths of G. With this measure, $f(G, \mathcal{P}) = 0$ if and only if the paths of $\mathcal{P}$ have a common vertex, and $f(G, \mathcal{P}) > 0$ otherwise (see the cited paper for details on calculating $f(G, \mathcal{P})$). Fujita et al. bounded $f(G, \mathcal{P})$ for a connected graph G with n vertices and a set $\mathcal{P}$ of three longest paths of G, showing that $f(G, \mathcal{P}) \leq (n + 6)/13$. Ekstein et al. [11] also investigated bounds on $f(G, \mathcal{P})$. They proved that, for every connected graph, every k longest paths have a common vertex for $3 \leq k \leq 6$ if and only if there exists a function g with certain properties such that $f(G, \mathcal{P}) \leq g(n)$ for every connected graph G of order n and every subset $\mathcal{P}$ of longest paths of G with $3 \leq |\mathcal{P}| \leq 6$; we refer the reader to the cited paper for further details of this function.

1.3.4.3 Vertices that are not in the vertex set of at least one longest path

Let $\mathcal{G}_k$ be the set of k-connected graphs, $k \geq 1$. For $j \geq 1$, let $\mathcal{G}_k^j \subseteq \mathcal{G}_k$ be the set of connected graphs G such that, for every set $S \subseteq V(G)$ of j vertices, there exists a longest path P of G for which $S \cap V(P) = \emptyset$. The following question was posed by Zamfirescu [42] in 1972 (restated): *What is* $\min\{|V(G)| : G \in \mathcal{G}_k^j\}$, *for* $j, k \geq 1$*?* Following Zamfirescu's notation, we denote this value P_k^j and define $\bar{P}_k^j$ similarly with the restriction that the graph is planar. If there exists no such graph then we write $P_k^j = \infty$. The graph on 12 vertices with no Gallai vertex, illustrated in Figure 1.6, shows that $P_1^1 \leq 12$. As mentioned previously, this is the smallest such graph with respect to the number of vertices, and hence $P_1^1 = 12$. The planar graph on 17 vertices with no Gallai vertex, illustrated in Figure 1.3, shows that $\bar{P}_1^1 \leq 17$. Other known results include $P_2^1 \leq 26$ (Skupien [33]) and $P_3^1 \leq 36$ (Zamfirescu [43]) and, for planar graphs, $\bar{P}_2^1 \leq 32$ (Zamfirescu [43]) and $\bar{P}_3^1 \leq 156$ (Jooyandeh et al. [24]). The well-known theorem from Tutte [36] which states that any 4-connected planar graph is Hamiltonian, shows that $\bar{P}_4^j = \infty$ for all j ($j \geq 1$). For a more comprehensive survey of these and related results, we refer the reader to Kensell [25] and Shabbir et al. [32].

Note that many of the questions discussed in Section 1.3 have also been investigated for longest cycles in a graph. The reader is referred to [21, 25, 30, 44] for more information.

1.4 Thesis structure

In this thesis, we answer Gallai's question in the positive for a specific class of graphs, and prove that Conjecture 1 holds for another class of graphs. We then use computational methods to approach Gallai's question and Conjecture 1 for several classes of graphs. The structure of this thesis is as follows.

In Chapter 2, we answer Gallai's question for a subclass of generalised theta graphs, as discussed in Section 1.2, showing that every graph of this class has a Gallai vertex. We categorise the longest paths of such a graph into different types depending on their endpoints. We present preliminary results using case analysis on these types of longest paths, before proving the main result of this chapter. In Chapter 4, we consider graphs with similar structure.

In Chapter 3, we prove that Conjecture 1 holds for graphs with n vertices and no more than $n + 5$

edges. We consider a minimal (with respect to edges) counterexample $\check{H}$ to Conjecture 1, as decribed in Section 1.3.2. We use a result of Axenovich [2] that, for such a graph $\check{H}$, each union of two of the longest paths P_0, P_1, and P_2 has two cycles. We present a number of preliminary results about such cycles, and prove that the presence of these six cycles in $\check{H}$ implies that $\check{H}$ has at least $n + 5$ edges. Additionally, we show that there exists a cycle of $\check{H}$ that is a cycle of the union of paths P_0, P_1, and P_2 but is not a cycle of any pairwise union of these longest paths, and prove that $\check{H}$ therefore has at least $n + 6$ edges. We conclude that, for any graph with n vertices and at most $n+5$ edges, every three longest paths have a common vertex. In Chapter 4, we independently verify this result using computational methods.

Finally, in Chapter 4, we use computational methods to approach Gallai's question and Conjecture 1 for several classes of graphs. We show that, for every simple graph G that is homeomorphic to a simple graph with at most 7 vertices, G has a Gallai vertex. We also show that every simple graph G homeomorphic to a simple graph with n vertices, $n + 6$ edges, and minimum vertex degree 3 has a Gallai vertex. This extends the result in Chapter 3. Let G be a simple graph, and let $\mathcal{G}$ be the infinite set of simple graphs homeomorphic to G. We use computational methods to find a finite set $\mathcal{S} \subset \mathcal{G}$ such that, if every graph in $\mathcal{S}$ has a Gallai vertex, then every graph in $\mathcal{G}$ has a Gallai vertex, and then determine whether each graph in $\mathcal{S}$ has a Gallai vertex. Using this process, we obtain the above two results. We then describe an alternative computational method and use this to obtain several classes of graphs for which Conjecture 1 holds. One such class is the class of graphs with n vertices and at most $n+5$ edges, independently confirming our result in Chapter 3. The other classes have a structure similar to the graphs in Chapter 2; we investigate graphs with exactly one non-trivial block. We also impose conditions on the numbers of vertices and edges of the graph, and the degrees of the vertices.

Chapter 2

Generalised theta graphs

2.1 Introduction

In this chapter, we investigate the class of generalised theta graphs, and show that generalised theta graphs with particular properties have a Gallai vertex.

Recall that a *generalised theta graph* is a graph G with at least one non-trivial block consisting of $k \geq 3$ internally disjoint paths with a common pair of distinct endpoints. Note that for such a block B, these two distinct endpoints are the only vertices $v \in V(B)$ such that $deg_B(v) \geq 3$. We call such blocks the *theta blocks* of G. An example of a generalised theta graph is shown in Figure 2.1.

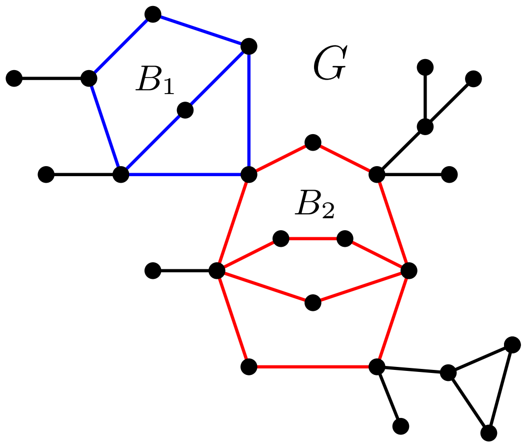

Figure 2.1: A generalised theta graph G with two theta blocks B_1 (with blue edges) and B_2 (with red edges).

In this chapter, we consider a subclass of generalised theta graphs. Let G be a generalised theta graph. We say that G is a *theta-Hamiltonian-tree graph* if G has at least one theta block $\hat{G}$ such that when we delete the edges of $\hat{G}$ from G to obtain subgraph G', each connected component C of G':

(i) has a Hamiltonian path with endpoint v, where $V(C) \cap V(\hat{G}) = \{v\}$, or

(ii) is a tree.

We call $\hat{G}$ a *core* of G. Furthermore, the components of G' are the *core-touching subgraphs* of core $\hat{G}$ of G, denoted $C(v)$ where $V(C(v)) \cap V(\hat{G}) = v$. (Note that since $\hat{G}$ is block of G, $|V(C(v)) \cap V(\hat{G})| = 1$.) If $V(C(v)) = \{v\}$, we call $C(v)$ a trivial core-touching subgraph of G.

An example of a theta-Hamiltonian-tree graph G with a unique core $\hat{G}$ is shown in Figure 2.2. The theta block B of G is not a core of G, and is the core-touching subgraph $C(a)$ of G. Subgraph $C(a)$ has a Hamiltonian path with endpoint a, and similarly $C(h)$ has a Hamiltonian path with endpoint h, while the core-touching subgraph $C(c)$ is a tree. The remaining core-touching subgraphs have both properties — for each $x \in \{b, d, e, f, g, u, v\}$, $C(x)$ is a path (possibly of length 0) and therefore $C(x)$ is a tree and has a Hamiltonian path with endpoint x.

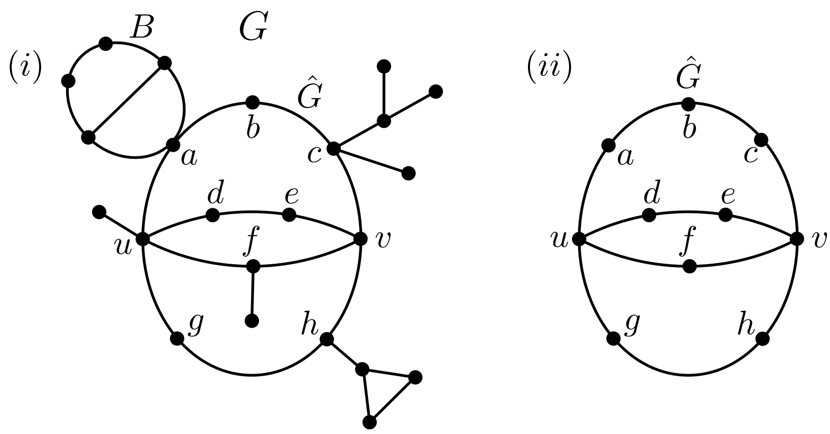

Figure 2.2: (i) A diagram of a theta-Hamiltonian-tree graph G with theta blocks $\hat{G}$ and B, where block $\hat{G}$ is the core of G. (ii) The core $\hat{G}$ of G.

In this chapter, we prove the following result about theta-Hamiltonian-tree graphs.

Theorem 2.1. *Every theta-Hamiltonian-tree graph has a Gallai vertex.*

The structure of the proof of Theorem 2.1 is as follows. We first consider theta-Hamiltonian-tree graphs with more than one core and prove that such graphs are Hamiltonian and therefore have a Gallai vertex. Turning our attention to theta-Hamiltonian-tree graphs with a unique core, we classify the longest paths of such a graph into two types: paths of Type (A) are paths of a core-touching subgraph of the graph, while paths of Type (B) have at least one edge that is an edge of the core of the graph. We then prove that if all of the longest paths of a theta-Hamiltonian-tree graph are of Type (A), or if there is at least one longest path of Type (A) and at least one longest path of Type (B), then the theta-Hamiltonian-tree graph has a Gallai vertex. This leaves the case in which all of the longest paths are of Type (B); these paths are then further classified into five subtypes. We present two forbidden configurations of Type (B) longest paths, then consider combinations of the subtypes of Type (B) longest paths and prove properties of their intersection. Using these results, we prove that if all of the longest paths of a theta-Hamiltonian-tree graph are of Type (B), then the graph has a Gallai vertex. The results on Type (A) and Type (B) paths are then used to show that every theta-Hamiltonian-tree graph with a unique core has a Gallai vertex.

2.1.1 Trees and the Helly property

Before turning to the proof of Theorem 2.1, we discuss our motivation for using trees as one of the options for the core-touching subgraphs of a theta-Hamiltonian-tree graph. This is motivated by the result that the set of paths of a tree has the *Helly property*, a property whose definition comes from set theory: a set $\mathcal{S}$ of sets has the *Helly property* if, for every non-empty subset $\mathcal{S}' \subseteq \mathcal{S}$ such that $A_1 \cap A_2 \neq \emptyset$ for all $A_1, A_2 \in \mathcal{S}'$, $\bigcap_{A \in \mathcal{S}'} A \neq \emptyset$. Restating this definition in terms of graphs, we let G be a graph and let $\mathcal{H}$ be a set of subgraphs of G. The set $\mathcal{S} = \{V(H) : H \in \mathcal{H}\}$ has the Helly property if, for every non-empty subset $\mathcal{S}' \subseteq \mathcal{S}$ such that $V(H_1) \cap V(H_2) \neq \emptyset$ for all $H_1, H_2 \in \mathcal{S}'$, $\bigcap_{H \in \mathcal{S}'} V(H) \neq \emptyset$. This means that, if every two subgraphs in a set $\mathcal{H}' \subseteq \mathcal{H}$ have a common vertex, then all of the subgraphs in $\mathcal{H}'$ have a common vertex. For brevity, we say that the set $\mathcal{H}$ has the Helly property.

To illustrate the Helly property on graphs, consider the graph G_1 in Figure 2.3(i). Let $\mathcal{H} = \{Q_1, Q_2, Q_3\}$. Then $\mathcal{H}$ has the Helly property since every two paths in $\mathcal{H}$ have a common vertex and $V(Q_1) \cap V(Q_2) \cap V(Q_3) = \{v\}$. However, for the graph G_2 in Figure 2.3(ii) with $\mathcal{H} = \{R_1, R_2, R_3\}$, the set $\mathcal{H}$ does not have the Helly property since every two paths in $\mathcal{H}$ have a common vertex but $V(R_1) \cap V(R_2) \cap V(R_3) = \emptyset$.

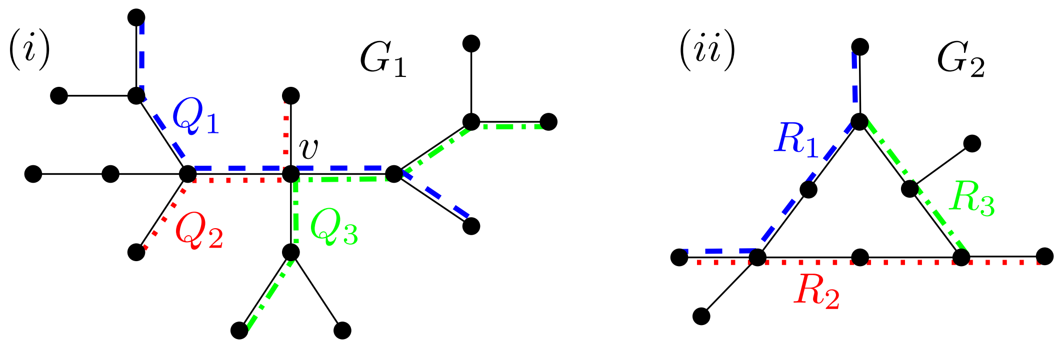

Figure 2.3: (i) A graph G_1 with three paths Q_1 (blue dashed lines), Q_2 (red dotted lines), and Q_3 (green dash-dotted lines) that have common vertex v. (ii) A graph G_2 with three paths R_1 (blue dashed lines), R_2 (red dotted lines), and R_3 (green dash-dotted lines) that do not have a common vertex.

The following result is restated from Golumbic [17, page 92].

Proposition 2.2. *Every set of subtrees of a tree has the Helly property.*

It follows from Proposition 2.2 that every set of paths of a tree has the Helly property. In fact, we have the following result.

Proposition 2.3. *Every set of paths of a graph G has the Helly property if and only if G is a tree.*

Proof. ($\leftarrow$) This follows directly from Proposition 2.2.

($\rightarrow$) Assume that G is not a tree. Then G has at least one cycle $C = v_1v_2 \ldots v_kv_1$, where $k \geq 3$ and $v_1, \ldots, v_k \in V(G)$. Let $P_1 = v_1v_2$, $P_2 = v_2v_3$, and $P_3 = v_3 \ldots v_kv_1$. Then P_1, P_2, and P_3 are three paths of C such that every two of these paths have a common vertex, but $V(P_1) \cap V(P_2) \cap V(P_3) = \emptyset$. Hence not every set of paths of G has the Helly property. $\square$

In Section 2.3, we will require that for a theta-Hamiltonian-tree graph G, each core-touching subgraph $C(v)$, where $v \in V(G)$, either has a Hamiltonian path with endpoint v or the set of paths of $C(v)$ has the Helly property. By Proposition 2.3, the latter condition is equivalent to the condition that $C(v)$ is a tree.

We now turn to the proof of Theorem 2.1, beginning with theta-Hamiltonian-tree graphs that have more than one core.

2.2 Graphs with more than one core

By definition, theta-Hamiltonian-tree graphs may have more than one core. We have the following result for such graphs.

Proposition 2.4. *Every theta-Hamiltonian-tree graph with more than one core has a Gallai vertex.*

Proof. Let G be a theta-Hamiltonian-tree graph with more than one core. We show that G has a Hamiltonian path.

Let $\hat{G}_1$ be a core of G, and let $x_1, \ldots, x_k$ ($k \geq 4$) be the vertices of $\hat{G}_1$. Then $C(x_1), \ldots, C(x_k)$ are the core-touching subgraphs of core $\hat{G}_1$. By definition, each $C(x_i)$, $1 \leq i \leq k$, is a tree or has a Hamiltonian path with endpoint x_i. Since G has more than one core, there exists another core $\hat{G}_2$ of G and exactly one x_i, $1 \leq i \leq k$, such that $\hat{G}_2$ is a subgraph of $C(x_i)$. Since $\hat{G}_2$ is not a tree, $C(x_i)$ has a Hamiltonian path H_1 with endpoints x_i and w, $w \in V(G)$. It follows that core $\hat{G}_2$ has at most two non-trivial core-touching subgraphs.

Let $y_1, \ldots, y_\ell$ ($\ell \geq 4$) be the vertices of $\hat{G}_2$. Then $C(y_1), \ldots, C(y_\ell)$ are the core-touching subgraphs of core $\hat{G}_1$. There exists exactly one y_j, $1 \leq j \leq \ell$, such that $\hat{G}_1$ is a subgraph of $C(y_j)$. Since $\hat{G}_1$ is not a tree, $C(y_j)$ has a Hamiltonian path H_2 with endpoints y_j and z, $z \in V(G)$. It follows that core $\hat{G}_1$ has at most two non-trivial core-touching subgraphs. Let w be the other endpoint of H_1, so that $H_1 = x_i H_1 w$, and let z be the other endpoint of H_2, so that $H_2 = y_j H_2 z$. An example of G with paths H_1 and H_2 is shown in Figure 2.4. Then $wH_1x_iH_2z$ is a Hamiltonian path of G. Therefore G has a Gallai vertex. $\square$

2.3 Graphs with a unique core

We can now restrict our attention to theta-Hamiltonian-tree graphs with a unique core. In this section, we prove the following result.

Proposition 2.5. *Every theta-Hamiltonian-tree graph with a unique core has a Gallai vertex.*

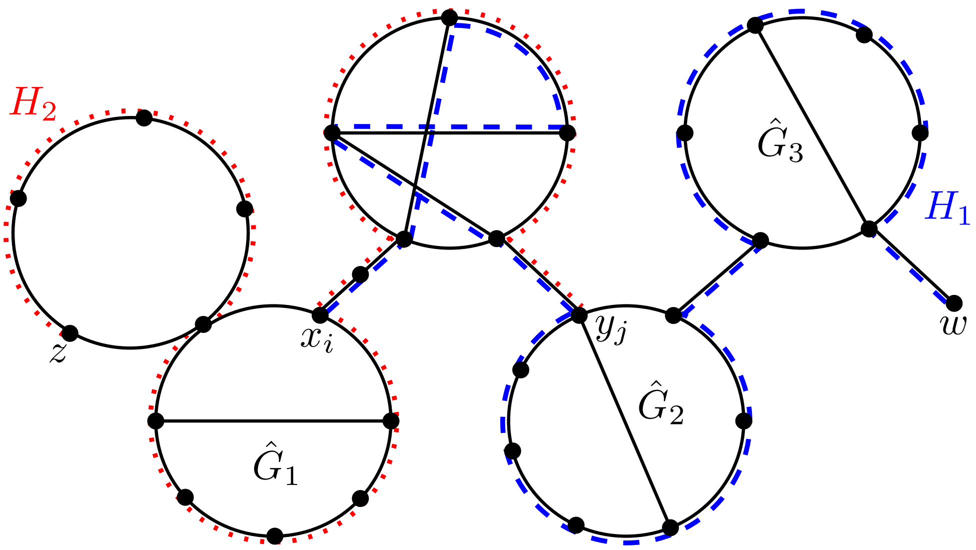

Figure 2.4: A theta-Hamiltonian-tree graph with three cores $\hat{G}_1, \hat{G}_2$, and $\hat{G}_3$. Path H_1 is shown by blue dashed lines, and path H_2 by red dotted lines.

2.3.1 Two types of longest paths

Let G be a theta-Hamiltonian-tree graph with unique core $\hat{G}$. Let $\mathcal{P}$ be the set of longest paths of G. For each path $P \in \mathcal{P}$, we classify P as one of two possible types of longest paths. Either

Type (A): $E(P) \cap E(\hat{G}) = \emptyset$ or

Type (B): $E(P) \cap E(\hat{G}) \neq \emptyset$.

In order to prove Proposition 2.5, we consider the following three cases and prove that G has a Gallai vertex in each case.

(1) Every longest path of G is of Type (A).

(2) At least one longest path of G is of Type (A) and at least one longest path of G is of Type (B).

(3) Every longest path of G is of Type (B).

2.3.2 Case (1): paths of Type (A)

We first require the following result.

Proposition 2.6. *Let G be a theta-Hamiltonian-tree graph with unique core $\hat{G}$, and let $x \in V(\hat{G})$. If the core-touching subgraph $C(x)$ of G has a Hamiltonian path with endpoint x, then, for every Type (A) longest path P of G, $V(P) \cap V(C(x)) = \emptyset$.*

Proof. Assume that $C(x)$ has a Hamiltonian path with endpoint x. Assume that there is a Type (A) longest path P of G such that $V(P) \cap V(C(x)) \neq \emptyset$. Then P is a path of $C(x)$ and, moreover, P is a Hamiltonian path of $C(x)$. Let Q be a Hamiltonian path of $C(x)$ with endpoint x. Let w be a vertex of $\hat{G}$ adjacent to x (which always exists since $|V(\hat{G})| \geq 4$). Let Q' be the path of G consisting of the path Q and the edge xw. Then $|Q'| > |P|$, a contradiction since P is a longest path of G. □

The proposition above therefore confines the Type (A) longest paths of G to core-touching subgraphs of G which are trees.

Lemma 2.7. *Let G be a theta-Hamiltonian-tree graph with a unique core. All of the Type (A) longest paths of G have a common vertex.*

Proof. Let $\mathcal{P}$ be the set of Type (A) longest paths of G, and let $\hat{G}$ be the unique core of G. Since every two paths in $\mathcal{P}$ have a common vertex by Lemma 1.1, and the paths in $\mathcal{P}$ are of Type (A), the paths in $\mathcal{P}$ are all paths of one core-touching subgraph $C(x)$, for some $x \in V(\hat{G})$. By Proposition 2.6, $C(x)$ is a tree. Hence, by Proposition 2.2, the paths in $\mathcal{P}$ have a common vertex. □

Corollary 2.8. *Let G be a theta-Hamiltonian-tree graph with a unique core. If every longest path of G is of Type (A), then G has a Gallai vertex.*

This follows directly from Lemma 2.7.

2.3.3 Case (2): paths of Type (A) and Type (B)

Let G be a theta-Hamiltonian-tree graph with unique core $\hat{G}$. Let Q be a path of G with endpoint x. If there exists $a \in V(\hat{G}) \cap V(Q)$ such that $x \in V(C(a))$ and $E(aQx) \subseteq E(C(a))$ for core-touching subgraph $C(a)$ of G, then aQx is a *tail* of Q. We call a a *base vertex* of Q. Note that it may be the case that $a = x$, in which case xQx is a *trivial tail*. Note that every Type (B) longest path of G has two tails, one or both of which may be trivial. Figure 2.5 shows an example of a theta-Hamiltonian-tree graph G and a path Q of G with trivial tail $aQx = xQx$ and non-trivial tail bQy, where a and b are the base vertices of Q and x and y are the endpoints of Q.

Lemma 2.9. *Let G be a theta-Hamiltonian-tree graph with a unique core. If at least one longest path of G is of Type (A) and at least one longest path of G is of Type (B), then G has a Gallai vertex.*

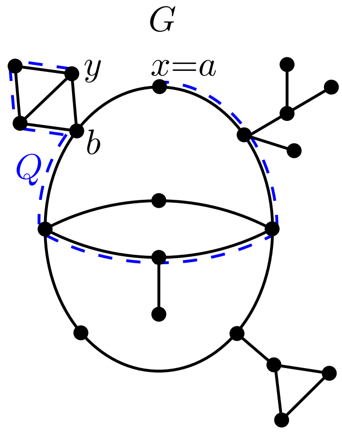

Figure 2.5: A theta-Hamiltonian-tree graph G with path Q (blue dashed lines) that has tails aQx and bQy.

Proof. Let $\hat{G}$ be the unique core of G. Let $\mathcal{P} = \mathcal{P}_A \cup \mathcal{P}_B$ be the set of longest paths of G, where $\mathcal{P}_A$ is the non-empty set of paths of Type (A) and $\mathcal{P}_B$ is the non-empty set of paths of Type (B). By Lemma 2.7, the paths in $\mathcal{P}_A$ have a common vertex and hence the paths in $\mathcal{P}_A$ are paths of one core-touching subgraph $C(x)$, for some $x \in V(\hat{G})$. By Lemma 1.1, each path in $\mathcal{P}_B$ has a common vertex with each path in $\mathcal{P}_A$ and hence $x \in V(P)$ for each path $P \in \mathcal{P}_B$, since there is at least one edge of P that is an edge of $\hat{G}$. If $x \in V(Q)$ for each path $Q \in \mathcal{P}_A$, then x is a Gallai vertex of G and we are done. If this is not the case, then there exists at least one path $P_1 \in \mathcal{P}_A$ that is a path of $C(x) - x$. Since each path in $\mathcal{P}_B$ has a common vertex with P_1, each path in $\mathcal{P}_B$ has a non-trivial tail that is a path of $C(x)$. Let T be the set of these tails, and let $S = T \cup \mathcal{P}_A$. Then S is a set of paths of $C(x)$ and every two paths in S have a common vertex. By Proposition 2.6, $C(x)$ is a tree, and hence, by Proposition 2.2, the paths in S have a common vertex. Therefore the paths in $\mathcal{P}$ have a common vertex. $\square$

2.3.4 Case (3): paths of Type (B)

Let G be a theta-Hamiltonian-tree graph with unique core $\hat{G}$, and let u and v be the two vertices of $\hat{G}$ such that $deg_{\hat{G}}(u) \geq 3$ and $deg_{\hat{G}}(v) \geq 3$. be the pair of common endpoints of the internally disjoint paths whose union is $\hat{G}$. Let P be a Type (B) longest path of G with tails aPx and bPy, where $a, b \in V(\hat{G})$ ($a \neq b$) and x and y are the endpoints of P. We classify Type (B) longest paths P as exactly one of five possible subtypes:

(a) $u, v \notin V(P)$;

(b) $u, v \in V(P)$, and there exists a uv–path U_1 such that a and b are internal vertices of U_1;

(c) $u, v \in V(P)$, and there exist uv–paths U_2 and U_3 such that $U_2 \neq U_3$ and a and b are internal vertices of U_2 and U_3 respectively;

(d) $u \in V(P)$ but $v \notin V(P)$; or

(e) $v \in V(P)$ but $u \notin V(P)$.

If a Type (B) longest path of G is of Subtype (a), we say that the path is of Type (B)(a), and similarly for Subtypes (b) – (e). Examples of the five subtypes of Type (B) longest paths are shown in Figure 2.6.

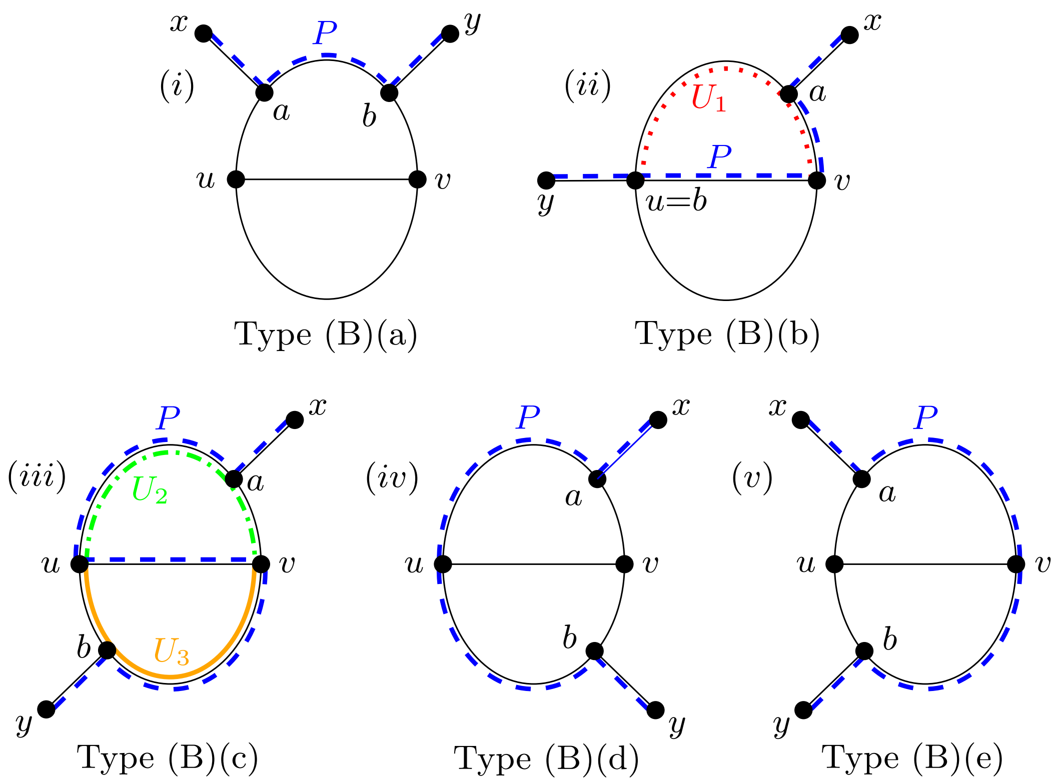

Figure 2.6: A selection of theta-Hamiltonian-tree graphs with examples of a Type (B) longest path P (blue dashed line) of each of Subtypes (a) – (e) shown in (i) – (v) respectively. In (ii), path U_1 is the red dotted line and in (iii), path U_2 is the green dash-dotted line and path U_3 is the orange thick solid line.

The main result of this section is the following lemma.

Lemma 2.10. *Let G be a theta-Hamiltonian-tree graph with a unique core. If every longest path of G is of Type (B), then G has a Gallai vertex.*

We present two forbidden configurations of Type (B) longest paths, followed by several results on the properties of Type (B) longest paths, before proving Lemma 2.10 at the end of this section.

2.3.4.1 Forbidden configurations of Type (B) longest paths

Let G be a theta-Hamiltonian-tree graph with unique core $\hat{G}$. Let u and v be the two vertices of $\hat{G}$ such that $deg_{\hat{G}}(u) \geq 3$ and $deg_{\hat{G}}(v) \geq 3$. We define two configurations whose existence is forbidden in G.

Configuration 1: Two Type (B) longest paths P_1 and P_2 such that there is an a_1a_2–path Q of G with length at least one where a_1 is a base vertex of P_1 and a_2 is a base vertex of P_2, and $V(Q) \cap V(P_1) = \{a_1\}$ and $V(Q) \cap V(P_2) = \{a_2\}$.

Configuration 2: A Type (B)(d) longest path P_1 and a Type (B)(e) longest path P_2.

Examples of these two configurations are shown in Figure 2.7

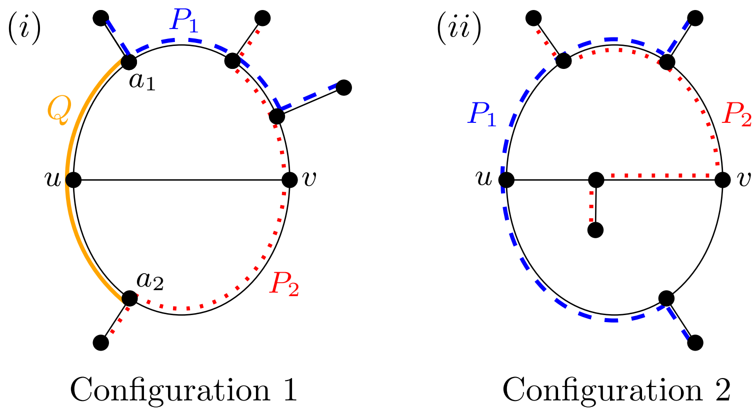

Figure 2.7: Examples of (i) Configuration 1 and (ii) Configuration 2 in a theta-Hamiltonian-tree graph, where P_1 is shown by the blue dashed line, P_2 is shown by the red dotted line, and in Configuration 1, Q is the a_1a_2–path shown by the orange solid line.

Lemma 2.11. *Let G be a theta-Hamiltonian-tree graph with a unique core. Then Configuration 1 does not exist in G.*

Proof. Let $\hat{G}$ be the unique core of G and let u and v be the two vertices of $\hat{G}$ such that $deg_{\hat{G}}(u) \geq 3$ and $deg_{\hat{G}}(v) \geq 3$. Assume that Configuration 1 does exist in G and let P_1, P_2, a_1, a_2, and Q be as defined in the definition of Configuration 1. Since P_1 and P_2 are both of Type (B), they each have two tails. Let $a_1P_1x_1$ and $b_1P_1y_1$ be the tails of P_1, where x_1 and y_1 are the endpoints of P_1 and $b_1 \in V(\hat{G})$. Similarly, let $a_2P_2x_2$ and $b_2P_2y_2$ be the tails of P_2, where x_2 and y_2 are the endpoints of P_2 and $b_2 \in V(\hat{G})$. An example of G is shown in Figure 2.8(i).

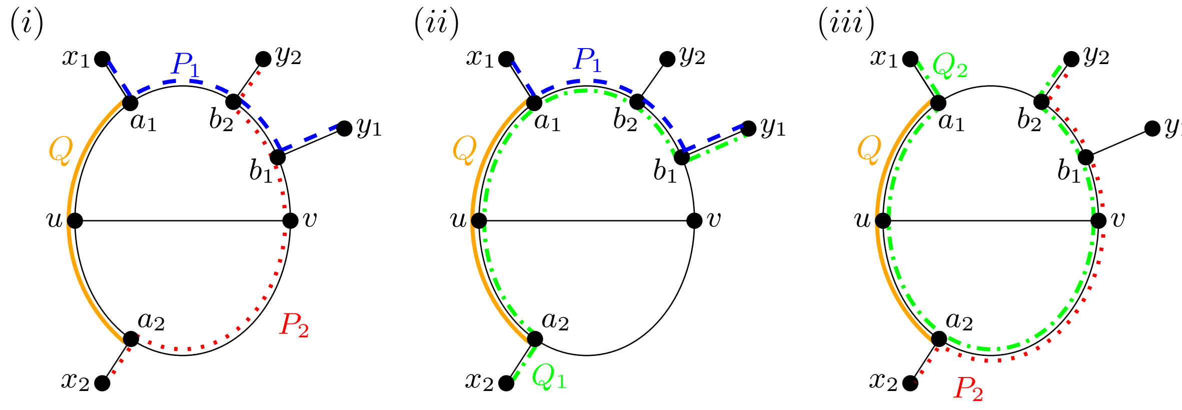

Figure 2.8: Three diagrams of a theta-Hamiltonian-tree graph showing (i) longest paths P_1 and P_2 (blue dashed and red dotted lines respectively) and a_1a_2–path Q (orange solid line); (ii) path Q_1 (green dash-dotted line) and paths P_1 and Q; and (iii) path Q_2 (green dash-dotted line) and paths P_2 and Q.

Let Q_1 be the path $x_2P_2a_2Qa_1P_1y_1$ of G and let Q_2 be the path $x_1P_1a_1Qa_2P_2y_2$ of G; an example of these paths is shown in Figure 2.8(ii) and (iii) respectively. Since P_1 is a longest path of G,

$$\begin{aligned} |P_1| &\geq |Q_1| \\ |x_1P_1a_1P_1y_1| &\geq |x_2P_2a_2Qa_1P_1y_1| \\ |x_1P_1a_1| &\geq |x_2P_2a_2| + |Q|. \end{aligned} \tag{2.1}$$

Similarly, since P_2 is a longest path of G,

$$\begin{aligned} |P_2| &\geq |Q_2| \\ |x_2P_2a_2P_2y_2| &\geq |x_1P_1a_1Qa_2P_2y_2| \\ |x_2P_2a_2| &\geq |x_1P_1a_1| + |Q|. \end{aligned} \tag{2.2}$$

However, from 2.1 and 2.2 we have $|Q| \leq 0$, a contradiction since $|Q| \geq 1$. □

Lemma 2.12. *Let G be a theta-Hamiltonian-tree graph with a unique core. Then Configuration 2 does not exist in G.*

Proof. Let $\hat{G}$ be the unique core of G and let u and v be the two vertices of $\hat{G}$ such that $deg_{\hat{G}}(u) \geq 3$ and $deg_{\hat{G}}(v) \geq 3$. Assume that Configuration 2 does exist in G and let P_1 and P_2 be as defined in the definition of Configuration 2. Let $a_1P_1x_1$ and $b_1P_1y_1$ be the tails of P_1, where x_1 and y_1 are the endpoints of P_1 and $a_1, b_1 \in V(\hat{G})$. Similarly, let $a_2P_2x_2$ and $b_2P_2y_2$ be the tails of P_2, where x_2 and y_2 are the endpoints of P_2 and $a_2, b_2 \in V(\hat{G})$. Since P_1 is of Type (B)(d), $u \in V(a_1P_1b_1)$, and, since P_2 is of Type (B)(e), $v \in V(a_2P_2b_2)$. By Lemma 1.1, P_1 and P_2 have a common vertex, and hence there exists a uv–path Q of G such that a_1 or b_1 is in $V(Q)$ and a_2 or b_2 is in $V(Q)$; suppose that $a_1, a_2 \in V(Q)$ without loss of generality. Then a_1Qa_2 is a subpath of both P_1 and P_2. Note that $b_1, b_2 \notin V(Q)\backslash\{u, v\}$ and that it is possible that $a_1 = a_2$.

There are two cases to be considered: (i) there exist two uv–paths S_1 and S_2 of G ($S_1 \neq S_2$) such that $b_1 \in V(S_1)$ and $b_2 \in V(S_2)$ and (ii) there exists a uv–path R_1 of G such that $b_1, b_2 \in V(R_1)$. Examples of these two cases are shown in Figure 2.9.

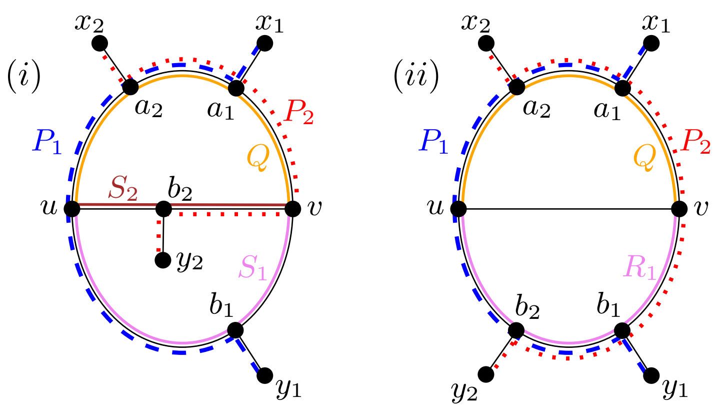

Figure 2.9: Two examples of a theta-Hamiltonian-tree graph showing longest paths P_1 of Type (B)(d) (blue dashed line) and P_2 of Type (B)(e) (red dotted line) along with uv–path Q (orange solid line), where b_1 and b_2 are (i) vertices of uv–paths S_1 and S_2 respectively (purple and brown solid lines respectively) and (ii) vertices of uv–path R_1 (purple solid line).

First consider case (i). Since $v \notin V(P_1)$, $|b_1S_1v| \geq 1$. Let Q_1 be the path $x_2P_2vS_1b_1P_1y_1$ of G, an example of which is shown in Figure 2.10(i). Since P_2 is a longest path of G,

$$|P_2| \geq |Q_1|$$

$$|x_2P_2vP_2y_2| \geq |x_2P_2vS_1b_1P_1y_1|$$

$$|vP_2y_2| \geq |vS_1b_1| + |b_1P_1y_1|. \tag{2.3}$$

Let Q_2 be the path $x_1P_1b_1S_1vP_2y_2$ of G, an example of which is shown in Figure 2.10(ii). Since P_1 is a longest path of G,

$$|P_1| \geq |Q_2|$$

$$|x_1P_1b_1P_1y_1| \geq |x_1P_1b_1S_1vP_2y_2|$$

$$|b_1P_1y_1| \geq |b_1S_1v| + |vP_2y_2|. \tag{2.4}$$

From 2.3 and 2.4 we have $|b_1S_1v| \leq 0$, a contradiction.

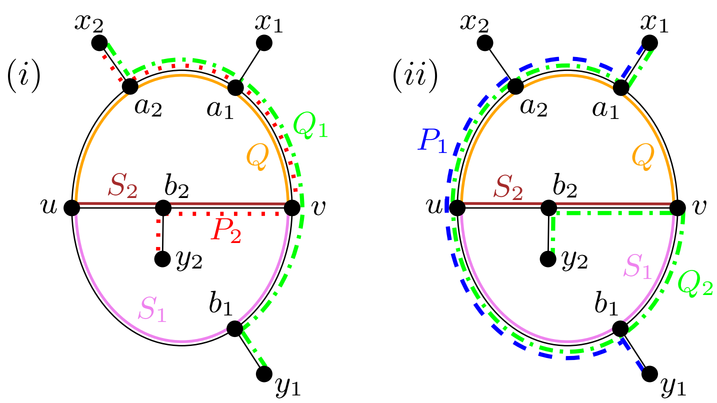

Figure 2.10: Two diagrams of a theta-Hamiltonian-tree graph showing (i) longest path P_2 of Type (B)(e) (red dotted line) and path Q_1 (green dash-dotted line) and (ii) longest path P_1 of Type (B)(d) (blue dashed line) and path Q_2 (green dash-dotted line). In both diagrams, paths Q, S_1, and S_2 are uv–paths of the graph (orange, purple, and brown solid lines respectively).

Next, consider case (ii). By Lemma 2.11, path $b_1R_1b_2$ is a subpath of both P_1 and P_2. Let R_2 be a uv–path of Q of maximum length, with $R_2 \neq Q$ and $R_2 \neq R_1$ (if there exists more than one such path of G, we pick one without loss of generality). Note that $|R_2| \geq 1$ since $u \neq v$.

Let Q_3 be the path $x_2P_2vR_2uP_1y_1$ of G, an example of which is shown in Figure 2.11(i). Since P_2 is

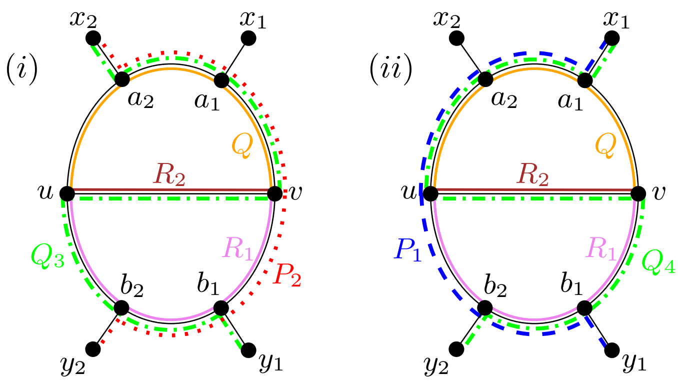

Figure 2.11: Two diagrams of a theta-Hamiltonian-tree graph showing (i) longest path P_2 of Type (B)(e) (red dotted line) and path Q_3 (green dash-dotted line) and (ii) longest path P_1 of Type (B)(d) (blue dashed line) and path Q_4 (green dash-dotted line). In both diagrams, paths Q, R_1, and R_2 (orange, purple, and brown solid lines respectively) are uv–paths of the graph.

a longest path of G,

$$\begin{aligned} |P_2| &\geq |Q_3| \\ |x_2P_2vP_2y_2| &\geq |x_2P_2vR_2uP_1y_1| \\ |vP_2y_2| &\geq |vR_2u| + |uP_1y_1|. \end{aligned} \tag{2.5}$$

Let Q_4 be the path $x_1P_1uR_2vP_2y_2$ of G, an example of which is shown in Figure 2.11(ii). Since P_1 is a longest path of G,

$$\begin{aligned} |P_1| &\geq |Q_4| \\ |x_1P_1uP_1y_1| &\geq |x_1P_1uR_2vP_2y_2| \\ |uP_1y_1| &\geq |uR_2v| + |vP_2y_2|. \end{aligned} \tag{2.6}$$

From 2.5 and 2.6 we have $|uR_2v| = |R_2| \leq 0$, a contradiction. We conclude that P_2 is not of Type (B)(e). An analogous argument can be used to show that if P_1 is of Type (B)(e), then there does not exist a longest path of G of Type (B)(d). $\square$

2.3.4.2 Properties of paths of Type (B)

Lemma 2.13. *Let G be a theta-Hamiltonian-tree graph with a unique core and let P_1 and P_2 be Type (B) longest paths of G. Let a_1 and b_1 be the base vertices of P_1 and let a_2 and b_2 be the base vertices of*

P_2. If P_1 and P_2 are both of Subtype (a), then $a_1P_1b_1$ is a subpath of $a_2P_2b_2$ or $a_2P_2b_2$ is a subpath of $a_1P_1b_1$.

Proof. Assume that P_1 and P_2 are Type (B)(a) longest paths of G. Let $\hat{G}$ be the unique core of G and let u and v be the two vertices of $\hat{G}$ such that $deg_{\hat{G}}(u) \geq 3$ and $deg_{\hat{G}}(v) \geq 3$. Since P_1 and P_2 have a common vertex by Lemma 1.1 and are of Type (B)(a), paths $a_1P_1b_1$ and $a_2P_2b_2$ are subpaths of a uv–path Q of G and have at least one common vertex. Without loss of generality, we assume that $a_1 \in V(uQb_1)$ and $a_2 \in V(uQb_2)$.

Assume that $E(a_1P_1b_1)\backslash E(a_2P_2b_2) \neq \emptyset$ and $E(a_2P_2b_2)\backslash E(a_1P_1b_1) \neq \emptyset$. Then $a_2 \in V(a_1Qb_1)$ or $b_2 \in V(a_1Qb_1)$. Suppose that $a_2 \in V(a_1Qb_1)$; then $a_1 \neq a_2$ and $b_2 \notin V(a_1Qb_1)$. An example of G is shown in Figure 2.12(i). Note that it may be the case that $a_2 = b_1$. Additionally, since P_1 and P_2 are of Type (B)(a), $a_1 \neq u$ and $b_2 \neq v$.

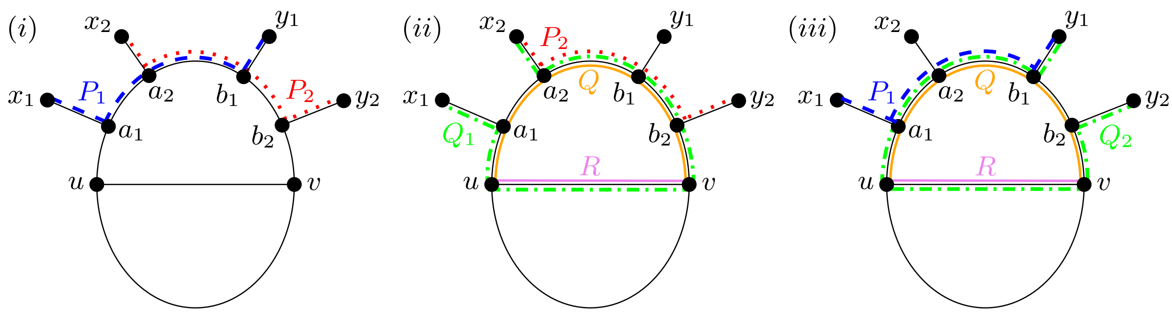

Figure 2.12: Three diagrams of a theta-Hamiltonian-tree graph showing (i) two longest paths P_1 and P_2 of Type (B)(a) (blue dashed and red dotted lines respectively), (ii) path Q_1 (green dash-dotted line) and path P_2, and (iii) path Q_2 (green dash-dotted line) and path P_1. In all three diagrams, paths Q and R (orange and purple solid lines respectively) are uv–paths of the graph.

Let $a_1P_1x_1$ and $b_1P_1y_1$ be the tails of P_1, where x_1 and y_1 are the endpoints of P_1. Similarly, let $a_2P_2x_2$ and $b_2P_2y_2$ be the tails of P_2, where x_2 and y_2 are the endpoints of P_2. Let R be a uv–path of G where $R \neq Q$. Let Q_1 be the path $x_2P_2b_2QvRuQa_1P_1x_1$ of G, an example of which is shown in Figure

2.12(ii). Since P_2 is a longest path of G,

$$\begin{aligned} |P_2| &\geq |Q_1| \\ |x_2P_2b_2P_2y_2| &\geq |x_2P_2b_2QvRuQa_1P_1x_1| \\ |b_2P_2y_2| &\geq |b_2QvRuQa_1| + |a_1P_1x_1|. \end{aligned} \tag{2.7}$$

Let Q_2 be the path $y_2P_2b_2QvRuQa_1P_1y_1$ of G, an example of which is shown in Figure 2.12(iii). Since P_1 is a longest path of G,

$$\begin{aligned} |P_1| &\geq |Q_2| \\ |x_1P_1a_1P_1y_1| &\geq |y_2P_2b_2QvRuQa_1P_1y_1| \\ |x_1P_1a_1| &\geq |y_2P_2b_2| + |b_2QvRuQa_1|. \end{aligned} \tag{2.8}$$

From 2.7 and 2.8 we have $|b_2QvRuQa_1| \leq 0$. However, $b_2 \neq u$, $a_1 \neq u$, and $|R| \geq 1$, hence $|b_2QvRuQa_1| \geq 3$, a contradiction. An analogous argument can be used to obtain a contradiction when $a_1 \in V(a_2Qb_2)$, with $a_1 \neq a_2$, and $b_1 \notin V(a_2Qb_2)$. $\square$

Lemma 2.14. *Let G be a theta-Hamiltonian-tree graph with unique core $\hat{G}$. If $P_1, \ldots, P_k$ $(k \geq 2)$ are Type (B)(a) longest paths of G, then there exists a vertex $x \in V(\hat{G})$ that is a common vertex of $P_1, \ldots, P_k$. Furthermore, x is a base vertex of each of $P_1, \ldots, P_k$.*

Proof. Assume that $P_1, \ldots, P_k$ $(k \geq 2)$ are Type (B)(a) longest paths of G. Let $\mathcal{P} = \{P_1, \ldots, P_k\}$ and let u and v be the two vertices of $\hat{G}$ such that $deg_{\hat{G}}(u) \geq 3$ and $deg_{\hat{G}}(v) \geq 3$. Let $a_i, b_i \in \hat{G}$ be the base vertices of P_i, $1 \leq i \leq k$.

We first prove that every pair of paths in $\mathcal{P}$ have a common base vertex. Without loss of generality, consider paths $P_1, P_2 \in \mathcal{P}$. Since P_1 and P_2 have a common vertex by Lemma 1.1 and are of Type (B)(a), paths $a_1P_1b_1$ and $a_2P_2b_2$ are subpaths of a uv–path Q of G and have at least one common vertex. Without loss of generality, we assume that $a_1 \in V(uQb_1)$ and $a_2 \in V(uQb_2)$. By Lemma 2.13, $a_1P_1b_1$ is a subpath of $a_2P_2b_2$ or $a_2P_2b_2$ is a subpath of $a_1P_1b_1$. Suppose that $a_2P_2b_2$ is a subpath of $a_1P_1b_1$, an example of which is shown in Figure 2.13(i). Note that, since P_1 and P_2 are of Type (B)(a), $a_1 \neq u$ and $b_1 \neq v$.

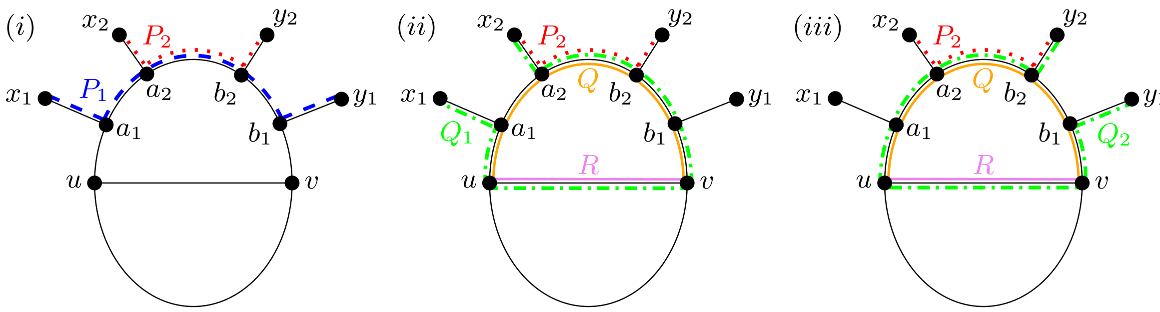

Figure 2.13: Three diagrams of a theta-Hamiltonian-tree graph showing (i) longest paths P_1 and P_2 of Type (B)(a) (blue dashed and red dotted lines respectively) (ii) path Q_1 (green dash-dotted line) and path P_2 and (iii) path Q_2 (green dash-dotted line) and path P_2. In all three diagrams, paths Q and R (orange and purple solid lines respectively) are uv–paths of the graph.

Assume that there does not exist a vertex of $\hat{G}$ that is a base vertex of both P_1 and P_2. Then $a_1 \neq a_2$ and $b_1 \neq b_2$. Let R be a uv–path of G where $R \neq Q$. Let Q_1 be the path $x_2P_2b_2QvRuQa_1P_1x_1$ of G and let Q_2 be the path $y_1P_1b_1QvRuQa_2P_2y_2$ of G, examples of which are shown in Figure 2.13(ii) and (iii) respectively. Since P_2 is a longest path of G,

$$\begin{aligned} |P_2| &\geq |Q_1| \\ |x_2P_2b_2| + |b_2P_2y_2| &\geq |x_2P_2b_2| + |b_2Q_1x_1| \\ |b_2P_2y_2| &\geq |b_2Q_1x_1|. \end{aligned} \tag{2.9}$$

Similarly,

$$\begin{aligned} |P_2| &\geq |Q_2| \\ |x_2P_2a_2| + |a_2P_2y_2| &\geq |y_1Q_2a_2| + |a_2P_2y_2| \\ |x_2P_2a_2| &\geq |y_1Q_2a_2|. \end{aligned} \tag{2.10}$$

Therefore

$$
\begin{aligned}
|P_2| &= |x_2P_2a_2| + |a_2P_2b_2| + |b_2P_2y_2| \\
&\geq |y_1Q_2a_2| + |a_2P_1b_2| + |b_2Q_1x_1| \quad \text{by 2.9 and 2.10} \\
&= |y_1P_1b_1| + |b_1Q_2a_1| + |a_1P_1a_2| + |a_2P_1b_2| + |b_2P_1b_1| + |b_1Q_1a_1| + |a_1P_1x_1| \\
&= |P_1| + 2|b_1Q_2a_1| \quad \text{since } |b_1Q_2a_1| = |b_1Q_1a_1| \\
&> |P_2|
\end{aligned}
$$

since $|P_1| = |P_2|$ and $|b_1Q_2a_1| > 0$, a contradiction. Hence P_1 and P_2 have a common base vertex. A similar argument shows that P_1 and P_2 have a common base vertex when $a_1P_1b_1$ is a subpath of $a_2P_2b_2$. Therefore, every pair of paths in $\mathcal{P}$ have a common base vertex.

We now prove that all of the paths in $\mathcal{P} = \{P_1, \ldots, P_k\}$ have a common base vertex when $k \geq 3$. Since every two paths in $\mathcal{P}$ have a common vertex by Lemma 1.1 and each path in $\mathcal{P}$ is of Type (B)(a), there exists a uv–path Q of G such that $a_iP_ib_i$ is a subpath of Q for all i, $1 \leq i \leq k$. Consider paths $P_1, P_2 \in \mathcal{P}$. By the above argument, P_1 and P_2 have a common base vertex. Without loss of generality, suppose that $a_1 = a_2$. By Lemma 2.13, $a_1P_1b_1$ is a subpath of $a_2P_2b_2$ or $a_2P_2b_2$ is a subpath of $a_1P_1b_1$. Suppose that $a_2P_2b_2$ is a subpath of $a_1P_1b_1$, an example of which is shown in Figure 2.14(i).

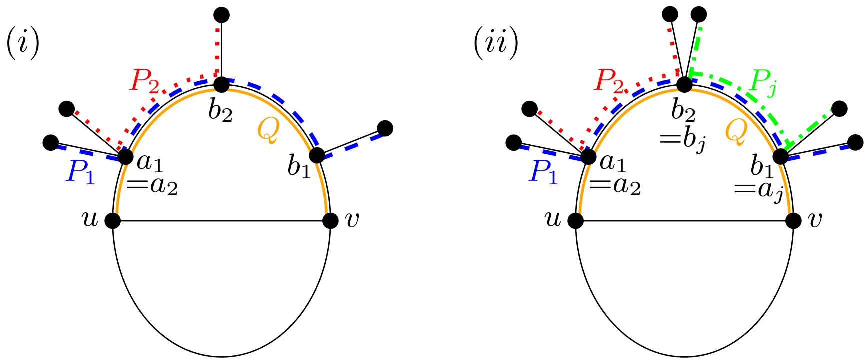

Figure 2.14: Examples of a theta-Hamiltonian-tree graph showing (i) two longest paths P_1 and P_2 of Type (B)(a) (blue dashed and red dotted lines respectively) that have a common base vertex, and (ii) another longest path P_j of Type (B)(a) (green dash-dotted line) that has a common base vertex with each of P_1 and P_2. In both diagrams, path Q (orange solid line) is a uv–path of the graph.

Suppose that there exists a path $P_j \in \mathcal{P}$, $3 \leq j \leq k$, such that $a_j \neq a_1$ and $b_j \neq a_1$. By the above argument, P_j has a common base vertex with each of P_1 and P_2. Without loss of generality, suppose that $a_j = b_1$ and $b_j = b_2$, an example of which is shown in Figure 2.14(ii). Since $a_jP_jb_j$ is a subpath of Q, $a_2 \notin V(P_j)$ and $a_j \notin V(P_2)$. However, by Lemma 2.13, $a_jP_jb_j$ is a subpath of $a_2P_2b_2$ or $a_2P_2b_2$ is a subpath of $a_jP_jb_j$, a contradiction. Hence, for every path P_j where $3 \le j \le k$, $a_j = a_1$ or $b_j = a_1$. Therefore a_1 is a base vertex of each path in $\mathcal{P}$. A similar argument shows that the result holds when $a_1P_1b_1$ is a subpath of $a_2P_2b_2$. □

Lemma 2.15. *Let G be a theta-Hamiltonian-tree graph with unique core $\hat{G}$ and let P_1 and P_2 be Type (B) longest paths of G. Let $a_1, b_1 \in V(\hat{G})$ be the two base vertices of P_1. If P_1 is of Subtype (a) and P_2 is of Subtype (d) or (e), then $a_1, b_1 \in V(P_2)$. Furthermore, $a_1P_1b_1$ is a subpath of P_2.*

Proof. Assume that P_1 is of Type (B)(a). Let u and v be the two vertices of $\hat{G}$ such that $deg_{\hat{G}}(u) \ge 3$ and $deg_{\hat{G}}(v) \ge 3$. Since P_1 is of Type (B)(a), there exists a uv–path Q of G such that $a_1P_1b_1$ is a subpath of Q. Without loss of generality, we assume that $a_1 \in V(uQb_1)$. Suppose that P_2 is of Type (B)(d). Since P_2 is of Type (B), P_2 has two base vertices $a_2, b_2 \in V(\hat{G})$. Since P_1 and P_2 have a common vertex by Lemma 1.1 and P_2 is of Type (B)(d), it follows that $a_1 \in V(P_2)$. Then $a_2 \in V(Q)$ or $b_2 \in V(Q)$. Without loss of generality, suppose that $a_2 \in V(Q)$. Let R be the uv–path of G such that $b_2 \in V(R)$. An example of G is shown in Figure 2.15.

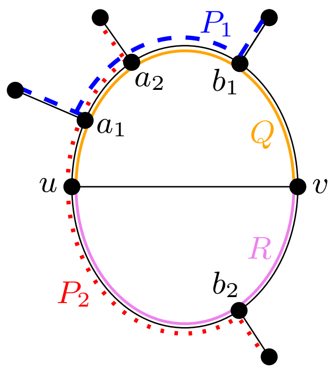

Figure 2.15: An example of a theta-Hamiltonian-tree graph with longest paths P_1 of Type (B)(a) (blue dashed line) and P_2 of Type (B)(d) (red dotted line). Paths Q and R (orange and purple solid lines respectively) are uv–paths of the graph.

Suppose that $b_1 \notin V(P_2)$. Then there exists a b_1b_2–path of G, namely b_1QvRb_2, such that $V(P_1) \cap$

$V(b_1QvRb_2) = \{b_1\}$ and $V(P_2) \cap V(b_1QvRb_2) = \{b_2\}$, a contradiction by Lemma 2.11. Therefore $b_1 \in V(P_2)$. Since $v \notin V(P_1) \cup V(P_2)$, it follows that $b_1 \notin V(uP_2b_2)$. Hence $b_1 \in V(uP_2a_2)$ and therefore $a_1P_1b_1$ is a subpath of P_2. A similar argument shows that the result holds when P_2 is of Type (B)(e). $\square$

Lemma 2.16. *Let G be a theta-Hamiltonian-tree graph with unique core $\hat{G}$ and let P_1 and P_2 be Type (B) longest paths of G. Let $a_1, b_1 \in V(\hat{G})$ be the two base vertices of P_1. If P_1 is of Subtype (a) and P_2 is of Subtype (b), then $a_1, b_1 \in V(P_2)$.*

Proof. Assume that P_1 is of Type (B)(a) and P_2 is of Type (B)(b). Let u and v be the two vertices of $\hat{G}$ such that $deg_{\hat{G}}(u) \geq 3$ and $deg_{\hat{G}}(v) \geq 3$. Since P_1 is of Type (B)(a) and $u, v \notin V(P_1)$, there exists a uv–path Q of G such that $a_1P_1b_1$ is a subpath of Q. Without loss of generality, we assume that these vertices are in the order u, a_1, b_1, v in Q. Since P_2 is of Type (B)(b), $u, v \in V(P_2)$ and P_2 has two base vertices $a_2, b_2 \in V(\hat{G})$. Without loss of generality, we assume that these vertices are in the order a_2, u, v, b_2 in P_2. If Q is a subpath of P_2, then $a_1, b_1 \in V(P_2)$ and we are done. It remains to consider the case in which $E(Q) \backslash E(P_2) \neq \emptyset$.

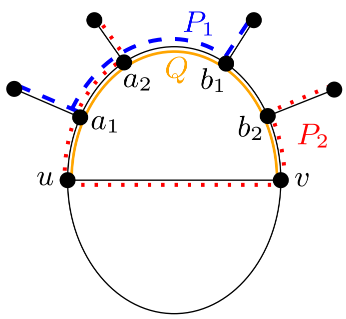

Figure 2.16: An example of a theta-Hamiltonian-tree graph with longest path P_1 (blue dashed line) of Type (B)(a) and longest path P_2 (red dotted line) of Type (B)(b). Path Q (orange solid line) is a uv–path of the graph.

Since P_2 is of Type (B)(b), a_2 and b_2 are vertices of one uv–path of G. Since P_1 and P_2 have a common vertex by Lemma 1.1, then $a_2, b_2 \in V(Q)$ and $a_1 \in V(P_2)$ or $b_1 \in V(P_2)$; assume without loss of generality that $a_1 \in V(P_2)$ and $b_1 \notin V(P_2)$. Then uP_2a_2 is a subpath of Q, and $a_1 \in V(uP_2a_2)$. An example of G is shown in Figure 2.16. If $b_1 \notin V(P_2)$, then there exists a b_1b_2–path of G, namely b_1Qb_2, such that $V(P_1) \cap V(b_1Qb_2) = \{b_1\}$ and $V(P_2) \cap V(b_1Qb_2) = \{b_2\}$, which is a forbidden configuration by

Lemma 2.11. Therefore $a_1, b_1 \in V(P_2)$. $\square$

Lemma 2.17. *Let G be a theta-Hamiltonian-tree graph with unique core $\hat{G}$ and let P_1 and P_2 be Type (B) longest paths of G. Let $a_1, b_1 \in V(\hat{G})$ be the two base vertices of P_1. If P_1 is of Subtype (a) and P_2 is of Subtype (c) then $a_1, b_1 \in V(P_2)$. Furthermore, $a_1P_1b_1$ is a subpath of P_2.*

Proof. Assume that P_1 is of Type (B)(a) and P_2 is of Type (B)(c). Let u and v be the two vertices of $\hat{G}$ such that $deg_{\hat{G}}(u) \geq 3$ and $deg_{\hat{G}}(v) \geq 3$. Since P_1 is of Type (B)(a), there exists a uv–path Q of G such that $a_1P_1b_1$ is a subpath of Q. If Q is a subpath of P_2, then $a_1P_1b_1$ is a subpath of P_2 and we are done. It remains to consider the case in which Q is not a subpath of P_2.

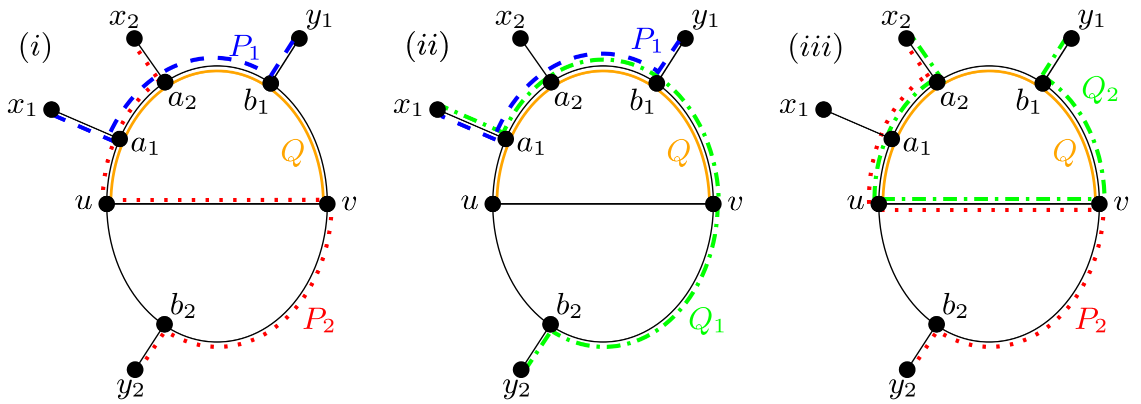

Figure 2.17: Three diagrams of a theta-Hamiltonian-tree graph showing (i) longest path P_1 of Type (B)(a) (blue dashed line) and longest path P_2 of Type (B)(c) (red dotted line), (ii) path Q_1 (green dash-dotted line) and path P_1, and (iii) path Q_2 (green dash-dotted line) and path P_2. In all three diagrams, path Q (orange solid line) is a uv–path of the graph.

Let $a_1P_1x_1$ and $b_1P_1y_1$ be the tails of P_1, where x_1 and y_1 are the endpoints of P_1. Without loss of generality, we assume that $a_1 \in V(uQb_1)$. Let $a_2P_2x_2$ and $b_2P_2y_2$ be the tails of P_2, where x_2 and y_2 are the endpoints of P_2 and $a_2, b_2 \in V(\hat{G})$. Since P_1 and P_2 have a common vertex by Lemma 1.1 and P_2 is of Type (B)(c), it follows that uQa_1 or vQb_1 is a subpath of P_2. Suppose that uQa_1 is a subpath of P_2. Then $a_2 \in V(Q)$ or $b_2 \in V(Q)$; without loss of generality, we assume that $a_2 \in V(Q)$. An example of G is shown in Figure 2.17(i).

Suppose that $b_1 \notin V(P_2)$. Let Q_1 be the path $x_1P_1b_1QvP_2y_2$ of G, an example of which is shown in

Figure 2.17(ii). Since P_1 is a longest path of G,

$$|P_1| \geq |Q_1|$$
$$|x_1P_1b_1P_1y_1| \geq |x_1P_1b_1QvP_2y_2|$$
$$|b_1P_1y_1| \geq |b_1Qv| + |vP_2y_2|. \tag{2.11}$$

Let Q_2 be the path $x_2P_2vQb_1P_1y_1$ of G, an example of which is shown in Figure 2.17(iii). Since P_2 is a longest path of G,

$$|P_2| \geq |Q_2|$$
$$|x_2P_2vP_2y_2| \geq |x_2P_2vQb_1P_1y_1|$$
$$|vP_2y_2| \geq |vQb_1| + |b_1P_1y_1|. \tag{2.12}$$

From 2.11 and 2.12 we have $|vQb_1| \leq 0$, a contradiction since $v \notin V(P_1)$. Hence $b_1 \in V(P_2)$. Since $v \notin V(P_1)$, then $b_1 \notin V(uP_2b_2)$, and hence $b_1 \in V(uP_2a_2)$. Therefore $a_1P_1b_1$ is a subpath of P_2. An analogous argument shows that the result holds when vQb_1 is a subpath of P_2. □

2.3.4.3 Proof of Lemma 2.10

Recall that Lemma 2.10 states that if all of the longest paths of a theta-Hamiltonian-tree graph G with a unique core are of Type (B), then G has a Gallai vertex.

Proof of Lemma 2.10. Let G be a theta-Hamiltonian-tree graph and let $\hat{G}$ be the unique core of G. Let u and v be the two vertices of $\hat{G}$ such that $deg_{\hat{G}}(u) \geq 3$ and $deg_{\hat{G}}(v) \geq 3$. Let $\mathcal{P}$ be the set of longest paths of G, and assume that every path in $\mathcal{P}$ is of Type (B). We consider the subtypes of the paths in $\mathcal{P}$.

First, by Lemma 2.12, there does not exist both a path of Subtype (d) and path of Subtype (e) in $\mathcal{P}$. Suppose that there does not exist a path in $\mathcal{P}$ of Subtype (e), so each path in $\mathcal{P}$ is of Subtype (a), (b), (c), or (d).

We then consider two cases: (i) there is no path in $\mathcal{P}$ of Subtype (a) and (ii) there is at least one path in $\mathcal{P}$ of Subtype (a). In case (i), u is a common vertex of all of the paths in $\mathcal{P}$, and we are done. Consider case (ii) and let $\mathcal{P}' \subseteq \mathcal{P}$ be the set of longest paths of G of Subtype (a). By Lemma 2.14, there exists a vertex $x \in V(\hat{G})$ that is a base vertex of each path $P \in \mathcal{P}'$. If $\mathcal{P}' = \mathcal{P}$, that is, every path in $\mathcal{P}$

is of Subtype (a), then x is a common vertex of all of the paths in $\mathcal{P}$, and we are again done. Suppose instead that there is at least one path in $\mathcal{P}\backslash\mathcal{P}'$. Each of the longest paths in $\mathcal{P}\backslash\mathcal{P}'$ is of Subtype (b), (c), or (d). By Lemmas 2.15 to 2.17, for each path $Q \in \mathcal{P}\backslash\mathcal{P}'$, $x \in V(Q)$. Hence x is a Gallai vertex of G.

An analogous argument shows that the result holds when there does not exist a path in $\mathcal{P}$ of Subtype (d). □

2.3.5 Proof of Proposition 2.5

We return to Proposition 2.5, and prove that every theta-Hamiltonian-tree graph with a unique core has a Gallai vertex.

Proof of Proposition 2.5. Let G be a theta-Hamiltonian-tree graph with a unique core. Then the longest paths of G are all of Type (A), all of Type (B), or at least one of Type (A) and at least one of Type (B). Then, by Corollary 2.8, Lemma 2.10, and Lemma 2.9, G has a Gallai vertex, as required. □

2.4 Proof of Theorem 2.1

We return to Theorem 2.1, and prove that every theta-Hamiltonian-tree graph has a Gallai vertex.

Proof of Theorem 2.1. Let G be a theta-Hamiltonian-tree graph. If G has more than one core, then by Proposition 2.4, G has a Gallai vertex. If G has a unique core, then by Proposition 2.5, G has a Gallai vertex. □

2.5 Concluding remarks

In this chapter, we proved that every theta-Hamiltonian-tree graph has a Gallai vertex (Theorem 2.1). There are two natural avenues for further investigation. We can consider other subclasses of generalised theta graphs similar to theta-Hamiltonian-tree graphs, where the cores are theta blocks but the core-touching subgraphs may have different properties, for example where the core-touching subgraphs may be series-parallel graphs or cactus graphs. We can also consider graphs similar to theta-Hamiltonian-tree graphs where the cores are not theta graphs but, for example chordal graphs or series-parallel graphs.

Another possibility is to combine the two options above, defining a class of graphs similar to theta-Hamiltonian-tree graphs where the cores are of a different class of graphs and the core-touching subgraphs have different properties. In Chapter 4, we consider graphs similar to theta-Hamiltonian-tree graphs in which the cores of the graph are non-trivial blocks, and the core-touching subgraphs are all trees. Such graphs have a unique core, and it follows that this class is the class of graphs that have exactly one non-trivial block. We use computational methods to investigate Conjecture 1 for these graphs, restricting number of vertices with particular degree in the core of the graph. For example, we consider such graphs where the core has at most 9 vertices of degree 3 or more (in the subgraph consisting of the core), and find that every three longest paths have a common vertex in these graphs.

We investigated a class of graphs defined similarly to theta-Hamiltonian-tree graphs with the addition of an edge inserted into the core, however, our methods do not naturally extend to this class of graphs. Let G be a theta-Hamiltonian-tree graph, and consider the case in which G has a unique core $\hat{G}$. Let u and v be the two vertices of $\hat{G}$ such that $deg_{\hat{G}}(u) \geq 3$ and $deg_{\hat{G}}(v) \geq 3$, and let $\mathcal{Q}$ be the set of internally disjoint paths of $\hat{G}$ with endpoints u and v. Let G' be the graph obtained from G by inserting an edge xy into $\hat{G}$, where x and y are interior vertices of distinct uv–paths in $\mathcal{Q}$. An example of such a graph is shown in Figure 2.18(i). Note that G is a subgraph of G' and so $\mathcal{Q}$ is a set of internally disjoint paths of G'.

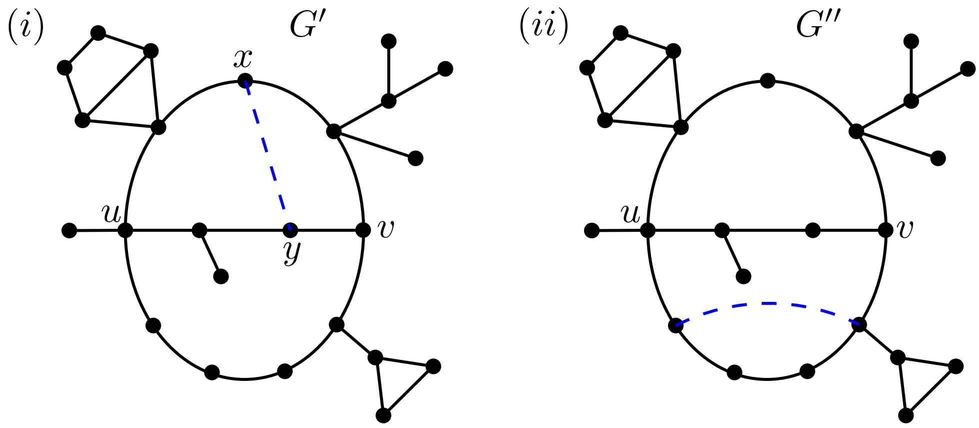

Figure 2.18: (i) A graph G' that is a theta-Hamiltonian-tree graph with an inserted edge xy (blue dashed line). (ii) A graph G'' that is a theta Hamiltonian-tree graph with an inserted edge (blue dashed line) whose endpoints are internal vertices of one uv–path of G''.

We can define two types of longest paths of G' analogous to those defined in Section 2.3.1 for a theta-Hamiltonian-tree graph, where Type (A) longest paths of G' have no edge that is an edge of the core of G', and Type (B) longest paths of G' have at least one edge that is an edge of the core of G'. We can then also define five subtypes of Type (B) longest paths of G' analogous to Subtypes (a) – (e) defined in Section 2.3.4 for theta-Hamiltonian-tree graphs. However, there are more cases for each of these subtypes of Type (B) longest paths of G' than there were for G, due to the additional edge xy. It may be possible to use the techniques of this chapter on these cases, however, careful consideration of these new cases is needed to obtain inequalities similar to those used in the proof in Section 2.3.4. New methods may perhaps be required for these cases, if indeed such graphs G' have a Gallai vertex.

Graphs that may be more amenable to our approach are theta-Hamiltonian-tree graphs G with an edge inserted whose endpoints are internal vertices of one of the uv–paths of G, as shown in the example in Figure 2.18(ii). More generally, the class of graphs similar to theta-Hamiltonian-tree graphs where the cores are two-terminal series-parallel graphs may be approachable using the methods in this chapter.

Chapter 3

Graphs with cyclomatic number 6

3.1 Introduction

In this chapter, we add to the classes of graphs for which Conjecture 1 holds, that is, the classes of graphs for which every three longest paths have a common vertex.

The *cyclomatic number* of a graph G is $|E(G)|-|V(G)|+1$ and is also known as the *circuit rank*, *cycle rank*, or *nullity* of the graph. Hence, a graph with n vertices and cyclomatic number c has $(n-1)+c$ edges or, equivalently, c more edges than its spanning tree. An equivalent definition is that the cyclomatic number of a graph G is the minimum size subset of edges of $E(G)$ whose removal from G produces a tree. The cyclomatic number of a graph relates to the number of cycles of the graph, as we will explore below and in Section 3.2. In the rest of this chapter, all graphs are connected unless stated otherwise.

The main result of this chapter is the following theorem.

Theorem 3.1. *Let G be a graph. If G has cyclomatic number at most 6 then every three longest paths of G have a common vertex.*

The relationship between a graph's cyclomatic number and its number of cycles is central to this chapter. Let $\mathcal{S}$ be a set of subgraphs of a graph G. The graphs in $\mathcal{S}$ are *independent* if, for each $S_1 \in \mathcal{S}$, there exists $e \in E(S_1)$ such that $e \notin E(S_2)$ for every $S_2 \in \mathcal{S}\backslash\{S_1\}$. Theorem 3.1 relies on the result that the maximum number of independent cycles in a graph is exactly its cyclomatic number [see 4, p. 27–29]. In order to prove Theorem 3.1, we suppose that there is a minimal (with respect to edges) counterexample $\check{H}$ to Conjecture 1 and show that such a graph has at least seven independent cycles. The graph $\check{H}$ has three longest paths that do not have a common vertex and is the union of these three longest paths. In Section 3.3 we use a result from Axenovich [2] to show that $\check{H}$ has a set $\mathcal{C}$ of at least six cycles, two in each pairwise union of its three longest paths. We then show that there is a seventh cycle C that is not in any pairwise union of the three longest paths of $\check{H}$. Lemmas 3.16, 3.18, 3.19 and 3.20 in Section 3.5 show that any three, four, five, or six of the cycles in $\mathcal{C}$ are independent, and then Lemma 3.21 shows that the seventh cycle C is also independent of the cycles of $\mathcal{C}$. Hence, a minimal counterexample to Conjecture 1 has at least seven independent cycles, and Theorem 3.1 follows.

We begin with some preliminary definitions and results.

3.2 Preliminaries

3.2.1 The mod 2 sum operation

In this section, we define an operation on subgraphs of a graph which will be used throughout this chapter, particularly with regard to the cycles of a graph, the *mod 2 sum* (or *symmetric difference*) operation. First note that the set of subgraphs of a graph together with this mod 2 sum operation forms an abelian group [see 34] where the identity element is the graph with no vertices or edges and the inverse of a subgraph is itself.

Let G be a graph. For two subgraphs G_1 and G_2 of G, the *mod 2 sum* of G_1 and G_2 is $G_1 \oplus G_2 = S$, where S is the edge-induced subgraph of G with edge set $E(S) = (E(G_1) \cup E(G_2)) \setminus (E(G_1) \cap E(G_2))$, that is, the edges of G that are edges of G_1 or G_2, but not both. For convenience, we define the *mod 2 sum* of a single subgraph G_1 of G to be G_1.

Consider the graph in Figure 3.1(i) with cycles $C_1 = v_2v_3v_4v_5v_2$ (blue dashed line), $C_2 = v_3v_6v_5v_4v_3$ (red dotted line), and $C_3 = v_2v_3v_6v_5v_2$, (green dash-dotted line). In this graph, $C_1 \oplus C_2 = C_3$.

Since each subgraph is its own inverse, if $G_1 \oplus G_2 = G_3$, then $G_1 \oplus G_3 = G_2$ and $G_2 \oplus G_3 = G_1$, as seen for cycles C_1, C_2, and C_3 in the example in Figure 3.1(i). This property will be used frequently throughout this chapter.

If subgraphs G_1 and G_2 of G are edge-disjoint, then $G_1 \oplus G_2 = G_1 \cup G_2$. An example of this is shown in Figure 3.1(ii), with cycles $C_1 = v_1v_2v_4v_3v_1$ (blue dashed line) and $C_2 = v_5v_6v_7v_5$ (red dotted line), where $C_1 \oplus C_2 = C_1 \cup C_2$.

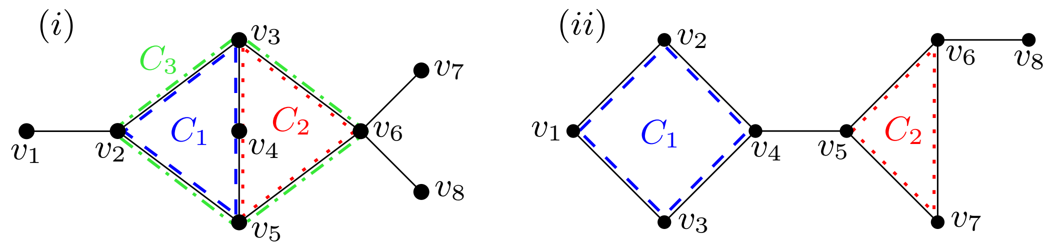

Figure 3.1: (i) A graph with three cycles C_1, C_2, and C_3 (shown with blue dashed, red dotted, and green dash-dotted lines respectively), with $C_1 \oplus C_2 = C_3$. (ii) A graph with two cycles C_1 and C_2 (shown with blue dashed and red dotted lines respectively), with $C_1 \oplus C_2 = C_1 \cup C_2$.

Let $G_1, G_2, \ldots, G_k$ $(k \geq 3)$ be subgraphs of G. The *mod 2 sum* of $G_1, G_2, \ldots, G_k$ is $(((G_1 \oplus G_2) \oplus G_3) \oplus \cdots \oplus G_k) = S_k$ where S_k is the edge-induced subgraph of G with edge set $E(S_k) = (E\big(((G_1 \oplus G_2) \oplus G_3) \oplus \cdots \oplus G_{k-1}\big) \cup E(G_k)) \setminus (E\big(((G_1 \oplus G_2) \oplus G_3) \oplus \cdots \oplus G_{k-1}\big) \cap E(G_k))$. Equivalently, the mod 2 sum of the subgraphs $G_1, G_2, \ldots, G_k$ of G is the subgraph of G consisting of exactly the edges of G that are edges of an odd number of the subgraphs $G_1, G_2, \ldots, G_k$. Note that the order in which we take the mod 2 sum of the subgraphs $G_1, G_2, \ldots, G_k$ does not matter as the mod 2 sum operation is both commutative and associative.

Consider the graph in Figure 3.2(i), with cycles $C_1 = v_1v_2v_4v_5v_1$ (shown by the blue dashed line), $C_2 = v_2v_3v_6v_4v_2$ (shown by the red dotted line), $C_3 = v_4v_6v_7v_5v_4$ (shown by the green dash-dotted line), and $C_4 = v_1v_2v_3v_6v_7v_5v_1$ (shown by the yellow solid line). In this graph, $C_1 \oplus C_2 \oplus C_3 = C_4$. The graph in Figure 3.2(ii) has cycles $C_1 = v_1v_2v_3v_4v_5v_1$ (shown by the blue dashed line), $C_2 = v_3v_6v_4v_3$ (shown by the red dotted line), $C_3 = v_2v_7v_5v_4v_3v_2$ (shown by the green dash-dotted line), and $C_4 = v_1v_2v_7v_5v_1$ (shown by the yellow solid line). In this graph, $C_1 \oplus C_2 \oplus C_3 = C_2 \cup C_4$.

3.2.2 Cycle decomposition of a graph

A graph G has a *cycle decomposition* if there is a set $\mathcal{C}$ of pairwise edge-disjoint cycles of G such that $E(G) = \bigcup_{C \in \mathcal{C}} E(C)$. For example, the graph in Figure 3.3(i) has a cycle decomposition consisting of the cycles $C_1 = v_1v_2v_4v_1$ (blue dashed lines), $C_2 = v_2v_3v_5v_2$ (red dotted lines), and $C_3 = v_4v_5v_6v_4$ (green

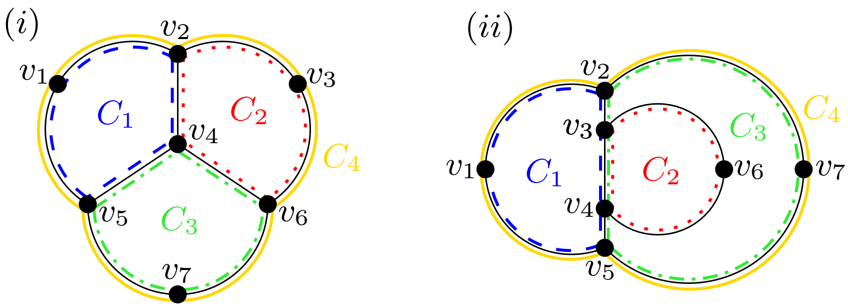

Figure 3.2: (i) A graph with cycles C_1, C_2, C_3, and C_4 (shown with blue dashed, red dotted, green dash-dotted, and yellow solid lines respectively), with $C_1 \oplus C_2 \oplus C_3 = C_4$. (ii) A graph with cycles C_1, C_2, C_3, and C_4 (shown with blue dashed, red dotted, green dash-dotted, and yellow solid lines respectively), with $C_1 \oplus C_2 \oplus C_3 = C_2 \cup C_4$.

dash-dotted lines). However, the graph in Figure 3.3(ii) does not have a cycle decomposition since there is no set $\mathcal{C}$ of pairwise edge-disjoint cycles of the graph in which every edge of the graph is an edge of a cycle in $\mathcal{C}$.

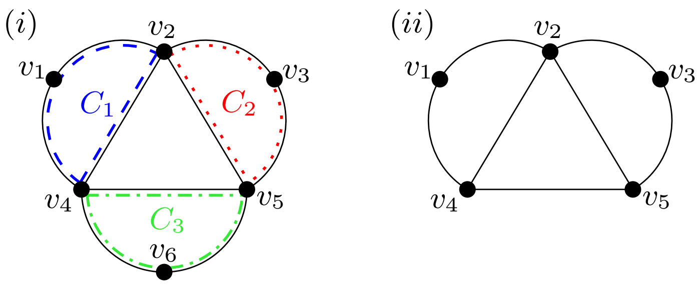

Figure 3.3: (i) A graph with cycle decomposition $\{C_1, C_2, C_3\}$ (blue dashed, red dotted, and green dash-dotted lines respectively). (ii) A graph with no cycle decomposition.

The following theorem is due to Veblen [37].

Theorem 3.2. *A graph G has a cycle decomposition if and only if every vertex of G has even degree.*

We use this theorem to prove the following lemma.

Lemma 3.3. *Let G be a graph. For every non-empty set $\mathcal{C}$ of cycles of G, the mod 2 sum G' of the cycles in $\mathcal{C}$ has a cycle decomposition.*

Proof. We show that every vertex of G' has even degree, using induction on the size of $\mathcal{C}$. If $|\mathcal{C}| = 1$ then G' is a cycle, and hence every vertex of G' has degree two.

Assume that if $1 \leq |\mathcal{C}| \leq k$, then every vertex of G' has even degree. Now suppose that $|\mathcal{C}| = k+1$. Let $C \in \mathcal{C}$, and let A be the mod 2 sum of the k cycles in $\mathcal{C}\backslash\{C\}$. By our earlier assumption, every vertex of A has even degree. If A and C are vertex-disjoint then $A \oplus C = A \cup C$ and hence every vertex of $A \oplus C$ has even degree. It remains to consider the case in which A and C have at least one common vertex.

Let $G' = A \oplus C$. Let $A' = A \cap G'$ and let $C' = C \cap G'$. Then $G' = A' \cup C'$. Let $v \in V(G')$. If $v \in V(A')\backslash V(C')$ then $\deg_{G'}(v) = \deg_A(v)$ which is even. If $v \in V(C')\backslash V(A')$ then $\deg_{G'}(v) = \deg_C(v) = 2$. Suppose $v \in V(A') \cap V(C')$.

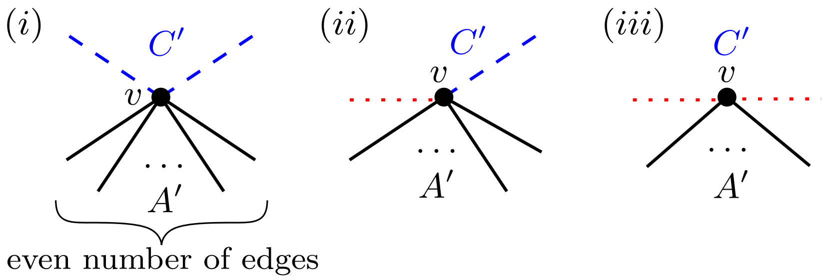

Figure 3.4: Examples of a vertex $v \in V(A') \cap V(C')$ and its incident edges, which are edges of A' (black solid lines), edges of C' (blue dashed lines), or edges of neither A' nor C' (red dotted lines), in the cases where (i) $\deg_{C'}(v) = 2$, (ii) $\deg_{C'}(v) = 1$, and (iii) $\deg_{C'}(v) = 0$.

The value of $\deg_{C'}(v)$ is 0, 1, or 2. First, suppose that $\deg_{C'}(v) = 2$. Then there exists no edge $e \in E(G)$ incident with v such that $e \in E(A) \cap E(C)$, as shown in the example in Figure 3.4(i). It follows that $\deg_{G'}(v) = \deg_{A'}(v) + \deg_{C'}(v) = \deg_A(v) + \deg_C(v) = \deg_A(v) + 2$ which is even. Next, suppose that $\deg_{C'}(v) = 1$. Then there exists exactly one edge $e \in E(G)$ incident with v such that $e \in E(A) \cap E(C)$, as shown in the example in Figure 3.4(ii). It follows that $\deg_{G'}(v) = \deg_{A'}(v) + \deg_{C'}(v) = \deg_A(v) - 1 + \deg_C(v) - 1$ which is even. Finally, suppose that $\deg_{C'}(v) = 0$. Then there exist exactly two edges $e, f \in E(G)$ incident with v such that $e, f \in E(A) \cap E(C)$, as shown in the example in Figure 3.4(iii). If $\deg_A(v) = 2$ then $v \notin V(G')$, a contradiction. If $\deg_A(v) > 2$ then $v \in V(G')$ and $\deg_{G'}(v) = \deg_{A'}(v) + \deg_{C'}(v) = \deg_A(v) - 2 + \deg_C(v) - 2 = \deg_A(v) - 2$ which is even.

Therefore, every vertex of G' has even degree, and hence, by Theorem 3.2, G' has a cycle decomposition, as required. □

3.2.3 Independent cycles

We return to the relationship between the number of cycles of a graph and its cyclomatic number.

We redefine the independence of a set of cycles in terms of the mod 2 sum operation on cycles. Let G be a graph with at least one cycle. A set $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ $(k \geq 1)$ of cycles of G is *dependent* if there exists a cycle $C_i \in \mathcal{C}$ $(1 \leq i \leq k)$ such that C_i is the mod 2 sum of all of the cycles in $\mathcal{C}\backslash\{C_i\}$. Otherwise, the set of cycles $\mathcal{C}$ is independent. For example, in Figure 3.1, the set of cycles $\{C_1, C_2, C_3\}$ is dependent since $C_1 \oplus C_2 = C_3$, but each pair of these cycles is independent.

The following lemma is from [4, p. 27-29].

Lemma 3.4. *The cyclomatic number of a graph G is equal to the maximum number of independent cycles of G.*

For example, in Figure 3.1(i), the graph has two independent cycles and has cyclomatic number $9 - 8 + 1 = 2$, and in Figure 3.2(i), the graph has three independent cycles and cyclomatic number $9 - 7 + 1 = 3$.

Note that a maximum-size set of independent cycles of a graph G need not be unique. For example, in Figure 3.1(a), each pair of the three cycles C_1, C_2, and C_3 is a maximum size set of independent cycles of the graph.

3.2.4 Graph colouring

In this chapter we use a broader definition of graph colouring than is usual. In a *partial colouring* of a graph G, each edge and vertex of G is either left uncoloured or assigned one or more colours from a set of colours. Such a graph is called a *partially coloured* graph. If G has no edges or vertices left uncoloured, then this is a *colouring* of G and G is a *coloured* graph. In a *painting* of G with colour c, each edge and vertex of G is assigned the colour c. Such a graph is called a *painted* graph, and G is said to be *painted* with colour c. Hence if G is painted with one or more colours, G is a coloured graph.

In a (partially) coloured graph G, for each $e \in E(G)$, let $\text{col}(e)$ be the set of colours assigned to e, and for each $v \in V(G)$, let $\text{col}(v)$ be the set of colours assigned to v. Our notion of colouring allows the colour set of an edge of G to be different to the colour sets of its endpoints. Moreover, if $e = uv$ and G

is (partially) coloured by painting one of more of its subgraphs, then $\mathrm{col}(e) \subseteq \mathrm{col}(v)$ and $\mathrm{col}(e) \subseteq \mathrm{col}(u)$. The *colour set* of G is

$$\mathrm{col}(G) = \left(\bigcup_{e \in E(G)} \mathrm{col}(e) \right) \cup \left(\bigcup_{v \in V(G)} \mathrm{col}(v) \right).$$

Suppose that G is a coloured graph. If there exists $S \subseteq \mathrm{col}(G)$ such that every edge and vertex of G is coloured with at least one colour from S, then G is an *S–graph*. For example, if $\mathrm{col}(G) = \{\mathrm{blue}, \mathrm{red}, \mathrm{green}\}$ and every edge and vertex of G is coloured blue or red, then G is a $\{\mathrm{blue}, \mathrm{red}\}$–graph. If G is coloured in such a way that G is also painted with each of the colours of S, then G is a *totally S–graph*. So a graph G which is painted red and painted blue is a totally $\{\mathrm{blue}, \mathrm{red}\}$–graph. If G is an S–graph for some $S \subseteq \mathrm{col}(G)$ and $|S| = 1$, then G is *uni-coloured*; if $|S| = 2$, then G is *bi-coloured*; and if $|S| = 3$, then G is *tri-coloured*. If G is a totally S–graph, then G is *totally uni-coloured*, *totally bi-coloured*, and *totally tri-coloured* respectively in these three cases. These definitions naturally extend to subgraphs of G.

The graph in Figure 3.5 is a $\{\mathrm{blue}, \mathrm{red}\}$–graph. The path tuv is a totally $\{\mathrm{blue}, \mathrm{red}\}$–path, and the path vxz is a $\{\mathrm{red}\}$–path.

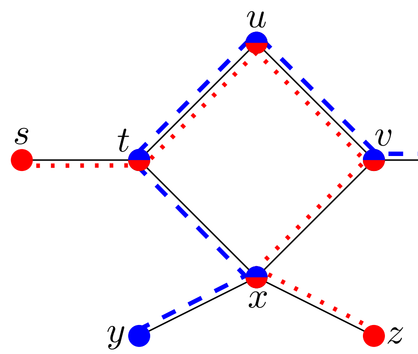

Figure 3.5: A $\{\mathrm{blue}, \mathrm{red}\}$–graph (blue line dashed, red line dotted).

If G is a $\{c_1, c_2\}$–graph, for some colours c_1 and c_2, that has j maximal $\{c_1\}$–paths and k maximal $\{c_2\}$–paths, $j, k \geq 1$, then G is a $[j\text{-}c_1, k\text{-}c_2]$–graph. For example, in Figure 3.5, the cycle $tuvxt$ is a $[1\text{-blue}, 1\text{-red}]$–cycle. When the number of maximal paths of a particular colour is not known or not relevant, the prefix may be omitted.

In the remaining figures in this chapter, we may show some, but not necessarily all, vertices or edges of a graph. Additionally, only a subset of the colour set of each vertex or edge may be shown.

3.3 Properties of a minimal counterexample

In our approach to proving Theorem 3.1, we consider the properties of a minimal (with respect to edges) counterexample $\check{H}$ to Conjecture 1. Recall that $\check{H}$ is a connected graph with three longest paths, P_1, P_2, and P_3, that do not have a common vertex, and since $\check{H}$ is minimal, $\check{H}$ is the union of these three longest paths; that is, $\check{H} = P_1 \cup P_2 \cup P_3$. We prove that $\check{H}$ has at least seven independent cycles and therefore cyclomatic number at least seven.

Throughout the rest of this chapter, we use a colouring of $\check{H}$ obtained by painting P_1, P_2, and P_3 with the colours blue, red, and green respectively. The following proposition summarises some basic properties of $\check{H}$.

Proposition 3.5.

(i) *$\check{H}$ has no tri-coloured vertices and no tri-coloured edges.*

(ii) *$\check{H}$ has no uni-coloured cycles.*

(iii) *Let $v \in V(\check{H})$ and let $E' \subseteq E(\check{H})$ be the set of edges incident with v. Then the colour set of v is the union of the colour sets of its incident edges, that is, $col(v) = \bigcup_{e \in E'} col(e)$.*

(iv) *For each vertex $v \in V(\check{H})$, $1 \leq |col(v)| \leq 2$, and for each edge $e \in E(\check{H})$, $1 \leq |col(e)| \leq 2$.*

Proof.

(i) Since P_1, P_2, and P_3 do not have a common vertex, $\check{H}$ has no tri-coloured vertices or tri-coloured edges.

(ii) Since P_1, P_2, and P_3 are paths painted with distinct colours, $\check{H}$ has no uni-coloured cycles.

(iii) Paths P_1, P_2, and P_3 each have length at least one since they do not all have a common vertex and $\check{H}$ is connected. Therefore, by the colouring of $\check{H}$, if $c \in \mathrm{col}(v)$ for some $c \in \mathrm{col}(\check{H})$, then there exists at least one edge $e \in E'$ such that $c \in \mathrm{col}(e)$. Additionally, since P_1, P_2, and P_3 are painted, then $\mathrm{col}(e) \subseteq \mathrm{col}(v)$ for every $e \in E'$. Hence $\mathrm{col}(v) = \bigcup_{e \in E'} \mathrm{col}(e)$.

(iv) Since $\check{H} = P_1 \cup P_2 \cup P_3$, then $|\mathrm{col}(v)| \geq 1$ for each $v \in V(\check{H})$ and $|\mathrm{col}(e)| \geq 1$ for each $e \in E(\check{H})$. By (i), $|\mathrm{col}(v)| \leq 2$ for each $v \in V(\check{H})$ and $|\mathrm{col}(e)| \leq 2$ for each $e \in E(\check{H})$. $\square$

3.3.1 Forbidden configurations in $\check{H}$

Let G be a graph and let P_1, P_2, and P_3 be three longest paths of G. Suppose that these three paths do not have a common vertex. Recall that, for a path P and vertices $x, y \in V(P)$, xPy denotes the subpath of P with endpoints x and y. Axenovich [2] defines two configurations that are forbidden in the union of these three longest paths of G, restated here.

Configuration 1: A cycle of G which is the union of three internally disjoint subpaths S_1, S_2, and S_3 of P_1, P_2, and P_3 respectively, such that

(i) for every interior vertex u of S_1 or S_3, $u \notin V(S_2)$, and

(ii) for every interior vertex v of S_2 or S_3, $v \notin V(P_1)$.

Configuration 2: A subpath xP_1y of P_1 such that

(i) $x \in V(P_2)$, $y \in V(P_3)$;

(ii) for every internal vertex v of xP_1y, $v \notin V(P_2) \cup V(P_3)$; and

(iii) $P_2 - x$ is the union of two paths P_2' and P_2'', and $P_3 - y$ is the union of two paths P_3' and P_3'', such that $V(P_2' \cup P_3') \cap V(P_2'' \cup P_3'') = \emptyset$ or $V(P_2' \cup P_3'') \cap V(P_2'' \cup P_3') = \emptyset$.

Examples of these two forbidden configurations are shown in Figure 3.6. The following lemma is restated from [2, Lemma 1].

Lemma 3.6. *Let G be a graph and let P_1, P_2, and P_3 be three longest paths of G. Suppose that these three paths do not have a common vertex. Then $P_1 \cup P_2 \cup P_3$ does not have a subgraph with Configuration 1 or Configuration 2.*

We colour G by painting P_1 blue, P_2 red, and P_3 green and define another configuration that is forbidden in the union of paths P_1, P_2, and P_3 of G.

Configuration 3: A path Q of G that is a subpath of a $\{c_1, c_2\}$–cycle and a $\{c_1, c_3\}$–cycle, where $c_1, c_2, c_3 \in \mathrm{col}(G)$, such that:

(i) there is at least one vertex $u \in V(Q)$ with $\mathrm{col}(u) = \{c_2, c_3\}$ and

(ii) there is at least one vertex $v \in V(Q)$ with $c_1 \in \mathrm{col}(v)$.

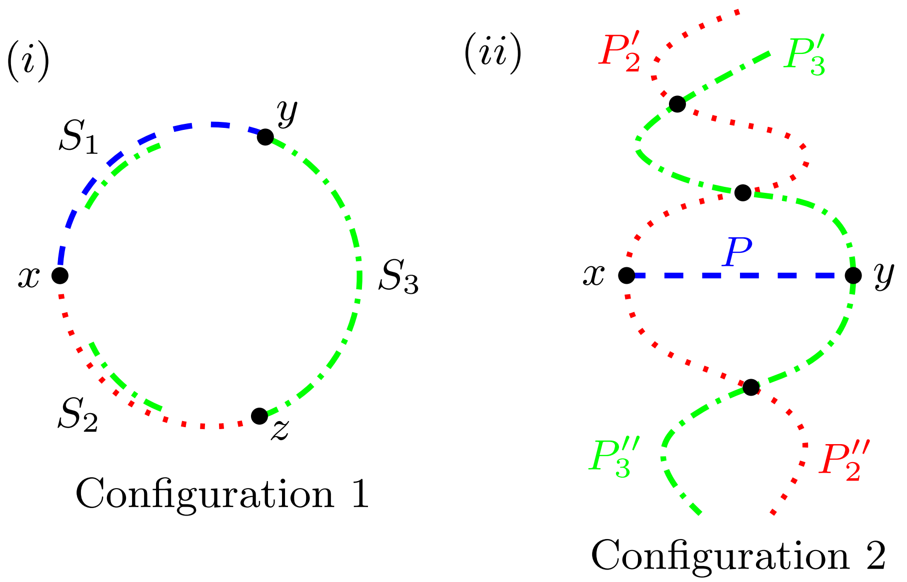

Figure 3.6: Subpaths of P_1, P_2, and P_3 shown by blue dashed, red dotted, and green dash-dotted lines respectively. (i) An example of Configuration 1, where S_1 is the {blue}–path with endpoints x and y, S_2 is the {red}–path with endpoints x and z, and S_3 is the {green}–path with endpoints y and z. (ii) An example of Configuration 2.

An example of this configuration is shown in Figure 3.7.

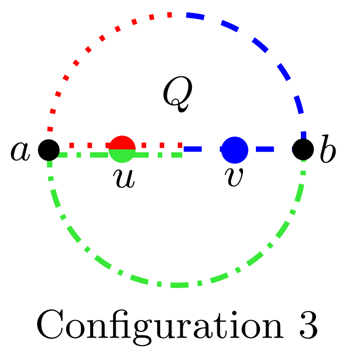

Figure 3.7: Subpaths of P_1, P_2, and P_3 shown by blue dashed, red dotted, and green dash-dotted lines respectively. This figure shown an example of Configuration 3 where c_1, c_2, and c_3 are blue, red, and green respectively. Path Q, with endpoints a and b, is a subpath of a $\{\text{blue}, \text{red}\}$–cycle and a subpath of a $\{\text{blue}, \text{green}\}$–cycle.

Lemma 3.7. *Let G be a graph and let P_1, P_2, and P_3 be three longest paths of G. Suppose that these three paths do not have a common vertex. Colour G by painting P_1 blue, P_2 red, and P_3 green. Then*

$P_1 \cup P_2 \cup P_3$ *does not have a subgraph with Configuration 3.*

Proof. Assume that $P_1 \cup P_2 \cup P_3$ has a subgraph with Configuration 3. Let path Q and vertices u and v be as defined in the definition of Configuration 3, and assume without loss of generality that c_1, c_2, and c_3 are blue, red, and green respectively. Let uQw, $w \in V(uQv)$, be the maximal totally $\{\text{red}, \text{green}\}$–subpath of uQv with endpoint u. Since v is not a tri-coloured vertex of G by Proposition 3.5(i), then $w \neq v$. It follows that there exists an edge $e \in E(wQv)$ incident to w. Then $\text{col}(e) \neq \{\text{red}, \text{green}\}$ since uQw is maximal. Additionally, $\text{col}(e) \neq \{\text{green}\}$ since uQv is a path of a $\{\text{blue}, \text{red}\}$–cycle, and $\text{col}(e) \neq \{\text{red}\}$ as uQv is a path of a $\{\text{blue}, \text{green}\}$–cycle. Hence blue $\in \text{col}(e)$ and therefore w is a tri-coloured vertex of G, a contradiction. □

3.3.2 The number of cycles of $\check{H}$

Using Configurations 1 and 2, Axenovich proves the lemma restated below [2, Lemma 3], This is a key result which is used throughout this chapter.

Lemma 3.8. *Each pairwise union of P_1, P_2, and P_3 of $\check{H}$ has at least two bi-coloured cycles.*

We have the following result regarding tri-coloured cycles of $\check{H}$.

Lemma 3.9. *There exists at least one cycle of $\check{H}$ that is a tri-coloured cycle but is not a bi-coloured cycle.*

Proof. Every two of the longest paths P_1, P_2, and P_3 of $\check{H}$ have a common vertex by Lemma 1.1. Let $x, y, z \in V(\check{H})$ be three such vertices where, without loss of generality, we assume $\text{col}(x) = \{\text{blue}, \text{red}\}$, $\text{col}(y) = \{\text{blue}, \text{green}\}$, and $\text{col}(z) = \{\text{red}, \text{green}\}$. Then there exists a tri-coloured cycle C of $\check{H}$ that is the union of the $\{\text{blue}\}$–path xP_1y, the $\{\text{red}\}$–path xP_2z, and the $\{\text{green}\}$–path yP_3z, an example of which is shown in Figure 3.8.

Suppose that C is also a bi-coloured cycle. Without loss of generality, suppose that C is a $\{\text{blue}, \text{red}\}$–cycle. Consider the $\{\text{green}\}$–path yP_3z of C. For each edge $e \in E(yP_3z)$, either $\text{col}(e) = \{\text{blue}, \text{green}\}$ or $\text{col}(e) = \{\text{red}, \text{green}\}$. If $\text{col}(e) = \{\text{blue}, \text{green}\}$ for every edge $e \in E(yP_3z)$, then z is a tri-coloured vertex of $\check{H}$, a contradiction by Proposition 3.5(i). Similarly, if $\text{col}(e) = \{\text{red}, \text{green}\}$ for every edge $e \in E(yP_3z)$, then y is a tri-coloured vertex of $\check{H}$, a contradiction. Finally, if there is at least one edge

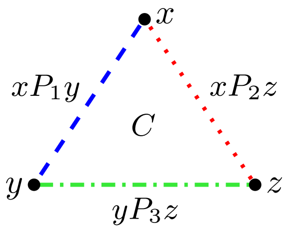

Figure 3.8: Subpaths of P_1, P_2, and P_3 shown by blue dashed, red dotted, and green dash-dotted lines respectively. An example of the tri-coloured cycle C that is the union of $\{blue\}$–path xP_1y, $\{red\}$–path xP_2z, and $\{green\}$–path yP_3z.

$e \in E(yP_3z)$ with $\mathrm{col}(e) = \{\mathrm{blue}, \mathrm{green}\}$ and at least one edge $f \in E(yP_3z)$ with $\mathrm{col}(f) = \{\mathrm{red}, \mathrm{green}\}$, then there exists an interior vertex v of yP_3z such that v is tri-coloured, a contradiction. We conclude that cycle C is not a bi-coloured cycle of $\check{H}$, as required. □

3.4 Preliminary results on bi-coloured cycles

In this section, we present three results about the bi-coloured cycles of $\check{H}$.

Lemma 3.10. *Let C be a bi-coloured cycle of $\check{H}$. Let $c_1, c_2, c_3 \in col(G)$. If C is a $\{c_1, c_2\}$–cycle, then C is neither a $\{c_1, c_3\}$–cycle nor a $\{c_2, c_3\}$–cycle.*

Proof. Assume without loss of generality that c_1, c_2, and c_3 are blue, red, and green respectively, and that C is a $\{\mathrm{blue}, \mathrm{red}\}$–cycle. By Lemma 3.8, cycle C will always exist. By Proposition 3.5(i) and (ii), there exists an edge $e \in E(C)$ such that $\mathrm{col}(e) = \{\mathrm{red}\}$, and hence C is not a $\{\mathrm{blue}, \mathrm{green}\}$–cycle . Similarly, there exists an edge $f \in E(C)$ such that $\mathrm{col}(f) = \{\mathrm{blue}\}$, and hence C is not a $\{\mathrm{red}, \mathrm{green}\}$–cycle. □

Lemma 3.11. *Let C be a bi-coloured cycle of $\check{H}$. Let $c_1, c_2 \in col(H)$. If C is a $[k\text{-}c_1, c_2]$–cycle for $k \geq 1$, then C is the mod 2 sum of two or more $[1\text{-}c_1, c_2]$–cycles of $\check{H}$.*

For example, in Figure 3.9, the $[2\text{-blue}, \mathrm{red}]$–cycle $S_1 \cup S_3 \cup S_2 \cup S_5$ is the mod 2 sum of the $[1\text{-blue}, \mathrm{red}]$–cycle $S_2 \cup S_4$ and the $[1\text{-blue}, \mathrm{red}]$–cycle $S_1 \cup S_3 \cup S_4 \cup S_5$.

Proof. Assume without loss of generality that c_1 and c_2 are blue and red respectively and that C is a $[k\text{-blue}, \mathrm{red}]$–cycle. By Lemma 3.8, cycle C will always exist. Cycle C is a $[k\text{-blue}, \mathrm{red}]$–cycle of $\check{H}$ for

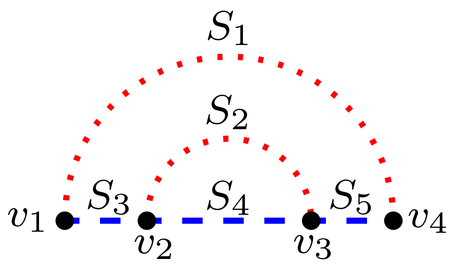

Figure 3.9: Subpaths of P_1 and P_2 shown by the blue dashed and red dotted lines respectively. A graph with {red}–paths S_1 and S_2 and {blue}–paths S_3, S_4, and S_5, where S_1 has endpoints v_1 and v_4, S_2 has endpoints v_2 and v_3, S_3 has endpoints v_1 and v_2, S_4 has endpoints v_2 and v_3, and S_5 has endpoints v_3 and v_4.

some $k \geq 1$. Note that a maximal {blue}–path of C may be a single vertex. In Figure 3.10(i), the cycle C has four maximal {blue}–paths, Q_1, Q_2, Q_3, and Q_4, where Q_2 is a single vertex.

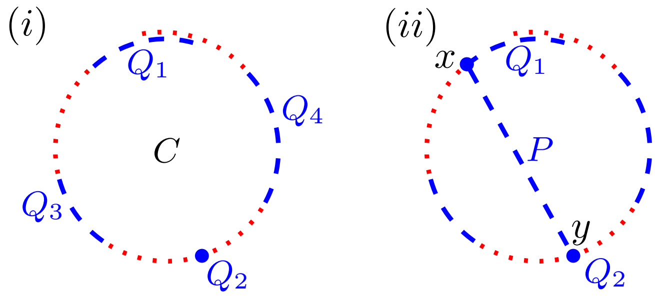

Figure 3.10: Subpaths of P_1 and P_2 shown by the blue dashed and red dotted lines respectively. (i) An example of a [4-blue, red]–cycle C where Q_1, Q_2, Q_3, and Q_4 are the four maximal {blue}–paths of C, one of which is a single vertex (Q_2). (ii) A {blue}–path P that has a common endpoint x with Q_1 and a common endpoint y with Q_2.

We use proof by induction on k. If $k = 1$, then C is a [1-blue, red]–cycle.

For $k \geq 2$, assume that every {blue, red}–cycle of $\check{H}$ that has at most $k - 1$ maximal {blue}–paths is the mod 2 sum of [1-blue, red]–cycles. Suppose that C is a [k-blue, red]–cycle, and let $\mathcal{Q}$ be the set of k maximal {blue}–paths of C. Let $P \notin \mathcal{Q}$ be a {blue}–path of $\check{H}$ such that there exists $Q_1, Q_2 \in \mathcal{Q}$ where P has a common endpoint x with Q_1 and a common endpoint y with Q_2 and, for every $Q \in \mathcal{Q}$, $(V(P) \backslash \{x, y\}) \cap V(Q) = \emptyset$. An example of this is shown in Figure 3.10(ii). Note that such paths P, Q_1, and Q_2 always exist as the paths in $\mathcal{Q}$ are vertex-disjoint subpaths of the {blue}–path P_1 of $\check{H}$.

Let R_1 and R_2 be the two paths of cycle C with endpoints x and y, such that $C = R_1 \cup R_2$. Let $C_1 = R_1 \cup P$ and $C_2 = R_2 \cup P$. Then C_1 and C_2 are two cycles of $\check{H}$, where $C = C_1 \oplus C_2$. Let k_1 $(1 \leq k_1 \leq k)$ be the number of maximal {blue}–paths of R_1. Let R_1' (respectively R_1'') be the maximal {blue}–path of R_1 with endpoint x (respectively y). Then $R_1' P R_1''$ is a maximal {blue}–path of cycle C_1, and hence C_1 has $k_1 - 1 \leq k - 1$ maximal {blue}–paths. An analogous argument can be used to show that cycle C_2 has at most $k-1$ maximal {blue}–paths. Then, by our earlier assumption, cycles C_1 and C_2 are each the mod 2 sum of [1-blue, red]–cycles. Hence $C = C_1 \oplus C_2$ is the mod 2 sum of two or more [1-blue, red]–cycles, as required. □

Corollary 3.12. *Let S be a maximal bi-coloured subgraph of $\check{H}$ and suppose, without loss of generality, that S is a* {blue, red}*–subgraph of $\check{H}$. Then S has two cycles that are* [1-blue, red]*–cycles and two cycles that are* [blue, 1-red]*–cycles.*

In Figure 3.11, A and B are [1-blue, red]–cycles, and $A \oplus B$ and B are [blue, 1-red]–cycles.

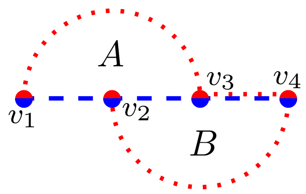

Figure 3.11: Subpaths of P_1 and P_2 are shown by the blue dashed and red dotted lines respectively. An example of a [1-blue, 2-red]–cycle $A = v_1v_3v_2v_1$ and a [1-blue, 1-red]–cycle $B = v_2v_3v_4v_2$, where $A \oplus B$ is a [2-blue, 1-red]–cycle.

3.5 Independence of cycles of $\check{H}$

In this section, we show that $\check{H}$ has seven independent cycles — six bi-coloured cycles, and one tri-coloured cycle.

3.5.1 Sets of cycles of $\check{H}$

By Lemma 3.8, $\check{H}$ has at least two bi-coloured cycles in each pair of its longest paths P_1, P_2, and P_3. From Corollary 3.12, we have the following proposition.

Proposition 3.13. *There is at least one set $\mathcal{B}$ of bi-coloured cycles of $\check{H}$ consisting of two* $\{\text{blue}, \text{red}\}$*–cycles, two* $\{\text{blue}, \text{green}\}$*–cycles, and two* $\{\text{red}, \text{green}\}$*–cycles where, for each pair $c_1, c_2 \in col(G)$, the two* $\{c_1, c_2\}$*–cycles in $\mathcal{B}$ are both* $[1\text{-}c_1, c_2]$*–cycles or both* $[c_1, 1\text{-}c_2]$*–cycles.*

An example of $\check{H}$ and such a set $\mathcal{B}$ of six bi-coloured cycles of $\check{H}$ is shown in Figure 3.12, where A and B are [1-blue, 1-red]–cycles, C and D are [1-blue, green]–cycles, and E and F are [1-red, 1-green]–cycles.

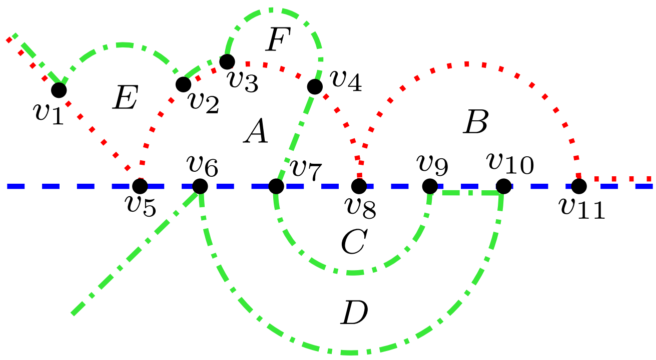

Figure 3.12: An example of $\check{H}$ where P_1, P_2, and P_3 are shown by the blue dashed, red dotted, and green dash-dotted lines respectively. The figure shows two [1-blue, 1-red]–cycles $A = v_5v_2v_3v_4v_8v_7v_6v_5$ and $B = v_8v_{11}v_{10}v_9v_8$, two [1-blue, green]–cycles $C = v_7v_8v_9v_7$ and $D = v_6v_7v_8v_9v_{10}v_6$, and two [1-red, 1-green]–cycles $E = v_1v_2v_5v_1$ and $F = v_3v_4v_3$.

Let $\mathfrak{B}$ be the set of all such sets $\mathcal{B}$ of bi-coloured cycles of $\check{H}$. We will prove that for each set $\mathcal{B} \in \mathfrak{B}$ the cycles in $\mathcal{B}$ are independent. We do this by proving that there does not exist a cycle $A \in \mathcal{B}$ that is the mod 2 sum of two, three, four, or five of the cycles in $\mathcal{B}\backslash\{A\}$.

Next, by Proposition 3.9, there exists a cycle of $\check{H}$ that is a tri-coloured cycle but not a bi-coloured cycle. We show that there exists such a cycle of $\check{H}$ that is independent of the cycles of a set $\mathcal{B} \in \mathfrak{B}$, and hence there is a set of seven independent cycles of $\check{H}$. Lastly, we put these results together to obtain the main result, Theorem 3.1.

3.5.2 The mod 2 sum of two cycles

Let $\mathcal{C}$ be a set of six bi-coloured cycles of $\check{H}$ consisting of two $\{\text{blue}, \text{red}\}$–cycles, two $\{\text{blue}, \text{green}\}$–cycles, and two $\{\text{red}, \text{green}\}$–cycles. In this section, we show that no cycle in such a set $\mathcal{C}$ is the mod 2 sum of two other cycles in the set, and hence every set of three cycles in $\mathcal{C}$ is independent.

Lemma 3.14. *Let A and B be two bi-coloured cycles of $\check{H}$, where A is a $\{c_1, c_2\}$–cycle, and B is a $\{c_1, c_3\}$–cycle for $c_1, c_2, c_3 \in col(\check{H})$. If $A \cap B$ is a path of $\check{H}$ with length at least one, then $A \oplus B$ is not a bi-coloured subgraph of $\check{H}$.*

Proof. Assume without loss of generality that c_1, c_2, and c_3 are blue, red, and green respectively. By Lemma 3.8, cycles A and B will always exist. Assume that $A \cap B$ is a path of $\check{H}$ with length at least one and $A \oplus B$ is a bi-coloured subgraph of $\check{H}$. Then $A \oplus B$ is a cycle. By Corollary 3.10, $A \oplus B$ is neither a $\{\text{blue}, \text{red}\}$–cycle nor a $\{\text{blue}, \text{green}\}$–cycle and hence $A \oplus B$ is a $\{\text{red}, \text{green}\}$–cycle.

Let P be the path of $\check{H}$ with edge set $E(A)\backslash E(B)$ and let Q be the path of $\check{H}$ with edge set $E(B)\backslash E(A)$. Let u and v be the common endpoints of P and Q. Then $A \oplus B = P \cup Q$.

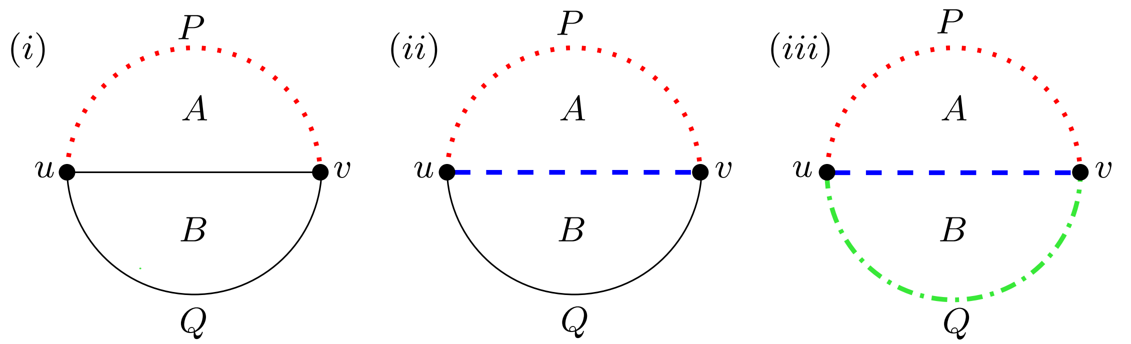

Figure 3.13: Subpaths of P_1, P_2, and P_3 shown by the blue dashed, red dotted, and green dash-dotted lines respectively. Examples of cycles A and B where $A \cap B$ is a path, path P has edge set $E(A)\backslash E(B)$, path Q has edge set $E(B)\backslash E(A)$, and u and v are the common endpoints of P and Q. In (i) P is a $\{\text{red}\}$–path P, in (ii) $A \cap B$ is a $\{\text{blue}\}$–path, and in (iii) Q is a $\{\text{green}\}$–path.

By Proposition 3.5(i) and (ii), there exists at least one edge $e \in E(A)$ such that $\text{col}(e) = \{\text{red}\}$. Since $A \cap B$ is a path of $\{\text{blue}, \text{green}\}$–cycle B, then $e \notin E(A \cap B)$, and hence $e \in E(P)$. By Lemma 3.7, since P is a path of both $\{\text{blue}, \text{red}\}$–cycle A and $\{\text{red}, \text{green}\}$–cycle $A \oplus B$, it follows that P is a $\{\text{red}\}$–path, as shown in Figure 3.13(i).

Similarly, by Proposition 3.5(i) and (ii), there exists at least one edge $f \in E(A)$ such that $\text{col}(f) = \{\text{blue}\}$. Since P is a $\{\text{red}\}$–path, then $f \notin E(P)$, and hence $f \in E(A \cap B)$. By Lemma 3.7, since $A \cap B$ is a path of both $\{\text{blue}, \text{red}\}$–cycle A and $\{\text{blue}, \text{green}\}$–cycle B, it follows that $A \cap B$ is a $\{\text{blue}\}$–path, as shown in Figure 3.13(ii).

Again, by Proposition 3.5(i) and (ii), there exists at least one edge $g \in E(B)$ such that $\mathrm{col}(g) = \{\text{green}\}$. Since $A \cap B$ is a $\{\text{blue}\}$–path, then $g \notin E(A \cap B)$, and hence $g \in E(Q)$. By Lemma 3.7, since Q is a path of both $\{\text{blue}, \text{green}\}$–cycle A and $\{\text{red}, \text{green}\}$–cycle $A \oplus B$, it follows that Q is a $\{\text{green}\}$–path, as shown in Figure 3.13(iii). However, u and v are now tri-coloured vertices of $\check{H}$, a contradiction by Proposition 3.5(i). □

Lemma 3.15. *Let A and B be two bi-coloured cycles of $\check{H}$, where A is a $\{c_1, c_2\}$–cycle and B is a $\{c_1, c_3\}$–cycle for $c_1, c_2, c_3 \in col(\check{H})$. If $A \cap B$ is a disconnected graph whose components are all paths, then $A \oplus B$ is not a bi-coloured subgraph of $\check{H}$.*

Proof. By Lemma 3.8, cycles A and B will always exist. Assume without loss of generality that c_1, c_2, and c_3 are blue, red, and green respectively. Assume that $A \cap B$ is a disconnected graph whose components are all paths and that $A \oplus B$ is a bi-coloured subgraph of $\check{H}$. By Lemma 3.10, $A \oplus B$ is neither a $\{\text{blue}, \text{red}\}$–subgraph nor a $\{\text{blue}, \text{green}\}$–subgraph of $\check{H}$, and hence $A \oplus B$ is a $\{\text{red}, \text{green}\}$–subgraph of $\check{H}$.

Let $R_1, R_2, \ldots, R_k$, $k \geq 2$, be the components of $A \cap B$, each of which is a path (possibly of length 0), and let $\mathcal{R} = \{R_1, R_2, \ldots, R_k\}$. Note that the endpoints of each path in $\mathcal{R}$ are vertices of $A \oplus B$. Let A' be the edge-induced subgraph of $\check{H}$ with edge set $E(A) \backslash E(B)$. Then A' consists of k maximal paths $S_1, S_2, \ldots, S_k$ such that each S_i, $1 \leq i \leq k$, has no internal vertex that is a vertex of a path in $\mathcal{R}$. Let $\mathcal{S} = \{S_1, S_2, \ldots, S_k\}$. Cycle A then consists of the k paths in $\mathcal{R}$ alternating with the k paths in $\mathcal{S}$. Let B' be the edge-induced subgraph of $\check{H}$ with edge set $E(B) \backslash E(A)$. Then B' consists of k maximal paths $Q_1, Q_2, \ldots, Q_k$ such that each Q_i, $1 \leq i \leq k$, has no internal vertex that is a vertex of a path in $\mathcal{R}$. Cycle B then consists of the k paths in $\mathcal{R}$ alternating with the k paths in $\mathcal{Q}$. Let $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_k\}$. Now $A \oplus B = \bigcup_{1 \leq i \leq k} S_i \cup Q_i$. Since the paths in $\mathcal{R}$ are the k components of $A \cap B$, no two paths in $\mathcal{R}$ have a common vertex, and hence each path in $\mathcal{S}$ or $\mathcal{Q}$ has length at least one.

For each $R_i \in \mathcal{R}$, if R_i has length at least one, then each endpoint of R_i is the endpoint of one path in $\mathcal{S}$ and one path in $\mathcal{Q}$. If R_i has length zero, then R_i is a single vertex that is the endpoint of two paths in $\mathcal{S}$ and two paths in $\mathcal{Q}$.

We next make an observation about $\{\text{blue}\}$–vertices and $\{\text{blue}\}$–paths of $\check{H}$. Let $R \in \mathcal{R}$ and let u be an endpoint of R. Suppose blue $\in \mathrm{col}(u)$. By Lemma 3.7, since every path in $\mathcal{R}$ is a path of $\{\text{blue}, \text{red}\}$–

cycle A and {blue, green}–cycle B, then R is a {blue}–path. Let $S \in \mathcal{S}$ and $Q \in \mathcal{Q}$ with endpoint u. Since $u \in E(A \oplus B)$, which is a {red, green}–subgraph of $\check{H}$, then red $\in$ col(u) or green $\in$ col(u), and hence col$(u) =$ {blue, red} or col$(u) =$ {blue, green}. Suppose that col$(u) =$ {blue, red}. Then by Lemma 3.7, since Q is a path of {blue, green}–cycle B and {red, green}–subgraph $A \oplus B$ of $\check{H}$, it follows that Q is a totally {blue, red}–path. Suppose that col$(u) =$ {blue, green}. Then by Lemma 3.7, since S is a path of {blue, red}–cycle A and {red, green}–subgraph $A \oplus B$ of $\check{H}$, it follows that S is a totally {blue, green}–path. Using this, we show that $\check{H}$ has a {blue}–cycle.

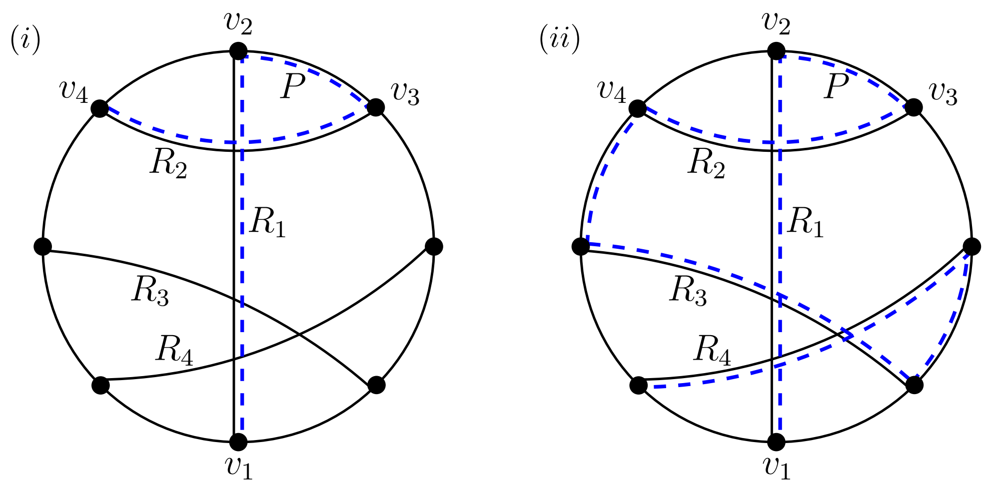

Figure 3.14: Subpaths of P_1 shown by blue dashed lines. Examples of cycle $A \oplus B$, which is the outer cycle in (i) and (ii), and paths $R_1, \ldots, R_4$ of $A \cap B$, showing (i) a {blue}–path $v_1 R_1 v_2 P v_3 R_2 v_4$ of $A \cup B$ and (ii) a {blue}–path of $A \cup B$ with subpaths $R_1, \ldots, R_4$.

Since A is a {blue, red}–cycle, by Proposition 3.5(i) and (ii) there exists an edge $e \in E(A)$ such that col$(e) =$ {blue}. Then $e \notin E(A \oplus B)$, and hence $e \in E(A \cap B)$. Without loss of generality, let R_1 be the path in $\mathcal{R}$ such that $e \in E(R_1)$. Since R_1 is a path of $A \cap B$, it follows by Lemma 3.7 that R_1 is a {blue}–path. Let v_1 and v_2 be the endpoints of R_1. As observed previously, there exists at least one {blue}–path in $\mathcal{Q} \cup \mathcal{S}$ with endpoint v_2. Let P be such a path, and let v_3 be the other endpoint of P. Without loss of generality, let R_2 be the path in $\mathcal{R}$ with endpoint v_3, and let v_4 be the other endpoint of R_2. Similarly to R_1, R_2 is a {blue}–path. An example of $\check{H}$ is shown in Figure 3.14(i). If $V(R_2) \cap V(R_1) \neq \emptyset$, then $\check{H}$ has a blue cycle, a contradiction by Proposition 3.5(ii). As before, there exists at least one {blue}–path in $\mathcal{Q} \cup \mathcal{S}$ with endpoint v_4. Continuing in this manner, we follow a {blue}–path of $\check{H}$ consisting of paths in $\mathcal{R}$ alternating with paths in $\mathcal{Q} \cup \mathcal{S}$, an example of which is shown in Figure 3.14(ii). We stop when we return to a path in $\mathcal{R}$ that is already a subpath of this {blue}–path. Since there are a finite number of paths in $\mathcal{R}$, this will always occur. Then $\check{H}$ has a {blue}–cycle, a contradiction by Proposition 3.5(ii). $\square$

Lemma 3.16. *Let $\mathcal{C}$ be a set of six bi-coloured cycles of $\check{H}$ consisting of two* $\{\text{blue}, \text{red}\}$*–cycles, two* $\{\text{blue}, \text{green}\}$*–cycles, and two* $\{\text{red}, \text{green}\}$*–cycles. Let $c_1, c_2 \in col(\check{H})$. Let $C \in \mathcal{C}$ be a $\{c_1, c_2\}$–cycle, and let $A, B \in \mathcal{C}\backslash\{C\}$. Then $A \oplus B$ is not a $\{c_1, c_2\}$–subgraph of $\check{H}$.*

Proof. By Lemma 3.8, set $\mathcal{C}$ and cycles A, B, and C will always exist. Assume without loss of generality that c_1 and c_2 are red and green respectively. Since there is exactly one $\{\text{red}, \text{green}\}$–cycle in $\mathcal{C}\backslash\{C\}$, at least one of A or B is a $\{\text{blue}, \text{red}\}$– or $\{\text{blue}, \text{green}\}$–cycle. Without loss of generality, suppose that A is a $\{\text{blue}, \text{red}\}$–cycle. If A and B are edge-disjoint, then $A \oplus B = A \cup B$. By Lemma 3.10, A is not a $\{\text{red}, \text{green}\}$–cycle, and hence $A \oplus B$ is not a $\{\text{red}, \text{green}\}$–subgraph of $\check{H}$.

Suppose instead that A and B are not edge-disjoint. Suppose that B is a $\{\text{blue}, \text{red}\}$–cycle. Then $A \oplus B$ is a $\{\text{blue}, \text{red}\}$–subgraph of $\check{H}$, and hence, by Lemma 3.10, $A \oplus B$ is not a $\{\text{red}, \text{green}\}$–subgraph of $\check{H}$. Next, suppose that B is a $\{\text{red}, \text{green}\}$–cycle. By Proposition 3.5(i) and (ii), there exists an edge $e \in E(A)$ such that $\text{col}(e) = \{\text{blue}\}$. However, $e \notin E(B)$, and hence $e \in E(A \oplus B)$. Therefore $A \oplus B$ is not a $\{\text{red}, \text{green}\}$–subgraph of $\check{H}$. Finally, suppose that B is a $\{\text{blue}, \text{green}\}$–cycle. Then, by Lemmas 3.14 and 3.15, $A \oplus B$ is not a $\{\text{red}, \text{green}\}$–subgraph of $\check{H}$, as required. $\square$

3.5.3 The mod 2 sum of three cycles

In this section, we show that no cycle in a set $\mathcal{B} \in \mathfrak{B}$ of six bi-coloured cycles of $\check{H}$ is the mod 2 sum of three other cycles in the set.

Lemma 3.17. *Let A, B, and C be three bi-coloured cycles of $\check{H}$ and let $c_1, c_2, c_3 \in col(\check{H})$. Let A and B be $\{c_1, c_2\}$–cycles and let C be a $\{c_1, c_3\}$–cycle that is a $[1\text{-}c_1, c_3]$–cycle, a $[c_1, 1\text{-}c_3]$–cycle, the mod 2 sum of two $[1\text{-}c_1, c_3]$–cycles, or the mod 2 sum of two $[c_1, 1\text{-}c_3]$–cycles. Then $A \oplus B \oplus C$ is not a $\{c_2, c_3\}$–subgraph of $\check{H}$.*

Proof. By Lemma 3.8, cycles A, B, and C will always exist in $\check{H}$. Assume without loss of generality that c_1, c_2, and c_3 are blue, green, and red respectively. Assume that $A \oplus B \oplus C$ is a $\{\text{red}, \text{green}\}$–subgraph of $\check{H}$. There are two possible configurations of cycle C.

(a) There exists a maximal {blue}–path Q_1 of C and a maximal {red}–path R_1 of C such that $C = Q_1 \cup R_1$.

(b) There exist two maximal {blue}–paths Q_1 and Q_2 in C and two maximal {red}–paths R_1 and R_2 in C such that $C = Q_1 \cup R_1 \cup Q_2 \cup R_2$. Additionally, C does not have Configuration (a).

Examples of these two configurations are shown in Figure 3.15.

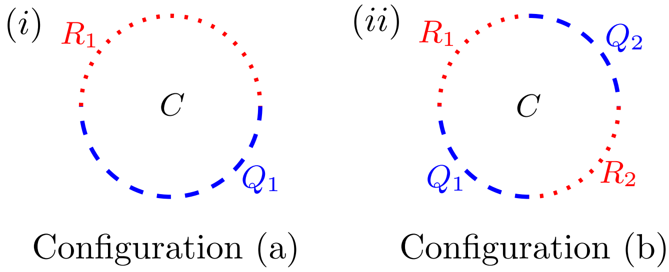

Figure 3.15: Subpaths of P_1 and P_2 shown by the blue dashed and red dotted lines repectively. Examples of (i) Configuration (a) of cycle C and (ii) Configuration (b) of cycle C.

To aid exposition in the proof, if C has Configuration (b), then we assign the labels Q_1, Q_2, R_1, and R_2 to paths of C as follows. Let Q_1 and Q_2 be the two maximal {blue}–paths of C that each have at least one edge e where $\mathrm{col}(e) = \{\text{blue}\}$. Then there exists a {blue}–path Q of $\check{H}$ that has a common endpoint with Q_1 and a common endpoint with Q_2, such that the other endpoints of Q_1 and Q_2 are not in $V(Q)$. Let q and r be the endpoints of Q_1, where $q \in V(Q)$. Let R_1 be the maximal {red}–path of C such that $r \in V(R_1)$ and let R_2 be the maximal {red}–path of C such that $q \in V(R_2)$, as shown in the example in Figure 3.16.

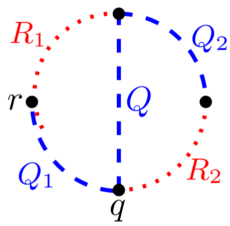

Figure 3.16: Subpaths of P_1 and P_2 shown by the blue dashed and red dotted lines repectively. An example of a {blue, red}–cycle C with Configuration (b) and {blue}–path Q, showing maximal {blue}–paths Q_1 and Q_2 of C, endpoints q and r of Q_1, and maximal {red}–paths R_1 and R_2 of C.

In the remainder of this proof, we simultaneously consider the case where C has Configuration (a) and the case where C has Configuration (b), noting differences where required.

Consider cycle C, which may have Configuration (a) or (b). Let u be an endpoint of the path R_1 where $u \in V(Q_1)$. Let $e_1 = uv$ be the edge of Q_1 incident to u where $e_1 \notin E(R_1)$. Then $\text{col}(e_1) = \{\text{blue}\}$. It follows that $e_1 \notin E(A \oplus B \oplus C)$, and hence $e_1 \in E(A)$ without loss of generality. Let P be the maximal path of $A \cap C$ such that $e_1 \in E(P)$, and let w and x be the endpoints of P, where these vertices are in the order w, u, v, x in P (note that we may have $w = u$ or $x = v$). An example of this is shown in Figure 3.17(i). By Lemma 3.7, since P is a path of $\{\text{blue}, \text{red}\}$–cycle C and a path of $\{\text{blue}, \text{green}\}$–cycle A, then P is a $\{\text{blue}\}$–path. Hence P is a subpath of Q_1.

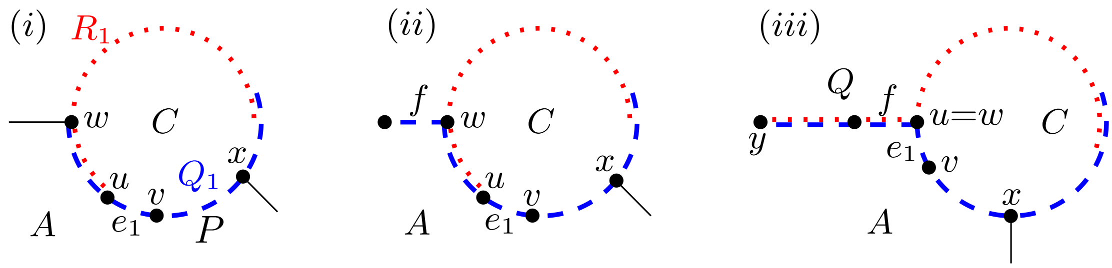

Figure 3.17: Subpaths of P_1 and P_2 shown by the blue dashed and red dotted lines respectively. Examples of cycle C with edge $e_1 = uv$ where $\text{col}(e) = \{\text{blue}\}$ along with (i) path P of $A \cap C$ with endpoints w and x, maximal $\{\text{blue}\}$–path Q_1 of C, and maximal $\{\text{red}\}$–path R_1 of C; (ii) edge $f \in E(A) \backslash E(C)$ incident to w; and (iii) totally $\{\text{blue}, \text{red}\}$–path Q of A with endpoints y and w.

By Proposition 3.5(i) and (ii), there exists an edge $e_2 \in E(R_1)$ such that $\text{col}(e_2) = \{\text{red}\}$. Then R_1 is not a subpath of P since P is a $\{\text{blue}\}$–path. Hence w is an internal vertex of R_1 or $w = u$. It follows that the path wPu is a path of $Q_1 \cap R_1$ and is a totally $\{\text{blue}, \text{red}\}$–path. Let $f \in E(A) \backslash E(C)$ be incident to w. Since $\text{col}(w) = \{\text{blue}, \text{red}\}$, then green $\notin \text{col}(f)$, and since A is a $\{\text{blue}, \text{green}\}$ cycle, then blue $\in \text{col}(f)$, as shown in the example in Figure 3.17(ii). We consider two cases: (1) $f \notin E(B)$ and (2) $f \in E(B)$.

First, consider case (1), where $f \notin E(B)$. Let Q be the maximal path of cycle A such that $f \in E(Q)$ and no interior vertices of Q are vertices of $B \cup C$. Then Q is a path of $A \oplus B \oplus C$ and has endpoints w and y, where $y \in V(\check{H})$. Since Q is a path of $\{\text{blue}, \text{green}\}$–cycle A and $\{\text{red}, \text{green}\}$–subgraph $A \oplus B \oplus C$ of $\check{H}$, it follows by Lemma 3.7 that Q is a totally $\{\text{blue}, \text{red}\}$–path, as shown in the example in Figure 3.17(iii).

Consider vertex y. First, suppose that $y \in V(C)$. If $y \in V(R_1)$ (or if $y \in V(R_2)$ in the case where C has Configuration (b)) then there is a $\{\text{blue}\}$–cycle in $\check{H}$, a contradiction by Proposition 3.5(ii). If, instead, y is an interior vertex of Q_1 (or of Q_2 in the case where C has Configuration (b)), then y is incident with three $\{\text{red}\}$–edges, a contradiction by the colouring of $\check{H}$. Next, suppose that $y \notin V(C)$. Then $y \in V(B)$ and there exist two edges $g_1, g_2 \in E(B)$ incident to y, where $g_1, g_2 \notin E(Q)$. Since $\text{col}(y) = \{\text{blue}, \text{red}\}$, then, without loss of generality, $\text{col}(g_1) = \{\text{red}\}$, a contradiction since B is a $\{\text{blue}, \text{green}\}$–cycle.

It remains to consider case (2), in which $f \in E(B)$. Recall that P is a $\{\text{blue}\}$–path with endpoints w and x and that $\text{blue} \in \text{col}(f)$, as shown in the example in Figure 3.17(ii). Let $h_1 \in E(B)$ be incident to w, where $h_1 \neq f$. Since $\text{col}(w) = \{\text{blue}, \text{red}\}$, then $\text{green} \notin \text{col}(h_1)$, and since B is a $\{\text{blue}, \text{green}\}$ cycle, then $\text{blue} \in \text{col}(h_1)$. Since w is incident to at most two $\{\text{blue}\}$–edges, and P is a $\{\text{blue}\}$–path, then $h_1 \in E(P)$. Recall that $\text{col}(e_1) = \{\text{blue}\}$. If $e_1 \in E(B)$, then $e_1 \in E(A \oplus B \oplus C)$, a contradiction since $A \oplus B \oplus C$ is a $\{\text{red}, \text{green}\}$–subgraph of $\check{H}$. Hence there exists a vertex $z \in V(wPu)$, where $z \neq w$, with incident edge $h_2 \in E(B) \backslash E(A \cup C)$. Since wPu is a totally $\{\text{blue}, \text{red}\}$–path, then $\text{green} \notin \text{col}(h_2)$, and since z is an interior vertex of P, then $\text{blue} \notin \text{col}(h_2)$. Hence $\text{col}(h_2) = \{\text{red}\}$, a contradiction since $h_1 \in E(B)$. □

Lemma 3.18. *Let $\mathcal{B} \in \mathfrak{B}$ be a set of six bi-coloured cycles of $\check{H}$. Let $D \in \mathcal{B}$ be a $\{c_1, c_2\}$–cycle of $\check{H}$ where $c_1, c_2 \in col(\check{H})$, and let $A, B, C \in \mathcal{B} \backslash \{D\}$. Then $A \oplus B \oplus C$ is not a $\{c_1, c_2\}$–subgraph of $\check{H}$.*

Proof. By Proposition 3.13, set $\mathcal{B}$ and cycles A, B, C, and D will always exist. Assume without loss of generality that c_1 and c_2 are red and green respectively. Let $\mathcal{C} = \{A, B, C\}$ and let $S = A \oplus B \oplus C$.

First suppose that one of the cycles in $\mathcal{C}$ is a $\{\text{red}, \text{green}\}$–cycle, say C without loss of generality. Since $A \oplus B \oplus C = S$, then $A \oplus B = C \oplus S$. Since C and S are both $\{\text{red}, \text{green}\}$–subgraphs of $\check{H}$, then $C \oplus S$ is also a $\{\text{red}, \text{green}\}$–subgraph of $\check{H}$. However, by Lemma 3.16, $A \oplus B$ is not a $\{\text{red}, \text{green}\}$–subgraph of $\check{H}$, a contradiction.

Suppose, instead, that none of the cycles in $\mathcal{C}$ is a $\{\text{red}, \text{green}\}$–cycle. If A and B are both $\{\text{blue}, \text{green}\}$–cycles, and C is a $\{\text{blue}, \text{red}\}$–cycle, then, by Lemma 3.17, S is not a $\{\text{red}, \text{green}\}$–subgraph of $\check{H}$. Similarly, if A and B are $\{\text{blue}, \text{red}\}$–cycles, and C is a $\{\text{blue}, \text{green}\}$–cycle, then S is not a $\{\text{red}, \text{green}\}$–subgraph of $\check{H}$, as required. □

3.5.4 The mod 2 sum of four cycles

In this section, we show that no cycle in a set $\mathcal{B} \in \mathfrak{B}$ of six bi-coloured cycles of $\check{H}$ is the mod 2 sum of four other cycles in the set.

Lemma 3.19. *Let $\mathcal{B} \in \mathfrak{B}$ be a set of six bi-coloured cycles of $\check{H}$. Let $E \in \mathcal{B}$ be a $\{c_1, c_2\}$–cycle where $c_1, c_2 \in col(\check{H})$ and let $A, B, C, D \in \mathcal{B} \backslash \{E\}$. Then $A \oplus B \oplus C \oplus D$ is not a $\{c_1, c_2\}$–subgraph of $\check{H}$.*

Proof. By Proposition 3.13, set $\mathcal{B}$ and cycles A, B, C, D, and E will always exist. Assume without loss of generality that c_1 and c_2 are red and green respectively. Let $\mathcal{C} = \{A, B, C, D\}$. Let S be the mod 2 sum of the cycles in $\mathcal{C}$ and assume that S is a $\{\text{red}, \text{green}\}$–subgraph of $\check{H}$.

First suppose that one of the cycles in $\mathcal{C}$ is a $\{\text{red}, \text{green}\}$–cycle, say D without loss of generality. Since $A \oplus B \oplus C \oplus D = S$, then $A \oplus B \oplus C = S \oplus D$, which is a $\{\text{red}, \text{green}\}$–subgraph of $\check{H}$. However, by Lemma 3.18, $A \oplus B \oplus C$ is not a $\{\text{red}, \text{green}\}$–subgraph of $\check{H}$, a contradiction. It remains to consider the case in which none of the cycles in $\mathcal{C}$ is a $\{\text{red}, \text{green}\}$–cycle. Then, without loss of generality, A and B are $\{\text{blue}, \text{red}\}$–cycles, and C and D are $\{\text{blue}, \text{green}\}$–cycles.

If $E(A) \cap E(B) \neq \emptyset$, then by Lemma 3.17, S is not a $\{\text{red}, \text{green}\}$–subgraph of $\check{H}$, a contradiction. We similarly obtain a contradiction if $E(C) \cap E(D) \neq \emptyset$. Suppose instead that cycles A and B are edge-disjoint, and cycles C and D are edge-disjoint.

Cycle A has a maximal $\{\text{blue}\}$–path A_1 and a maximal $\{\text{red}\}$–path A_2 such that $A = A_1 \cup A_2$. Similarly, cycle B has a maximal $\{\text{blue}\}$–path B_1 and a maximal $\{\text{red}\}$–path B_2 such that $B = B_1 \cup B_2$. Cycle C has a maximal $\{\text{blue}\}$–path C_1 and a maximal $\{\text{green}\}$–path C_2 such that $C = C_1 \cup C_2$. Similarly, cycle D has a maximal $\{\text{blue}\}$–path D_1 and a maximal $\{\text{green}\}$–path D_2 such that $D = D_1 \cup D_2$.

We next show that there is exactly one maximal path of $A \cap C$ such that every edge e of $A \cap C$ with $\text{col}(e) = \{\text{blue}\}$ is an edge of this path. Similarly for $A \cap D$, $B \cap C$, and $B \cap D$. We will refer to this as *Property 1.* Suppose that $A \cap C$ is a disconnected graph, and note that each component of $A \cap C$ is a path. Suppose that two of the maximal paths of $A \cap C$, R_1 and R_2, each have at least one edge e such that $\text{col}(e) = \{\text{blue}\}$. Let R_1' be a maximal subpath of R_1 where, for every edge $e \in E(R_1')$, $\text{col}(e) = \{\text{blue}\}$. Similarly define a subpath R_2' of R_2. Then, by definition of A_1 and C_1, paths R_1' and R_2' are subpaths of A_1 and subpaths of C_1. It follows that there exists a $\{\text{blue}\}$–subpath A_1' of A_1

(respectively C_1' of C_1) such that A_1' (C_1') has a common endpoint with R_1' and a common endpoint with R_2' and no other vertices of R_1' or R_2' are vertices of A_1' (C_1'). If $A_1' = C_1'$ then R_1' and R_2' are subpaths of one maximal {blue}–path of $A \cap C$, a contradiction. Hence $E(A_1')\backslash E(C_1') \neq \emptyset$ and $E(C_1')\backslash E(A_1') \neq \emptyset$. Therefore, $R_1' \cup R_2' \cup A_1' \cup C_1'$ has a {blue}–cycle, a contradiction by Proposition 3.5(ii). An analogous argument can be used to show that Property 1 holds for $A \cap D$, $B \cap C$, and $B \cap D$.

We now show that there is a blue cycle in $\check{H}$ by considering the colours of the edges of cycles A, B, C, and D. Recall that $S = A \oplus B \oplus C \oplus D$ is a $\{\text{red}, \text{green}\}$–subgraph of $\check{H}$. First consider cycle A. Let x be an endpoint of A_2, and let w be the vertex adjacent to x such that $e_1 = wx \in E(A_1)\backslash E(A_2)$. Then $\text{col}(e_1) = \{\text{blue}\}$. It follows that $e_1 \notin E(S)$ and, since A and B are edge-disjoint, then $e_1 \in E(C)$ without loss of generality. Let R_1 be the maximal path of $A \cap C$ such that $e_1 \in E(R_1)$ and let v_1 and v_2 be the endpoints of R_1, where these vertices are in the order v_1, w, x, v_2 in R_1 (note that we may have $v_1 = w$ or $v_2 = x$), as shown in the example in Figure 3.18. By Lemma 3.7, R_1 is a {blue}–path, and hence xR_1v_2 is a totally $\{\text{blue}, \text{red}\}$–path.

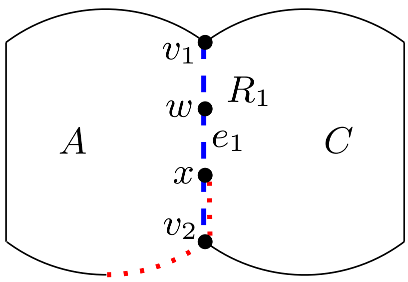

Figure 3.18: Subpaths of P_1 and P_2 shown by the blue dashed and red dotted lines respectively. An example of cycles A and C and the {blue}–path R_1 of $A \cap C$ with endpoints v_1 and v_2.

Consider the edges of $E(C)\backslash E(R_1)$. By Proposition 3.5 (i) and (ii), there exists an edge $f_1 \in E(C)\backslash E(R_1)$ such that $\text{col}(f_1) = \{\text{green}\}$. Then, since $\text{col}(v_2) = \{\text{blue}, \text{red}\}$, it follows by Proposition 3.5 (i) that there exists an edge $e_2 \in E(C)\backslash E(R_1)$ such that $\text{col}(e_2) = \{\text{blue}\}$. Edge $e_2 \notin E(A)$ by Property 1, $e_2 \notin E(D)$ since C and D are edge-disjoint, and $e_2 \notin E(S)$ since S is a $\{\text{red}, \text{green}\}$–subgraph of $\check{H}$. Hence $e_2 \in E(B)$.

Let R_2 be the maximal path of $C \cap B$ such that $e_2 \in E(R_2)$, and let v_3 and v_4 be the endpoints of R_2, where these vertices are in the order v_1, v_2, v_3, v_4, v_1 in C. By Lemma 3.7, R_2 is a {blue}–path. Let

Q_1 (respectively Q_2) be the path of C with endpoints v_1 and v_4 (respectively v_2 and v_3) that has no interior vertices that are vertices of R_1 or R_2. Then $C = R_1 \cup Q_2 \cup R_2 \cup Q_1$.

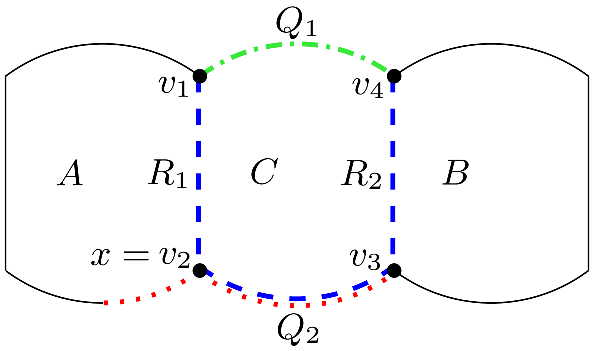

Figure 3.19: Subpaths of P_1, P_2, and P_3 shown by the blue dashed, red dotted, and green dash-dotted lines respectively. An example of cycles A, B and C. The $\{\text{blue}\}$–paths R_1 and R_2 are maximal paths of $A \cap C$ and $C \cap B$ respectively. Also shown are the $\{\text{green}\}$–path Q_1 of C and $\{\text{blue}, \text{red}\}$–path Q_2 of C, which are internally disjoint from R_1 and R_2.

By Property 1, there does not exist an edge $e \in E(Q_1 \cup Q_2)$ such that $\text{col}(e) = \{\text{blue}\}$. Since $\text{col}(v_2) = \{\text{blue}, \text{red}\}$ and C is a $\{\text{blue}, \text{green}\}$–cycle, it follows by Proposition 3.5 (i) that Q_2 is a totally $\{\text{blue}, \text{red}\}$–path. Recall that $\text{col}(f_1) = \{\text{green}\}$. Since $f_1 \notin E(R_1) \cup E(R_2) \cup E(Q_2)$, then $f_1 \in E(Q_1)$. It follows by Proposition 3.5 (i) that Q_1 is a $\{\text{green}\}$–path. An example of cycles A, B, and C is shown in Figure 3.19. Note that Q_2 may be a single vertex, and that $E(Q_1 \cup Q_2) \cap E(A \cup B)$ may be non-empty.

Next, we follow a similar line of deductions for cycle B. Consider the edges of $E(B) \backslash E(R_2)$. By Proposition 3.5 (i) and (ii), there exists an edge $e_3 \in E(B) \backslash E(R_2)$ such that $\text{col}(e_3) = \{\text{blue}\}$. Since $e_3 \notin E(A) \cup E(C) \cup E(S)$, then $e_3 \in E(D)$. Let R_3 be the maximal path of $B \cap D$ such that $e_3 \in E(R_3)$, and let v_5 and v_6 be the endpoints of R_3, where these vertices are in the order v_3, v_4, v_5, v_6, v_3 in B. By Lemma 3.7, R_3 is a $\{\text{blue}\}$–path. Let Q_3 (respectively Q_4) be the path of B with endpoints v_3 and v_6 (respectively v_4 and v_5) that has no interior vertices that are vertices of R_2 or R_3. Then $B = R_2 \cup Q_3 \cup R_3 \cup Q_4$.

By Property 1, there does not exist an edge $e \in E(Q_3) \cup E(Q_4)$ such that $\text{col}(e) = \{\text{blue}\}$. Since $\text{col}(v_4) = \{\text{blue}, \text{green}\}$ and B is a $\{\text{blue}, \text{red}\}$–cycle, it follows by Proposition 3.5 (i) that Q_4 is a totally $\{\text{blue}, \text{green}\}$–path. By Proposition 3.5 (i) and (ii), there is an edge $f_2 \in E(B)$ such that $\text{col}(f_2) = \{\text{red}\}$.

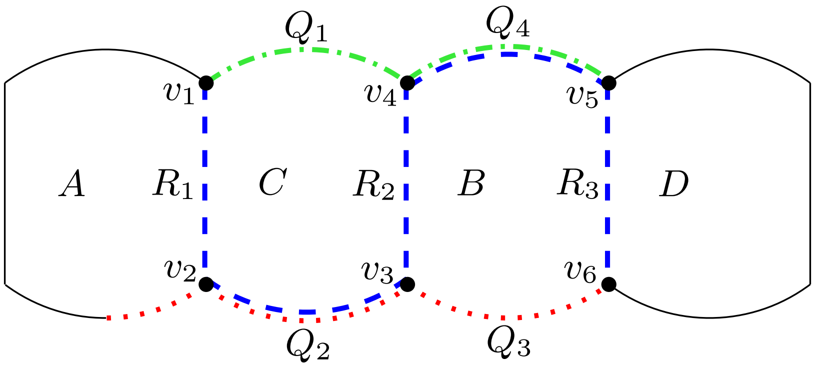

Figure 3.20: Subpaths of P_1, P_2, and P_3 shown by the blue dashed, red dotted, and green dash-dotted lines respectively. An example of cycles A, B, C and D. The {blue}–paths R_1, R_2, and R_3 are maximal paths of $A \cap C$, $C \cap B$, and $B \cap D$ respectively. The paths $Q_1, \ldots, Q_4$ and $R_1, \ldots R_3$ are all internally disjoint. The colours of paths Q_1 and Q_2 of C and paths Q_3 and Q_4 of B are also shown.

Then $f_2 \in E(Q_3)$ and hence Q_3 is a {red}–path. An example of cycles A, B, C, and D is shown in Figure 3.20. Note that Q_4 may be a single vertex, and that $E(Q_3 \cup Q_4) \cap E(C \cup D)$ may be non-empty.

Next, consider the edges of $E(D) \backslash E(R_3)$. As with cycle C and path R_1, there exists an edge $e_4 \in E(D) \backslash E(R_3)$ such that $\mathrm{col}(e_4) = \{\mathrm{blue}\}$. Since $e_4 \notin E(C) \cup E(B) \cup E(S)$, then $e_4 \in E(A)$. Let R_4 be the maximal path of $D \cap A$ such that $e_4 \in E(R_4)$, and let v_7 and v_8 be the endpoints of R_4, where these vertices are in the order v_5, v_6, v_7, v_8, v_5 in D. By Lemma 3.7, R_4 is a {blue}–path. Let Q_5 (respectively Q_6) be the path in D with endpoints v_5 and v_8 (respectively v_6 and v_7) that has no interior vertices that are vertices of R_3 or R_4. Then $D = R_3 \cup Q_6 \cup R_4 \cup Q_5$.

By Property 1, there does not exist an edge $e \in E(Q_5 \cup Q_6)$ such that $\mathrm{col}(e) = \{\mathrm{blue}\}$. Similarly to paths Q_2 and Q_1 respectively, Q_6 is a {blue, red}–path and Q_5 is a {green}–path. Note that Q_6 may be a single vertex, and that $E(Q_5 \cup Q_6) \cap E(A \cup B)$ may be non-empty.

Let Q_7 (respectively Q_8) be the path in A with endpoints v_7 and v_2 (respectively v_8 and v_1) that has no interior vertices that are vertices of R_4 or R_1. Then $A = R_1 \cup Q_7 \cup R_4 \cup Q_8$. Similarly to path Q_4, Q_8 is a totally {blue, green}–path. An example of cycles A, B, C, and D is shown in Figure 3.21. However, now $\check{H}$ has a blue cycle $v_1 R_1 v_2 Q_2 v_3 R_2 v_4 Q_4 v_5 R_3 v_6 Q_6 v_7 R_4 v_8 Q_8 v_1$, a contradiction. $\square$

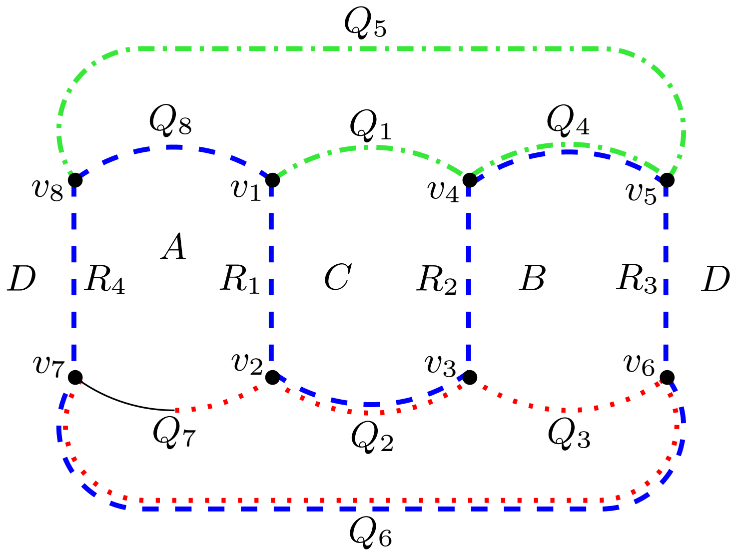

Figure 3.21: Subpaths of P_1, P_2, and P_3 shown by the blue dashed, red dotted, and green dash-dotted lines respectively. An example of cycles A, B, C and D, where D is the outer cycle, showing a {blue} cycle in the graph. The {blue}–paths R_1, R_2, R_3, and R_4 are maximal paths of $A \cap C$, $C \cap B$, $B \cap D$, and $D \cap A$ respectively. The paths $Q_1, \ldots, Q_8$ and $R_1, \ldots, R_4$ are all internally disjoint.

3.5.5 The mod 2 sum of five cycles

In this section, we show that no cycle in a set $\mathcal{B} \in \mathfrak{B}$ of six bi-coloured cycles of $\check{H}$ is the mod 2 sum of the other five cycles in the set.

Lemma 3.20. *Let $\mathcal{B} \in \mathfrak{B}$ be a set of six bi-coloured cycles of $\check{H}$. Let $F \in \mathcal{B}$ be a $\{c_1, c_2\}$–cycle where $c_1, c_2 \in col(\check{H})$ and let $A, B, C, D, E \in \mathcal{B} \backslash \{F\}$. Then $A \oplus B \oplus C \oplus D \oplus E$ is not a $\{c_1, c_2\}$–subgraph of $\check{H}$.*

Proof. By Proposition 3.13, set $\mathcal{B}$ and cycles A, B, C, D, E, and F will always exist. Assume without loss of generality that c_1 and c_2 are red and green respectively. Without loss of generality, let E be the $\{\text{red}, \text{green}\}$–cycle in $\mathcal{B} \backslash \{F\}$. Let $S = A \oplus B \oplus C \oplus D \oplus E$ and let $S' = A \oplus B \oplus C \oplus D$. By Lemma 3.19, S' is not a $\{\text{red}, \text{green}\}$–subgraph of $\check{H}$. Then there is an edge $e \in E(S')$ with $\text{col}(e) = \{\text{blue}\}$. However, e is not an edge of cycle E, and hence e is an edge of $S = S' \oplus E$. Therefore S is not a $\{\text{red}, \text{green}\}$–subgraph of $\check{H}$. $\square$

3.5.6 Independent tri-coloured cycle

We now consider a tri-coloured cycle of $\check{H}$ (which always exists by Lemma 3.9) and prove that this cycle is independent of a set $\mathcal{B} \in \mathfrak{B}$ of six bi-coloured cycles of $\check{H}$.

Lemma 3.21. *There exists a tri-coloured cycle T of $\check{H}$ that is not a bi-coloured cycle and there exists a set $\mathcal{B} \in \mathfrak{B}$ of six bi-coloured cycles of $\check{H}$ such that $\mathcal{B} \cup \{T\}$ is independent.*

Proof. By Lemma 3.9 and Proposition 3.13 respectively, cycle T and set $\mathfrak{B}$ will always exist in $\check{H}$. By Lemma 1.1, each pair of the longest paths P_1, P_2, and P_3 have a common vertex. Let $x, y, z \in V(\check{H})$ be three such vertices, where, without loss of generality, we assume that col(x)={blue, red}, col(y) = {blue, green}, and col(z) = {red, green}. Then there is a tri-coloured cycle T of $\check{H}$ that is the union of {blue}–path xP_1y, {red}–path xP_1z, and {green}–path yP_2z, which are internally disjoint. An example of cycle T is shown in Figure 3.22 (which is the same as Figure 3.8, repeated here for reference).

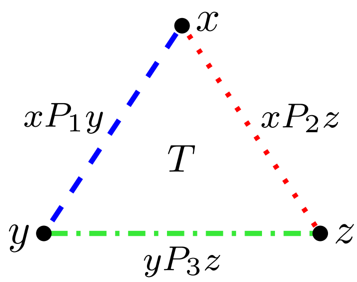

Figure 3.22: Subpaths of P_1, P_2, and P_3 shown by the blue dashed, red dotted, and green dash-dotted lines respectively. An example of cycle T consisting of {blue}–path xP_1y, {red}–path xP_2z, and {green}–path yP_3z.

By Lemma 3.9, cycle T is not a bi-coloured cycle. We call cycles that are the union of a {blue}–path, a {red}–path, and a {green}–path, but are not bi-coloured, *3-segment cycles.*

Assume that it is not the case that there exists a tri-coloured cycle T of $\check{H}$ and a set $\mathcal{B} \in \mathfrak{B}$ of six bi-coloured cycles of $\check{H}$ such that $\mathcal{B} \cup \{T\}$ is independent. Then for every tri-coloured cycle T of $\check{H}$ and every set $\mathcal{B} \in \mathfrak{B}$ of of six bi-coloured cycles of $\check{H}$, the set $\mathcal{B} \cup \{T\}$ is dependent, that is, T is the mod 2 sum of a subset of the cycles in $\mathcal{B}$.

Consider $\check{H}$ and 3-segment cycle T. Let $\mathcal{C}$ be the set of all bi-coloured cycles of $\check{H}$ Let the endpoints of path P_1 of $\check{H}$ be u_1 and v_1. Consider the maximal {blue}–path of T with subpath xP_1y, and let its endpoints be x_1 and y_1, where these vertices are in the order u_1, x_1, x, y, y_1, v_1 in P_1, as shown in the example in Figure 3.23. Similarly define u_2, v_2, x_2, and z_2 for P_2, where these vertices are in the order u_2, x_2, x, z, z_2, v_2 in P_2. Similarly define u_3, v_3, y_3, and z_3 for P_3, where these vertices are in the order u_3, y_3, y, z, z_3, v_3 in P_3.

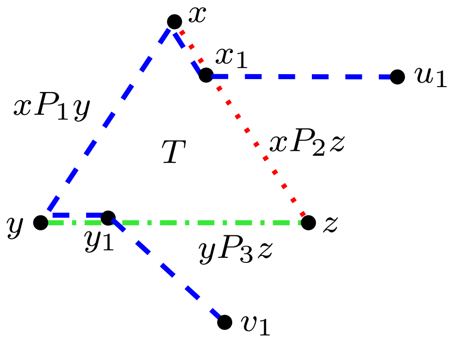

Figure 3.23: Subpaths of P_1, P_2, and P_3 shown by the blue dashed, red dotted, and green dash-dotted lines respectively. An example of cycle T and {blue}–path P_1, showing vertices x_1 and y_1 that are the endpoints of the maximal {blue}–path of T with subpath xP_1y and endpoints u_1 and v_1 of the path P_1.

Note that $ys_1 \in V(yP_3z)$. Let $e_1 \in E(y_1P_3z_2)$ incident to y_1; then $\mathrm{col}(e_1) = \{\text{green}\}$. Let C be a bi-coloured cycle of $\check{H}$ such that $e_1 \in E(C)$. Such a cycle exists since $T \cup \mathcal{B}$ is dependent for every $\mathcal{B} \in \mathfrak{B}$. We consider all configurations of this cycle, and show that, in each case that is possible, there is a 3-segment cycle T_1 such that $T = T_1 \oplus C \oplus \mathcal{S}$ where $\mathcal{S}$ is a (possibly empty) set of bi-coloured cycles in $\mathcal{C}\backslash\{C\}$. Since there is a set of at most six bi-coloured cycles in $\mathcal{C}$ such that T is the mod 2 sum of these cycles, then T_1 is the mod 2 sum of at most five bi-coloured cycles in $\mathcal{C}\backslash\{C\}$. We then repeat this process for T_1 and subsequent 3-segment cycles, obtaining a 3-segment cycle that is (the mod 2 sum of) a bi-coloured cycle.

Case 1: Suppose that C is a {red, green}–cycle. Since $y_3P_3y_1$ is a totally {blue, green}–path, neither $x_2P_2u_2$ nor $z_2P_2v_2$ has a vertex that is a vertex of $y_3P_3y_1$. Therefore there is a maximal subpath of $y_3P_3u_3$ with endpoint y_3 that is a subpath of C. Let w be the other endpoint of this subpath. Then $w \in V(P_2)$.

Case 1a: Suppose that $w \in V(x_2P_2z_2)$. Let $t \in V(x_2P_1y_3) \cap V(y_3P_3w)$ where $V(x_2P_1t) \cap V(y_3P_3w) = \{t\}$. Then $C = wP_2zP_3w$ and $T_1 = wP_3tP_1xP_2w$. An example of cycles T, T_1, and C in this case is shown in

Figure 3.24(i).

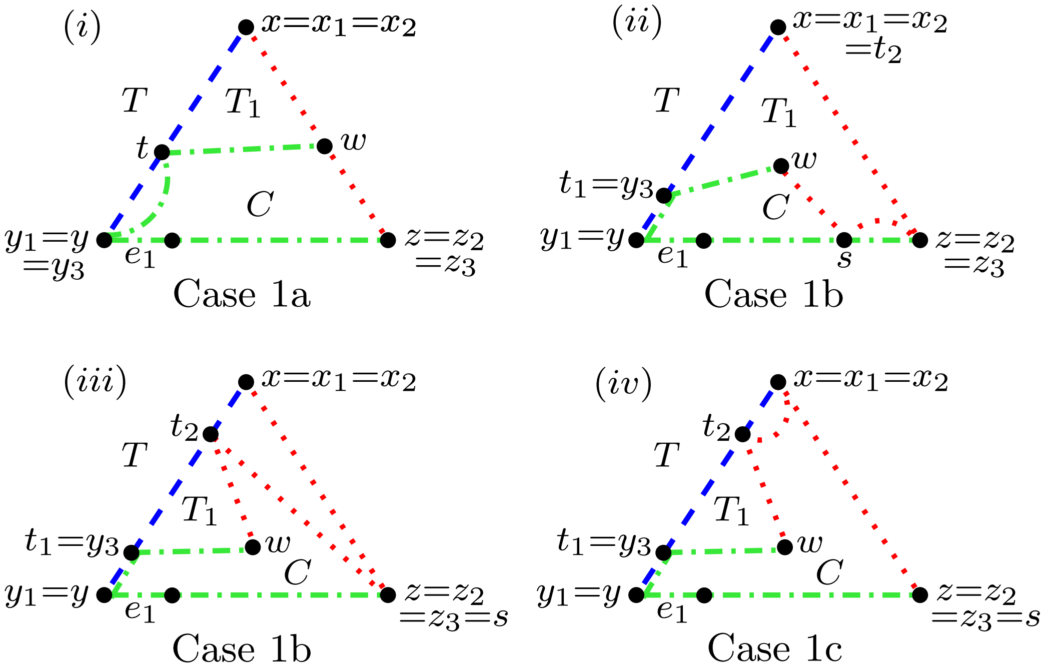

Figure 3.24: Subpaths of P_1, P_2 and P_3 shown by the blue dashed, red dotted, and green dash-dotted lines respectively. Examples of cycle T (outer cycle in (i) – (iv)), $\{\text{red}, \text{green}\}$–cycle C, and 3-segment cycle T_1 in (i) case 1a, (ii) and (iii) case 1b, and (iv) case 1c.

Case 1b: Suppose that $w \in V(z_2P_2v_2)$. Let $t_1 \in V(x_2P_1y_3) \cap V(y_3P_3w)$ where $V(x_2P_1y_y) \cap V(t_1P_3w) = \{t_1\}$. Let $t_2 \in V(x_2P_1y_3) \cap V(x_2P_2w)$ where $V(t_1P_1t_2) \cap V(x_2P_2w) = \{t_2\}$. Let $s \in V(z_2P_2w) \cap V(y_3P_3z_2)$ where $V(sP_2w) \cap V(y_3P_3z_2) = \{s\}$. Then $C = wP_2sP_3w$ and $T_1 = wP_3t_1P_1t_2P_2w$. Two examples of cycles T, T_1, and C in this case are shown in Figure 3.24(ii) and (iii).

Case 1c: Suppose that $w \in V(x_2P_2u_2)$. Let $t_1 \in V(x_2P_1y_3) \cap V(y_3P_3w)$ where $V(x_2P_1y_3) \cap V(t_1P_3w) = \{t_1\}$. Let $t_2 \in V(x_2P_1y_3) \cap V(x_2P_2w)$ where $V(t_1P_1t_2) \cap V(x_2P_2w) = \{t_2\}$. Let $s \in V(x_2P_2w) \cap V(y_3P_3z_2)$ where $V(sP_2w) \cap V(y_3P_3z_2) = \{s\}$. Then $C = wP_2sP_3w$ and $T_1 = wP_3t_1P_1t_2P_2w$. An example of cycles T, T_1, and C in this case is shown in Figure 3.24(iv)s.

Case 2: Suppose that C is a $\{\text{blue}, \text{green}\}$–cycle.

Case 2a: Suppose that $E(z_3P_3v_3) \cap E(x_2P_1y_3) \neq \emptyset$. Then there is a vertex $w \in V(z_3P_3v_3) \cap V(x_2P_1y_3)$ where $V(z_3P_3w) \cap V(x_2P_1y_3) = \{w\}$. Let $t \in V(x_1P_2z_3) \cap V(z_3P_3w)$ where $V(x_1P_2t) \cap V(y_3P_3w) = \{t\}$. Then $C = wP_1yP_3w$ and $T_1 = wP_3tP_2xP_1w$. An example of cycles T, T_1, and C in this case is shown in

Figure 3.25(i).

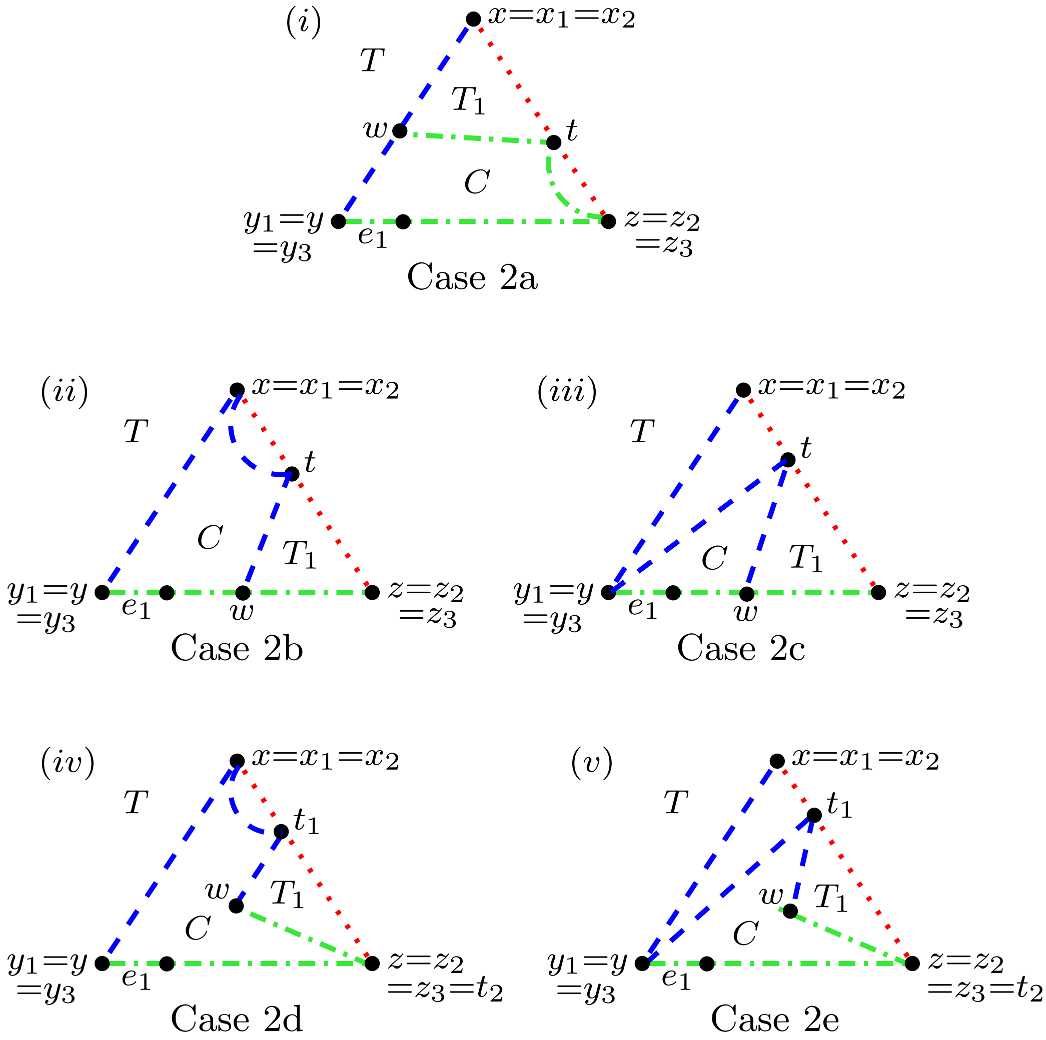

Figure 3.25: Subpaths of P_1, P_2, and P_3 shown by the blue dashed, red dotted, and green dash-dotted lines respectively. An example of cycle T (outer cycle in (i) – (v)), $\{\text{blue}, \text{green}\}$–cycle C, and 3-segment cycle T_1 in each of cases 2a – 2e, shown in (i) – (v) respectively.

Case 2b: Suppose that $V(x_1P_1u_1) \cap V(y_1P_3z_2) \neq \emptyset$ and case 2a does not occur. Then there is a vertex $w \in V(x_1P_1u_1) \cap V(y_1P_3z_2)$ where $V(y_1P_3z_2) \cap V(x_1P_1w) = \{w\}$. Let $t \in V(x_1P_2z_3) \cap V(x_1P_1w)$ where $V(z_3P_2t) \cap V(x_1P_1w) = \{t\}$. Then $C = wP_1yP_3w$ and $T_1 = wP_3zP_2tP_1w$. An example of cycles T, T_1, and C in this case is shown in Figure 3.25(ii).

Case 2c: Suppose that $V(y_3P_1v_1) \cap V(y_1P_3z_2) \neq \emptyset$ and cases 2a and 2b do not occur. Then there is a vertex $w \in V(y_3P_1v_1) \cap V(y_1P_3z_2)$ where $V(y_1P_3z_2) \cap V(y_3P_1w) = \{w\}$. Let $t \in V(x_1P_2z_3) \cap V(y_3P_3w)$ where $V(z_3P_2t) \cap V(y_3P_3w) = \{t\}$. Then $C = wP_1yP_3w$ and $T_1 = wP_3zP_2tP_1w$. An example of cycles

T, T_1, and C in this case is shown in Figure 3.25(iii).

Case 2d: Suppose that $V(x_1P_1u_1) \cap V(z_2P_3v_2) \neq \emptyset$ and cases 2a, 2b, and 2c do not occur. Then there is a vertex $w \in V(x_1P_1u_1) \cap V(z_2P_3v_2)$ where $V(x_1P_1w) \cap V(z_2P_3v_2) = \{w\}$. Let $t_1 \in V(x_1P_1w) \cap V(x_1P_2z_3)$ where $V(t_1P_1w) \cap V(x_1P_2z_3) = \{t_1\}$. Let $t_2 \in V(z_3P_3w) \cap V(x_1P_2z_3)$ where $V(t_2P_3w) \cap V(x_1P_2z_3) = \{t_2\}$. Then $C = wP_1yP_3w$ and $T_1 = wP_3t_2P_2t_1P_1w$. An example of cycles T, T_1, and C in this case is shown in Figure 3.25(iv).

Case 2e: Suppose that $V(y_1P_1v_1) \cap V(z_2P_3v_2) \neq \emptyset$ and cases 2a, 2b, 2c, and 2d do not occur. Then there is a vertex $w \in V(x_1P_1u_1) \cap V(z_2P_3v_2)$ where $V(y_1P_1w) \cap V(z_2P_3v_2) = \{w\}$. Let $t_1 \in V(y_1P_1w) \cap V(x_1P_2z_3)$ where $V(t_1P_1w) \cap V(x_1P_2z_3) = \{t_1\}$. Let $t_2 \in V(z_3P_3w) \cap V(x_1P_2z_3)$ where $V(t_2P_3w) \cap V(x_1P_2z_3) = \{t_2\}$. Then $C = wP_1yP_3w$ and $T_1 = wP_3t_2P_2t_1P_1w$. An example of cycles T, T_1, and C in this case is shown in Figure 3.25(v).

If there is a $\{\text{blue}, \text{green}\}$–cycle C_2 with $e_1 \in E(C_2)$ such that there is an edge $f \in E(C_2)$ with $f \in E(y_3P_3u_3)$ and $f \notin E(P_1) \cup E(P_2)$, then one of the previous subcases of case 2 also occurs.

Now, in every case, there is a bi-coloured cycle C and a 3-segment cycle T_1 such that $T = T_1 \oplus C \oplus \mathcal{S}$ where $\mathcal{S}$ is a (possibly empty) set of bi-coloured cycles in $\mathcal{C} \backslash \{C\}$. Since there is a set of at most six bi-coloured cycles in $\mathcal{C}$ such that T is the mod 2 sum of these cycles, then T_1 is the mod 2 sum of at most five bi-coloured cycles in $\mathcal{C} \backslash \{C\}$. Repeating this process for T_1 and subsequent 3-segment cycles, we obtain a 3-segment cycle that is (the mod 2 sum of) a bi-coloured cycle, a contradiction. □

3.5.7 Proof of Theorem 3.1

We first prove the following theorem.

Theorem 3.22. *The graph $\check{H}$ has cyclomatic number at least 7.*

Proof. By Proposition 3.13 there exists at least one set $\mathcal{B} \in \mathfrak{B}$ of six bi-coloured cycles of $\check{H}$ consisting of two $\{\text{blue}, \text{red}\}$–cycles, two $\{\text{blue}, \text{green}\}$–cycles, and two $\{\text{red}, \text{green}\}$–cycles such that, for each pair $c_1, c_2 \in \text{col}(\check{H})$, the two $\{c_1, c_2\}$–cycles in $\mathcal{B}$ are both $[1\text{-}c_1, c_2]$–cycles or both $[c_1, 1\text{-}c_2]$–cycles. Let $F \in \mathcal{B}$ be a $\{\text{red}, \text{green}\}$–cycle without loss of generality. By Lemmas 3.16 and 3.18 to 3.20, there is no $\mathcal{B}' \subseteq \mathcal{B} \backslash \{F\}$ such that F is the mod 2 sum of the cycles in $\mathcal{B}'$. Hence the six cycles in $\mathcal{B}$ are independent for every set $\mathcal{B} \in \mathfrak{B}$. By Lemmas 3.9 and 3.21 there is a tri-coloured cycle C of $\check{H}$ that is not a bi-coloured cycle and there exists a set $\mathcal{B} \in \mathfrak{B}$ such that C is independent of the cycles in $\mathcal{B}$. Then by Lemma 3.4, $\check{H}$ has cyclomatic number at least 7. $\square$

We now return to the main result of this chapter, restated below.

Theorem 3.1. *Let G be a graph. If G has cyclomatic number at most 6 then every three longest paths of G have a common vertex.*

Proof. Assume that there exist three longest paths P_1, P_2, and P_3 of G that do not have a common vertex. Let G' be the subgraph $P_1 \cup P_2 \cup P_3$ of G. Then G' is a counterexample to Conjecture 1. By Theorem 3.22, a minimal (with respect to edges) counterexample to conjecture Conjecture 1 has cyclomatic number at least 7, and hence G' has cyclomatic number at least 7. Since G' is a subgraph of G, then G also has cyclomatic number at least 7. Therefore, in every graph with cyclomatic number at most 6, every three longest paths have a common vertex. $\square$

3.6 Concluding Remarks

In this chapter, we have shown that Conjecture 1 holds for graphs with cyclomatic number at most 6, that is, we have proved that in a graph with n vertices and at most $n + 5$ edges, every three longest paths have a common vertex. In the course of proving this, we also prove that a minimal (with respect to edges) counterexample to Conjecture 1 has cyclomatic number at least 7.

We first observe that our result cannot be extended to all longest paths. The graph shown in Figure 3.26, which was found by Schmitz [31] and is discussed in Section 1.2, has cyclomatic number 3 and does not have a Gallai vertex.

It is natural to ask whether our result can be extended to graphs with higher cyclomatic number. Recall that $\check{H}$ is a graph that is the union of its three longest paths, P_1, P_2, and P_3, and these paths do not have a common vertex. There are two steps to extending the methods used in this chapter: first, proving that there are more cycles in $\check{H}$ and, second, proving that these cycles are independent of the existing set of seven independent cycles of $\check{H}$.

Step 1: One way to show that there are more cycles in $\check{H}$ is to show that there is a seventh bi-coloured cycle in $\check{H}$, that is, to show that there are at least three cycles in at least one pair of the three longest

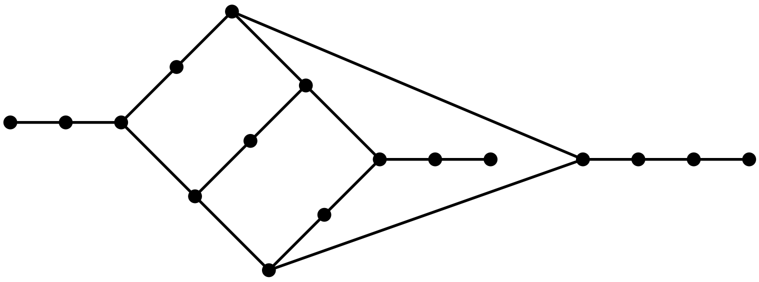

Figure 3.26: A planar graph with cyclomatic number 3 that does not have a Gallai vertex, found by Schmitz [31].

paths of $\check{H}$. This would extend Axenovich's result, Lemma 3.8, that there are at least two cycles in every pair of the longest paths of $\check{H}$. However, from our initial investigations, this result is not easily extended. Another way to show that there are more cycles in $\check{H}$ would be to show that there is more than one tri-coloured cycle of $\check{H}$ that is not a bi-coloured cycle.

Step 2: If it can be proved that there are more cycles in $\check{H}$, it then remains to prove that these cycles are independent of the existing set of seven independent cycles of $\check{H}$. This gets progressively more difficult the more cycles $\check{H}$ has. For additional bi-coloured cycles, results are required for the sum of more than five bi-coloured cycles. There are also extra configurations to be considered in several of the existing results in Section 3.5, which become increasingly complex and numerous, and the methods used in this chapter do not easily extend to these cases. The proof that there is a tri-coloured cycle in $\check{H}$ that is independent of a set of six bi-coloured cycles of $\check{H}$ is also not easily extended to more tri-coloured cycles. Since the mod 2 sum of a tri-coloured cycle and a bi-coloured cycle that are not edge-disjoint is a tri-coloured cycle, which is central to the proof of Lemma 3.21, proving that there is a tri-coloured cycle of $\check{H}$ that is independent of the existing seven independent cycles of $\check{H}$ requires a different approach. Therefore, additional techniques are required to show that there are more than seven independent cycles of $\check{H}$.

In Chapter 4, we develop techniques that allow us to independently verify Theorem 3.1 using computational methods.

Chapter 4

Computational Investigations

4.1 Introduction

In this chapter, we use computational methods to determine whether every graph has a Gallai vertex for two specific classes of graphs, and whether every three or six longest paths have a common vertex for several other classes of graphs. In 2013, Brinkman and van Cleemput (see [32]) used computational methods to show that every graph with at most 11 vertices has a Gallai vertex. Additionally, McKay [28] has used computers to show that there is only one graph with 12 vertices that does not have a Gallai vertex, the graph discussed in Section 1.3.3.

Our approach allows us to obtain results about an infinite set of graphs by testing a finite subset of that set. Recall that two graphs G_1 and G_2 are *homeomorphic* if there exists a subdivision of G_1 that is isomorphic to a subdivision of G_2. Equivalently, G_1 and G_2 are homeomorphic if there exists a graph H such that G_1 and G_2 are subdivisions of H. An example of such graphs is shown in Figure 4.1. Note that this implies that G_1 and H are homeomorphic, as are G_2 and H. All graphs in this chapter are connected unless stated otherwise. Let G be a graph and let $\mathcal{G}$ be the set of graphs that are homeomorphic to G. If H is the smallest (with respect to edges) simple graph in $\mathcal{G}$, then we call H a *simple reduced* graph. A simple reduced graph H of a set $\mathcal{G}$ can be used to represent all of the simple graphs in $\mathcal{G}$ by assigning a weight to each of its edges, as described further in Section 4.2.1.

Let G be a graph, let $\mathcal{G}$ be the set of simple graphs homeomorphic to G, and let H be a simple

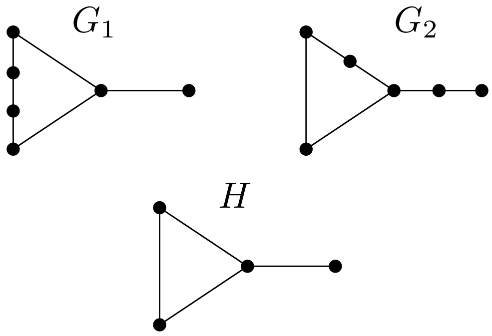

Figure 4.1: Two homeomorphic graphs G_1 and G_2 that are subdivisions of the graph H.

reduced graph of $\mathcal{G}$. We generate a set of inequalities on the weights of the edges of H and walks of H that are maximal with respect to particular properties, and use a program that enumerates the vertices of the polyhedron satisfying this set of inequalities. We then check each vertex of the polyhedron, which corresponds to a graph in $\mathcal{G}$, to determine whether the longest paths of the graph have a common vertex. If every graph corresponding to a vertex of the polyhedron has a Gallai vertex, then every simple graph in $\mathcal{G}$ has a Gallai vertex. If this is not the case, we can investigate further to determine which of the graphs in $\mathcal{G}$ do not have a Gallai vertex. The theory behind this method is detailed further in Section 4.2, with a summary of the implementation in Section 4.3, and results presented in Section 4.4.

In Section 4.5, we describe an alternative method that uses linear programming instead of the vertex enumeration of a polyhedron. We provide a number of properties of a (not necessarily simple) reduced graph that has a subdivision G with three longest paths that have no common vertex and where G is minimal with respect to edges. We then present several results using this linear programming method. Lastly, we provide concluding remarks in Section 4.6.

4.2 Theory

Let G be a simple graph. The opposite of subdivision, *smoothing* a degree two vertex $w \in V(G)$, is the operation of deleting w and its incident edges uw and wv and inserting an edge uv. A *chain* of G is a path of G whose interior vertices have degree two. Two distinct chains of G are *parallel* if they have a pair of common endpoints. Note that two parallel chains are internally disjoint by definition of a chain. Let G_1 be a simple graph homeomorphic to G such that G is a subdivision of G_1. Then G can be obtained from G_1 by deleting each edge $e = uv$ of G_1 and inserting a chain Q of length at least one with endpoints u and v. We say that edge e of G_1 *corresponds* to chain Q of G, and Q *corresponds* to e. Note that $V(G_1) \subseteq V(G)$ and $\deg_{G_1}(v) = \deg_G(v)$ for all $v \in V(G_1)$.

4.2.1 Homeomorphic graphs

Let G be a simple graph and let H be a simple reduced graph homeomorphic to G. The graph H can be obtained from G by repeatedly smoothing degree two vertices of G where such smoothing does not result in a non-simple graph. Additionally, H is unique, up to isomorphism. To see this, observe that the only case in which the order of smoothing vertices of degree two matters is when smoothing degree two vertices of a set of parallel chains. Suppose that G has a set of $k \geq 2$ parallel chains with a pair of common endpoints $u, v \in V(G)$ and that there is no edge $uv \in E(G)$. Then one of these k chains corresponds to the edge uv of H, while the remaining $k-1$ chains correspond to chains of length two of H. However, regardless of which of these chains of G corresponds to the edge uv of H, H has an edge uv and $k-1$ chains of length two with a pair of common endpoints u and v, and therefore H is unique up to isomorphism.

Suppose that H has n vertices and m edges, and let $e_1, \dots, e_m$ be the edges of H. Assign to each edge $e_i \in E(H)$ $(1 \leq i \leq m)$ a positive integer weight $w(e_i)$. Let $\mathcal{G}_H$ be the set of simple graphs homeomorphic to H, and let $G \in \mathcal{G}_H$. The graph G corresponds uniquely to a set of positive integer edge weights w on the edges of H, where G can be obtained from H by replacing each edge $e \in E(H)$ with a chain of length $w(e)$. For example, in Figure 4.2, the graphs G_1 and G_2 correspond to the edge weighted graphs directly below. To obtain G_1 from H, the edge with weight 3 is replaced by a chain of length 3. Note that every set of positive integer edge weights on H corresponds uniquely to a graph in $\mathcal{G}_H$.

We next define a set of walks of H that corresponds to a set of maximal paths of a graph in $\mathcal{G}_H$.

4.2.2 Maximal walks and paths

Let H be a simple reduced graph. We consider walks $W = v_1 v_2 \dots v_k$ of H $(k \geq 1)$ where the vertices and edges of W are unique except that:

(i) v_1 is one of $v_2, \dots, v_{k-1}$ unless it is a leaf,

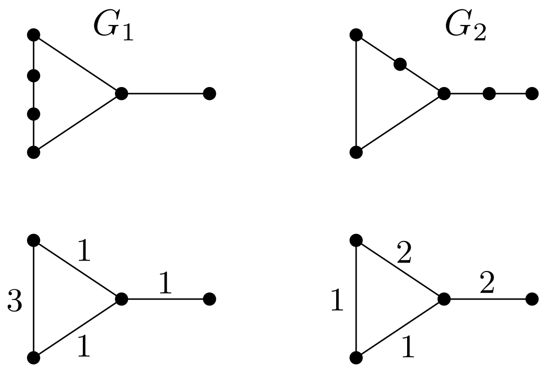

Figure 4.2: The two graphs G_1 and G_2 in $\mathcal{G}_H$ from Figure 4.1, along with their corresponding edge weights on the graph H.

(ii) v_k is one of $v_2, \ldots, v_{k-1}$ unless it is a leaf, and

(iii) we may have $v_1 v_2 = v_{k-1} v_k$.

We define $\mathcal{W}(H)$ to be the set of such walks W that are maximal with respect to these conditions. Let $W \in \mathcal{W}(H)$. If (iii) holds for W, we say that the ends of W *overlap*. Note that we do not have $v_1 = v_k$ as then W is not maximal; we instead have $v_1 = v_{k-1}$ and $v_k = v_2$. An endpoint of W that is not a leaf is *repeated*.

We differentiate five types of walks in $\mathcal{W}(H)$ based on properties of their endpoints. For a walk $W = v_1 v_2 \ldots v_k$ $(k \geq 1)$ in $\mathcal{W}(H)$:

(a) both v_1 and v_k are leaves,

(b) v_1 or v_k is a leaf and the other is repeated,

(c) both v_1 and v_k are repeated and $v_1 = v_k$,

(d) both v_1 and v_k are repeated but $v_1 \neq v_k$ and the ends of W do not overlap, or

(e) both v_1 and v_k are repeated and the ends of W overlap.

Figure 4.3 shows examples of these five different types of walks in $\mathcal{W}(H)$ for a graph H.

Let $\mathcal{G}_H$ be the set of simple graphs homeomorphic to H. Let $G \in \mathcal{G}_H$ and let $W \in \mathcal{W}(H)$. Then there is a walk W_G of G corresponding to the walk W of H, obtained by replacing each edge of W with the corresponding chain of G. Examples of this are shown in Figure 4.4 in the left and middle columns. Let $\mathcal{W}(G)$ be the set of such walks of G obtained from the walks in $\mathcal{W}(H)$. Note that the five types of walks in $\mathcal{W}(H)$ can be defined analogously for the walks in $\mathcal{W}(G)$, and that a walk $W \in \mathcal{W}(H)$ and

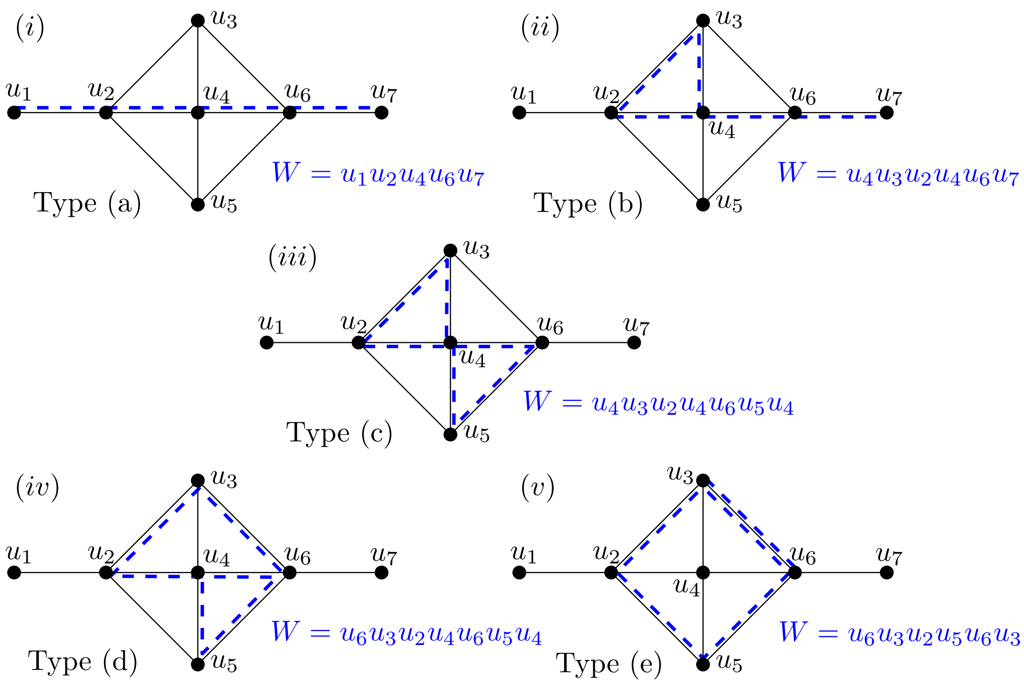

Figure 4.3: Examples of the five different types of walks (a) – (e) in $\mathcal{W}(H)$ shown in (i) – (v) respectively for a graph H (walks shown with blue dashed lines).

the corresponding walk $W_G \in \mathcal{W}(G)$ are of the same type. To obtain the walk $W \in \mathcal{W}(H)$ from the corresponding walk $W_G \in \mathcal{W}(G)$, replace each minimal subpath of W_G whose endpoints are in $V(H)$ with an edge.

We obtain a path P of G from a walk $W_G \in \mathcal{W}(G)$ by the following operations, performed once each in order.

(1) If the ends of W_G do not overlap, then for each of the two endpoints of W_G, if the endpoint is repeated, then remove the endpoint, as shown in the examples in Figure 4.4(i) and (ii).

(2) If the ends of W_G overlap, then we have the case $v_1 = v_{k-1}$ and $v_k = v_2$. Remove vertices alternately from the two ends of W_G, stopping when no vertices are repeated, as shown in the example in Figure 4.4(iii). If there is more than one possible resulting path, pick one without loss of generality (as discussed later, in Lemma 4.3).

We say that P is a path of G corresponding to W and W_G. Note that the vertices in $V(H) \cap V(W)$ are also vertices of W_G and P. Let $\mathcal{P}(G)$ be the set of all such paths P that are maximal paths of G.

Note that not all paths P obtained by performing operations (1) and (2) on a walk in $\mathcal{W}(G)$ are maximal paths of G. If at least one endpoint of P is a vertex of $V(H)$ that is not a leaf of G, then P may not be a maximal path of G. This may occur when at least one end of W_G is repeated but the ends do not overlap, and when the ends of W_G overlap and the chain of G with endpoints v_1 and v_2 has length at most two. An example of the latter situation is shown in Figure 4.4(iv). However, if P is not a maximal path of G, there is a maximal path Q of G such that P is a subpath of Q. Furthermore, there exists such a Q where $Q \in \mathcal{P}(G)$. To see this, we construct a walk W_1 of G such that W_1 corresponds to a walk $W_2 \in \mathcal{W}(H)$ and corresponds to path Q. Let y_1 and y_2 be the endpoints of Q. If y_1 is a leaf, let $Q_1 = y_1$. If y_1 is an interior vertex of a minimal chain of G whose endpoints are in $V(H)$, let Q_1 be this chain. If y_1 is a vertex of $V(H)$, then since Q is maximal there is a vertex x_1 of G incident to y_1 such that $x_1 \in V(H) \cap V(Q)$. In this case, let $Q_1 = y_1x_1$. Similarly define Q_2 for y_2. Now $W_1 = Q_1y_1Qy_2Q_2$ is a walk of G whose endpoints are in $V(H)$, and the corresponding walk W_2 of H satisfies the conditions of a walk in $\mathcal{W}(H)$.

To obtain a walk in $\mathcal{W}(H)$ corresponding to a path in $\mathcal{P}(G)$, we reverse this process. Let $P \in \mathcal{P}(G)$, and let its two endpoints be u and v. First note that since G is a subdivision of H, then $V(H) \subseteq V(G)$. Since P is a maximal path of G, then every vertex of G adjacent to u is in $V(P)$. Let $x \in V(G)$ be adjacent to u with $xu \notin E(P)$. If $x \notin V(H)$, then x has degree two, hence x is not an interior vertex of P, and $x = v$. Therefore $x \in V(H)$ or $x = v$. Note that if P has length one, then $G = K_2 = H$ and every graph homeomorphic to H has a Gallai vertex, so we do not consider this case. We obtain a walk $W_G \in \mathcal{W}(G)$ from P as follows:

(1) If $uv \in E(G)\backslash E(P)$, then we have the case in which the ends of W_G overlap. Let Q be the path of G with $u, v \in V(Q)$ and endpoints $y, z \in V(H)$ where $V(Q) \cap V(H) = \{y, z\}$, and assume without loss of generality that these vertices are in the order y, u, v, z in Q. (Note that it may be the case that $u = y$ or $v = z$.) Then let W_G be the walk $zQuPvQy$.

(2) If $uv \notin E(G)$, then we have one of the other four types of walks in $\mathcal{W}(G)$. If u is not a leaf of G, let $y \in V(G)$ adjacent to u where $uy \notin E(P)$. By our earlier reasoning, $y \in V(H)$. If there is more than one such vertex y, pick one without loss of generality (as discussed later, in Lemma 4.3). Let W_1 be the walk $P \cup uy$. If u is a leaf of G, let $W_1 = P$. If v is not a leaf of G, define z similarly to y, and let W_G be the path $W_1 \cup vz$. As before, if there is more than one such vertex z, pick one

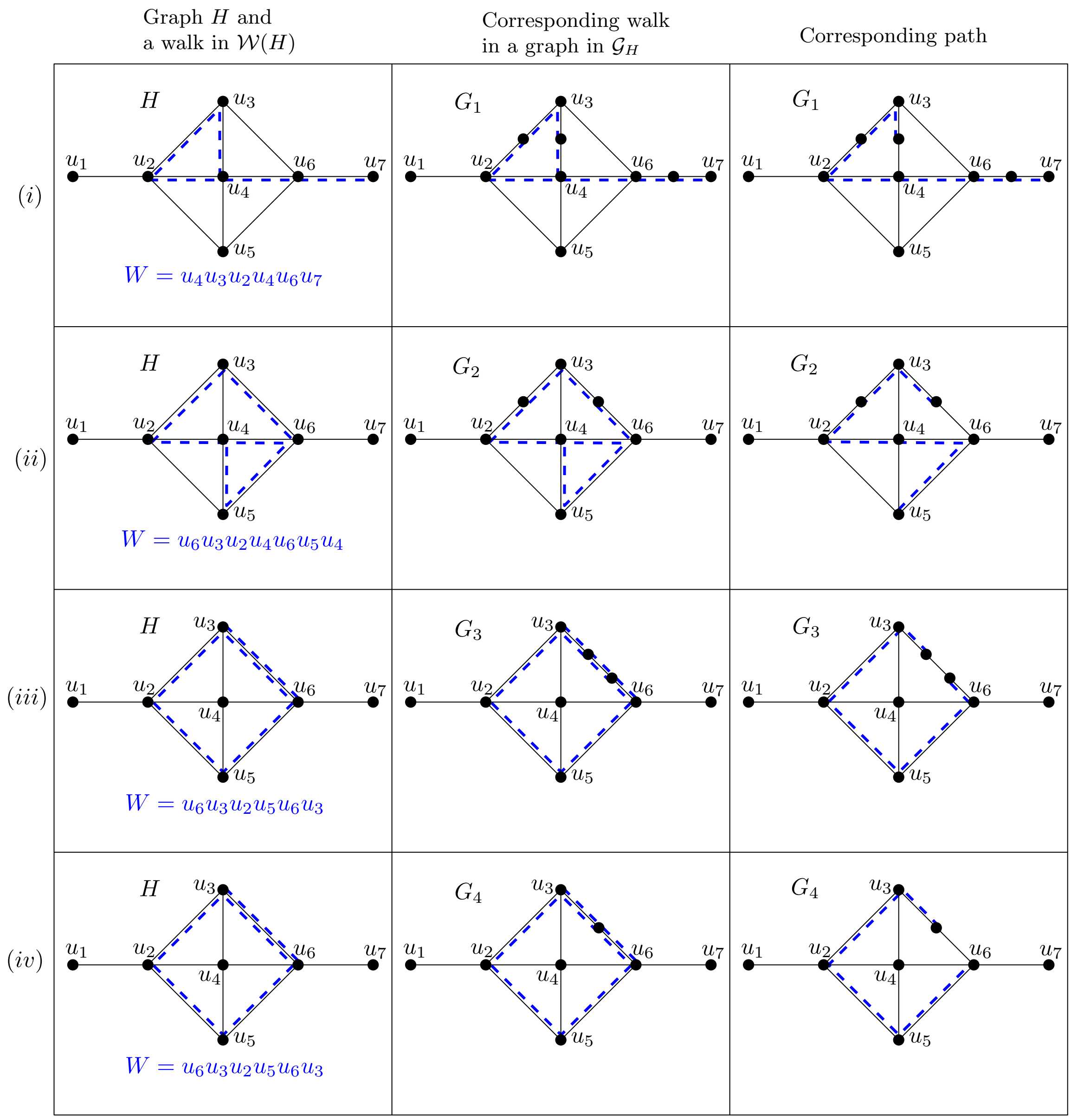

Figure 4.4: (i), (ii), (iii), and (iv) show walks in $\mathcal{W}(H)$ in the graph H (left, walks shown with blue dashed lines), the corresponding walks in the graphs G_1, G_2, G_3, and G_4 in $\mathcal{G}_H$ respectively (middle; only the vertices of H are labeled), and the corresponding paths in G_1, G_2, G_3, and G_4 (right).

without loss of generality. If v is a leaf of G, let $W_G = W_1$.

Now W_G is a walk in $\mathcal{W}(G)$ corresponding to P.

4.2.3 Polyhedra

In this section, we describe a system of linear inequalities representing the edge weights of the graph H and the walks in $\mathcal{W}(H)$. We then describe a geometrical object defined by these inequalities, whose properties are crucial to our method.

We first require the following definition. A set X of vectors in $\mathbb{R}^n$ is a *(convex) polyhedron* if $X = \{\underline{x} \mid A\underline{x} \leq \underline{b}\}$ for some matrix A and vector $\underline{b}$. We say that $A\underline{x} \leq \underline{b}$ determines or defines X.

For a walk $W = v_1 v_2 \dots v_k$ in $\mathcal{W}(H)$ $(k \geq 1)$, let $\alpha(W) \in \{0, 1, 2\}$ be defined by:

$$\alpha(W) = \begin{cases} 0 & \text{if both } v_1 \text{ and } v_k \text{ are leaves;} \\ 1 & \text{if exactly one of } v_1 \text{ or } v_k \text{ is a leaf, or the ends of } W \text{ overlap;} \\ 2 & \text{otherwise.} \end{cases}$$

As discussed in Section 4.2.1, a set of values for the weight variables $w(e)$, $e \in E(H)$, corresponds to a graph $G \in \mathcal{G}_H$. For a walk $W \in \mathcal{W}(H)$, the length of a path of G corresponding to W is $\sum_{e \in W} w(e) - \alpha(W)$. We also require another variable ℓ, representing an upper bound on the length of the paths of G. We now have the following system of inequalities representing the graph H and the set of walks $\mathcal{W}(H)$.

Definition 1. *The* polyhedron of H *is the set of vectors in* $\mathbb{R}^{m+1}$*, where* $m = |E(H)|$*, satisfying the following system of inequalities (constraints).*

Variables: $w(e)$ *for each* $e \in E(H)$*, and* ℓ

Constraint 1: $w(e) \geq 1$ *for each edge* e *of* H*,*

Constraint 2: $\ell \geq 1$*,*

Constraint 3: $\left(\sum_{e \in W} w(e)\right) - \alpha(W) \leq \ell$ *for each walk* $W \in \mathcal{W}(H)$*.*

By the well-known decomposition theorem for polyhedra (see [45, Theorem 1.2], for example), a polyhedron is determined by its set of vertices $\underline{v}^i$ $(i \geq 1)$ and its set of rays $\underline{r}^j$ $(j \geq 0)$. Then, a point $\underline{x}$ in the polyhedron of H can be expressed as the sum of a convex combination of the vertices and a linear combination of the rays of the polyhedron:

$$\underline{x} = \sum_i \lambda_i \underline{v}^i + \sum_j \mu_j \underline{r}^j$$

where $\lambda_i \geq 0, \sum_i \lambda_i = 1$, and $\mu_j \geq 0$.

Let the edges of H be $e_1, e_2, \ldots, e_m$ ($m \geq 1$). A point $\underline{x}$ in the polyhedron is an $(m+1)$-dimensional vector $(w(e_1), \ldots, w(e_m), \ell)$. A walk $W \in \mathcal{W}(H)$ is *extremal* at $\underline{x}$ if Constraint 3 is satisfied with equality for W and the values of $w(e_k)$ in $\underline{x}$.

For a point $\underline{x}$ in the polyhedron whose entries are all positive integers, the values of the $w(e_k)$ in $\underline{x}$ correspond to a unique graph $G_{\underline{x}} \in \mathcal{G}_H$, and the value of ℓ in $\underline{x}$, denoted $\ell_{\underline{x}}$ ($= x_{m+1}$), is an upper bound on the length of the paths of $G_{\underline{x}}$. If $\ell_{\underline{x}}$ is a tight upper bound on the length of the paths of $G_{\underline{x}}$, that is, there is at least one extremal walk $W \in \mathcal{W}(H)$ at the point $\underline{x}$, then the set of walks in $\mathcal{W}(H)$ that are extremal at $\underline{x}$ corresponds to the set of all longest paths of $G_{\underline{x}}$.

Note that for each graph $G \in \mathcal{G}_H$, there is a point $\underline{y}$ in the polyhedron such that $G_{\underline{y}} = G$ and $\ell_{\underline{y}}$ is the length of the longest path(s) of G (that is, $\ell_{\underline{y}}$ is a tight upper bound on the lengths of the paths of G). For a walk $W \in \mathcal{W}(H)$, let $s_{\underline{x}}(W)$ be the length of the corresponding walk in $\mathcal{W}(G_{\underline{x}})$, that is,

$$s_{\underline{x}}(W) = \sum_{k: e_k \in W} x_k.$$

Then the length of the corresponding path in $\mathcal{P}(G_{\underline{x}})$ is

$$s_{\underline{x}}(W) - \alpha(W) = \Big(\sum_{k: e_k \in W} x_k\Big) - \alpha(W).$$

From this, we can determine which of the paths in $\mathcal{P}(G_{\underline{x}})$ are of length $\ell_{\underline{x}}$ and then determine whether these paths have a common vertex.

The following lemma is essential to our method as it allows us to restrict our interest to the vertices of the polyhedron, a finite subset of the points in the polyhedron.

Lemma 4.1. *Let H be a simple reduced graph. Let the polyhedron of H have vertices $\underline{v}^i$ ($i \geq 1$) and rays $\underline{r}^j$ ($j \geq 0$). Let $\underline{x}$ be a point in the polyhedron of H,*

$$\underline{x} = \sum_i \lambda_i \underline{v}^i + \sum_j \mu_j \underline{r}^j$$

where $\lambda_i \geq 0$, $\sum_i \lambda_i = 1$, and $\mu_j \geq 0$.

For each i, if $\lambda_i > 0$, then the set of walks in $\mathcal{W}(H)$ that are extremal at $\underline{x}$ is a subset of the set of walks in $\mathcal{W}(H)$ that are extremal at $\underline{v}^i$.

Proof. Let $W \in \mathcal{W}(H)$. Let

$$f_W(\underline{x}) = \ell_{\underline{x}} - s_{\underline{x}}(W)$$

and note that this function is linear. Suppose that W is extremal at $\underline{x}$. Then

$$\ell_{\underline{x}} - (s_{\underline{x}}(W) - \alpha(W)) = 0$$

and therefore

$$f_W(\underline{x}) + \alpha(W) = 0.$$

Note that, for every point $\underline{w}$ in the polyhedron, since $\ell_{\underline{w}}$ is a upper bound on $s_{\underline{w}}(W) - \alpha(W)$, it is always the case that $f_W(\underline{w}) + \alpha(W) \geq 0$.

Suppose that the vertices and rays defining the polyhedron include at least one ray. Let $\lambda_i \geq 0$, $\sum_i \lambda_i = 1$, and $\mu_j \geq 0$ and let $\underline{x}_1$ be the point in the polyhedron

$$\underline{x}_1 = \sum_i \lambda_i \underline{v}^i + \sum_j \mu_j \underline{r}^j.$$

We show that the set of walks in $\mathcal{W}(H)$ that are extremal at $\underline{x}_1$ are extremal at $\sum_i \lambda_i \underline{v}^i$. Suppose that there is at least one ray $\underline{r}^j$, $j \geq 1$, such that $\mu_j \neq 0$. Consider another point in the polyhedron

$$\underline{x}_2 = \sum_i \lambda_i \underline{v}^i + 2\sum_j \mu_j \underline{r}^j.$$

Then $f_W(\underline{x}_2) + \alpha(W) \geq 0$ since this is true everywhere. Now consider the point in the polyhedron

$$\underline{x}_3 = \sum_i \lambda_i \underline{v}^i$$

and consider $f_W(\underline{x}_3)$. First observe that $\underline{x}_3 = 2\underline{x}_1 - \underline{x}_2$. Then

$$\begin{aligned}
f_W(\underline{x}_3) &= f_W(2\underline{x}_1 - \underline{x}_2) \\
&= 2f_W(\underline{x}_1) - f_W(\underline{x}_2) \quad \text{since } f_W \text{ is linear} \\
&= -2\alpha(W) - f_W(\underline{x}_2) \quad \text{since } f_W(\underline{x}_1) + \alpha(W) = 0 \\
&\leq -2\alpha(W) + \alpha(W) \quad \text{since } f_W(\underline{x}_2) \geq -\alpha(W) \\
&= -\alpha(W).
\end{aligned}$$

Now $f_W(\underline{x}_3) \leq -\alpha(W)$ and hence $f_W(\underline{x}_3) + \alpha(W) \leq 0$. However, $f_W(\underline{w}) + \alpha(W) \geq 0$ for all $\underline{w}$ in the polyhedron, and hence $f_W(\underline{x}_3) + \alpha(W) = 0$ and W is extremal at $\underline{x}_3$. Therefore every walk $W \in \mathcal{W}(H)$ that is extremal at $\underline{x}_1$ is extremal at $\underline{x}_3$. Hence, we only need to consider the case in which $\mu_j = 0$ for all j.

Suppose that $\underline{x} = \sum_i \lambda_i \underline{v}^i$ where $\lambda_i \geq 0$ and $\sum_i \lambda_i = 1$, and let $W \in \mathcal{W}(H)$ be extremal at $\underline{x}$. Then

$$f_W(\underline{x}) + \alpha(W) = 0$$

so

$$f_W\big(\sum_i \lambda_i \underline{v}^i\big) + \alpha(W) = 0.$$

Then

$$\big(\sum_i \lambda_i f_W(\underline{v}^i)\big) + \alpha(W) = 0$$

since f_W is linear, and then

$$\sum_i \lambda_i (f_W(\underline{v}^i) + \alpha(W)) = 0$$

since $\sum_i \lambda_i = 1$. Recall that $f_W(\underline{w}) + \alpha(W) \geq 0$ for every point $\underline{w}$ in the polyhedron, and that $\lambda_i \geq 0$ for all i ($i \geq 1$). Therefore, for each i, if $\lambda_i > 0$, then $f_W(\underline{v}^i) + \alpha(W) = 0$. Hence, every walk $W \in \mathcal{W}(H)$ that is extremal at $\underline{x}$ is extremal at $\underline{v}^i$ for each i where $\lambda_i > 0$. □

4.2.4 An optimisation

Lemma 4.2. *Let H be a simple reduced graph, and let $W, X \in \mathcal{W}(H)$. For each simple graph G homeomorphic to H, the walks W and X have a common vertex if and only if the corresponding paths in $\mathcal{P}(G)$ have a common vertex.*

Proof. Let $W_G, X_G \in \mathcal{P}(G)$ correspond to W and X respectively. Suppose that W and X have a common vertex $v \in V(H)$. Then $v \in V(W_G) \cap V(X_G)$, as required. Suppose that W_G and X_G have a common vertex $u \in V(G)$ and there is no vertex $x \in V(H)$ where x is a common vertex of W_G and X_G. Let Q be the minimal chain of G with $u \in V(Q)$ whose endpoints are vertices of H. Then the endpoints of Q are vertices of W_G and X_G (by definition of W_G and X_G, whether or not W_G and X_G are maximal paths of G), a contradiction. Hence there is a vertex $x \in V(H)$ where $x \in V(W_G) \cap V(X_G)$ and therefore $x \in V(W) \cap V(X)$. □

Corollary 4.3. *Let H be a simple reduced graph, and let G be a simple graph homeomorphic to H. Let $W, X \in \mathcal{W}(H)$ and let $P, Q, R \in \mathcal{P}(G)$ where P and Q correspond to W, and R corresponds to X. Paths P and R have a common vertex if and only if paths Q and R have a common vertex.*

Lemma 4.4. *Let H be a simple reduced graph. Let $W, X \in \mathcal{W}(H)$. If $E(W) = E(X)$ then $\alpha(W) = \alpha(X)$.*

Proof. Assume that $E(W) = E(X)$. Then $V(W) = V(X)$ and, for each $v \in V(W)$, $\deg_W(v) = \deg_X(v)$. First suppose that $\alpha(W) = 0$. Then W is a maximal path of H, and hence $X = W$ and therefore $\alpha(X) = 0$. Next, suppose that $\alpha(W) = 1$. Then W has exactly one leaf or the ends of W overlap. In the former case, X also has exactly one leaf, and hence $\alpha(X) = 1$. In the latter case, $E(W)$ is the edge set of a cycle in H, and similarly for $E(X)$. Therefore the ends of X overlap, and hence $\alpha(X) = 1$. Lastly, suppose that $\alpha(W) = 2$. Then there exists $v \in V(W)$ with $\deg_W(v) = 4$ or there exist $y, z \in V(W)$ with $\deg_W(y) = \deg_W(z) = 3$. These vertices have the same degree in X. In either case, both ends of X are repeated, but the ends do not overlap, and hence $\alpha(X) = 2$. $\square$

By Corollary 4.3 and Lemma 4.4, if the edge set of a walk $X \in \mathcal{W}(H)$ is a subset of the edge set of a walk $W \in \mathcal{W}(H)$, then, since H is simple, replacing the set of walks $\mathcal{W}(H)$ with the set of walks $\mathcal{W}(H)\backslash\{X\}$ does not change the polyhedron of H or affect whether G contains a Gallai vertex or satisfies Conjecture 1. Therefore, in the previous sections, instead of $\mathcal{W}(H)$, we can use a subset $\mathcal{W}'(H)$ of $\mathcal{W}(H)$ where there is no walk in $\mathcal{W}'(H)$ whose edge set is a subset of, or equal to, another walk in $\mathcal{W}'(H)$, and, for each walk $W \in \mathcal{W}(H)$, there is a walk $W' \in \mathcal{W}'(H)$ such that $E(W) \subseteq E(W')$.

4.2.5 Worked example

In this section, we work through an example of the entire process, from the graph H, to the polyhedron of H, to the intersection of maximal paths in graphs in $\mathcal{G}_H$. This example is a very simple one but it serves to illustrate the process without being unduly long or complicated.

Consider the graph G in Figure 4.5 and the smallest simple graph homeomorphic to G, the graph H in the same figure. Let $\mathcal{G}_H$ be the set of simple graphs homeomorphic to H, a few of which are shown in Figure 4.5.

Consider the set $\mathcal{W}(H)$ of walks in H, as defined previously, and let $W, X \in \mathcal{W}(H)$. By Lemma 4.2, for each graph $G \in \mathcal{G}_H$, the walks W and X have a common vertex of H if and only if their corresponding paths of G have a common vertex. As mentioned previously, if $E(X) \subseteq E(W)$, we can consider the subset $\mathcal{W}(H)\backslash\{X\}$ instead of $\mathcal{W}(H)$ without changing, for any graph $G \in \mathcal{G}_H$, whether G contains a Gallai vertex or satisfies Conjecture 1. Let $\mathcal{W}'(H) \subseteq \mathcal{W}(H)$ be a set of walks as defined in the

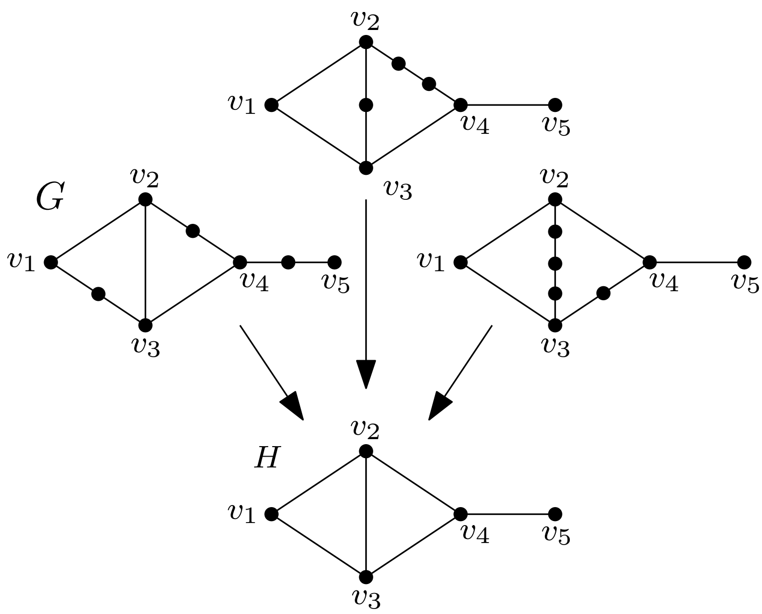

Figure 4.5: A graph H and three of the graphs in $\mathcal{G}_H$ homeomorphic to H.

previous section. For the graph H in Figure 4.5, we do not need to consider the walk $v_5v_4v_3v_1v_2v_4$, for example, since its edge set is equal to that of the walk $v_5v_4v_2v_1v_3v_4$. The set $\mathcal{W}'(H)$ therefore consists of five walks. Without loss of generality, let these five walks be $W_1 = v_5v_4v_2v_3v_4$, $W_2 = v_5v_4v_2v_1v_3v_2$, $W_3 = v_3v_4v_2v_1v_3v_2$, $W_4 = v_5v_4v_2v_1v_3v_4$, and $W_5 = v_5v_4v_3v_2v_1v_3$, as shown in Figure 4.6.

Let the six edges of H be $e_1 = v_1v_2, e_2 = v_1v_3, e_3 = v_2v_3, e_4 = v_2v_4, e_5 = v_3v_4$, and $e_6 = v_4v_5$. For the graph and walks in Figure 4.6, the inequalities are as follows:

$$\begin{aligned}
w(e_i) &\geq 1 \quad \text{for } i \text{ where } 1 \leq i \leq 6 \\
\ell &\geq 1 \\
w(e_3) + w(e_4) + w(e_5) + w(e_6) - 1 &\leq \ell \\
w(e_1) + w(e_2) + w(e_3) + w(e_4) + w(e_6) - 1 &\leq \ell \\
w(e_1) + w(e_2) + w(e_3) + w(e_4) + w(e_5) - 2 &\leq \ell \\
w(e_1) + w(e_2) + w(e_4) + w(e_5) + w(e_6) - 1 &\leq \ell \\
w(e_1) + w(e_2) + w(e_3) + w(e_5) + w(e_6) - 1 &\leq \ell
\end{aligned}$$

This results in a polyhedron with two vertices $\underline{v}^1$ and $\underline{v}^2$, and nine rays $\underline{r}^1, \ldots, \underline{r}^9$, shown in the table below.

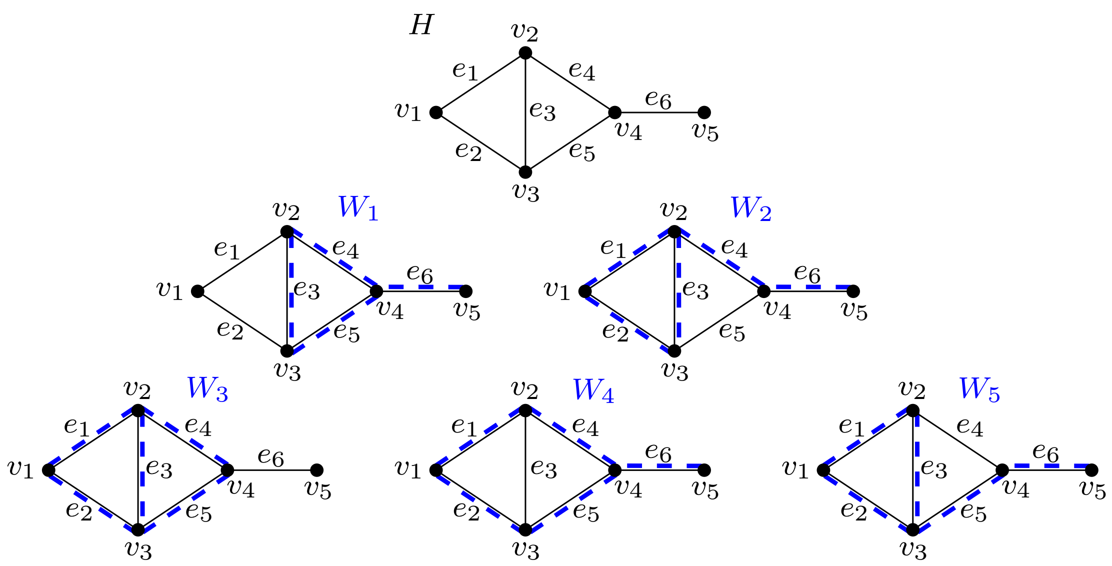

Figure 4.6: An example of a graph H and five walks in $\mathcal{W}'(H)$ (shown with blue dashed lines), where $W_1 = v_5v_4v_2v_3v_4$, $W_2 = v_5v_4v_2v_1v_3v_2$, $W_3 = v_3v_4v_2v_1v_3v_2$, $W_4 = v_5v_4v_2v_1v_3v_4$, and $W_5 = v_5v_4v_3v_2v_1v_3$.

	e_1	e_2	e_3	e_4	e_5	e_6	ℓ
$\underline{v}^1$	1	1	1	1	1	1	4
$\underline{v}^2$	1	1	2	2	2	1	6
$\underline{r}^1$	1	0	0	0	0	0	1
$\underline{r}^2$	0	1	0	0	0	0	1
$\underline{r}^3$	0	0	1	0	0	0	1
$\underline{r}^4$	0	0	0	1	0	1	1
$\underline{r}^5$	0	0	0	0	1	0	1
$\underline{r}^6$	0	0	0	0	0	1	1
$\underline{r}^7$	0	1	1	1	1	1	4
$\underline{r}^8$	1	0	1	1	1	1	4
$\underline{r}^9$	0	0	0	0	0	0	1

Vertices $\underline{v}^1$ and $\underline{v}^2$ correspond to the graphs $G_{\underline{v}^1}$ and $G_{\underline{v}^2}$ in Figure 4.7. In $G_{\underline{v}^1}$, the walks $W_2, W_4, W_5 \in \mathcal{W}'(H)$ correspond to the paths $P_2, P_4, P_5 \in \mathcal{P}(G_{\underline{v}^1})$ respectively, each of which has length $\ell_{\underline{v}^1}(= 4)$, as

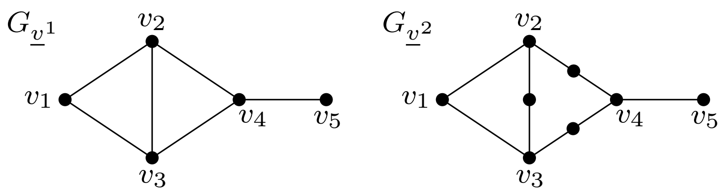

Figure 4.7: Graphs $G_{\underline{v}^1}$ and $G_{\underline{v}^2}$, where only the vertices that are also in H are labeled.

shown in Figure 4.8. Since W_2, W_4, and W_5 have a common vertex, then the corresponding paths P_2, P_4, and P_5 have a common vertex. Therefore $G_{\underline{v}^1}$ has a Gallai vertex. Similarly, in $G_{\underline{v}^2}$, the walks $W_1, W_2, W_3, W_4, W_5 \in \mathcal{W}'(H)$ correspond to paths in $\mathcal{P}(G_{\underline{v}^2})$ of length $\ell_{\underline{v}^2}(= 6)$, as shown in Figure 4.9, and since these five walks have a common vertex, then the corresponding paths have a common vertex. Therefore $G_{\underline{v}^1}$ has a Gallai vertex. Now, at every vertex $\underline{v}$ of the polyhedron, the graph $G_{\underline{v}}$ has a Gallai vertex, and hence every graph in $\mathcal{G}_H$ has a Gallai vertex.

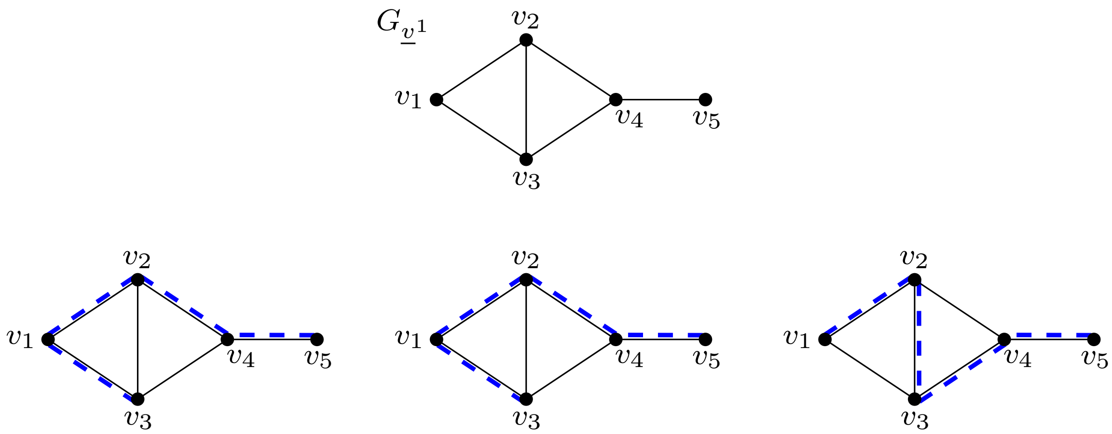

Figure 4.8: Graph $G_{\underline{v}^1}$ and the paths in $\mathcal{P}(G_{\underline{v}^1})$ of length $\ell_{\underline{v}^1}$.

4.3 Method of computation

4.3.1 Overview

We present a method of examining classes of graphs to determine whether every graph in the class has a Gallai vertex. We first generate a set of simple graphs using the `geng` program written by McKay [29], which is part of the `nauty` package. We then generate, for each graph H in this set, all of the

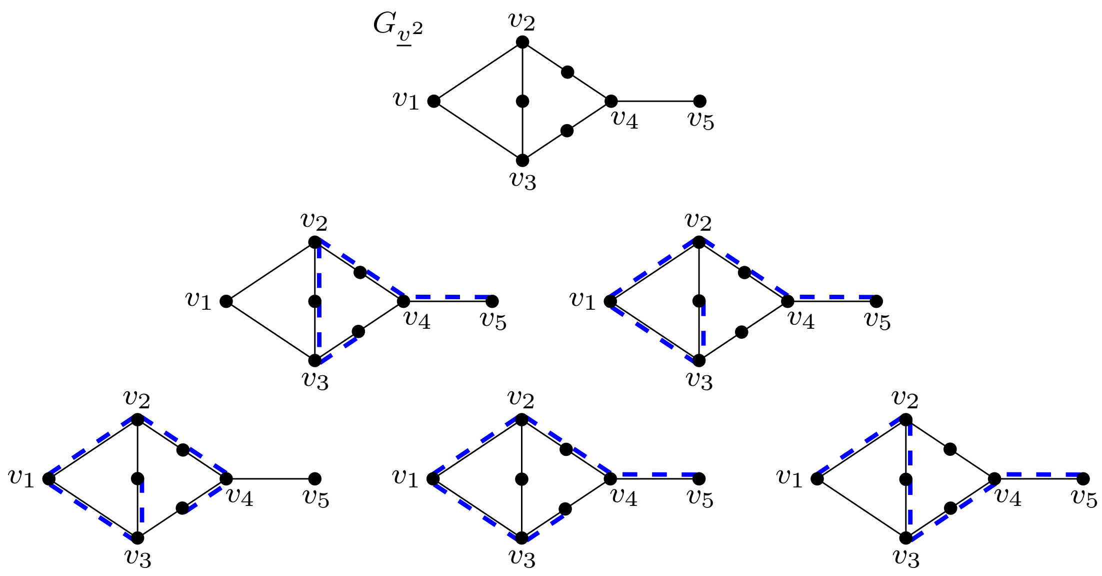

Figure 4.9: Graph $G_{\underline{v}^2}$ and the paths in $\mathcal{P}(G_{\underline{v}^2})$ of length $\ell_{\underline{v}^2}$, where only the vertices that are also in H are labeled.

walks in $\mathcal{W}(H)$ described in Section 4.2.2, and from these generate a set of inequalities. Next, we use the program `lrslib` written by Avis [1] to calculate, for each graph H, the vertices and rays of the polyhedron satisfying the set of inequalities for H. Our final program then processes each vertex of each polyhedron to determine which paths of the corresponding graph are of maximum length, and check whether these paths have a common vertex. Each of these steps is detailed more thoroughly in the following sections. We started by examining connected graphs with just four vertices, and then increasing the number of vertices, up to the limits of the computational time available.

4.3.2 Generating graphs

The first step is to generate a set of simple graphs. To do this, we use two programs that can be found in the `nauty` package written by McKay and Piperno [29]. We first use `geng` to generate graphs with a specified number of vertices (and other conditions as necessary), and then use `listg` to output the graphs to a file as a list of adjacencies. This output is then used by the next program.

4.3.3 Finding maximal walks

This step uses the `maximalwalks.py` Python 3 program in Appendix B. It takes as input a file from the previous program, containing the adjacency lists of a set of graphs. The program outputs two files for each graph. One file contains a set of inequalities representing the lengths of maximal paths of the graph, to be used by the next program, `lrslib`. The other file contains the graph and a set of its walks, output using the pickle library in Python 3, so that these walks can be recovered later in the same data structures and do not have to be recalculated or processed from a text file. A sample input file and sample output file for `lrslib` are shown in Appendix B.

Algorithm 1 below gives an overview of the `maximalwalks.py` program. We note a few things about this algorithm. In line 3, if a graph H has a degree two vertex whose neighbours are not themselves adjacent, then H is homeomorphic to the simple graph obtained by smoothing this degree two vertex, and so its set of homeomorphic graphs has already been considered when we examined graphs with fewer vertices. In line 7, finding the walks in $\mathcal{W}(H)$ uses a modification of a standard recursive function for finding the maximal paths of the graph H, with some adjustments to allow for repeated or overlapping ends, as described in Section 4.2.2. If all of the walks in $\mathcal{W}(H)$ have a common vertex, then we can stop since all of the maximal paths in every simple graph homeomorphic to H have a common vertex. In line 16, we obtain the set $\mathcal{W}'(H)$ from $\mathcal{W}(H)$ as discussed in Section 4.2.4

4.3.4 Calculating the polyhedron

The files of inequalities generated by the previous program are now processed using the programs `lrs` and `mplrs` (the multithreaded version of `lrs`) in the package `lrslib` written by Avis [1]. For each graph H, `lrs` calculates the polyhedron satisfying the system of inequalities, and outputs a file containing the vertices and rays of the polyhedron of H. Note that the vertex enumeration of a polyhedron is slow, and this step is the limiting factor in our computations. The complexity of `lrs` is discussed in [1].

4.3.5 Finding intersections of paths

The last step is to process the output from `lrs`, using the `postprocessing.py` Python 3 program in Appendix C. Let H be a simple reduced graph and consider the polyhedron of H. Note that the program

Algorithm 1 maximalwalks.py

Input: File of graphs, containing a list of adjacencies for each graph

Output: For each graph, a file of inequalities and a pickled file of the graph and its maximal walks

for each graph H in the input file **do**
    read list of adjacencies of H into an array
    if H has a degree two vertex whose two neighbours are not themselves adjacent **then**
        return
    end if
    for each vertex v in H **do**
        find all walks in $\mathcal{W}(H)$ starting at v
    end for
    if all walks in $\mathcal{W}(H)$ have a common vertex **then**
        return
    end if
    assign each edge of H a unique index from 0 to $|E(H)| - 1$
    for each walk W in $\mathcal{W}(H)$ **do**
        turn W into a binary vector representing which edges of H are in W
    end for
    remove walks from $\mathcal{W}(H)$ whose edge set is a subset of another walk in $\mathcal{W}(H)$
    call the resulting set of walks $\mathcal{W}'(H)$
    if all walks in $\mathcal{W}'(H)$ have a common vertex **then**
        return
    end if
    for each walk W in $\mathcal{W}'(H)$ **do**
        calculate $\alpha(W)$ as in Section 4.2.3
        turn the set of edges of W and the value $\alpha(W)$ into an inequality, as in Section 4.2.3
        write this inequality to the inequalities output file for H
    end for
    write H and the walks in $\mathcal{W}'(H)$ to a pickled file for H
end for

`lrs` finds the vertices of the polyhedron of H exactly, even if they have non-integer rational coordinates. We analyse the vertices using floating-point arithmetic with a generous allowance for rounding error.

For each vertex $\underline{v}$ of the polyhedron of H, we calculate the set $\mathcal{S} \subset \mathcal{W}'(H)$ of walks of H that are extremal at $\underline{v}$. We then check whether the walks in $\mathcal{S}$ have a common vertex and, if they do not, we then investigate whether every three walks in $\mathcal{S}$ have a common vertex.

The program `postprocessing` takes a file containing the output from `lrslib` for a graph H and another file containing its pickled walks $\mathcal{W}'(H)$. The output is a file containing the graph H and vertices of the polyhedron of H for which the set of extremal walks in $\mathcal{W}'(H)$ do not have a common vertex, or for which a set of three extremal walks do not have a common vertex, and a list of these walks for each such vertex of the polyhedron. A sample input file from `lrslib` and a sample output file are shown in Appendix C.

Algorithm 2 gives an overview of the post-processing program. This program contains a function (not included in the algorithm) that uses the multiprocessing Python 3 library to parallelise the processing of the vertices of the polyhedron in order to speed up the program, as there may be tens or hundreds of thousands of vertices of the polyhedron for one graph H.

This program is run on each `lrslib` output file, that is, on each original reduced graph being tested. The output files are then manually read (or searched using `grep`) to discover which graphs do not have a Gallai vertex, and whether every three longest paths have a common vertex for these graphs.

4.4 Results

Let H be a simple reduced graph and let $\mathcal{G}_H$ be the set of simple graphs homeomorphic to H. Observe that if H has exactly two or exactly three vertices, then every graph in $\mathcal{G}_H$ is a path or a cycle, respectively, and hence every graph in $\mathcal{G}_H$ has a Gallai vertex. Therefore, we start by examining simple reduced graphs H with $n = 4$ vertices, and examined all graphs H with at most $n = 7$ vertices, which is the limit of the computational time available to us. We obtained the following result.

Theorem 4.5. *Every simple connected graph that is homeomorphic to a simple connected graph with at most 7 vertices has a Gallai vertex.*

Algorithm 2 postprocessing.py

Input: Pickled file containing a graph H and a subset $\mathcal{W}'(H)$ of $\mathcal{W}(H)$ (as output by the program maximalwalks.py), and a file containing the vertices of the polyhedron of H.

Output: File containing notes of which longest paths, in which graphs, do not share a vertex.

unpickle H and $\mathcal{W}'(H)$
for each vertex $\underline{v}$ of the polyhedron **do** ▷ (multiple vertices may be processed in parallel)
  read $\underline{v}$ from the file
  for each walk W in $\mathcal{W}'(H)$ **do**
    calculate the length of the corresponding path in $\mathcal{P}(G_{\underline{v}})$
  end for
  determine the set $\mathcal{S}$ of walks in $\mathcal{W}'(H)$ that are extremal at $\underline{v}$ ▷ (check that the length of the walk is at least $0.999 \cdot \ell_{\underline{v}}$ to allow for floating-point rounding errors)
  calculate whether the walks in $\mathcal{S}$ all have a common vertex
  if the walks in $\mathcal{S}$ have a common vertex **then**
    return
  end if
  write to output file that these walks do not have a common vertex
  for each set of three walks in $\mathcal{S}$ **do**
    check whether the three paths have a common vertex
    if the three paths do not have a common vertex **then**
      write to output file that these paths do not have a common vertex
    end if
  end for
end for

This is equivalent to the following theorem.

Theorem 4.6. *Let G be a simple connected graph. Let a be the number of vertices $v \in V(G)$ with $\deg_G(v) \neq 2$. Let p be the number of chains of G that are parallel to at least one other chain, and let s be the number of maximal sets of parallel chains of G. If $a + p - s \leq 7$ then G has a Gallai vertex.*

Our test of all reduced graphs with cyclomatic number 7 (n vertices and $n + 6$ edges) and minimum degree 3, yields the result below.

Theorem 4.7. *Every simple connected graph homeomorphic to a simple connected graph with cyclomatic number 7 and minimum degree 3 has a Gallai vertex.*

This is equivalent to the following theorem.

Theorem 4.8. *Every simple connected graph with cyclomatic number 7, no leaves, and no parallel chains has a Gallai vertex.*

This also means that every three longest paths in these graphs have a common vertex, and so this result extends Theorem 3.1 in Chapter 3 to cyclomatic number 7 in the case where these restrictions hold. Again, this result is the limit of the computational time available.

4.5 Linear programming

In this section, we describe an alternative to the method in Sections 4.2 and 4.3, utilising linear programming solvers, which are faster than programs that enumerate the vertices of a polyhedron. Instead of determining whether all of the graphs in a certain class have a Gallai vertex, this method determines whether there is a graph in the class that has k ($k \geq 3$) longest paths that do not have a common vertex. As mentioned in Section 1.3, it is not known whether every k longest paths in a connected graph have a common vertex for $3 \leq k \leq 6$, though it has been conjectured that every three longest paths have a common vertex (Conjecture 1).

4.5.1 Linear program

Let H be a simple reduced graph and let $\mathcal{W}'(H)$ be a set of walks of H as defined in Section 4.2. For each set $S \subseteq \mathcal{W}'(H)$ with $|S| = k$ ($k \geq 3$) such that the walks in S do not have a common vertex, we have the linear program below. Note that exact arithmetic performs computations using rational numbers instead of floating-point numbers, and so has no rounding error.

Definition 2 (Linear program for exact arithmetic)**.**

Variables: $w(e)$ *for each edge* $e \in E(H)$*, and* ℓ*.*

Constraint 1: $w(e) \geq 1$ *for each edge* e *of* H*,*

Constraint 2: $\ell \geq 1$*,*

Constraint 3: $\left(\sum_{e \in W} w(e)\right) - \alpha(W) \leq \ell$ *for each walk* $W \in \mathcal{W}'(H) \backslash S$*,*

Constraint 4: $\left(\sum_{e \in W} w(e)\right) - \alpha(W) = \ell$ *for each walk* $W \in S$*.*

We then use a linear program solver with exact arithmetic to determine whether there is a feasible point for this set of (in)equalities, that is, whether there exists a simple graph G homeomorphic to H such that the k walks in S correspond to k longest paths of G, and these k longest paths do not have a common vertex.

However, linear program solvers using exact arithmetic are much slower than those using floating-point arithmetic. We present a faster method that allows the use of linear program solvers with floating-point arithmetic, only using the exact arithmetic solver when necessary. We loosen the constraints on each walk $W \in \mathcal{S}$ to $\ell - \epsilon \leq \left(\sum_{e \in W} w(e)\right) - \alpha(W) \leq \ell + \epsilon$ for some small ϵ. Now, for each set $S \subseteq \mathcal{W}'(H)$ with $|S| = k$ ($k \geq 3$) such that the walks in S do not have a common vertex, we have the linear program below, where ϵ is a small constant.

Definition 3 (Linear program for floating-point arithmetic)**.**

Variables: $w(e)$ *for each edge* $e \in E(H)$*, and* ℓ*.*

Constraint 1: $w(e) \geq 1$ *for each edge* e *of* H*,*

Constraint 2: $\ell \geq 1$*,*

Constraint 3: $\left(\sum_{e \in W} w(e)\right) - \alpha(W) \leq \ell$ *for each walk* $W \in \mathcal{W}'(H) \backslash S$*,*

Constraint 4: $\ell - \epsilon \leq \left(\sum_{e \in W} w(e)\right) - \alpha(W) \leq \ell + \epsilon$ *for each walk* $W \in S$*.*

We can then use a linear program solver with floating-point arithmetic to search for a feasible region, which is much faster, and then use an exact arithmetic linear program solver only if a feasible region is found. For the graphs we tested, no feasible region was found and we therefore did not need to use an exact arithmetic linear program solver.

4.5.2 Properties of a reduced graph

In this section, we consider graphs that are not necessarily simple. Let G be a simple graph and let H be the smallest (with respect to edges) graph homeomorphic to G. The graph H can be obtained from G by repeatedly smoothing degree two vertices of G until the resulting graph has no degree two vertices. Similarly to a simple reduced graph, H is unique up to isomorphism. Let B be a block of H. If B consists of a loop or a set of parallel edges, then we consider B to be a non-trivial block of H. Let G be a simple graph with k longest paths ($k \geq 3$) that do not have a common vertex, and let H be the smallest (with respect to edges) graph homeomorphic to G. We present several properties of G and thereby obtain results about properties of H, limiting the number of cases to be examined computationally for the existence of k longest paths that do not have a common vertex.

We focus on Conjecture 1, that is, the case $k = 3$. A family $\mathcal{F}$ of graphs is *monotone* if it is closed under the deletion of edges, that is, for every $G \in \mathcal{F}$ and every $e \in E(G)$, the graph G' obtained from G by deleting e is in $\mathcal{F}$. Axenovich [2] proves the following.

Lemma 4.9. *For a monotone family $\mathcal{F}$ of graphs, let $G \in \mathcal{F}$ be a simple connected graph with smallest $|V(G)| + |E(G)|$ having three longest paths with no common vertex. Then G has exactly one nontrivial block.*

The proof of this lemma shows that the result also holds when $G \in \mathcal{F}$ is minimal with respect to edges, instead of having the smallest $|V(G)| + |E(G)|$. This is stated in the following lemma.

Lemma 4.10. *For a monotone family $\mathcal{F}$ of graphs, let $G \in \mathcal{F}$ be a simple connected graph having three longest paths with no common vertex, and let G be minimal with respect to edges. Then G has exactly one nontrivial block.*

Note that if a graph G has exactly one non-trivial block, then the smallest (with respect to edges) graph H homeomorphic to G also has exactly one non-trivial block.

The following result is due to Klavžar and Petkovšek [26].

Lemma 4.11. *Every graph that is a cactus has a Gallai vertex.*

Lemma 4.12. *Let G be a simple connected graph with three longest paths that have no common vertex, and let G be minimal with respect to edges. Let H be the smallest (with respect to edges) graph homeomorphic to G. Then H has no loops.*

Proof. Suppose that H has a loop e. Let H' be the graph obtained from H by deleting e. If H' is a tree, then G is a cactus, and hence by Lemma 4.11, G has a Gallai vertex, a contradiction. If H' is not a tree, then G has more than one nontrivial block, which contradicts Lemma 4.10. □

Let G be a simple minimal (with respect to edges) counterexample to Conjecture 1, and let B be the only non-trivial block of G. Following Kensell's definition [25], a *branching point* of G is a vertex $x \in V(G)$ such that $\deg_B(x) \geq 3$. Kensell states the following lemma and provides a sketch of the proof:

Lemma 4.13. *Every cycle of G has at least 3 branching points [under the conditions above].*

Since a cycle of G with exactly two branching points consists of two parallel chains, which corresponds to a pair of parallel edges of H, Lemma 4.13 implies the following result:

Corollary 4.14. *Let G be a simple connected graph with three longest paths that have no common vertex, and let G be minimal with respect to edges. Let H be the smallest (with respect to edges) graph homeomorphic to G. Then H has no parallel edges.*

Proof. Suppose that H has a pair of parallel edges e_1 and e_2 with endpoints u and v. Let Q_1 and Q_2 be the corresponding paths of G. Note that the interior vertices of Q_1 and Q_2 all have degree two. Then $Q_1 \cup Q_2$ is a cycle of G with exactly two branching points, u and v, a contradiction by Lemma 4.13. □

Assuming that Lemma 4.13 holds, the following result follows directly from Lemma 4.12 and Corollary 4.14.

Lemma 4.15. *Let G be a simple connected graph with three longest paths that have no common vertex, and let G be minimal with respect to edges. Let H be the smallest (with respect to edges) graph homeomorphic to G. Then H is simple.*

Let G be a simple connected graph with three longest paths that have no common vertex, and let G be minimal with respect to edges. By Lemma 4.10, G has exactly one non-trivial block, B. We use our terminology from Chapter 2, and call B the core of G. Recall that the core-touching subgraphs of core B of G are the components of the graph obtained from G by deleting the edges of B. Since B is the only non-trivial block of G, the core-touching subgraphs of G are trees. Let H be the smallest (with respect to edges) graph homeomorphic to G. Since the number of non-trivial blocks of a subdivision of H is exactly the number of non-trivial blocks of H, it follows by Lemma 4.10 that H has exactly one non-trivial block. Similarly to G, the non-trivial block of H is the core of H, and the core-touching subgraphs of H are trees.

Lemma 4.16. *Let G be a simple connected graph with three longest paths that do not have a common vertex, and let G be minimal with respect to edges. Let B be the unique core of G. Then each core-touching subgraph of G is a path.*

The proof of this lemma follows similar reasoning to the proof of Lemma 2.9 in Chapter 2.

Proof. By Lemma 4.10, B exists. For each longest path P of G, (a) P is a subgraph of a core-touching subgraph of G or (b) P has at least one edge that is an edge of the core B of G. Let $\mathcal{P} = \mathcal{P}_a \cup \mathcal{P}_b$ be the set of longest paths of G, where $\mathcal{P}_a$ is the set of paths for which (a) holds and $\mathcal{P}_b$ is the set of paths for which (b) holds.

We first show that $\mathcal{P}_a = \emptyset$. Suppose that $\mathcal{P}_a \neq \emptyset$. By Lemma 1.1, every pair of paths in $\mathcal{P}_a$ have a common vertex, and hence there is a core-touching subgraph $C(x)$ of G, for some $x \in V(B)$, such that each path in $\mathcal{P}_a$ is a subpath of $C(x)$. If $\mathcal{P} = \mathcal{P}_a$, then by Proposition 2.2, G has a Gallai vertex, a contradiction. Hence $\mathcal{P}_b \neq \emptyset$. By Lemma 1.1, each path in $\mathcal{P}_b$ has a common vertex with each path in $\mathcal{P}_a$, and hence $x \in V(P)$ for each path $P \in \mathcal{P}_b$ since there is at least one edge of P that is an edge of B. If $x \in V(Q)$ for each path $Q \in \mathcal{P}_a$, then x is a Gallai vertex of G, a contradiction. If this is not the case, then there exists at least one path $P_1 \in \mathcal{P}_a$ that is a path of $C(x) - x$. Since each path in $\mathcal{P}_b$ has a common vertex with P_1, each path in $\mathcal{P}_b$ has a subpath of length at least one that is a path of $C(x)$. Let T be the set of these subpaths, and let $S = T \cup \mathcal{P}_a$. Then S is a set of paths of $C(x)$ and every two paths in S have a common vertex. Hence, by Proposition 2.2, the paths in S have a common vertex, and therefore G has a Gallai vertex, a contradiction. Hence $\mathcal{P} = \mathcal{P}_b$.

We next show that there is no core-touching subgraph of G that is a tree but not a path. Let $C(x)$

be a core-touching subgraph of G for some $x \in V(B)$. Suppose that $C(x)$ is a tree but not a path. Note that $|E(C(x))| \geq 2$. Let $Q_1, \ldots, Q_k$ $(k \geq 2)$ be the set of maximal paths of $C(x)$ with endpoint x and assume without loss of generality that $|Q_1| \geq |Q_2| \geq \cdots \geq |Q_k|$. If $E(P) \cap E(C(x)) = \emptyset$ for every $P \in \mathcal{P}$, then G is not minimal with respect to edges, a contradiction. Suppose there is at least one path $P \in \mathcal{P}$ with $E(P) \cap E(C(x)) \neq \emptyset$. For every such path P, $x \in V(P)$ and, since P is a longest path, there is a leaf of $C(x)$ that is an endpoint of P. Therefore each such path P can be assumed to have subpath Q_1. Let G' be the graph obtained from G by deleting the edges of $Q_2, \ldots, Q_k$ and any resulting isolated vertices, and note that $|E(G')| < |E(G)|$. We show that there are three longest paths of G' with no common vertex. Let $P_1, P_2, P_3 \in \mathcal{P}$ have no common vertex. Then x is a vertex of at most two of P_1, P_2, and P_3. If $x \in V(P_1)$, then let P_1' be the path obtained from P_1 by replacing the subpath of P_1 that is a subpath of $C(x)$ with the path Q_1 of $C(x)$. If $x \notin V(P_1)$, let $P_1' = P_1$. Similarly define P_2' and P_3'. Then P_1', P_2', and P_3' are three longest paths of G' with no common vertex and hence G is not minimal, a contradiction. □

Since a core-touching subgraph of G that is a path with length at least one corresponds to an edge of H that is not a loop, we have the following corollary.

Corollary 4.17. *Let G be a simple connected graph with three longest paths that have no common vertex, and let G be minimal with respect to edges. Let H be the smallest (with respect to edges) graph homeomorphic to G and let B be the unique core of H. Then each core-touching subgraph of H has at most one edge and is not a loop.*

Proof. By Lemma 4.10, G has exactly one nontrivial block. Hence H has exactly one non-trivial block, and therefore the unique core B of H always exists. By Lemma 4.12, H has no loops. A core-touching subgraph of G consisting of exactly one vertex corresponds to a vertex of H. A core-touching subgraph of G that is a path with length at least one corresponds to an edge of H that is not a loop. It follows that every core-touching subgraph of H has at most one edge and is not a loop. □

4.5.3 Results

Thesis supervisor Brendan McKay wrote a program that uses the method in Section 4.5.1 and GLPK [27] and ran this program on several classes of graphs, producing the following two lemmas.

Lemma 4.18. *Let H be a simple graph with no vertices of degree two and exactly one non-trivial block B, which is the core of H, where each core-touching subgraph of H has at most one edge. If*

(i) $|E(B)| - |V(B)| \leq 6$, *or*

(ii) $|V(B)| \leq 9$, *or*

(iii) $|V(B)| \leq 16$ *and B has maximum degree at most three*

then every three walks in $\mathcal{W}(H)$ have a common vertex.

Lemma 4.19. *Let H be a simple graph with no vertices of degree two that has a connected subgraph A with minimum degree two where each component of the graph H' obtained from H by deleting the edges of A has at most one edge. If*

(i) $|V(A)| \leq 11$ *and every vertex in $V(A)$ has maximum degree at most three, or*

(ii) $|V(A)| \leq 6$ *and every vertex in $V(A)$ has maximum degree at most four,*

then every six walks in $\mathcal{W}(H)$ have a common vertex.

From these lemmas, we deduce results about graphs homeomorphic to such graphs H in the following two sections.

4.5.3.1 Three longest paths

From Lemma 4.18(i), we obtain the following result, which verifies Theorem 3.1 in Chapter 3.

Lemma 4.20. *For every graph with cyclomatic number at most 6, every three longest paths have a common vertex.*

Proof. Assume that there is a graph with cyclomatic number at most 6 that has three longest paths with no common vertex. Then there is such a graph G that is minimal with respect to edges. The set of graphs with cyclomatic number at most 6 is monotone. To see this, observe that deleting an edge of a graph decreases its number of edges by one and decreases its number of vertices by at most one, and hence the cyclomatic number of the graph does not increase. Therefore, by Lemma 4.10, G has exactly one non-trivial block. Let H be the smallest (with respect to edges) graph homeomorphic to G. Then H has exactly one non-trivial block and has no vertices of degree two. Since subdividing an edge of a graph increases the number of edges by one and the number of vertices by one, this operation does not change the cyclomatic number of a graph. Hence the cyclomatic number of H is exactly the cyclomatic number of G, and therefore is at most 6. Additionally, by Lemmas 4.12 and 4.13, H is simple. Since G has three longest paths with no common vertex, there are three walks in $\mathcal{W}(H)$ that do not have a common vertex. However, by Lemma 4.18, for such a graph H, every three walks in $\mathcal{W}(H)$ have a common vertex, a contradiction. $\square$

Let G be a simple graph with no parallel chains and exactly one non-trivial block B. Let $\mathcal{G}_B$ be the class of all such graphs G. From Lemma 4.18(ii) and (iii) respectively, we have the following two results.

Lemma 4.21. *Let $G \in \mathcal{G}_B$ such that there are at most 9 vertices $v \in V(B)$ with $\deg_B(v) \geq 3$. Then every three longest paths of G have a common vertex.*

Lemma 4.22. *Let $G \in \mathcal{G}_B$ such that there are at most 16 vertices $v \in V(B)$ with $\deg_B(v) = 3$ and no vertices $x \in V(B)$ with $\deg_B(x) > 3$. Then every three longest paths of G have a common vertex.*

Note that if G is a simple graph with exactly one non-trivial block B and this block is a cycle, then the reduced graph H may not be simple, but G is a cactus graph and hence by Lemma 4.11, G has a Gallai vertex.

4.5.3.2 Six longest paths

Let G be a simple graph with no parallel chains, no cycle with exactly one cut vertex of G, and for which there exists a connected subgraph A of G such that every component of the graph G' obtained from G by deleting the edges of A is a path. Let $\mathcal{G}_A$ be the set of all such graphs G. From Lemma 4.19, we have the following two results.

Lemma 4.23. *Let $G \in \mathcal{G}_A$ such that there are at most 11 vertices $v \in V(A)$ with $\deg_A(v) = 3$ and no vertices $x \in V(A)$ with $\deg_A(x) > 3$. Then every six longest paths of G have a common vertex.*

Lemma 4.24. *Let $G \in \mathcal{G}_A$ such that here are at most 6 vertices $v \in V(A)$ with $3 \leq \deg_A(v) \leq 4$ and no vertices $x \in V(A)$ with $\deg_A(x) > 4$. Then every six longest paths of G have a common vertex.*

The method described in this section requires running a linear program solver on every set of k walks in $W'(H)$ ($k \geq 3$) that do not have a common vertex, for a reduced graph H. This is in contrast to the method described in Section 4.3 that required running `lrs` once for a simple reduced graph H. However, there are linear program solvers, particularly those using floating-point arithmetic (rather than exact arithmetic) that are far faster than any available vertex enumeration programs like `lrs`, and so the former approach is much faster. For $k = 3$, we may also be able to use the results in Section 4.5.2 to restrict the reduced graphs to be tested for a particular set of conditions, further speeding up the process. However, for $k > 3$ many of the results n Section 4.5.2 do not apply and hence there are more reduced graphs to be tested for a particular set of conditions. Additionally, the linear programming method does not tell us whether the graphs in $\mathcal{G}_H$ have a Gallai vertex, but only whether every k longest paths have a common vertex for a specified $k \geq 3$.

4.6 Concluding remarks

In this chapter, we used computational methods to check graphs for the presence of a Gallai vertex or for the presence of three or six longest paths with no common vertex.

We used vertex enumeration of a polyhedron to determine that every graph that is homeomorphic to a simple connected graph with at most 7 vertices has a Gallai vertex, and that every simple connected graph with cyclomatic number 7 and minimum degree 3 has a Gallai vertex. The approach used here allows us to determine results about an infinite set of homeomorphic graphs by checking a finite subset of these graphs for the presence of a Gallai vertex. We start with the smallest simple graph in the set of homeomorphic graphs, produce a set of inequalities representing the graphs in this set, process this set of inequalities, and obtain a finite set of graphs to check for the presence of a Gallai vertex. However, this method relies on the vertex enumeration of a polyhedron, for which there are no efficient algorithms or programs. Avis [1] notes that certain graphs may run faster with the program `cdd` written by Fukuda [13], which is therefore a promising method for future investigations.

In Section 4.5.1, we give an alternative method that uses linear program solvers instead of a program that enumerates the vertices of a polyhedron. Using this method, thesis supervisor McKay tested simple reduced graphs with certain conditions for the presence of three or six longest paths that do not have a common vertex. We used the results in Section 4.5.2 and these computations to determine that every three longest paths have a common vertex in every graph with cyclomatic number at most 6, among other results stated in Section 4.5.3. This result on graphs with cyclomatic number at most 6 verifies Theorem 3.1 in Chapter 3. The main advantage of this linear programming approach is that there are much faster linear program solvers available than programs that enumerate the vertices of a polyhedron. Even though, for a particular simple reduced graph, we have a separate linear program for every set of k longest paths ($k \geq 3$), running a solver on each of these linear programs is much faster than the polyhedron method for the graphs tested in this chapter. Note that as k increases, if the number of maximal walks in the graph is sufficiently large, then the number of combinations of k maximal walks increases greatly and so the running time increases accordingly. It is therefore possible that there may be graphs and values of k for which the polyhedron method is faster. One downside to the linear programming method is that it does not allow us to specify that any subset of the maximal walks of the reduced graph may be longest paths of an unreduced graph (as we do with the polyhedron method), without testing each such subset individually. However, as the main questions of interest are whether every three longest paths in a graph have a common vertex, and whether there is a graph with a set of k longest paths, $3 \leq k \leq 6$, that do not have a common vertex, the linear programming method is likely to be much faster for examining these questions than the polyhedron method. Additionally, if more results can be proved about the structure of the reduced graph H for a minimal (with respect to edges) graph G with k longest paths that do not have a common vertex, this will decrease the number of reduced graphs to be tested and speed up the process further.

Chapter 5

Conclusion

In this thesis we investigate the conjecture that every three longest paths of a graph have a common vertex (Conjecture 1). We also investigate the question of whether all longest paths of a graph have a common vertex (Gallai's question).

5.1 Results

In this thesis, we add to the graph classes for which Conjecture 1 holds and to the graph classes that have a Gallai vertex. Figures 5.1 and 5.2 show the diagrams of graphs classes from Chapter 1 for three longest paths and all longest paths respectively with the addition of the results in this thesis, identified by a red outline. We also present two small graph classes for which every six longest paths have a common vertex. Recall that a *chain* of a graph G is a path of G whose interior vertices have degree two, and a set of at least two chains are *parallel* if they have a pair of common endpoints.

Conjecture 1 (three longest paths): Our main result for Conjecture 1 is that every three longest paths have a common vertex in:

- Graphs with cyclomatic number at most 6 (Theorem 3.1).

The entirety of Chapter 3 and part of Chapter 4 are dedicated to proving this result. Chapter 3 also shows that a minimal (with respect to edges) counterexample to Conjecture 1 has cyclomatic number at least 7, adding to the known properties of such a counterexample. In Chapter 4, we prove two smaller results. Recall that a graph G is in the graph class $\mathcal{G}_B$ if G has no parallel chains and there exists exactly one non-trivial block B of G. We prove that every three longest paths have a common vertex in the following graph classes:

- Graphs in $\mathcal{G}_B$ for which there are at most 9 vertices $v \in V(B)$ with $deg_B(v) \geq 3$ (Lemma 4.21).
- Graphs in $\mathcal{G}_B$ for which there are at most 16 vertices $v \in V(B)$ with $deg_B(v) = 3$ and there does not exist a vertex $x \in V(B)$ with $deg_B(x) > 3$ (Lemma 4.22).

Figure 5.1 shows the diagram of graph classes from Figure 1.5 in Section 1.3.2 with the addition of the results above, highlighted with a red outline. Note that Conjecture 1 also holds for the graph classes listed below.

Gallai's question (all longest paths): Our main result for Gallai's question is that every graph in the following graph class has a Gallai vertex:

- Θ_H, theta-Hamiltonian-tree graphs (Theorem 2.1).

Chapter 2 is dedicated to proving this result. In Chapter 4, we prove that every graph in the following two smaller graph classes has a Gallai vertex:

- Graphs with cyclomatic number 7, no leaves, and no parallel chains (Theorem 4.8).
- Graphs for which the number of vertices of degree 3 plus the number of chains that are parallel to at least one other chain minus the number of maximal sets of parallel chains is at most seven (Theorem 4.6).

Figure 5.2 shows the diagram of graph classes from Figure 1.7 in Section 1.3.3 with the addition of the results above, highlighted with a red outline.

Six longest paths: Every six longest paths have a common vertex for the graph classes that have a Gallai vertex listed above. In Chapter 4, we present results on two smaller graph classes. Recall from Chapter 4 that a graph G is in the graph class $\mathcal{G}_A$ if G has no parallel chains, no cycle with exactly one cut vertex of G, and there exists a connected subgraph A of G such that every component of the graph G' obtained from G by deleting the edges of A is a path. We prove that every six longest paths have a common vertex for every graph in the following graph classes:

- Graphs in $\mathcal{G}_A$ for which there are at most 11 vertices $v \in V(A)$ with $deg_A(v) = 3$ and there does not exist a vertex $x \in V(A)$ with $deg_A(x) > 3$ (Lemma 4.23).

- Graphs in $\mathcal{G}_A$ for which there are at most 6 vertices $v \in V(A)$ with $3 \leq deg_A(v) \leq 4$ and there does not exist a vertex $x \in V(A)$ with $deg_A(x) > 4$ (Lemma 4.24).

5.2 Methods and extensions

In this section, we discuss the methods used in Chapters 2, 3, and 4 and the limitations and possible extensions of these methods.

In Chapter 2, we use case analysis to prove that every theta-Hamiltonian-tree graph has a Gallai vertex. We investigate extending our result to theta-Hamiltonian-tree graphs with a unique core and a single edge inserted into the core with restrictions on the placement of this edge, as discussed in Section 2.5. However, our methods are not easily extended to these graphs, if indeed such graphs have a Gallai vertex. There are many additional cases and considerably more work is required in each case to obtain a set of inequalities on the maximal paths of the graph similar to those in Section Lemma 2.10.

In Chapter 3, we use case analysis to prove that Conjecture 1 holds for graphs with cyclomatic number at most 6. We investigated extending this result to graphs with cyclomatic number 7. This requires proving that there is a seventh independent bi-coloured cycle or a second independent tri-coloured cycle in a minimal (with respect to edges) counterexample to Conjecture 1. However, we were unable to prove that there is another bi-coloured cycle or another tri-coloured cycle in this graph. Additionally, if such a cycle does exist, the methods we use in this chapter to show that these cycles are independent are not easily extended to further cycles, as discussed in Section 3.6. More sophisticated techniques are required to extend our results to a higher cyclomatic number. Note that our result cannot be extended to all longest paths since there are graphs with cyclomatic number at most 6 that do not have a Gallai vertex, as discussed in Section 3.6.

In Chapter 4, we take a different approach and use computational methods to investigate Gallai's question and Conjecture 1 for particular graph classes. Similarly to Chapter 2, we use inequalities on the lengths of the maximal paths of a graph. For a simple reduced graph H, we generate a set of inequalities on the maximal walks of H which correspond to the maximal paths of a graph homeomorphic to H. We then use two different methods for solving this system of inequalities. In the first method, we use

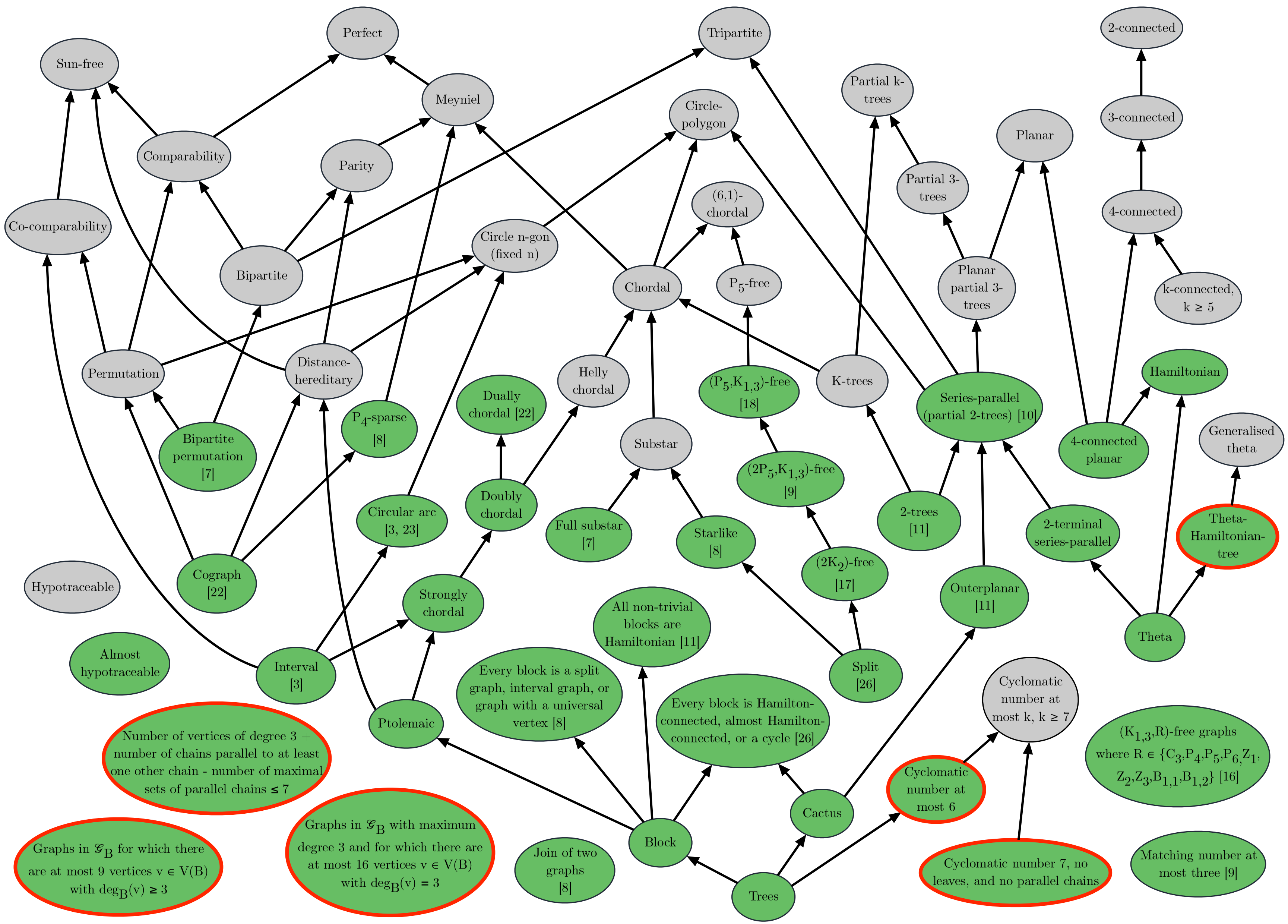

Figure 5.1: A diagram showing relationships between a number of graph classes for which Conjecture 1 holds (green) and classes for which it has not yet been determined whether Conjecture 1 holds (grey). A red outline indicates results proved in this thesis. Arrows indicate subclass to superclass relationship. This diagram does not include all such relationships of the classes shown.

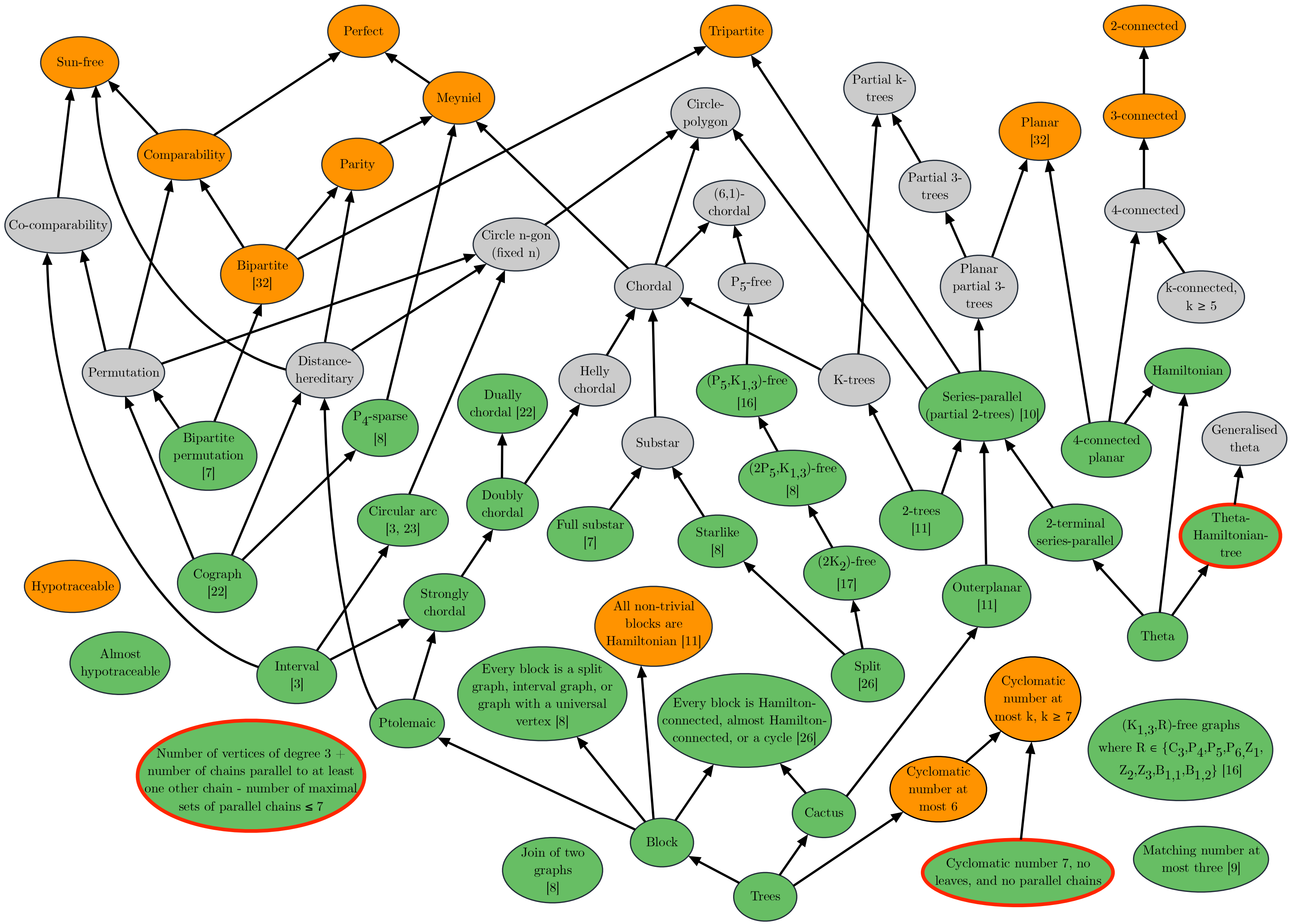

Figure 5.2: A diagram showing relationships between a number of graph classes for which every graph has a Gallai vertex (green), classes for which it has not yet been determined whether every graph has a Gallai vertex (grey), and classes for which there exists a graph with no Gallai vertex (orange). A red outline indicates results proved in this thesis. Arrows indicate subclass to superclass relationship. This diagram does not include all such relationships of the classes shown.

a program to enumerate the vertices of the polyhedron satisfying this system of inequalities. We then process each vertex of the polyhedon to determine whether every simple graph homeomorphic to H has a Gallai vertex. Using this method, we obtained results for two graph classes, as in the list in Section 5.1. Our second method uses linear programming and separately considers each set of k (for some $k \geq 3$) inequalities on the maximal walks of H. Our approach is to suppose that these k inequalities hold with equality and use a linear program solver to determine whether there is a feasible region for the given constraints. Using this second method, we independently confirmed our result from Chapter 3 and obtained several other results, as listed in Section 5.1. For the polyhedron method, the limiting factor is the program that enumerates the vertices of the polyhedron, for which there is no efficient algorithm. However, the linear programming method does not have the same limitations as there are considerably more efficient algorithms for solving linear programs. Therefore, the linear programming method has the potential to be extended, for example to graphs with cyclomatic number 7.

While we were not able to extend our case analysis techniques in Chapters 2 and 3 to the graph classes discussed above, these methods may be able to be applied to other graph classes. For instance, graphs similar to theta-Hamiltonian-tree graphs, but with a series parallel core, may be amenable to these methods. In Chapter 4, while the polyhedron method is slow and therefore difficult to extend, the linear programming method is considerably faster and has the potential for extension. It would be of interest to investigate whether every four, five, or six longest paths have a common vertex for graphs with cyclomatic number at most 6, and the linear programming method could be applied to this problem. We also believe that using the linear programming method could yield results in the investigation of whether every three longest paths have a common vertex in graphs with cyclomatic number at most 7.

Appendices

Appendix A

Graph classes for which every graph has a Gallai vertex

Every graph in the following graph classes has a Gallai vertex. The graph classes are listed here in chronological order by the year the paper was published. Note that some of the results are superseded by later results. Additionally, this list may not be complete.

It is well-known that trees have a Gallai vertex [see 10, 32]. All Hamiltonian graphs have a Gallai vertex since every longest path of the graph is a Hamiltonian path.

- **4-connected planar graphs**, since all 4-connected planar graphs are Hamiltonian, as shown by Tutte [36].

- **Split graphs and graphs in which every block is Hamilton-connected, almost Hamilton-connected, or a cycle**, proved in 1990 by Klavžar and Petrovšek [26]. The result on split graphs was later superseded by the result on $2K_2$–free graphs below. A graph is called *Hamilton-connected* if for every pair of vertices x, y there is a Hamiltonian path with endpoints x and y. A bipartite graph with vertex partition V_1 and V_2 is called *almost Hamilton-connected* if for every pair of vertices x, y there is a path with endpoints x and y that contains all of the vertices of either V_1 or V_2. These results imply that every cactus graph and every block graph have a Gallai vertex.

- **Circular arc graphs**, proved in 2004 by Balister et al. [3], with a gap in the proof closed by Joos [23] in 2015.

- **Outerplanar graphs**, proved in 2013 by de Rezende et al. [10], extending a result by Axenovich [2] that every three longest paths in an outerplanar graph have a common vertex (later superseded by the result on series-parallel graphs below). A planar drawing of a graph partitions the plane into regions called *faces*, and every planar drawing of a graph has exactly one unbounded face called the *outer face*. An *outerplanar graph* is a planar graph in which all vertices belong to the outer face. It is also shown in this paper that, for graphs in which all non-trivial blocks are Hamiltonian, every three longest paths have a common vertex.

- **2-trees**, proved in 2013 by de Rezende et al. [10] (later superseded by the result on series-parallel graphs below). A graph G is a *k-tree* if and only if either G is a complete graph with k vertices or G has a vertex v with degree k such that v together with the set of vertices adjacent to v forms a clique, and $G - v$ is a k-tree.

- **Graphs with matching number at most three**, proved in 2015 by F. Chen [8]. A *matching* of a graph is a set of pairwise nonadjacent edges in the graph. The *matching number* of a graph is the number of edges in a maximum matching of the graph.

- **Dually chordal graphs and connected cographs**, proved in 2016 by Jobson et al. [22]. The result for connected cographs latter was later superseded by the result on P_4–sparse graphs below.

- **Series-parallel graphs**, proved in 2017 by G. Chen et al. [9], supserseding the results on outerplanar graphs and 2-trees mentioned above.

- **$(2K_2)$–free graphs**, proved in 2018 by Golan and Shan [16]. Note that this supersedes the result on split graphs.

- **Starlike graphs, P_4–sparse graphs, $(2P_5, K_{1,3})$–free graphs, graphs that are the join of two graphs, and graphs in which every block is a split graph, interval graph, or graph with a universal vertex**, shown in 2019 by Cerioli and Lima [7] (first published without the proofs in 2016). A *star* is a graph in which all but one vertex v is a leaf, and v is adjacent to every leaf. A *starlike graph* is the intersection graph of substars of a star. This class generalises split graphs. A P_4–*sparse graph* is a graph G in which, for every set S of five vertices of G, the induced subgraph of G with vertex set S has at most one P_4 (path with four vertices). This result supersedes the result on cographs, which are (P_4)–free graphs. The graph P_5 is the path with five vertices. The graph $K_{1,3}$ is the graph with four vertices where three are leaves and the other vertex is adjacent to the three leaves. The *join* of two graphs G_1 and G_2 is the graph with vertex set $V(G_1) \cup V(G_2)$ and edge set $E(G_1) \cup E(G_2) \cup \{uv : u \in V(G_1), v \in V(G_2)\}$. A *universal vertex* is a vertex of a graph G that is adjacent to all other vertices of G.

- $(K_{1,3}, R)$–**free graphs where** $R \in \{C_3, P_4, P_5, P_6, Z_1, Z_2, Z_3, B_{1,1}, B_{1,2}\}$, proved in 2021 by Gao and Shan [15]. The graph C_n is a cycle on n vertices and P_n is a path on n vertices. The graph Z_n is the graph formed from K_3 and a path of length n by identifying one vertex of K_3 with one endpoint of the path. The graph $B_{n,m}$ is the graph formed from K_3, a path P of length n, and a path Q of length m by identifying a vertex of K_3 with one endpoint of P and a different vertex of K_3 with one endpoint of Q.

Appendix B

Python program maximalwalks.py

B.1 Program code

```python
'''Takes a file of graphs, each of which is given by its adjacencies.
Ignores graphs for which further processing is not needed.
For the remaining graphs, creates two files,
one containing a set of inequalities for the maximal paths in lrs format,
and the other a pickled file containing the graph and its maximal paths.

Usage: python3 maximalwalks.py n inFile
n : number of vertices in each graph. '''

from itertools import combinations
import sys
import os
import pickle

'''Create dictionary assigning each edge of graph an index value that can be
looked up. Edges stored as a tuple, both ways eg (0,1) and (1,0). (Used to
give each edge an index in an array.)'''
def make_edge_dict(graph):
    edge_dict = {}
    index = 0
    for vertex in range(len(graph)):
        for adj_v in graph[vertex]:
            if (vertex, adj_v) not in edge_dict:
                edge_dict[(vertex, adj_v)] = index
                edge_dict[(adj_v, vertex)] = index
                index += 1
    return edge_dict

'''Create a tuple of 0s and -1s indicating which edges are in each path, based
on indices in edge_dict, and remove duplicates. Note -1 means edge is in the
path since inequalities are reversed. Does not remove subvectors.'''
def unique_path_edges(edge_dict, all_paths):
    m = len(edge_dict) // 2 # Number of edges
    all_edges = {}
```

```python
    for path in all_paths:
        edges = [0] * m
        for i in range(len(path) - 1):
            # Take two vertices of path and look up 'edge' in dictionary.
            # Set this edge to -1.
            edges[edge_dict[tuple(path[i:i+2])]] = -1
        edges = tuple(edges)
        if edges not in all_edges:
            all_edges[edges] = path
    return all_edges

'''Remove any tuple of edges that is a subset of another tuple of edges.
Returns a dictionary with the edge tuples as keys and the corresponding path as the value.'''
def remove_subvectors(all_edges):
    num_edges = len(next(iter(all_edges)))
    length_buckets = {i: [] for i in range(1, num_edges + 1)}
    for edge_vect in all_edges:
        vect_len = -sum(edge_vect)
        length_buckets[vect_len].append(edge_vect)
    for super_ind in range(num_edges, 1, -1):
        for sub_ind in range(super_ind - 1, 0, -1):
            # Loop backwards so that pop(i) doesn't mess with indices.
            for i in range(len(length_buckets[sub_ind]) - 1, -1, -1):
                if any(is_sub(length_buckets[sub_ind][i], x)
                       for x in length_buckets[super_ind]):
                    length_buckets[sub_ind].pop(i)
    return {v: all_edges[v] for v in sum(length_buckets.values(), [])}

'''Check whether the edges in sub are all in sup.'''
def is_sub(sub, sup):
    return all(x == 0 or y == -1 for x, y in zip(sub, sup))

'''Calculate the number (0, 1, or 2) to be subtracted from the length of the
path to adjust for the start and/or end possibly being repeated.'''
def minus_factor(path):
    minus_num = 0
    first_repeated = path[0] in path[1:]
    last_repeated = path[-1] in path [:-1]
    if first_repeated:
        minus_num = 1
        if last_repeated:
            # If first and last edge are the same, minus_num stays as 1.
            # Otherwise (as below) it's 2.
            if path[0] != path[-2] or path[1] != path[-1]:
                minus_num = 2
    elif last_repeated:
        minus_num = 1
    return minus_num

'''Write inequalities to LRS file.'''
def write_lrs(f, graph, num_vert, graph_num, all_edges):
    f.write(f"Graph{num_vert}v_{graph_num}.ine\n")
    f.write("H-representation \n")
    f.write("begin\n")

    num_edges = len(next(iter(all_edges)))
```

```
    num_ineq = num_edges + len(all_edges)
    num_var = num_edges + 2
    f.write(f"{num_ineq} {num_var} integer\n")

    # write inequality edgeLen >= 1 for each edge so -1 + edgelen >= 0, then no L (0)
    # order: constant, edges, L
    for i in range(num_edges):
        f.write(f"-1 {'0 '*i}1 {'0 '*(num_edges-i-1)}0\n")

    # write inequality for each path: segments - alpha <= L, so alpha - segments + L >= 0,
    for edge_vect in all_edges:
        minus_fact = minus_factor(all_edges[edge_vect])
        # last variable is L (length of longest path)
        f.write(f"{minus_fact} {' '.join(map(str, edge_vect))} 1\n")
    f.write("end\n\n")

'''Determine whether there is a vertex that is in all of the max length paths.'''
def has_intersection(all_paths):
    hits = {}
    for index in range(len(all_paths)):
        path_vertices = set(all_paths[index])
        for v in path_vertices:
            hits[v] = hits.get(v, 0) + 1
    if len(all_paths) in hits.values():
        return True
    else:
        return False

'''A recursive function to generate all maximal "paths" in a graph, where the
endpoints may be repeated. visited[] keeps track of vertices in current path.'''
def gen_all_paths(graph, prefix=(), visited=None, start_visits=0):
    if not prefix:
        for v in range(len(graph)):
            yield from gen_all_paths(graph, (v,), [i == v for i in range(len(graph))], 1)
        return
    for v in graph[prefix[-1]]: # Neighbors of the last vertex
        if visited[v]:
            if len(prefix) > 1 and v == prefix[-2]:
                # We just came from here! Ignore this vertex.
                continue
            elif v == prefix[0]:
                # Repeating the first vertex of the path.
                if start_visits >= 2:
                    # We have already repeated this vertex once, so stop. (Figure eight.)
                    yield prefix + (v,)
                else:
                    # First time repeating this vertex.
                    start_visits += 1
                    yield from gen_all_paths(graph, prefix + (v,), visited, start_visits)
            else:
                # Ending in a cycle or is a cycle. Stop.
                yield prefix + (v,)
                continue
        else:
            # Normal path. Try to recurse if not it's a leaf.
            visited_v = visited[:]
```

```
            visited_v[v] = True
            extensions = gen_all_paths(graph, prefix + (v,), visited_v, start_visits)
            # Check whether there is an element in extensions by tying to get the first one
            # and get None otherwise.
            first_extension = next(extensions, None)
            if first_extension is not None:
                yield first_extension
                yield from extensions
            else:
                # Leaf
                yield prefix + (v,)

'''Check graph for a degree two vertex in which its adj vertices are not themselves adj
(i.e. graph can be reduced). Return True if found. (This graph will then be removed.)'''
def is_reducible(graph):
    for row in graph:
        if len(row) == 2:
            if row[1] not in graph[row[0]]:
                return True
    return False

'''Process a single graph. Stops and returns 0 if the graph is reducible or all (longest)
paths have a common vertex. Otherwise, creates two output files and returns 1.'''
def process_graph(graph, num_vert, graph_num):
    if is_reducible(graph):
        print('reducible')
        return 0
    all_paths = []
    for path in gen_all_paths(graph):
        all_paths.append(path)
    all_intersect = has_intersection(all_paths)
    if all_intersect:
        print('all paths intersect')
        return 0
    edge_dict = make_edge_dict(graph)
    all_edges = remove_subvectors(unique_path_edges(edge_dict, all_paths))
    paths = list(all_edges.values())
    paths_intersect = has_intersection(paths)
    if paths_intersect:
        print('maximal paths intersect')
        return 0
    print('create output files')
    lrsdir = f"pathslrs/lrsin{num_vert}v"
    os.makedirs(lrsdir, exist_ok=True)
    with open(f"{lrsdir}/lrsin{num_vert}v_{graph_num}.ine", "w") as lrs_file:
        write_lrs(lrs_file, graph, num_vert, graph_num, all_edges)
    pkldir = f"pathspkl/paths{num_vert}v"
    os.makedirs(pkldir, exist_ok=True)
    with open(f"{pkldir}/pkl{num_vert}v_{graph_num}.pkl", "wb") as pkl_file:
        pickle.dump(graph, pkl_file, -1)
        pickle.dump(paths, pkl_file, -1)
    return 1

'''Process the graphs in the input file.'''
def process_graphs(num_vert, graphs_file):
    graph = []
```

```
    num_graphs = 0
    for line in graphs_file:
        if line != '\n':
            line_array = line.strip().split()
            if line_array[0] == 'Graph':
                graph_num = int(line_array[1].strip('.'))
                print(graph_num)
            else:
                graph.append([int(x) for x in line_array])
        elif graph_num != 0:
            num_graphs += process_graph(graph, num_vert, graph_num)
            graph = []
    print(f"Total number of graphs: {graph_num}")
    print(f"Number of graphs remaining: {num_graphs}")

if __name__ == '__main__':
    num_vert = int(sys.argv[1])
    if sys.argv[2] != '-':
        with open(sys.argv[2], 'r') as graphs_file:
            process_graphs(num_vert, graphs_file)
    else:
        process_graphs(num_vert, sys.stdin)
```

B.2 Sample input

For Graph 1 and Graph 2 in Figure B.1, the input file, shown below, states the graph number followed by a space-separated list of the vertices that are adjacent to vertices 0, 1, 2, 3, 4, and 5 respectively on individual lines.

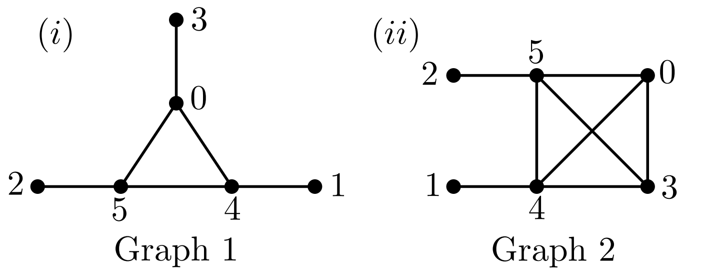

Figure B.1: Two graphs that each have six vertices, (i) Graph 1 and (ii) Graph 2.

```
Graph 1.
3 4 5
4
5
0
0 1 5
0 2 4

Graph 2.
3 4 5
4
```

```
5
0 4 5
0 1 3 5
0 2 3 4
```

B.3 Sample output: file for lrs

For Graph 1 in Figure B.1(i), the output file for lrs is as below.

```
Graph6v_1.ine
H-representation
begin
15 8 integer
-1 1 0 0 0 0 0 0
-1 0 1 0 0 0 0 0
-1 0 0 1 0 0 0 0
-1 0 0 0 1 0 0 0
-1 0 0 0 0 1 0 0
-1 0 0 0 0 0 1 0
0 -1 -1 0 -1 0 0 1
0 0 0 0 -1 -1 -1 1
0 -1 0 -1 0 -1 0 1
1 -1 -1 -1 0 0 -1 1
0 0 -1 -1 -1 -1 0 1
1 0 -1 -1 -1 0 -1 1
0 -1 0 -1 -1 0 -1 1
1 0 -1 -1 0 -1 -1 1
0 -1 -1 0 0 -1 -1 1
end
```

Appendix C

Python program postprocessing.py

C.1 Program code

```
'''Takes output file from LRS for a single graph, processes each vertex of the polyhedron
to find any graphs in which the longest paths do not have a common vertex, and outputs
these graphs and paths.

Usage: python3 post-processing.py n lrsFile pickleFile
n : number of vertices in the graph.
lrsFile: file from lrs containing the output for one graph
pickleFile: pickled file containing the corresponding graph and its paths.
The pickle file is generated by longestpaths.py.
'''

import sys
import pickle
from itertools import combinations
import re
import os
from multiprocessing import Pool
from longestpaths import make_edge_dict, minus_factor
NUM_THREADS = os.cpu_count()

def eprint(*args, **kwargs):
    print(file=sys.stderr, *args, **kwargs)

''' Generator to give pairs of items from a pickled file, and catch the end of the file.'''
def unpickle_pairs(f):
    try:
        while True:
            yield pickle.load(f), pickle.load(f)
    except EOFError:
        eprint("End of pickle file")
        return

'''Turn a string of an integer or fraction into a number (int or float respectively).'''
def numeric(s):
```

```
    if s.isdigit():
        return int(s)
    if '/' in s:
        num, denom = s.split('/')
        return int(num) / int(denom)
    raise ValueError(f"Don't know how to handle number {s}")

'''Determine whether there is a vertex that is in all of the max length paths.'''
def has_max_paths_intersection(all_paths, max_path_indices):
    hits = {}
    for index in max_path_indices:
        path_vertices = set(all_paths[index])
        for v in path_vertices:
            hits[v] = hits.get(v, 0) + 1
    if len(max_path_indices) in hits.values():
        return True
    else:
        return False

'''Calculate the weight of a path.'''
def get_weight(path, edge_dict, edge_lengths):
    w = 0
    used = []
    for i in range(len(path) - 1):
        edge_index = edge_dict[tuple(sorted(path[i:i+2]))]
        if edge_index not in used:
            w += edge_lengths[edge_index]
            used.append(edge_index)
    w -= minus_factor(path)
    return w

'''Find the max weight (length) paths in the graph based on the edge weights.
Allow wiggle room of epsilon = 0.001 to account for rounding errors with floats.'''
def max_paths(all_paths, edge_dict, edge_lengths):
    max_path_indices = []
    max_weight = edge_lengths[-1]
    for index, path in enumerate(all_paths):
        w = get_weight(path, edge_dict, edge_lengths)
        if w >= 0.999 * max_weight:
            max_path_indices.append(index)
    return max_path_indices

'''Convert the max length paths to binary numbers giving the vertices in the path.'''
def binary_max_paths(all_paths, max_path_indices):
    binary_paths = []
    for path in max_path_indices:
        bin_num = 0
        for vertex in all_paths[path]:
            bin_num |= 1 << vertex
        binary_paths.append(bin_num)
    return binary_paths

'''Calculate the intersection of every 3 longest paths in the graph.
Return a string with every set of three paths that don't intersect.'''
def intersect_every_3(binary_paths):
    n = len(binary_paths)
```

```python
    s = ""
    for i in range(n):
        for j in range(i+1, n):
            int_ij = binary_paths[i] & binary_paths[j]
            for k in range(j+1, n):
                if (int_ij & binary_paths[k]) == 0:
                    s += "These 3 paths do not share a vertex."
                    s += f"Path numbers: {combination}, Intersection: {bin(intersection)}\n"
    return s

'''Check vertex of polyhedron to find corresponding longest paths in the graph and check
whether they have a common vertex. If they don't, check every three longest paths.
Return a string to be output.'''
def check_vertex(all_paths, edge_dict, edge_lengths):
    max_path_indices = max_paths(all_paths, edge_dict, edge_lengths)
    all_intersect = has_max_paths_intersection(all_paths, max_path_indices)
    if all_intersect:
        return ""
    s = write_intersection(all_paths, edge_lengths, max_path_indices)
    binary_paths = binary_max_paths(all_paths, max_path_indices)
    s += intersect_every_3(binary_paths)
    return s

'''Return a string to be output containing the paths that do not have a common vertex.'''
def write_intersection(all_paths, edge_lengths, max_path_indices):
    s = f"Edge lengths: {edge_lengths}\n"
    s += f"These {len(max_path_indices)} paths do not have a common vertex:\n"
    for i in max_path_indices:
        s += f"Path {i}: {all_paths[i]}\n"
    return s

''' Generator that gets the next matrix one line at a time, stripping the formatting and
putting each line of the matrix into an array. Takes only the rows that start with a 1
(the vertices of the polyhedron), and ignores those that start with 0 (the rays).'''
def get_matrix(f):
    for line in f:
        if line == 'begin\n':
            break
    else: # "begin" was not found
        return
    # skip over the line after "begin" and before the start of the matrix (gives matrix size)
    f.readline()
    for line in f:
        if line == 'end\n':
            break
        if line.startswith(" 1"):
            yield [numeric(x) for x in line[2:].strip().split()]

'''Helper function for multiprocessing.'''
def get_matrix_jobs(all_paths, edge_dict, lrs_file):
    for line in get_matrix(lrs_file):
        yield all_paths, edge_dict, line

'''Helper function for multiprocessing.'''
def check_vertex_star(args):
    return check_vertex(*args)
```

```python
'''Get a matrix and the corresponding graph and paths from the lrs_file and pickled paths_file,
and check each line of the matrix (check each vertex of the polyhedron).'''
def process_matrices(lrs_file, paths_file):
    graph = pickle.load(paths_file)
    all_paths = pickle.load(paths_file)
    pattern = re.compile(r'Graph\d+v_(?P<graph_num>\d+)[.]ine')
    for line in lrs_file:
        if m:= pattern.match(line):
            break
    else:
        raise Exception('No Graph line in file')
    graph_num = m.group('graph_num')
    eprint(f"Graph {graph_num}")
    print(f"Graph number {graph_num} \nGraph: {graph}")
    edge_dict = make_edge_dict(graph)

    # Multiprocessing
    with Pool(NUM_THREADS) as p:
        jobs = get_matrix_jobs(all_paths, edge_dict, lrs_file)
        for output in p.imap_unordered(check_vertex_star, jobs):
            if output:
                print(output)
    print("done \n")


if __name__ == '__main__':
    graph_vert = sys.argv[1]
    with open(sys.argv[2], "r") as lrs_file, open(sys.argv[3], "rb") as paths_file:
        process_matrices(lrs_file, paths_file)
```

C.2 Sample input: file from lrs

For Graph 1 in Figure B.1(i), the output file from the program `lrs`, which is one of the two input files for `postprocessing.py`, is below. The other input file for `postprocessing.py` is the picked file containing the graph and its maximal paths (not shown).

```
*lrs:lrslib v.7.1 2020.6.4(64bit,lrsmp.h)
Graph6v_1.ine
V-representation
begin
***** 8 rational
 1  1  2  2  1  1  2  6
 0  0  0  1  0  0  0  1
 0  0  1  0  0  0  0  1
 0  0  0  0  0  0  1  1
 0  1  1  1  1  1  1  4
 1  1  2  2  3  3  4  10
 0  0  0  1  1  1  1  3
 0  0  1  0  1  1  1  3
 0  0  0  0  0  0  1  1
 0  1  1  1  2  2  2  6
```

```
 1  1  1  2  2  3  3  8
 0  0  0  0  0  1  0  1
 0  1  0  1  1  2  1  4
 1  1  2  1  3  2  3  8
 0  0  0  0  1  0  0  1
 0  1  1  0  2  1  1  4
 1  1  1  1  2  2  2  6
 0  1  0  0  1  1  0  2
 1  3  2  4  1  3  2  10
 0  0  0  1  0  0  0  1
 0  1  1  1  0  1  0  3
 0  2  1  2  1  2  1  6
 0  1  0  1  0  1  1  3
 1  3  2  3  1  2  1  8
 0  2  1  1  1  1  0  4
 0  1  0  0  0  0  0  1
 1  1  1  2  1  2  2  6
 0  0  0  1  0  0  0  1
 1  2  1  3  1  3  2  8
 0  0  0  0  0  1  0  1
 0  1  0  1  1  2  1  4
 1  2  1  2  1  2  1  6
 0  1  0  0  1  1  0  2
 1  3  4  2  3  1  2  10
 0  1  1  1  1  0  0  3
 0  2  2  1  2  1  1  6
 0  0  1  0  0  0  0  1
 0  1  1  0  1  0  1  3
 1  3  3  2  2  1  1  8
 0  2  1  1  1  1  0  4
 0  1  0  0  0  0  0  1
 1  1  2  1  2  1  2  6
 0  0  1  0  0  0  0  1
 1  2  3  1  3  1  2  8
 0  0  0  0  1  0  0  1
 0  1  1  0  2  1  1  4
 1  2  2  1  2  1  1  6
 0  1  0  0  1  1  0  2
 1  2  2  2  1  1  1  6
 1  1  1  1  1  1  1  4
 0  0  0  0  0  0  1  1
 0  1  0  0  0  0  0  1
 0  0  0  0  0  0  0  1
end
*Note! Duplicate rays may be present
*Totals: vertices=17 rays=36 bases=26 integer_vertices=17  vertices+rays=53
*Dictionary Cache: max size= 6 misses= 0/25   Tree Depth= 5
*lrs:lrslib v.7.1 2020.6.4(64bit,lrsmp.h)
*0.003u 0.003s 1839104Kb 0 flts 0 swaps 0 blks-in 0 blks-out
```

C.3 Sample output

For Graph 1 in Figure B.1(i), the output file is as below. Since this output file does not contain details of any vertex of the polyhedron and corresponding graph and longest paths that do not have a common vertex, every graph that is a subdivision of Graph 1 has a Gallai vertex.

```
Graph number 1
Graph: [[3, 4, 5], [4], [5], [0], [0, 1, 5], [0, 2, 4]]
done
```

Bibliography

[1] Avis, D. (2000). A Revised Implementation of the Reverse Search Vertex Enumeration Algorithm. In Kalai, G. and Ziegler, G. M., editors, *Polytopes – Combinatorics and Computation*, pages 177–198. Birkhäuser Basel.

[2] Axenovich, M. (2009). When do three longest paths have a common intersection? *Discrete Math. Algorithms Appl.*, 1(1):115–120.

[3] Balister, P. N., Győri, E., Lehel, J., and Schelp, R. H. (2004). Longest paths in circular arc graphs. *Combin. Probab. Comput.*, 13:311–317.

[4] Berge, C. (2001). *The Theory of Graphs.* Dover Publications, Inc.

[5] Brandstädt, A., Le, V. B., and Spinrad, J. P. (1999). *Graph Classes: A Survey.* Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics.

[6] Cerioli, M. R., Fernandes, C. G., Gómez, R., Gutiérrez, J., and Lima, P. T. (2020). Transversals of longest paths. *Discrete Math.*, 343(3):111717.

[7] Cerioli, M. R. and Lima, P. (2019). Intersection of Longest Paths in Graph Classes. *Discrete Appl. Math.*, 281:139–142.

[8] Chen, F. (2015). Nonempty intersection of longest paths in a graph with a small matching number. *Czechoslov. Math. J.*, 65(2):545–553.

[9] Chen, G., Ehrenmüller, J., Fernandes, C. G., Heise, C. G., Shan, S., Yang, P., and Yates, A. N. (2017). Nonempty intersection of longest paths in series–parallel graphs. *Discrete Math.*, 340(3):287–304, 1310.1376.

[10] de Rezende, S. F., Fernandes, C. G., Martin, D. M., and Wakabayashi, Y. (2013). Intersecting longest paths. *Discrete Math.*, 313:1401–1408.

[11] Ekstein, J., Fujita, S., Kabela, A., and Teska, J. (2018). Bounding the distance among longest paths in a connected graph. *Discrete Math.*, 341(4):1155–1159.

[12] Fujita, S., Furuya, M., Naserasr, R., and Ozeki, K. (2019). A New Approach Towards a Conjecture on Intersecting Three Longest Paths. *J. Comb.*, 10(2):221–234, 1503.01219.

[13] Fukuda, K. (1993). cdd. `https://people.inf.ethz.ch/fukudak/cdd_home/`.

[14] Gallai, T. (1966). Problem 4. In Erdős, P. and Katona, G., editors, *Theory Graphs, Proc. Colloquium Held Tihany, Hungary, 1966*. Academic Press, New York.

[15] Gao, Y. and Shan, S. (2021). Nonempty intersection of longest paths in graphs without forbidden pairs. *Discrete Appl. Math.*, 304:76–83.

[16] Golan, G. and Shan, S. (2018). Nonempty intersection of longest paths in 2K2-free graphs. *Electron. J. Comb.*, 25(2):2–6, 1611.05967.

[17] Golumbic, M. (2004). *Algorithmic Graph Theory and Perfect Graphs*. Elsevier Science.

[18] Harris, J. M., Hirst, J. L., and Mossinghoff, M. J. (2008). *Combinatorics and Graph Theory*. Springer.

[19] Harvey, D. J. and Payne, M. S. (2020). Intersecting longest paths in chordal graphs. arXiv:2012.07221.

[20] Havet, F. (2004). Stable set meeting every longest path. *Discrete Math.*, 289(1-3):169–173.

[21] Jendrol, S. and Skupień, Z. (1997). Exact Numbers of Longest Cycles with Empty Intersection. *European J. Combin.*, 18:575–578.

[22] Jobson, A. S., Kézdy, A. E., Lehel, J., and White, S. C. (2016). Detour trees. *Discrete Appl. Math.*, 206:73–80.

[23] Joos, F. (2015). A note on longest paths in circular arc graphs. *Discuss. Math. Graph Theory*, 35(3):419–426.

[24] Jooyandeh, M., McKay, B. D., Pettersson, V. H., Osterg, P. R. J., and Zamfirescu, C. T. (2016). Planar hypohamiltonian graphs on 40 Vertices. *J. Graph Theory*, 84(2):121–133.

[25] Kensell, S. (2011). Intersection of longest paths. Master's thesis, Central European University.

[26] Klavžar, S. and Petkovšek, M. (1990). Graphs with nonempty intersection of longest paths. *Ars. Combin.*, 29:43–52.

[27] Makhorin, A. (2000). GLPK: GNU Linear Programming Kit. `https://www.gnu.org/software/glpk/`.

[28] McKay, B. D. (2020). Personal communication.

[29] McKay, B. D. and Piperno, A. (2014). Practical graph isomorphism, II. *J. Symbolic Comput.*, 60:94–112, 1301.1493.

[30] Rautenbach, D. and Sereni, J.-S. (2014). Transversals of longest paths and cycles. *SIAM J. Discrete Math.*, 28(1):335–341.

[31] Schmitz, W. (1975). Über längste Wege und Kreise in Graphen. *Rend. Semin. Mat. Univ. Padova*, 53:97–103.

[32] Shabbir, A., Zamfirescu, C. T., and Zamfirescu, T. I. (2013). Intersecting longest paths and longest cycles: a survey. *Electron. J. Graph Theory Appl.*, 1(1):56–76.

[33] Skupień, Z. (1996). Smallest sets of longest paths with empty intersection. *Combin. Probab. Comput.*, 5:429–436.

[34] Stoll, R. (2012). *Set Theory and Logic.* Dover Books on Mathematics. Dover Publications.

[35] Thomassen, C. (1974). Hypohamiltonian and hypotraceable graphs. *Discrete Math.*, 9(1):91–96.

[36] Tutte, W. T. (1956). A theorem on planar graphs. *Trans. Amer. Math. Soc.*, 82:99–116.

[37] Veblen, O. (1912). An application of modular equations in analysis situs. *Ann. of Math. (2)*, 14(1):86–94.

[38] Walther, H. (1969). Über die Nichtexistenz eines Knotenpunktes, durch den alle längsten Wege eines Graphen gehen. *J. Combin. Theory*, 6:1–6.

[39] Walther, H. and Voss, H. J. (1974). *Über Kreise in Graphen.* Deutscher Verlag der Wissenschaften, Berlin.

[40] Walther, H. J. (1970). Über die Nichtexistenz zweier Knotenpunkte eines Graphen, die alle längsten Kreise fassen. *J. Combin. Theory*, 8:330–333.

[41] Wei, B., Wu, H., and Zhao, Q. (2022). Bonds intersecting long paths in k-connected graphs. arXiv:2201.11245.

[42] Zamfirescu, T. (1972). A two-connected planar graph without concurrent longest paths. *J. Combin. Theory Ser. B*, 13(2):116–121.

[43] Zamfirescu, T. (1976). On longest paths and circuits in graphs. *Math Scand.*, 38:211–239.

[44] Zamfirescu, T. (2001). Intersecting longest paths or cycles: a short survey. *An. Univ. Craiova Ser. Mat. Inform.*, 28:1–9.

[45] Ziegler, G. (2012). *Lectures on Polytopes.* Graduate Texts in Mathematics. Springer, New York.