GRAPHS WITH MONOTONE CONNECTED MIXED SEARCH NUMBER OF AT MOST TWO

by

Micah J Best
Bachelor of Computer Science with Honours, University of Victoria, 2004

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
2007

APPROVAL

Name: Micah J Best

Degree: Master of Science

Title of thesis: Graphs with Monotone Connected Mixed Search Number of at Most Two

Examining Committee: Dr. Ramesh Krishnamurti
Professor, Computing Science
Simon Fraser University
Chair

Dr. Arvind Gupta,
Professor, Computing Science
Simon Fraser University
Senior Supervisor

Dr. Pavol Hell,
Professor, Computing Science
Simon Fraser University
Supervisor

Dr. Andrei Bulatov, External Examiner,
Assistant Professor, Computing Science
Simon Fraser University

Date Approved: Aug 1, 2007

Abstract

Graph searching is used to model a variety of problems and has close connections to variations of path-decomposition. This work explores Monotone Connected Mixed Search. Metaphorically, we consider this problem in terms of searchers exploring a network of tunnels and rooms to locate an opponent. In one turn this opponent moves arbitrarily fast while the searchers may only move to adjacent rooms. The objective is, given an arbitrary graph, to determine the minimum number of searchers for which there exist a valid series of moves that searches the graph. We show that the family of graphs requiring at most k searchers is closed under graph contraction.

We exploit the close ties between the contraction ordering and the minor ordering to produce a number of structural decomposition techniques and show that there are 172 obstructions in the contraction order for the set of graphs requiring at most two searchers.

keywords graph searching, connected monotone mixed search, graph contractions

This one is for my parents. Two searchers who never failed to come find me.

"You are in a maze of twisty little passages, all alike"

— *Crowther and Woods,* ADVENTURE, *1976*

Acknowledgments

Foremost I would like to thank Dimitrios Thilikos from the University of Athens who was kind enough to share his time and experience with me and develop the original foundations of this work. Without his assistance and guidance this work would not exist.

I would also be completely remiss if I did not gratefully acknowledge the support, guidance – and possibly undeserved patience – of my supervisor Arvind Gupta. I couldn't have asked for a better person to guide me through this degree with a combination of encouragement, imparted wisdom and well placed criticism and pressure. My time with him has made me a significantly more effective academic, and possibly a more effective person.

Finally I'd like to thank the staff, specifically Chieko, Asuka, Laurel, Tomo and Chidgdam, at Waves Coffee on Richards and Hastings, Vancouver. I spent many hours there drinking copious amounts of coffee while transforming this material from a rough sketch to a fully realized idea and writing this text. It has been said, either by or about Paul Erdős, that a mathematician is a machine for converting coffee into theorems. If this is true, then those that produce the coffee are critical to the process and deserve thanks.

Contents

List of Figures

Chapter 1

Introduction

Graph searching, as the name implies, is the process of examining a graph where this evaluation is conducted over a discrete series of 'turns', each following specific rules. It is differentiated from classical algorithmic techniques, such as Depth First Search, in that multiple vertices or edges may be examined simultaneously. Similar to the well known 'Cops and Robbers' games, there are a number of variations on problems which place restrictions on when a vertex can be examined. Different metaphors are used to help visualize the process, most involving a series of caves or rooms connected by passages or tunnels. In one of the more common the structure has become contaminated by poison gas and a number of agents are dispatched to decontaminate the area by visiting each room. In another there is an opponent agent, pursued by a number of searchers through the structure. In all cases an optimal solution is one that minimizes the number of agents, searchers or equivalents within the constraints of the search. Unless otherwise noted, it is this second metaphor that we will use when it may increase clarity.

The concept of graph searching in terms of multiple searchers was first defined by Parsons in 1976 [10]. Expanding on his work, Megiddo, Hakimi, Garey, Johnson and Papadimitriou refined the model presented by Parsons in order to prove its complexity [9]. Though this work did not explicitly give their model the name, it did define *edge search*. As the name suggests, in edge search the searcher 'slide' from one vertex to another along the edges of a graph until every edge has been visited. Node search, where the visiting of every vertex by a searcher defines a complete search, was defined by Kirousis and Papadimitriou [6]. The concept of mixed search, which combines the ideas of both edge and node search was defined simultaneously by Bienstock and Seymour [3] and Takahashi, Ueno and Kajitani [1]. A more thorough description of these searches will be defined in the next section.

An interesting extension to this body of work is to consider graph searching with the additional constraint that the searched area remains connected for the entire duration of the search [2]. This restriction allows the modeling of a different variety of problems, but makes a significant difference in the techniques required to study them.

In this work we define connected mixed search, develop a number of techniques for examining graph families closed under edge contraction, and give a complete characterization of all graphs that require at most two searchers to perform a valid monotone connected mixed search. We prove this characterization, for $k = 1$ and $k = 2$, by showing that there is a graph family $\mathbb{F}$ such that following are equivalent:

1. For all graphs $G \in F$, G requires at most k searchers in a valid search strategy.
2. There exists a set of graphs $\mathbb{O}_k$ such that if $G' \notin \mathbb{F}$ then there is a $H \in \mathbb{O}_k$ such that H can be obtained from G' by a series of zero or more edge contractions.
3. There is a formal description of the structure of every graph $\mathbb{F}$.

Chapter 2

Background

2.1 Domain of Graphs and Basic Notation

All graphs under consideration will be finite, simple and non-empty. For a graph $G = (V, E)$ we use $V(G)$ and $E(G)$ to refer to the vertex and edge sets of G, respectively. We use $G\backslash v$ to refer to the graph G with vertex v and any edges with v as an endpoint removed and $G\backslash\{u, \ldots, w\}$ to be the graph with vertices $u, \ldots, w$ and associated edges removed.

2.2 Search Models

In the literature there are several, and in most cases equivalent, definitions of the different searches. The 'flavors' of search we consider in this work can be expressed essentially as a series of turns where one of the following actions can be performed:

Place – a new searcher is placed on a unoccupied node, thus occupying it.

Remove – a searcher is removed from a node, thus leaving it unoccupied.

Move – a searcher is moved from one node to an adjacent node. The adjacent node is now occupied and the original node is left unoccupied.

Once a node has been occupied, it is considered to have been searched. As mentioned above, all searches are monotone. Therefore, a move is invalid if it would allow there to exist an edge such that both endpoints were unoccupied with one node searched and the other unsearched.

Different searches define the criteria to determine when a search has been completed and what variety of moves are allowed.

Edge search In *edge search* an edge is considered searched when a move has been performed across that edge. The search is completed when all edges have been searched.

Node search In *node search*, moving the searchers is disallowed and the search must be conducted with only placements and removals. A search is completed when every node has been searched. The node search number of a graph – the smallest number of searchers required – is exactly one greater than its path-width [6].

Mixed search In *mixed search* an edge is considered searched either when a searcher moves across it, as in edge search, or when two searchers each occupy one endpoint of the edge. The search is completed when all edges and nodes have been searched. The mixed search number of a graph is exactly equal to its proper-path-width. [1]

2.3 Monotone Search

A monotone search is one such that at no point during the search does the 'searched' area decrease in comparison to previous states. The searched area decreases when an unoccupied searched vertex is adjacent to an unsearched vertex. This is often referred to as 'recontamination' in reference to the poison gas metaphor. If there is no agent present to stop the gas, it will flood back into an already cleared area and will have to be cleared again.

LaPaugh showed that edge search allowing recontamination will not decrease the number of searchers [7]. The same result has also been shown for mixed and node searching [4] [3]. Since non-monotonic searches are outside the scope of this work all types of searches will be considered to have monotonicity as an attribute.

2.3.1 Connected Mixed search

Connected mixed search, the model we are concerned with in this work, imposes one additional constraint on the general mixed search described above. In connected mixed search the first placement can be arbitrary, but subsequent placements must be adjacent to an already searched node. This constraint guarantees that the searched area remains connected throughout the entire search.

2.4 Search Strategies and Search Numbers

To describe a monotone mixed search strategy we will use notation based on that presented by Bienstock and Seymour [3], modified slightly for our purposes.

Let S be a sequence of pairs

$$(A_0, Z_0), (A_1, Z_1), \ldots, (A_l, Z_l)$$

where $A_i \subseteq E(G)$ and $Z_i \subseteq V(G)$. In terms of the searcher metaphor, Z_j represents the position of searchers and A_j the edges cleared at turn j.

We say that S is a valid *mixed search strategy* if all of the following hold:

(1) $A_0 = \emptyset$ and $A_l = E(G)$ *(all edges must be searched)*

(2) $\bigcup_{i=0}^{l} Z_i = V(G)$ *(required for the single vertex graph)*

(3) for $0 \le i \le l$ if there is a vertex $v \in V(G)$ such that there exists an edge $(u, v) \in A_i$ for some $u \in V(G)$ and there exists an edge (v, w) for some $w \in V(G)$ such that $(v, w) \notin A_i$ this implies that $v \in Z_i$

(4) for $1 \le i \le l$ exactly one of the following holds:

  (a) $Z_i \supseteq Z_{i-1}$ and $|Z_i| - |Z_{i-1}| = 1$ *(an new searcher has been placed)*

  (b) $Z_i \subseteq z_{i-1}$ and $|Z_{i-1}| - |Z_i| = 1$ *(a searcher has been removed)*

  (c) $|Z_i| = |Z_{i-1}|$, $|Z_i \cap Z_{i-1}| = |Z_i| - 1$. This implies there must be a $(u, v) \in E(G)$ such that $u \in Z_{i-1} \backslash Z_i$ and $v \in Z_i \backslash_{i-1}$. *(a searcher has moved along an edge)*

  (d) $Z_i = Z_{i-1}$ *(no move has been made)*

(5) for each edge $(u, v) \in E(G)$ at least one of the following holds: *(node search and edge search together is mixed search)*

(a) there exists an i such that $\{u, v\} \subseteq Z_i$ *(with condition 2 defines node search)*

(b) there exists an j such that $u \in Z_{j-1}$ and $v \in Z_j$ *(defines edge search)*

A *search number* is the minimum number of searchers to search a given graph. For a valid mixed search strategy $S = (A_0, Z_0), \ldots (A_l, Z_l)$ for a graph G we define $\sigma(S)$ to be $\max\{|Z_0|, \ldots, |Z_l|\}$. This leads to a natural definition of $ms(G)$ as being the smallest value $\sigma(S')$ of any possible valid connected mixed search strategy S' for the graph G.

A valid *connected mixed search strategy* is a valid mixed search strategy with the following additional condition:

(6) for $0 \leq i \leq l$ the vertices $\bigcup_{j=0}^{i} Z_j$ must induce a connected subgraph in G *(enforces connectedness)*

We define $cms(G)$, the connected mixed search number of a graph G, in the same fashion as $ms(G)$ with connected mixed search strategies in place of mixed search strategies. Note that this additional connectedness condition also restricts the number of graphs with a finite connected mixed search number. It is easy to see that $cms(G)$ is defined if and only if G is connected.

2.5 Edge Contractions

An edge contraction is an operation on an edge (u, v) of a graph G that replaces both u and v with a single vertex w such that all vertices that were adjacent to either u or v are now adjacent to w and the self loop (w, w) is removed.

Formally, $G/(u, v)$ is defined as the graph produced by contracting the edge (u, v) in graph G [5]. If $G/(u, v) = \{V', E'\}$ and $w \notin V(G)$ then

$$\begin{aligned} V' &= V(G) \backslash \{u, v\} \cup \{w\} \\ E' &= \begin{array}{l} \{(p, q) \in E(G) | \{p, q\} \cap \{u, q\} = \emptyset\} \\ \cup \{(w, q) | (u, q) \in E(G) \backslash \{(u, v)\} \text{ or } (v, w) \in E(G) \backslash \{(u, v)\}\} \end{array} \end{aligned}$$

For convenience we will define $G/[(u, v) \rightarrow w]$ to be the graph $G/(u, v)$ such that the edge (u, v) is replaced with vertex w where $w \in V(G)$.

2.6 Contraction Relation

We present two alternate versions of the contraction relation, $\leq_c$, and show their equivalence.

Definition by Repeated Contraction Operations (Definition A) A graph H is a contraction of G ($H \leq_c G$) if and only if H can be obtained from G through a series of edge contractions.

Definition by Existence of Mapping (Definition B) For two graphs H and G, $H \leq_c G$ if and only if there exists a injective mapping $\tau : V(H) \longrightarrow 2^{V(G)}$ such that follow hold:

(1) for every pair $u, v \in V(H)$, $(v) \cap \tau(u) = \emptyset$ and $\bigcup_{v \in V(H)} \tau(v) = V(G)$ *(the image of τ forms a partitioning of $V(G)$)*

(2) for every $v \in V(H)$, $\tau(v)$ induces a connected subgraph of G

(3) if there exists an edge $(u, v) \in E(H)$ then there exists an edge $(u', v') \in E(G)$ such that $u' \in \tau(u)$ and $v' \in \tau(v)$ *(edges are preserved)*

(4) if for any pair of vertices $\{u, v\} \subseteq V(G)$ such that $(u, v) \notin E(H)$ then for all $u' \in \tau(u)$ and all $v' \in \tau(v)$ there does not exist an $(u', v') \in E(G)$. *(non-edges are preserved)*

We define any mapping that meets the above conditions to be a *valid contraction mapping.*

Lemma 2.6.1. *Definitions **A** and **B** of $\leq_c$ are equivalent.*

Proof. To show definition A implies definition B, we first show it for the case that H is obtained from G by one contraction.
For any two graphs G and H such that $(u, v) \in E(G)$ and $H = G/[(u, v) \rightarrow w]$ in H we can define an injective mapping τ as follows:

For all vertices $x \in V(H)$

$$\tau(x) = \begin{cases} \{u, v\} & \text{if } x = w \\ \{x\} & \text{otherwise} \end{cases}$$

Observe that mappings defined in **B** are closed under composition. Therefore by induction, if a graph H can be obtained from G by a series of edge contraction then there exists a mapping that meets the criteria in definition **B**.

Also, observe that for a connected graph G there is always a finite series of edge contractions that reduces G to a single vertex. By extension if C is a connected subgraph of G there must exist a finite series of edge contractions that produce a graph G' with C deleted and a new vertex w such that $w \in V(G')$, but $w \notin V(G)$. The structure of G' is such that $V(G')\backslash\{w\} = V(G)\backslash\{(V(C))\}$, $E(G') \cap E(G) = E(G)\backslash\{(E(C))\}$ and for every $(u, v) \in E(G)$ such that $u \notin V(C)$ and $v \in V(C)$ there is an edge $(u, w) \in E(G')$. In other words the vertices of the component C have been reduced to a single vertex. We call any series of contractions of this type a *reduction series* of C with respect to G.

Let G and H be two graphs such that there exists a mapping $\tau : V(H) \to 2^{V(G)}$ that satisfies the conditions of definition **B**. For each $v \in V(H)$, let R_v be the reduction series of $\tau(v)$ with respect to G if $|\tau(v)| > 1$ and empty otherwise. The concatenation of R_u, for all $u \in V(H)$ gives a series of edge contractions that will produce H if applied to G.

Thus the two definitions are equivalent. □

For convenience we say that for two graphs G and H with vertices $a \in V(H)$ and $b \in V(G)$, $H \leq_c^{(a \to b)} G$ if and only if $H \leq_c G$ and there exists a valid contraction mapping τ such that $b \in \tau(a)$.

2.7 Graph Minors

Given any graph G 'taking a minor' of G, in other words applying the minor operation to G, involves a series of the following changes to the structure of G:

Vertex Deletion Some $v \in V(G)$ is removed. All edges (u, v) such that $(u, v) \in E(G)$ are also removed.

Edge Deletion Some $e \in E(G)$ is removed.

Edge Contraction A pair of vertices u, v such that $(u, v) \in E(G)$ are contracted as described in section 2.5.

While there are several different equivalent characterizations of the minor relation, $\leq_m$, it will be most convenient to work with the following:

For two non-empty graphs H and G, we say that $H \leq_m G$, that is H is a minor of G, if and only if there exists a subgraph G' of G such that $H \leq_c G$' [5]. The empty graph is always a minor of any graph.

2.8 Closure

A family of graphs $\mathbb{F}$ is said to be closed under a operation $\oplus$ if and only if for every graph $G \in \mathbb{F} \Rightarrow \oplus(G) \in \mathbb{F}$.

2.9 Obstructions

Consider a graph family $\mathbb{F}$ which is closed under an operation $\oplus$ and the relation $\leq_\oplus$ such that for two graphs G and H, $H \leq_\oplus G$ if and only if H can be obtained from G by zero or more applications of $\oplus$ and $\leq_\oplus$ defines a partial ordering of all graphs. Let $\overline{\mathbb{F}}$ be all graphs G such that $G \notin \mathbb{F}$: this is the set of forbidden graphs for the family $\mathbb{F}$. Let $\mathbb{O} = \{G | G \in \overline{\mathbb{F}}$ *and if* $H \leq_\oplus G, H \neq G$ *then* $H \in \mathbb{F}\}$. This subset of $\overline{\mathbb{F}}$, $\mathbb{O}$, is the set of minimal forbidden graphs, obstructions, for that family with respect to that relation. Intuitively, $\mathbb{O}$ is the 'border' between $\mathbb{F}$ and $\overline{\mathbb{F}}$ with respect to the partial order defined by $\leq_\oplus$.

2.10 Graph Terminology

2.10.1 Articulation Point

An articulation point, also called a cut vertex, for a connected graph G is a vertex $v \in V(G)$ such that $G \backslash v$ is disconnected.

2.10.2 Biconnected graph

A biconnected, or 2-connected, graph G is a connected graph such that for every vertex $v \in V(G)$, $G \backslash v$ is connected [5]. In other words a biconnected graph is a connected graph that does not contain an articulation point.

2.10.3 Outerplanar Graph

A graph G is *outerplanar* if it has an embedding on the plane, without edge crossing, in which every vertex lies on the outer face [5]. That is to say that the vertices of G can be arranged on the plane such that the the border between the infinite area that lies 'outside' of any area of the plane bounded by the edges of G contains each of the vertices of G.

2.10.4 Biconnected Outerplanar Graphs

The definitions above imply a feature of biconnected outerplanar graphs that will become integral to the upcoming proofs.

Lemma 2.10.1. *For all biconnected outerplanar graphs G, G has a subgraph C such that C is a cycle and $V(C) = V(G)$.*

Proof. Assume not. At least on of the following must apply to G':

(1) **Case 1** *There exists a vertex v in $V(G')$ such that v has degree 1.* Therefore there must exist exactly one vertex u in $V(G')$ such that the edge (u, v) exists in $E(G')$. Therefore $G' \backslash u$ is disconnected and G' cannot be outerplanar.

(2) **Case 2** *There exists a pair of vertices $\{u, v\} \subset V(G')$ such that $G' \backslash \{u, v\}$ is disconnected and has at least three connected components.* Let $C'_1, \ldots, C'_3$ be three connected components in $G' \backslash \{u, v\}$. By the definition of an outerplanar graph, G' must have a planar embedding such that all vertices lie on the outer. So if the subgraph induced by the vertices $V(C'_1)$, $V(C'_2)$, u and v are embedded without edge crossing they must define an inner face distinct from the unbounded outer face. So it is impossible to augment this embedding to include C'_3 while maintaining planarity without either by placing it inside this inner face created by the first two, or creating a new outer face and containing one of the original components.

Since either case gives contradiction and so G' cannot be both biconnected and outerplanar and thus proving the lemma. $\square$

In order to facilitate discussion of the properties of biconnected outerplanar graphs we define $Outer(G)$ of a biconnected outerplanar graph G to be the subgraph C of G such that C is cycle and $V(C) = V(G)$. Similarly we define $Inner(G)$ to be the graph I such that $E(I) = E(G) \backslash E(Outer(G))$ and $V(I)$ to be all vertices v such that for some $u \in V(G)$, $(u, v) \in E(I)$.

Chapter 3

Preliminaries

In this work we will present a characterization of all graphs with a connected mixed search number less than or equal to two. We will do this by explicitly stating the obstructions in the contraction ordering for these families and proving that this set is complete. Before we get to the heart of the proof we need to show some facts about the contraction relation, both by itself and in connection to the minor relation. Additionally, we will introduce notation specific to this problem that will considerably condense the proofs.

3.1 Contractions, Minors, and Searching

3.1.1 Connected Mixed Searching Is Not Minor Closed

In chapter 2 we gave descriptions for three searches that do not require connectedness: edge, node, and mixed. It is known that each search is minor closed. For example, with mixed search for two graphs G and H if $ms(G) = k$ and $H \leq_m G$ then $ms(H) \leq k$. This is unfortunately not the case with connected mixed search. It has been shown in general that connected monotone searches are not minor closed [2]. For the specific case of connected mixed search (recall that all searches under examination are monotone) we can demonstrate this with a small counter example.

Since H' is a subgraph of H, H' is a minor of H. However a quick inspection will show that $cms(H) = 2$ and $cms(H') = 3$.

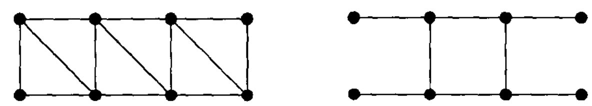


Figure 3.1: Counterexample for the minor closure of *cms*

3.1.2 Connected Mixed Searching Is Closed Under Contraction

Fortunately, for the purposes of our characterization, connected mixed searching is closed under the contraction operation defined in section 2.

Theorem 3.1.1. *The family of graphs G such that $cms(G) \leq k$ is closed under edge contraction.*

Proof. Consider a valid connected mixed search strategy $S = (A_0, Z_0), \ldots, (A_l, Z_l)$ for a graph G, and the graph $G' = G/(u, v)$ for an arbitrary $(u, v) \in E(G)$ where the vertex $w \notin V(G)$ replaces the edge (u, v) in G'. We define $S' = (A'_0, Z'_0), \ldots, (A'_l, Z'_l)$ to be a search strategy for G' constructed as follows, for $0 \leq i \leq l$:

(1) Let $R_i^E \subseteq A_i$ be the edges (x, y) such that either $x = u$ or $y = v$. Let $R_i^{E'}$ be the set of edges $(a, w) \subset E(G')$ such that $(a, u) \in R_i^E$ or $(a, v) \in R_i^E$ and $a \notin \{u, v\}$. *(This defines the set of edges that need to be replaced and their replacements. Note that (u, v) is not in $R_i^{E'}$).* $A'_i = (A_i \backslash R_i^E) \cup R_i^{E'}$.

(2) Let $R_i^V = Z_i \cap \{u, v\}$ *(the set of vertices for replacement)*. Let $Z'_i = (Z_i \backslash R_i^V) \cup \{w\}$ if $R_i^V \neq \emptyset$, and Z_i otherwise.

The construction above guarantees that $A'_l = E(G')$ and $\bigcup_{i=0}^{l} Z'_i = V(G')$, thus fulfilling conditions 1 and 2 for a valid connected mixed search strategy *(the graph is completely searched)*. The satisfaction of condition 3 can be seen by observing that if (a, b) was an edge in A_i and a was in Z_i then

- $(a, b) \in A'_i$ and $a \in Z'_i$ if $a \neq w$ and $b \neq w$
- $(a, w) \in A'_i$ and $a \in Z'_i$ if $a \notin \{u, v\}$ and $b \in \{u, v\}$
- $(w, b) \in A'_i$ and $w \in Z'_i$ if $a \in \{u, v\}$ and $b \notin \{u, v\}$

The construction of S' guarantees that $|Z_i'| = |Z_i| - 1$ if $\{u, v\} \subseteq Z_i$ and $|Z_i'| = |Z_i|$ otherwise, which satisfies all the cardinality constraints for condition 4. Since S is valid, it is impossible for there to be a pair Z_i, Z_{i+1} such that $Z_i \cap Z_{i+1} = \{u, v\}$. Then

- if $Z_i \subseteq Z_{i+1}$ and $a = Z_i \cap Z_{i+1}$ then $Z_i' \subseteq Z_{i+1}'$ and $Z_i' \cap Z_{i+1}' = a$ if $a \notin \{u, v\}$ and $Z_i' \cap Z_{i+1}' = w$ otherwise. *(a searcher has been placed)*

- if $Z_i \supseteq Z_{i+1}$ and $b = Z_i \cap Z_{i+1}$ then $Z_i' \supseteq Z_{i+1}'$ and $Z_i' \cap Z_{i+1}' = b$ if $b \notin \{u, v\}$ and $Z_i' \cap Z_{i+1}' = w$ otherwise. *(a searcher has been removed)*

- if $|Z_i| = |Z_{i+1}|$, $|Z_i \cap Z_{i+1}| = |Z_{i+1}| - 1$ and $a \in Z_i$, $a \notin Z_{i+1}$, $b \notin Z_i$, $b \in Z_{i+1}$ then

  - $a \in Z_i'$ and $b \in Z_{i+1}'$ if $a, b \cap \{u, v\} = \emptyset$
  - $a \in Z_i'$ and $w \in Z_{i+1}'$ if $a \notin \{u, v\}$ and $b \in \{u, v\}$
  - $w \in Z_i'$ and $b \in Z_{i+1}'$ if $a \in \{u, v\}$ and $b \notin \{u, v\}$
  - $Z_i' = Z_{i+1}'$ if $\{a, b\} = \{u, v\}$

  (a searcher has been moved).

- if $Z_i = Z_{i+1}$ then $Z_i' = Z_{i+1}'$ *(no move has been made)*

These satisfy the remainder of the constraints for condition 4.

Condition 5, the constraints that define node and edge search, can been seen to be satisfied by the following: For each edge $(a, b) \in E(G)$

- if $\{a, b\} \cap \{u, v\} = \emptyset$ then if for some j_1,$1 \leq j_1 \leq l$, $\{a, b\} \subseteq Z_{j_1}$ then $\{a, b\} \subseteq Z_{j_1}'$ and if for some j_2,$1 \leq j_2 \leq l$, $a \in Z_{j_2}$ and $b \in Z_{j_2+1}$ then $a \in Z_{j2}'$ and $b \in Z_{j2+1}'$

- if $a \notin \{u, v\}$ and $b \in \{u, v\}$ then if for some j_1,$1 \leq j_1 \leq l$, $\{a, b\} \subseteq Z_{j_1}$ then $\{a, w\} \subseteq Z_{j_1}'$ and if for some j_2, $1 \leq j_2 \leq l$, $a \in Z_{j_2}$ and $b \in Z_{j_2+1}$ then $a \in Z_{j2}'$ and $w \in Z_{j2+1}'$

- if $a \in \{u, v\}$ and $b \notin \{u, v\}$ then if for some j_1,$1 \leq j_1 \leq l$, $\{a, b\} \subseteq Z_{j_1}$ then $\{w, b\} \subseteq Z_{j_1}'$ and if for some j_2, $1 \leq j_2 \leq l$, $a \in Z_{j_2}$ and $b \in Z_{j_2+1}$ then $w \in Z_{j2}'$ and $b \in Z_{j2+1}'$

- if $\{a, b\} = \{u, v\}$ the edge (u, v) does not occur in $E(G')$

Condition 6, which enforces connectedness, can be seen to be satisfied by considering following: for all i, $0 \leq i \leq l$ let H_i be the graph induced in G by $\bigcup_{j=0}^{i} Z_j$, and H' be the graph induced in G' by $\bigcup_{j=0}^{i} Z'_j$. If $V(H_i) \cap \{u, v\} = \emptyset$ then H_i and H'_i are identical, and thus H' is connected. Recall that since S is a valid connected mixed search strategy then each H_i must be connected. If $|V(H_i) \cap \{u, v\}| = 1$ (one of u or v is in $V(H_i)$) then H_i and H'_i are isomorphic, and so H' is connected. If $\{u, v\} \subseteq V(H_i)$ then $H'_i = H_i/(u, v)$ and since edge contraction maintains connectedness, H'_i is connected. Thus condition 6 also holds and S' is valid.

Finally observe that for all i, $0 \leq i \leq l$ that $|Z'_i| \leq |Z_i|$ and we can conclude by extension that $cms(G') \leq cms(G)$. □

Thus we can conclude that for all $k > 1$ that the family of graphs with a connected mixed search number less than or equal to k has a set of obstructions in the contraction ordering, $\mathbb{O}_k$; however, we can not guarantee that for any particular k that $\mathbb{O}_k$ is finite. The well known Graph Minor Theorem, proved by Robertson and Seymour, shows that the minor relation defines a well-quasi-ordering and thus all sets of obstructions in the minor ordering for minor closed families are finite. For a more thorough overview of the Graph Minor Theorem and well-quasi-ordering see [5]. We can make no such claim for the contraction relation as it does not define a well-quasi-ordering of all graphs.

Theorem 3.1.2. *The contraction relation, $\leq_c$, does not define a well-quasi-ordering on the set of all graphs*

Proof. One of the properties of a well-quasi-ordering is that it does not contain an infinite antichain [5]. That is to say that the set on which the relation is defined does not contain an infinite subset such that any two elements in the subset are incomparable under the given relation. In order to prove the theorem we demonstrate an infinite antichain in the set of graphs under the contraction relation.

Define the graph N_1 such that $V(N_1) = \{a, b, v_1\}$ and $E(N_1) = \{(a, v_1), (b, v_1)\}$. For all $i > 1$ define N_i such that $V(N_i) = V(N_{i-1}) \cup \{v_i\}$ and $E(A_i) = E(A_{i-1}) \cup \{(a, v_i), (b, v_i)\})$.

Suppose that there existed a distinct m, n such that $N_m \leq_c N_n$. Given that any edge contraction of N_n must be either $N_n/(a, v_i)$ or $N_n/(b, v_i)$ for some i, $1 \leq i \leq n$. Since a

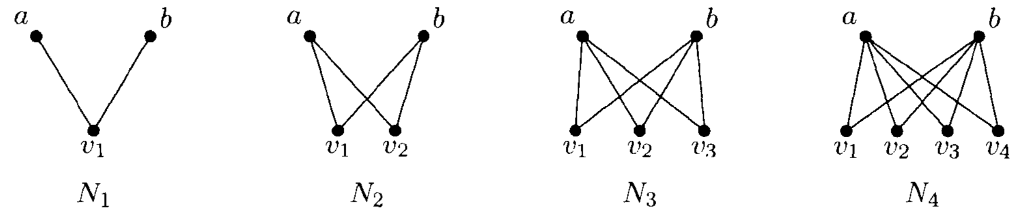


Figure 3.2: The first four elements of the counterexample to the well-quasi-ordering of $\leq_c$

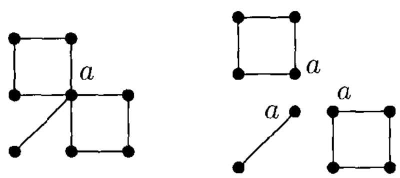
and b are interchangeable we can say without loss of generality that the first contraction is $N_m/[(a, v_i) \to a]$. N_m cannot be N_1 as any N_j, $j > 1$ has more vertices than N_1 and so $m < n$ and $n > 1$. Observe that $N_n/[(a, v_i) \to a]$ must contain a K_3 as a subgraph and so cannot be N_m. Consider now the series of contractions that follow the first. A single edge contraction can produce one of two results. If the edge (a, b) is contracted then the entire graph is reduced to a star, further contraction of which must also produce stars until a K_2 is obtained. Since K_2 has only two vertices there cannot be a j such that $N_j \leq_c K_2$ as all member of the series have at least 3 vertices. Otherwise, again, either $(a, v_{i'})$ or $(b, v_{i'})$ for some i', $1 \leq i' \leq n$ must be contracted. In either case the resulting graph will retain a K_3 as subgraph and so cannot possibly be a member of the series. After a further $m - 1$ of these contractions the graph will have been reduced to a K_2, and as above there is no $N_{j'}$ such that $N_{j'} \leq_c K_2$. This gives us a contradiction and proves the theorem. □

However, while not being well-quasi-ordered guarantees that some set of obstructions in the contraction order is infinite, it does not preclude some families from having a finite number of obstructions. We shall show that this is the case for many non-trivial graph families (see Corollary 3.3.2), and specifically for the class of graphs with $cms \leq 1$ and $cms \leq 2$.

3.2 Further Graph Operations

In the following proofs it will become necessary to examine different subgraphs of graph under consideration and their relation to each other. For the sake of brevity and clarity we now define six more graph operations that we will be using repeatedly.

3.2.1 Split

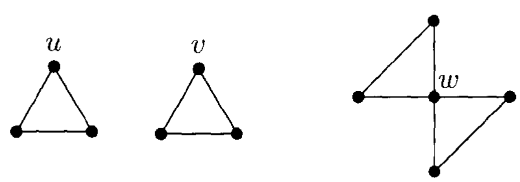


G_1 $\quad$ $Split(G_1, a)$

Figure 3.3: Example of *Split*

For a graph G with an articulation point $a \in V(G)$ we define $Split(G, a)$ as follows. Let $G'_1, \ldots, G'_n$ be the of $G \backslash a$ and for all $1 \leq i \leq n$, define G''_i such that $V(G''_i) = V(G'_i) \cup \{a\}$ and $E(G''_i) = E(G'_i) \cup \{(u, a) | (u, a) \in E(G) \text{ and } u \in V(G'_i)\}$. Then $Split(G, a) = \{G''_1, \ldots, G''_n\}$.

3.2.2 Join



G_2 $\quad$ H $\quad$ $Join(G_2, H, u, v, w)$

Figure 3.4: Example of *Join*

As the name suggests *Join* is the reverse operation of *Split*. Given two graphs G and H and three vertices u, v, w such that $u \in V(G)$, $v \in V(H)$ and $w \notin V(G) \cup V(H)$ we define $Join(G, H, u, v, w)$ to be the graph GH such that $V(GH) = V(G) \backslash \{u\} \cup V(H) \backslash \{v\} \cup \{w\}$. The edge set of GH is defined as follows: for all pairs of distinct vertices c and d such that $(c, d) \in E(G) \cup E(H)$, if $\{c, d\} \cap \{u, v\} = \emptyset$ then $(c, d) \in E(GH)$. Otherwise it follows that, without loss of generality, $c \in \{u, v\}$, and so $(w, d) \in E(GH)$.

3.2.3 Follow

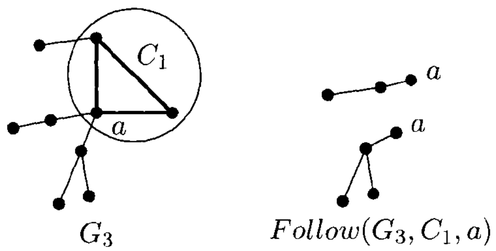


Figure 3.5: Example of *Follow*

Given a graph G with a subgraph C and an articulation point $a \in V(G)$ such that $a \in V(C)$ then let G' be the graph in $Split(G, a)$ such that $V(C) \subseteq V(G')$. Then $Follow(G, C, a) = Split(G, a) \backslash \{G'\}$, the set of subgraphs that 'follow' a with respect to C.

3.2.4 Prune

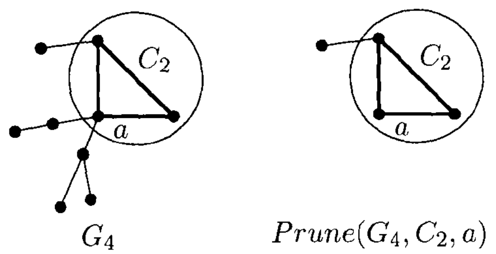


Figure 3.6: Example of *Prune*

Prune, in a certain respect, is the complementary operation to *Follow*. Given a graph G with a subgraph C and an articulation point $a \in V(G)$ such that $a \in V(C)$ then let G' be the graph in $Split(G, a)$ such that $V(C) \subseteq V(G')$. Then $Prune(G, C, a) = G'$.

3.2.5 Contract

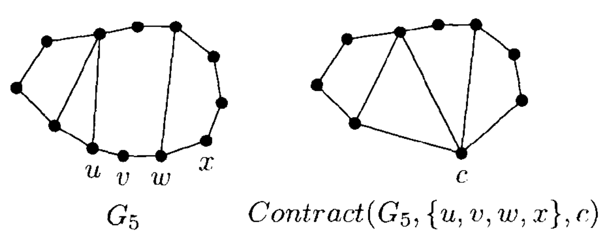


Figure 3.7: Example of *Contract*

For a graph G and a set $C \subseteq V(G)$ such that C induces a connected subgraph in G, $Contract(G, C, c)$ is the graph obtained by contracting the edges in the subgraph induced by C to a vertex labeled c. The fact that $Contract(G, C, c) \leq_c G$ follows directly from definition **B** of the contraction relation given in section 2.6.

3.2.6 Span

The *Span* operation is particular to biconnected outerplanar graphs. If G is a biconnected outerplanar graph with three vertices u, v, w such that $\{u, v, w\} \subseteq Outer(G)$ then $Span(G, u, v, w)$ is the subpath P of $Outer(G)$ between, but not including, u and v such that $w \notin V(P)$.

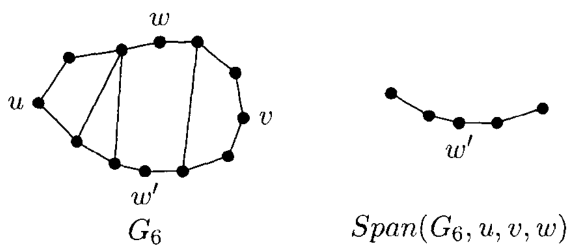


Figure 3.8: Example of *Span*

3.3 Relation Between Minors and Contractions

Theorem 3.3.1. *If $H \leq_m G$ then then there exists a H' such that $H' \leq_c G$ where $V(H') = V(H)$ and $E(H) \subseteq E(H')$.*

Proof. To prove this we will give a method to construct H' from H. The definition of the minor relation in Section 2.7 says that there must exist a subgraph G' of G such that $H \leq_c G'$. Therefore there exists an injective mapping $\tau : V(H) \to 2^V(G')$ that satisfies the conditions given in Section 2.6.

Let $R_1, \ldots, R_n$ be the connected components of $G \backslash V(G')$. For $1 \leq i \leq n$ let r_i be a vertex in $V(G')$, chosen arbitrarily, such that there exists a $b \in V(R_i)$ where $(r_i, b) \in E(G)$. We construct H' as follows. Let $V(H') = V(H)$ and let $E(H)$ be the set of edges such that $(u, v) \in E(H)$ if and only if one of the following holds:

(1) The edge (u, v) is in $E(H)$

(2) There is an edge $(u', v') \in E(G)$ such that $u' \in \tau(u)$ and $v' \in \tau(v)$

(3) For some i, $1 \leq i \leq n$, $u = r_i$ and there exists a path P in G from some $u' \in \tau(u)$ to some $v' \in \tau(v)$ such that $V(P) \backslash \{u', v'\} \subseteq V(R_i)$

It remains to show that H' is a contraction of G. Define $\tau' : V(H') \to 2^V(G)$ such that for all vertices $x \in V(H')$, $\tau'(x)$ is the subset $V(G)$ such that $w \in \tau'(x)$ if and only if either $w \in \tau(x)$ or $w \in R_j$, $1 \leq j \leq n$, where $r_j \in \tau(x)$.

Observe that since the union of $V(R_1), \ldots, V(R_n)$ and $V(G')$ is $V(G)$ and that each subgraph R_i, $1 \leq i \leq n$, is connected in G by an edge to a vertex in $V(G')$ that the image of τ' is partitioning of $V(G)$ and for all $w \in V(H')$, $\tau'(w)$ is connected. If $(u, v) \in E(H)$ there must be an edge between the graphs induced in G by $\tau'(u)$ and $\tau'(v)$ by the validity of τ. An edge $(u, v) \in E(H') \backslash E(H)$ must have been added only if u and v met condition 2 or 3 in the construction of H'. If the former then the edge between $\tau'(u)$ and $\tau'(v)$ is demonstrated explicitly in the construction. If the later then (u, v) would only have been added to $E(H')$ if there was an R_j, $1 \leq j \leq n$, such that there was a path P as a subgraph of G where the endpoints were vertices from $\tau(u)$ and $\tau(v)$ and the remainder of the vertices are a subset of $V(P)$. By the construction of τ', $V(R_j) \subset \tau'(u) \cup \tau'(v)$ and so there must be an edge between $\tau'(u)$ and $\tau'(v)$ in G.

Finally we must consider the condition of non-edge preservation. Suppose there is a pair of vertices $\{u, v\} \in V(H')$ such that $(u, v) \notin E(H')$ and there is an edge $(u', v') \in E(G)$ such that $u' \in \tau'(u)$ and $v' \in \tau'(v)$. At least one of $\{u', v'\}$ must be in $\tau(u) \cup \tau(v)$ as otherwise there would exist an R_i, $1 \leq i \leq n$ and an R_j, $1 \leq j \leq n$ where $i \neq j$ such that $u' \in V(R_i)$ and $v' \in V(R_j)$ and this would imply that R_i and R_j were subgraphs of the same connected component in $G \backslash V(G')$ which is contrary to the definition of $R_1, \ldots, R_n$. If $u' \in \tau(u)$ and $v' \in \tau(v)$ then the edge (u, v) would have been added $E(H')$ by condition 1 or 2. If $v' \in \tau(v)$ and $u' \notin \tau(u)$ then there is an R_j, $1 \leq j \leq n$, such that $u' \in V(R_j)$ and that $r_j \in \tau(u)$ and so the edge (u, v) would have been added $E(H')$ by condition 3. Since both cases show that (u, v) would have been added to $E(H')$ this gives us a contradiction and proves the Theorem.

□

For a minor closed graph family F notice that $\mathbb{F}$ is also closed under contraction. If $\mathbb{O_F}^M = \{M_1, \ldots M_n\}$ are the obstructions in the minor order for $\mathbb{F}$ for each M_i let $E_i^\star = 2^{E(\overline{M_i})}$ *(the set of all possible sets of non edges of M_i)*. Let $M_i^\star = \{H | V(H) = V(M_i)$ *and* $E(H) E(M_i)\}$. Let $\mathbb{M} = \bigcup_{i=1}^{n} M_i^\star$.

Corollary 3.3.2. *For any minor closed graph family $\mathbb{F}$ there exists a finite set of contraction obstructions, ${\mathbb{O}_\mathbb{F}}^C$, ${\mathbb{O}_\mathbb{F}}^C$ for $\mathbb{F}$ where ${\mathbb{O}_\mathbb{F}}^C \subseteq \mathbb{M}$.*

Proof. Assume not. Let O be an obstruction in the contraction order for $\mathbb{F}$ such that $O \notin \mathbb{M}$. Since $O \notin \mathbb{F}$ then there exists an $M_j \in {\mathbb{O}_\mathbb{F}}^M$ such that $M_j \leq_m O$ and so by Theorem 3.3.1 and the construction of $M_i^\star$ we know that $O' \in M_j^\star$. Thus $O' \in \mathbb{M}$, is a contradiction. Since ${\mathbb{O}_\mathbb{F}}^M$ is finite each $M_i^\star$ is finite and thus ${\mathbb{O}_\mathbb{F}}^C$ is finite. □

Additionally, any graph with an edge has K_2 as a minor and any graph with a cycle has K_3 as a minor. As both of these graphs are complete and permit no additional edges, Theorem 3.3.1 tell us that K_2 is a contraction of any graph with an edge, and K_3 is a contraction of any graph with a cycle.

3.4 Constructing Contraction Obstruction Sets From Minor Construction Sets

A procedure for constructing contraction obstruction sets from minor obstruction sets follows directly from the above. Let $M_i^\star$, $1 \leq i \leq n$ and the set $\mathbb{M}$ for the minor closed graph family $\mathbb{F}$ be defined as in the previous section. Since for each $H \in \mathbb{M}$ there exists an M_j for some $1 \leq j \leq n$ such that $M_j \leq_m H$ then $H \notin \mathbb{F}$. In other words every member of $\mathbb{M}$ is a forbidden contraction. Let $\mathbb{O}_F^C$ be all $O \in \mathbb{M}$ such that for all $O' \in$ where $O' \neq O$ then $O' \not\leq_c O$. Thus $\mathbb{O}_F^C$ is exactly the set of contraction obstructions to $\mathbb{F}$.

3.5 Articulation Points and Contractions

Lemma 3.5.1. *Given a graph G with a connected subgraph C and a set of articulation points $A = \{a_1, \ldots, a_n\}$ such that $A \subseteq V(C)$ and for every edge $(u, v) \in E(G)$, if $u \notin V(C)$ and $v \in V(C)$ then $v \in A$. Then $C \leq_c G$.*

Proof. We can demonstrate this simply, the proof follows the same form as Theorem 3.3.1, only simpler. Let $\tau : V(C) \to 2^{V(G)}$ be a mapping such that for all $x \in V(C)$:

$$\tau(x) = \begin{cases} \{x\} & x \notin A \\ V(Follow(G, C, x)) \cup \{x\} & \text{otherwise} \end{cases}$$

which is a valid contraction mapping for $C \leq_c G$.

□

Lemma 3.5.2. *Given a graph G with a C* **and an articulation point** a **such that** $a \in V(C)$**, let** H **be any graph with a vertex** b **such that for some** $F \in Follow(G, C, a)$, $H \leq_c^{(b \to a)} F$ **. Then** $Join(Prune(G, C, a), H, a, b, c) \leq_c G$**.**

Proof. Again we can can show this with a simpler version of Theorem 3.3.1. By the given conditions there exists a valid contraction mapping $\tau : V(H) \to 2^{V(F)}$ such that $a \in \tau(b)$. If we let $G' = Join(Prune(G, C, a), H, a, b, c)$ we can define $\tau' : V(G') \to 2^{V(G)}$ for all $x \in V(G')$:

$$\tau'(x) = \begin{cases} \{x\} & x \in V(Prune(G, C, a)) \text{ and } x \neq a \\ \tau(b) & \text{if } x = c \\ \tau(x) & \text{otherwise} \end{cases}$$

Which is a valid contraction mapping for $G' \leq_c G$. □

3.6 Articulation Points and Graph Demarcation

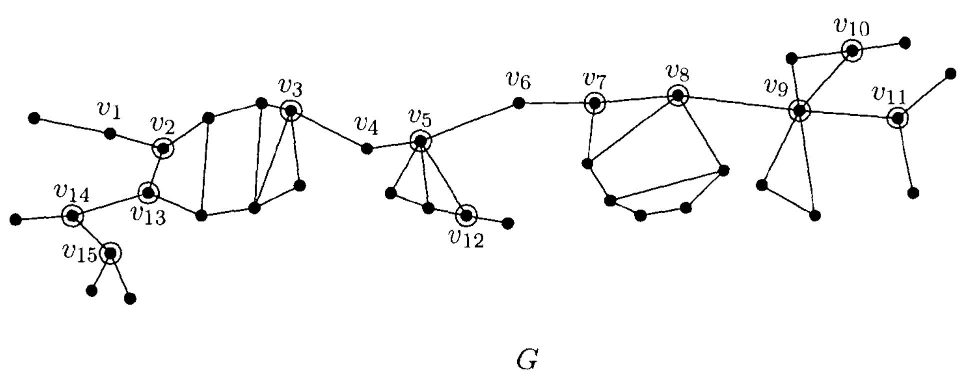


Figure 3.9: Outerplanar graph G with the members of $SA(G)$ circled

3.6.1 Absolute Labeled Articulation Points

Let $A(G) \subseteq V(G)$ be all articulation points in the graph G. By definition an articulation point must have at degree of at least 2. If an articulation point has degree of exactly 2 we refer to it as a *passing articulation point* and as a *significant articulation point* otherwise. Let $PA(G)$ be the passing articulation points of G and $SA(G)$ be the significant articulation points. If we observe G from figure 3.9 then $PA(G) = \{v_1, v_4, v_6\}$ and $SA(G) = \{v_2, v_3, v_5, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}, v_{15}\}$.

3.6.2 Demarcation Operation

It will be fundamental to the following proofs to examine the components separated by the significant articulation points. For this purpose we define the demarcation operation. For a graph G we define $Demarcate(G)$ as follows:

- Let $G'_1 \ldots G'_n$ be the of $G \backslash SA(G)$.
- For $1 \leq i \leq n$, let V'_i be the vertices v such that $v \in SA(G)$ and there exists an edge (u, v) for some $u \in V(G'_i)$. Let E'_i be the set of edges (u, v) such that $v \in V'_i$, $u \in V(G'_i)$, and $(u, v) \in E(G)$. Let G''_i be the graph such that $V(G''_i) = V(G'_i) \cup V'_i$ and $E(G''_i) = E(G'_i) \cup E'_i$.
- Let $L^\star$ be the set of graphs such that for every edge $(u, v) \in E(G)$ such that for $1 \leq i \leq n$, $(u, v) \notin E(G''_i)$ then the graph L such that $V(L) = \{u, v\}$ and $E(L) = \{(u, v)\}$ is in $L^\star$.
- $Demarcate(G) = \{G''_1, \ldots, G''_n\} \cup L^\star$.

For the graph G presented in figure 3.9 we show the resulting graphs of $Demarcate(G)$ in figure 3.10.

We refer to any graph $G' \in Demarcate(G)$ as a *demarcated component*. In figure 3.10 C is a demarcated component. Finally for any $G' \in Demarcate(G)$ we define $DSA(G, G')$, the significant articulation points from G in demarcated component G', to be $SA(G) \cap V(G')$. In figure 3.10 $DSA(G, C) = \{v_2, v_3, v_{13}\}$.

3.6.3 Relatively Labeled Articulation Points

We now address relatively labeled articulation points. Given a demarcated component C in $Demarcate(G)$ with and a vertex a such that $a \in SA(G, C)$, we say a is *weak* with respect to C if $Follow(G, C, a)$ has one member and that single member is a path. Otherwise a is *strong* with respect to C. If a is strong with respect to C and there exists some

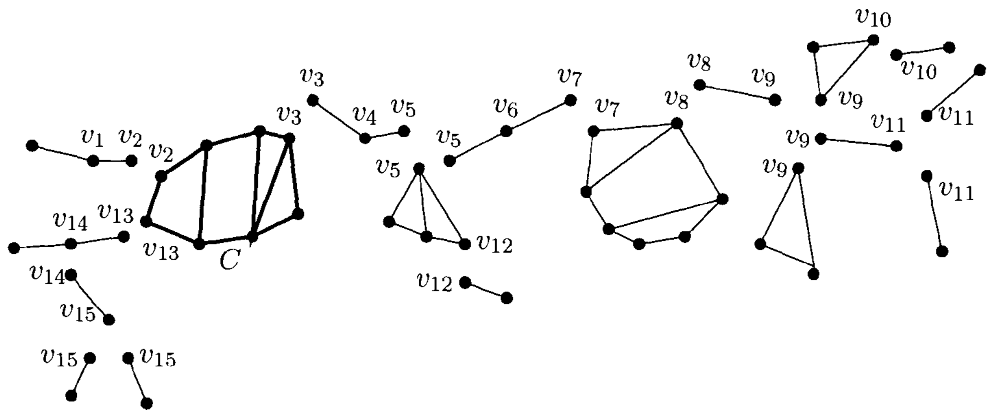


Figure 3.10: The graphs of *Demarcate*(*G*) with a demarcated component *C* identified

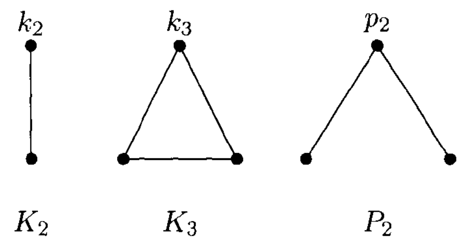


Figure 3.11: Vertex labeling of K_2, K_3 and P_2

$G' \in Follow(G, C, a)$ such that there is a cycle as subgraph of G' we say that a is also *complete* with respect to C, otherwise we say it is *split* with respect to G'. In G from figure 3.9 with the identified demarcated component C from figure 3.10, we can see that v_2 is weak with respect to C, v_{13} is strong and split with respect to C, and v_3 is strong and complete with respect to C.

3.7 Compound Operations on Relatively Labeled Articulation Points

Now that we have defined relative articulation points we will use the operations defined earlier to produce a set of compound operations that will be used frequently in the proofs bellow. Initially we take K_2 and K_3, the complete graphs on two and three vertices respectively, and label one vertex on each with k_2 and k_3 respectively. The choice of vertex is not important as both are vertex transitive. Additionally we take P_2, the path on three vertices, and give the vertex with degree two the label p_2. This is shown in figure 3.11. With these labelings we can define the following for a graph G with a demarcated component C and a significant articulation point a such that $a \in DSA(G, C)$.

3.7.1 $Replace_{K_2}$

$Replace_{K_2}(G, C, a) = Join(Prune(G, C, a), K_2, a, k_2, k_2)$. It follows from Lemma 3.5.2 that if a is a significant articulation point, strong or weak, with respect to C in G then $Replace_{K_2}(G, C, a) \leq_c G$.

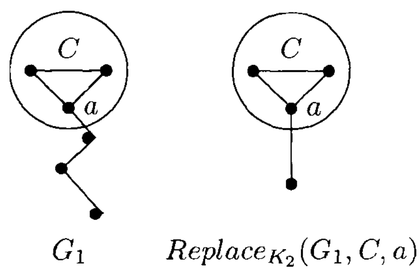


Figure 3.12: Example of $Replace_{K_2}$

3.7.2 $Replace_{P_2}$

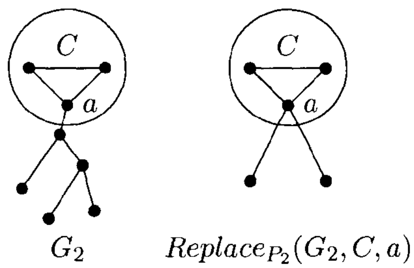


Figure 3.13: Example of $Replace_{K_3}$

$Replace_{P_2}(G, C, a) = Join(Prune(G, C, a), P_2, a, p_2)$. It follows from Lemma 3.5.2 that if a is strong and split with respect to C in G then $Replace_{P_2}(G, C, a) \leq_c G$.

3.7.3 $Replace_{K_3}$

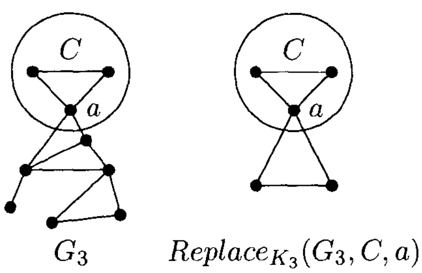


Figure 3.14: Example of $Replace_{P_2}$

$Replace_{K_3}(G, C, a) = Join(Prune(G, C, a), K_3, a, k_3)$. It follows from Lemma 3.5.2 that if a is strong and complete with respect to C then $Replace_{K_3}(G, C, a) \leq_c G$ and $Replace_{K_3}(G, C, a) \leq_c G$.

3.8 Plans and Outerplanar Completion

Lastly, we introduce a notational convenience that will save a great deal of space. An *outerplanar plan*, referred to from here on simply as a *plan*, is an outerplanar graph such that the edges are one of two types either *static* or *potential*. Graphically the static edges will be represented as solid lines and potential edges as dashed lines, an example of which

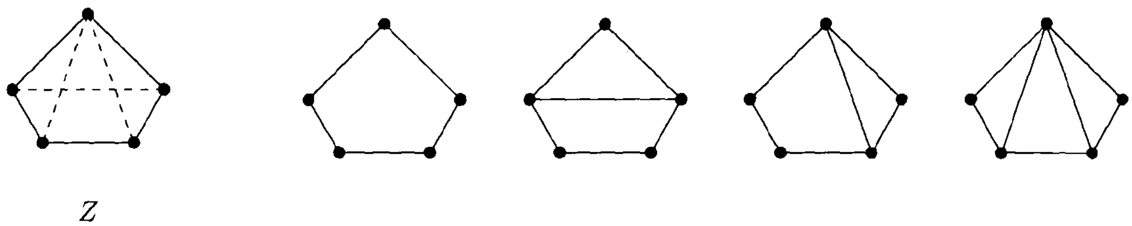


Figure 3.15: The plan Z and its outerplanar completions

is given in figure 3.15. If for a plan Z such that $E_S \subseteq E(Z)$ are the static edges and $E_P \subseteq E(Z)$ are the potential edges, the outerplanar completions are all outerplanar graphs Z' such that $V(Z') = V(Z)$, $E_S \subseteq E(Z')$ and $E(Z') \subseteq E_S \cup E_P$. We will us $Complete(Z)$ to represent all the outerplanar completions of plan Z.

3.9 A Note on the Obstructions

The contraction obstructions for graphs with $cms \leq 1$ ($\mathbb{O}_1$) and $cms \leq 2$ ($\mathbb{O}_2$) can be found in Appendices A and B respectively. The proofs in presented in the next two chapters rely heavily on the fact that each obstruction presented has a certain connected mixed search number. For each $G_1 \in \mathbb{O}_1$ and $G_2 \in \mathbb{O}_2$ it has been verified, both mechanically and by inspection, that $cms(G_1) = 2$ and $cms(G_2) = 3$ and for all $H \leq_c G_1$, $H \neq G_1$ that $cms(H) \leq 1$ and for all $H \leq G_2$, $H \neq G_2$ that $cms(H) \leq 2$.

Chapter 4

Graphs with Connected Mixed Search Number One

While the results in the section are not particularly hard to derive, we present them as an overview of the techniques that we will be using to tackle the considerably more complex structures in the next section.

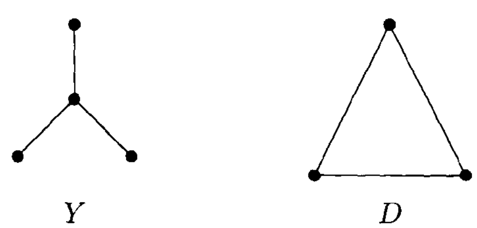


Figure 4.1: The plans for $\mathbb{O}_1$

The plans for $\mathbb{O}_1$, the obstructions in the contraction order for all graphs with $cms \leq 1$, are presented in Figure 4.1.

Lemma 4.0.1. *Let $\mathbb{T}$ be all graphs T such that $D \not\leq_c T$. Then $\mathbb{T}$ is the family of trees.*

Proof. It is well known that K_3 is the obstruction in the minor order for trees. This can be observed by considering that if a G is not a tree then it must contain a cycle C as a subgraph, and a finite series of edge contractions will always reduce any cycle to K_3. If G is a tree then every connected subgraph of G is also a tree and so such C can exist.

If $K_3 \leq_m H$ by theorem 3.3.1 there must exist a K' such that $V(K') = V(K_3)$ and $E(K_3) \subseteq E(K')$. Since K_3 is complete it follows the $E(K') = E(K_3)$ and K' is isomorphic to K_3. Thus K_3 is a contraction obstruction to the family of trees. Suppose there is a contraction obstruction A to the family of trees such that A is non-isomorphic to K_3. It follows that A must not be a tree and thus $K_3 \leq_m A$ and so by the logic above $D \leq_c A$. □

Lemma 4.0.2. *Let $\mathbb{P}$ be all graphs, $P \in \mathbb{T}$ such that $Y \nleq_c P$. Then $\mathbb{P}$ is the family of paths.*

Proof. Suppose not. Therefore there would exist a tree T' such that T' is not a path and $Y \nleq_c T'$. Since T' is not a path there must exist a vertex $a \in V(T')$ such that a has a degree greater than two. Thus $a \in SA(T')$. Let P_1 and P_2 be paths in T' such that a is the endpoint of each path, and $V(P_1) \cap V(P_2) \cup \{a\}$. Let T'' be the subgraph of T' induced by the vertices $V(P_1) \cup V(P_2)$. Let $P_a = Replace_{K_2}(T', T'', a)$. Since a is either strong and split or weak with respect to T'' then $P_a \leq_c T'$. Let P'_a be the subgraph of P_a induced by the vertices $(V(P_a)) \backslash V(P_1)) \cup \{a\}$. Let $P_b = Replace_{K_2}(P_a, P'_a, a)$. Since a is either strong and split or weak with respect to P_a then $P_b \leq_c T'$. Let P'_b be the subgraph of P_b induced by the vertices $(V(P_b) \backslash V(P_2)) \cup \{a\}$. Let $P_c = Replace_{K_2}(P_b, P'_b, a)$. Since a is either strong and split or weak with respect to P_b, $P_c \leq_c T'$. From the above we can see that $Y \leq_c P_c$ which is a contradiction. □

Lemma 4.0.3. *There exists a valid connected mixed search strategy for paths using only 1 searcher.*

Proof. For any path P let $p_1, \ldots, p_n$ be all the vertices of P such that p_1 and p_n are the end points of P and for each p_i, $1 < i < n$, p_1 is adjacent to p_{i-1} and p_{i+1}. Define a search strategy $S = (A_0, Z_0), \ldots, (A_n, Z_n)$ as follows: $A_0 = Z_0 = \emptyset$, $A_1 = \emptyset$, $Z_1 = \{p_1\}$ and $Z_j = p_j$, $A_j = A_{j-1} \cup \{(p_{j-1}, p_j)\}$ for all i, $1 < i \leq n$. Effectively a single searcher begins at one end of the path and 'slides' from edge to edge until the other endpoint is reached. A quick inspection will show S to be valid for P, and since for any non-empty graph at least one searcher is required, it can be concluded that $cms(P) = 1$. □

Theorem 4.0.4. *The graphs in $\mathbb{O}_1$ comprise the entire set of contraction obstruction for graphs with $cms \leq 1$.*

Proof. Lemmas 4.0.1, 4.0.2 and 4.0.3 show that all graphs that forbid a member of $\mathbb{O}_1$ have $cms \leq 1$. Suppose there was a graph G and an $H \in \mathbb{O}_1$ such that $cms(G) \leq 1$ and $H \leq_c G$. For every $H' \in \mathbb{O}_1$ $cms(H') = 2$, and so by Theorem 3.1.1 $cms(G) \geq 2$, giving a contradiction and proving the theorem. □

Chapter 5

Graphs with Connected Mixed Search Number Two

We now proceed to the main result, the contraction obstructions for $cms \leq 2$.

In order to simplify the proof, we will define a number of intermediate graph families. Each family will be the result of forbidding a group of graphs, therefore restricting the previous family. The graphs forbidden in each step will comprise the totality of $\mathbb{O}_2$, the plans for the obstructions will be given as part of the proofs and the complete set of obstructions themselves are given in Appendix B. Finally we will show that each graph from the most restricted family, with all members of $\mathbb{O}_2$ excluded as contraction, has $cms \leq 2$ and that $\mathbb{O}_2$ is complete.

5.1 Outerplanar Graphs

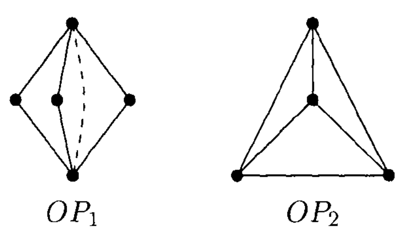


Figure 5.1: Plans for the contraction obstructions for the family of outerplanar graphs

Let $\mathbb{OP}$ be the family of graphs such that for all $G \in \mathbb{OP}$ there does not exist an H such that $H \in Complete(OP_1) \cup Complete(OP_2)$ and $H \leq_c G$.

Lemma 5.1.1. *The family of outerplanar graphs is exactly* $\mathbb{OP}$.

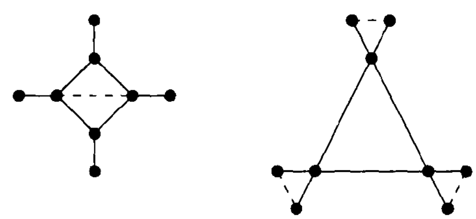
Proof. It is known that K_4 and $K_{2,3}$ are the minor obstructions for outerplanar graphs [5]. Since the family of outerplanar graphs is minor closed then there must exist a finite set of obstructions in the contraction order by Corollary 3.3.2. By following the procedure in section 3.4 the set of contraction obstructions, $\mathbb{O}_{\mathbb{OP}}{}^C$, to the family of outerplanar graphs can be obtained. Notice that, for any graph H, $H \in \mathbb{O}_{\mathbb{OP}}{}^C$ if and only if $H \in Complete(OP_1) \cup Complete(OP_2)$. □

5.2 Family Alpha

We define α to be the set of graphs G such that $G \in \mathbb{OP}$ and for each $G' \in Demarcate(G)$ one of the following holds:

path G' is a path.

biconnected component G' is biconnected, $|DSA(G, G')| \leq 3$ and if $DSA(G, G') = 3$ there must exist an $a \in DSA(G, G')$ such that a is weak with respect to G'.



FW *TS*

Figure 5.2: Plans for the contraction obstructions for family α

Lemma 5.2.1. *For any graph* $G \notin \alpha$ *implies there exists an* $O \in Complete(FW) \cup Complete(TS)$ *such that* $O \leq_c G$.

Proof. Notice that by the definition of *Demarcate* and the nature of outerplanar graphs that any $C \in Demarcate(G)$ is either a path or is biconnected.

Assume there is a graph $G \notin \alpha$ such that there does not exist an $O \in Complete(FW) \cup Complete(TS)$ such that $O \leq_c G$. Then there must exist a $C \in Demarcate(G)$ such that C is biconnected and violates the conditions given for family α. One of the follow conditions occurs:

Case 1 $DSA(G,C) > 3$. Let $\{a_1,\ldots,a_n\} = DSA(G,C)$, $n > 3$, such that $D = DSA(G,C)$ be vertices such that they occur in order in $Outer(C)$. We can then apply the following operations to G:

- Let $G^1_0 = G$. If $n > 4$ then let $G^1_i = Prune(G^1_{i-1}, C, a_{i+4})$, $1 \leq i \leq n-4$. *(Remove all but 4 subgraphs that follow a demarcated articulation point of C)*
- Let $G^2_0 = G^1_{n-4}$. For $1 \leq i \leq 4$ let $G^2_i = Replace_{K_2}(G^2_{i-1}, a_i)$. *(Replace each remaining following subgraph with an edge)*
- Let $G^3_0 = G^2_4$. $1 \leq i \leq 4$ let $G^3_i = Contract(G^3_{i-1}, V(Span(G, C, a_i, a_{(i \bmod 4)+1}, a_{(i \bmod 4)+2})) \cup \{a_i\}, a_i)$. *(Contract C so that Outer(C) has only 4 vertices)*

The operations guarantee $G^3_4 \leq_c G$ and that $G^3_4 \in Complete(FW)$.

Case 2 $DSA(G,C) = 3$ **and for each** $a' \in DSA(G,C)$**,** a' **is strong** ***wrt*** C. Let $\{a_1,\ldots,a_n\} = DSA(G,C)$. We apply the following operations:

- Let $G^1_0 = G$. For $1 \leq i \leq 3$ let $G^2_i = Replace_{P_2}(G^1_{i-1}, a_i)$ if a_i is split with respect to C and $G^1_i = Replace_{K_3}(G^1_{i-1}, a_i)$ otherwise. *(Replace the following subgraph with either two edges or a K_3 according to the nature of the articulation point)*
- Let $G^2_0 = G^1_3$. $1 \leq i \leq 3$ let

  $$G^2_i = Contract(G^2_{i-1}, V(Span(G, C, a_i, a_{(i \bmod 3)+1}, a_{(i \bmod 3)+2})) \cup \{a_i\}, a_i)$$

  (Contract C so that Outer(C) has only 3 vertices)

The operations guarantee $G^3_4 \leq_c G$ and that $G^3_4 \in Complete(TS)$.

Since all possible cases give a contradiction, the lemma is proved. □

5.3 Fans

For any graph G, a graph $G' \in Split(G,a)$ where $a \in SA(G)$ is called a *fan* if there exists a path P that is a subgraph of G' where $V(P) = V(G')$, a is an endpoint of P, and for all

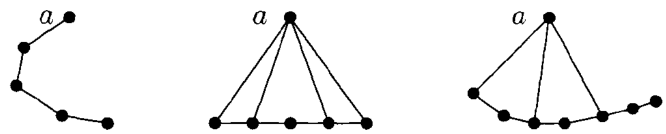


Figure 5.3: Examples of fans

edges $(u, v) \in E(G')$, either $(u, v) \in E(P)$ or $a \in \{u, v\}$.

For any graph G, define $Fans(G)$ to be all $D \in Demarcate(G)$ such that there exists an $a \in SA(G)$ such that D is subgraph of of some fan $G' \in Split(G, a)$. Finally we put a further restriction on the articulation points. Define $NFSA(G)$ such that for a vertex $v \in V(G)$ $v \in NFSA(G)$ if an only if $v \in SA(G)$ and there exits a $C \in Demarcate(G)$ such that $v \in V(C)$ and $C \notin Fans(G)$. Thus $NFSA(G)$ is the set of significant articulation points that are not completely contained within the fans of G.

5.4 Family Beta

We define $\beta \subseteq \alpha$ to be the set of graphs G such that $G \in \alpha$ and for all $a \in SA(G)$ if $|Split(G, a)| \geq 3$ then for any three distinct graphs G_1, G_2 and G_3 such that $\{G_1, G_2, G_3\} \subseteq Split(G, a)$, at least one G_i, $1 \leq i \leq 3$ is a fan.

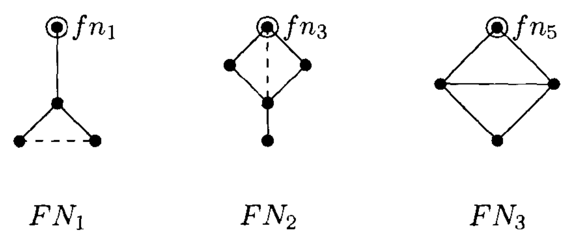


Figure 5.4: Plan used to construct the contraction obstructions for family β

The plans FN_1, FN_2 and FN_3 are given in figure 5.4. Let $FN_i^\star = Complete(FN_i)$, $1 \leq i \leq 3$. We define the family $\mathbb{FN}$ as all the unique graphs produced by selecting three, not necessarily non-isomorphic, graphs G_1, G_2, G_3 from $FN_1^\star \cup FN_2^\star \cup FN_3^\star$ and if $G_i \in FN_j^\star$

then $a_i = fn_j$ and performing the operation $Join(G_1, Join(G_2, G_3, a_2, a_3, a_3)a_1, a_3, a)$.

Lemma 5.4.1. *For any graph $G \in \alpha$, $G \notin \beta$ implies that there exists an $O \in \mathbb{FN}$ such that $O \leq_c G$.*

Proof. First we show that for any graph G with a significant articulation point a that if $G' \in Split(G, a)$ and G' is not a fan then there exists an $F' \in FN_i^\star$, for some $1 \leq i \leq 3$, such that $F' \leq_c^{(fn_i \to a)} G'$. Notice that there is a unique $G'' \in Demarcate(G')$ such that $a \in G$". In other words, G'' is the demarcated component that separates, in G the graph induced by $V(G) \backslash V(G')$ from the graph induced by of $V(G') \backslash V(G'')$.

The cases are as follows:

G'' is a path There must exist an $a' \in (V(G') \cap SA(G))$ such that a, a' are the endpoints of G''. Since $a' \in SA(G)$ then a' is strong with respect to G'' in G'. If a' is split then let $H^1 = Replace_{P_2}(G'', a')$ and let $H^1 = Replace_{K_3}(G'', a')$ otherwise. Next we contract the path to a single edge by letting $H^2 = Contract(G', V(G'') \backslash \{a\}, a')$. By this we can see that $H^2 \leq G'$ and $H^2 \in FN_1^\star$.

G'' is biconnected There are two further cases. Firstly, if there exists an edge $(u, v) \in Inner(G)$ such that $a \notin \{u, v\}$ we can apply the following:

- If $|DSA(G', G'')| > 0$ then let $\{a'_1, \dots a'_n\} = DSA(G', G'')$, $H_0^1 = G'$ and $H_i^1 = Prune(H_{i-1}^1, G', a'_i)$. *(Remove everything from G' but G'').*
- If $G' = G''$ then let $H_0^2 = H_0^1$. Otherwise let $H_0^2 = H_n^1$. Further more let:

$$H_1^2 = Contract(H_0^2, V(Span(H_0^2, u, v, a)), d)$$

$$H_2^2 = Contract(H_1^2, V(Span(H_1^2, u, a, v)) \cup \{u\}, u)$$

and finally

$$H_3^2 = Contract(H_2^2, V(Span(H_2^2, v, a, u)) \cup \{v\}, v)$$

(Contract so that $Outer(G'')$ is a cycle on 4 vertices)

Observe that $H_3^2 \leq_c^{(a \to a)} G''$ and $H_3^2 \in FN_3^\star$. Otherwise if no such edge exists then $|DSA(G', G'')| > 0$, if not then $G'' = G'$, and if we disregard the edges in $Inner(G'')$

and one edge (a, u) for some $u \in V(G'')$ then G'' is a path and all the disregarded edges are of the form (a, v) for some $v \in V(G'')$, and so G' is a fan.

Given this fact there are two possible remaining cases.

There is an $a' \in DSA(G', G'')$ such that a' is strong with respect to G''

We can then apply the following:

- If $|DSA(G', G'')| > 1$ then let $\{a'_1, \dots a'_n\} = DSA(G', G'')\{a'\}$, $H^1_0 = G'$ and $H^1_i = Prune(H^1_{i-1}, a'_i)$. *(Remove everything from G' but G'' and the subgraph that follows a').*
- If $|DSA(G', G'')| = 0$ then let $H^2_0 = H^1_0$, otherwise let $H^2_0 = H^1_n$. If a' is split with respect to G' then let $H^2_1 = Replace_{P_2}(H^2_0, G'', a')$ and let $H^2_2 = Replace_{P_2}(H^2_1, G'', a')$.
- Let $H^3 = Contract(H^2_2, V(G'')\backslash\{a\}, a')$ *(Reduce G'' to the single edge, (a, a'))*

By this we can see that $H^3 \leq_c^{(a \to a)} G'$ and $H^3 \in FN^\star_1$

There is a single vertex $a' \in DSA(G', G'')$ where a' is weak *w.r.t.* G''

It follows that $G''' = Follow(G', G'', a')$ is a path with a' as an endpoint. Note that if (a, a') is an edge in $Outer(G'')$ and we refer to the other endpoint of G''' as p, we can find a path in G'' from a to p by disregarding the edges in $Inner(G'')$ and (a, a'). Since all the disregarded edges are of the form (a, u) for some $u \in (V(G))$ then G' is a fan. So we know that a and a' are non adjacent in $Outer(G'')$ and we can apply the following:

- Let $H^1 = Replace_{K_2}(G', G'', a')$. *(Replace G''' with a single edge)*
- Let $c \in V(G'')$ be a vertex such that $c \notin \{a, a'\}$. Let $H^2_0 = Contract(H^1, V(Span(G'', a, a', c)), c')$ and let $H^2_1 = Contract(H^2_0, V(Span(G'', a, a', c')), c)$. *(Contract so that $Outer(G'')$ is a cycle on 4 vertices).*

Therefore $H^2_1 \leq_c^{(a \to a)} G'$ and $H^2 \in FN^\star_1$.

If $\{a'_1, \dots a'_n\} = DSA(G', G'')$, $n \geq 2$ and each a_i is weak with respect to G''

Once again we apply a set of operations to reach the desired results:

- $H^1_0 = G'$. If $n > 2$ then $H^1_i = Prune(H^1_{i-1}, a'_{i+1})$ for $1 \leq i \leq n-2$. *(Remove everything from G' but G'' and the paths that follow a'_1 and a'_2)*

- If $n = 2$ then let $H_0^2 = H_0^1$. Otherwise let $H_0^2 = H_{n-2}^1$ and let $H_1^2 = Replace_{K_2}(H_0^2, G'', a_1')$ and $H_2^2 = Replace_{K_2}(H_1^2, G'', a_2')$. *(Reduce the remaining paths to single edges)*
- Let $H^3 = Contract(G', V(G'')\backslash\{a\}, a')$ *(Reduce G'' to the single edge, (a, a'), thus bringing together the edges produced in the last step to share a common endpoint)*

By this we can see that $H^3 \leq_c^{(a \to a)} G'$ and $H^3 \in FN_1^\star$

So by the above if we have a graph G with an $a \in SA(G)$ such that there are three graphs $\{G_1, G_2, G_3\} \subseteq Split(G, a)$ where no G_i is a fan, we can show that there exists an $O \in \mathbb{FN}$ such that $O \leq_c G$. Let G' be the graph induced by the vertices $V(G_1) \cup V(G_2) \cup V(G_3)$. We isolate these three graph by letting $H^1 = Prune(G, G', a)$. By the above for each G_i there exists an $G_i' \in FN_j^\star$ for some i, j, $1 \leq i \leq 3$ and $1 \leq j \leq 3$ such that $G_i' \leq_c^{(fn_j \to a)} G_i$. If $G_i \in FN_j$ then let $a_i = fn_j$ and $G'' = Join(G_1, Join(G_2, G_3, a_2, a_3, a_3)a_1, a_3)$. Then $G'' \leq_c G' \leq_c$, and since G'' must be in $\mathbb{FN}$ the lemma is proved. □

5.5 Entrance and Exit Vertices

The final step in this series of lemmas will give a search strategy for the final family. An important observation can be made at this point. The following will be a general explanation in order to provide motivation to the reader; the details will be provided as the proofs progress. The construction of family α dictates that a demarcated component can only have at most two strong articulation points and family β dictates that any significant articulation point can have at most two non-fans that follow it. Observe that the graphs that lead from weak articulation are paths. So if we consider a graph G from family β (the families defined below will all be subsets of β), and we disregard all the paths that lead from weak articulation points and the fans of G, we will be left with a 'string' of biconnected subgraphs, connected to each other, arranged in a 'path-like' manner with one demarcated component in this reduced graph connected to at most two others. The final search strategy will begin at one 'end' of this path and search to the other. This allows us to order the vertices in $NFSA(G)$.

If we start with a demarcated component with only one vertex in $NFSA(G)$, (the above implies that there are at most two) we call this vertex the first. If we examine demarcated components that share this 'first' vertex with our initially selected component, it will contain at most two members of $NFSA(G)$. One of them is the already defined 'first' vertex and the other, if it exits, will be 'second'. By repeating this we progress along our reduced graph until we have ordered all of $NFSA(G)$. This is the order that the members of $NFSA(G)$ will be visited as the search progresses. Notice though that there were possibly two candidates for the initial component. This suggests that there may be two orderings of the vertices of $NFSA(G)$ in this way. We will show that for some graphs with $cms \leq 2$ both orderings are valid, but not for all. Some demarcated components can only be searched as part of the larger graph if their two strong articulation points are encountered in a specific order. The following will define family γ, a further refinement on family β. This further refinement will place additional restrictions on the biconnected components, additionally it will define *Entrance* and *Exit* for each of these components. These definitions are not necessary for the definition of family γ, but they are crucial to the definition of the following family, family δ. To avoid redundancy we will introduce them as part of the definition of family γ.

Formally, *Entrance* is a mapping of the demarcated components $C \in Demarcate(G)$ to the members of $NFSA(G)$. However it is neither necessarily injective or surjective and may be undefined. *Exit* is defined in exactly the same manner.

5.6 Family Gamma

We define $\gamma \subseteq \beta$ to be the set of graphs G such that $G \in \beta$ and for each $C \in Demarcate(G)$ either C is a path in $Fans(G)$, or is biconnected and meets the conditions described below.

For each C that is not a path and not in $Fans(G)$ there must exist two vertex disjoint subpaths P_x and P_y of $Outer(G)$ with endpoints x_1,x_2 and y_1,y_2 respectively, such that (x_1, y_1) and (x_2, y_2) are edges in $Outer(C)$, and the sets $V(P_x)$ and $V(P_y)$ form a bipartition of $Inner(C)$. Depending on the nature of the demarcated component, there may be further restrictions on P_x and P_y. The complete set of restrictions is as follows:

$V(C)$ contains no significant articulation points that are strong *w.r.t.* C

For all cases where $V(C)$ has no significant articulation points that are strong with respect to C both $Entrance(C)$ and $Exit(C)$ are undefined. Let $W = V(C) \cap NFSA(G)$.

If W is non-empty then we refer to the vertices of W as $w_1, \ldots, w_{|W|}$. There are further restrictions depending on the number of significant articulation points in $V(C)$ that are weak with respect to C. They are as follows:

$|W| = 0$ There are no further restriction on P_1 or P_2. .

$|W| = 1$ In this case $w_1 \in \{x_1, x_2\}$.

$|W| = 2$ In this case $w_1 \in \{x_1, x_2\}$ and $w_2 \in \{y_1, y_2\}$.

$|W| = 3$ At least two members of W must be adjacent in $Outer(C)$. Without loss of generality say $(w_1, w_2) \in E(C)$. Additionally, $x_1 = w_1$, $y_1 = w_2$ and $w_3 \in \{x_2, y_2\}$.

$V(C)$ contains one significant articulation point s_1 which is strong *w.r.t.* C Let $W = V(C) \cap NFSA(G)$. If W is non-empty then we label the vertices $w_1, \ldots, w_{|W|}$. There are further restrictions depending on the cardinality of W, which are as follows:

$|W| = 0$ In this case $x_1 = s_1$ and both $Entrance(C)$ and $Exit(C)$ are undefined.

$|W| = 1$ If w_1 and s_1 are non-adjacent in $Outer(C)$ then $s_1 \in \{x_1, x_2\}$, $w_1 \in \{y_1, y_2\}$, $Entrance(C)$ and $Exit(C)$ are undefined. Otherwise either $V(P_x) = \{s_1, w_1\}$ and both $Entrance(C)$ and $Exit(C)$ are undefined or $w_1 = x_1$ and $s_1 = y_1$, $Exit(G, C) = s_1$ and $Entrance(C)$ is undefined.

$|W| = 2$ At least one pair of the significant articulation points in $V(C)$ must be adjacent in $Outer(C)$. If (w_1, w_2) is in $E(Outer(C))$ and both w_1 and w_2 are non-adjacent to s_1 in $Outer(C)$ then $\{x_1, x_2\} = \{w_1, s_1\}$, $w_2 \in \{y_1, y_2\}$, $Entrance(C) = s_1$ and $Exit(C)$ is undefined. Otherwise, without loss of generality, say s_1 and w_1 are adjacent in $Outer(C)$ and there are four further cases:

w_2 is non-adjacent to both s_1 and w_1 in $Outer(C)$ The restrictions are that $s_1 = x_1$, $w_1 = y_1$, $w_2 \in \{x_2, y_2\}$, $Exit(C) = s_1$ and $Entrance(C)$ is undefined.

w_2 is adjacent to both s_1 and w_1 in $Outer(C)$ In this case C is a K_3 and $V(P_x) = \{s_1, w_1\}$, $V(P_2) = \{w_1\}$ and both $Entrance(C)$ and $Exit(C)$ are undefined.

w_2 is adjacent to s_1 in $Outer(C)$ Either $V(P_x) = \{s_1, w_1\}$ or $V(P_x) = \{s_1, w_2\}$, $Exit(C) = s_1$ and $Entrance(C)$ is undefined.

w_2 is adjacent to w_1 in $Outer(C)$ The restrictions in this case are that either $V(P_x) = \{s_1, w_1\}$, $Entrance(G,C) = s_1$ and $Exit(G,C)$ is undefined or $V(P_x) = \{w_1, w_2\}$, $Exit(G,C) = s_1$ and $Entrance(G,C)$ is undefined. However if either choice of $V(P_x)$ leads to a valid partitioning then both $Entrance(G,C)$ and $Exit(G,C)$ are undefined.

$\{s_1, s_2\} \subseteq DSA(G,C)$ such that s_1 and s_2 are both strong with respect to C Since all members of family β are members of family α then we know that there does not exist an s_3 where $s_3 \in V(C) \cap DSA(G)$, s_3 is strong with respect to C and $s_3 \notin \{s_1, s_2\}$ Once again, let $W = V(C) \cap NFSA(G)$. Either $|W| = 0$ or $|W| = 1$ and refer the single vertex in W as w_1.

$|W| = 0$ If s_1 and s_2 are non-adjacent in $Outer(C)$ then $s_1 \in \{x_1, x_2\}$ and $s_2 \in \{y_1, y_2\}$ and both $Entrance(C)$ and $Exit(G)$ are undefined. Otherwise s_1 and s_2 are adjacent in $Outer(C)$ and $V(P_x) = \{s_1, s_2\}$ both $Entrance(C)$ and $Exit(G)$ are undefined.

$|W| = 1$ In this case w_1 must be adjacent in $Outer(C)$ to at least one of s_1 or s_2. Without loss of generality say that w_1 is adjacent to s_1. This leads to a further four cases to consider:

s_2 is non-adjacent to both w_1 and s_1 in $Outer(C)$ In this case the restrictions are such that $s_1 = x_1$, $w_1 = y_1$, $s_2 \in \{x_2, y_2\}$, $Exit(C) = s_1$ and $Entrance(C)$ is undefined.

s_2 is adjacent to both w_1 and s_1 in $Outer(C)$ Once again this implies that C is a K_3 and the restrictions are that $V(P_x) = \{s_1, s_2\}$, $V(P_y) = \{w_1\}$ and both $Entrance(C)$ and $Exit(C)$ are undefined.

s_2 is adjacent to s_1, but not to w_1 In this case the restrictions are such that $V(P_x) = \{s_1, s_2\}$, $Entrance(C) = s_2$ and $Exit(C) = s_1$.

s_2 is adjacent to w_1, but not to s_1 It is the case that either $V(P_x) = \{s_1, w_1\}$ and $Entrance(C) = s_1$, $Exit(C) = s_2$ or $V(P_x) = \{s_2, w_1\}$, $Entrance(C) = s_2$ and $Exit(C) = s_1$. However if either choice of $V(P_x)$ leads to a valid partitioning then $Exit(C) = w_1$ and $Entrance(C)$ is undefined.

The plans for the proposed obstructions are given in figure 5.5.

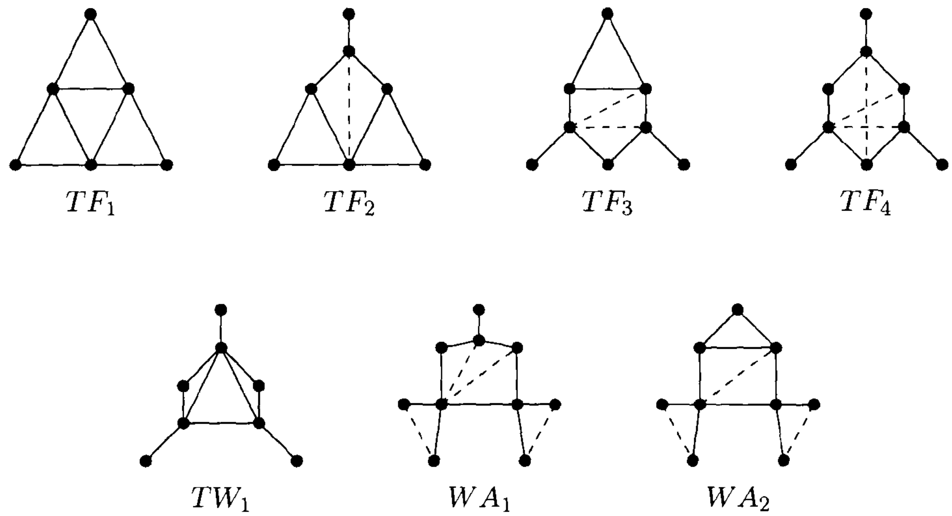


Figure 5.5: Plans for the contraction obstructions of family γ

Lemma 5.6.1. *For any graph G in family β, if G is not in family γ it implies that there exists an $O \in Complete(TF_1) \cup Complete(TF_2) \cup Complete(TF_3) \cup Complete(TF_4) \cup Complete(TW_1) \cup Complete(WA_1) \cup Complete(WA_1)$ such that $O \leq_c G$*

Proof. If G is in β, but not in γ then there must be a $C \in Demarcate(G)$ that violates the conditions given in the definition of γ. We will refer to such a C as violating component. Since $G \in \beta$ then C must be biconnected and outerplanar.

Recall that the definition of family γ states that for C there must exist two vertex disjoint subpaths P_x and P_y of $Outer(C)$ with endpoints x_1,x_2 and y_1,y_2 respectively such that (x_1, y_1) and (x_2, y_2) are edges in $Outer(C)$, and the sets $V(P_x)$ and $V(P_y)$ form a bipartition of $Inner(C)$. If C does not meet this criteria then there must exist three vertex disjoint edges $\{(u_1, v_1), (u_2, v_2), (u_3, v_3)\} \subseteq E(Inner(C))$ such that the vertices u_1,v_1 u_2 v_2, u_3 and v_3 occur in the order listed in $Outer(C)$. Since each edge is in $E(Inner(C))$ it follows that $V(Span(C, u_i, v_i, u_{(i \bmod 3+1)}))$ is non-empty for $1 \leq i \leq 3$. If $D = V(C) \cap NFSA(G)$ and if $|D| > 0$, we refer to the members of D as $a_1, \ldots, a_{|D|}$. We can then apply the following operations:

- Let $H_0^1 = G$. If $|D| > 0$ then $H_i^1 = Prune(H_{i-1}^1, C, a_i)$ for all i, $1 \leq i \leq |D|$. *(Remove* *everything but C from G)*

- Let $H_0^2 = H_{|D|}^1$. Let $H_i^2 = Contract(H_{i-1}^2, Span((C, u_i, v_i, u_{i \bmod 3+1})), uv_i)$ for all i, $1 \leq i \leq 3$. *(Contract so there are three K_3s with the vertices u_i, u_i and uv_i)*
- Let $H_0^3 = H_3^2$. Let $H_i^3 = Contract(H_{i-1}^2, V(Span(C, v_i, u_{i+1}, v_{i \bmod 3+1})), vu_i)$ for all i, $1 \leq i \leq 3$. *(Bring the endpoints of the edges together)*

So, $H_3^3 \leq G$ and $H_3^3 \in Complete(TF_1)$.

All demarcated components that contain three significant articulation points have the requirement that at least two of them are adjacent in O_i. If C contains three significant articulation points a_1, a_2 and a_3 such that they are all non-adjacent, we can apply the following operations:

- Let $H_0^1 = G$ and $H_i^1 = Replace_{K_2}(H_{i-1}^1, C, a_i)$ for all i, $1 \leq i \leq 3$. *(Reduce all the graphs that follow C to a single edge)*
- Let $H_0^2 = G$ and $H_i^2 = Contract(H_{i-1}^1, V(Span(C, a_i, a_{(i \bmod 3)+1}, a_{(i \bmod 3)+2})), m_i)$ for all i, $1 \leq i \leq 3$. *(Reduce all the paths in $Outer(C)$ between the significant articulation points to a single vertex)*

Let $M \subseteq \{a_1, a_{2,3}\}$ be the set such that $a_i \in M$ if an only if $(m_{(i-2 \bmod 3)+1}, m_i)$ is an edge in H_3^2. If $|M| = 0$ then $H_3^2 \in Complete(TF_4)$. Otherwise there exists an $O \in Complete(TF_{4-|m|})$ such that $O \leq_c H_3^2$.

If C is not a violating component by the above, C must contain no more than three significant articulation points by it's membership in family β so we enumerate the possible nature and placements of these points:

$V(C)$ contains no significant articulation points that are strong *w.r.t.* C

Let $W = V(C) \cap NFSA(G)$. If W is non-empty then we label the vertices of $w_1, \ldots, w_|W|$. There are further restrictions depending on the cardinality of W, which are as follows:

$|W| = 0$ Family γ puts no further restriction on this case. If C does not fail the test above, it cannot be a violating component.

$|W| = 1$ Family γ requires that $w_1 \in \{x_1, x_2\}$. If C is a violating component, there must exist two distinct edges, (u_1, v_1) and (u_2, v_2) in $E(Inner(C))$ such that $w_1 \notin \{u_1, v_1, u_2, v_2\}$ and assume without loss of generality that u_1, v_1, v_2 and u_2 occur in that order around $Outer(C)$ and in $V(Span(C, u_1, u_2, v_1))$. We then apply the following operations:

- Let $H^1 = Replace_{K_2}(G, C, w_1)$. *(Reduce the path to a single edge)*
- Let $H_1^2 = Contract(H^1, V(Span(C, u_1, v_1, u_2), uv_1))$ and $H_2^2 = Contract(H^1, V(Span(C, u_2, v_2, u_1)), uv_2)$. *Reduce the graph such that are two K_3s on the vertices $\{u_1, v_1, uv_1\}$ and $\{u_2, v_2, uv_2\}$)*
- Let $H^3 = Contract(H^1, V(Span(C, v_1, v_2, u_2)) \cup \{v_1, v_2\}, vv)$ *(Bring together the endpoints of the edges)*
- Let $H_1^4 = Contract(H^3, V(Span(w_1, u_1, u_2)) \cup \{u_1\}), u_1)$ and

$$H_2^4 = Contract(H_1^4, V(Span(w_1, u_2, u_1)) \cup \{u_2\}), u_2)$$

. *(Contract so that u_1 and u_2 are adjacent to w_1)*

By this we know that $H_2^4 \leq_c G$ and $H_2^4 \in Complete(TF_2)$

$|W| = 2$ Family γ requires that $w_1 \in \{x_1, x_2\}$ and $w_2 \in \{y_1, y_2\}$. If C is a violating component then w_1 and w_2 must be non-adjacent in $Outer(C)$. If there exist two edges (u_1, v_1) and (u_2, v_2) such that u_1, v_1, v_2 and u_2 occur in that order around $Outer(C)$ and some w_i, $1 \leq i \leq 2$ occurs in $V(Span(C, u_1, u_2,))$ then for $w_j \neq w_i$ let $G' = Prune(G, C, w_j)$ and the procedure given for the case where $|W| = 1$ apply to G'. Otherwise, in order for C to be a violating component, there must exist a vertex in $v' \in V(C)$ and an edge $(u, v) \in E(Outer(C))$ such that $v' \notin \{w_1, w_2\}$ and $\{u, v\} \subseteq V(Span(C, w_1, w_2, v'))$. Assume without loss of generality that $u \in V(Span(C, w_1, v, w_2))$. Given this we can apply the following operations:

- Let $H_0^1 = Replace_{K_2}(G, C, w_1)$ and $H_1^1 = Replace_{K_2}(H_0^1, C, w_2)$. *(Reduce the paths to single edges)*
- Let $H^2 = Contract(H_1^1, V(Span(C, w_1, w_2, u_2), v'))$ *(Reduce the subpath containing v' to a single vertex)*
- Let $H_0^3 = Contract(H^2, V(Span(C, w_1, u, v')) \cup \{u\}, u))$ and

$$H_1^3 = Contract(H_0^3, V(Span(C, w_2, v, v')) \cup \{v\}, v))$$

. *(Make w_1 and w_2 adjacent to u and v)*

- Let $H^4 = Contract(H_1^3, V(Span(C, u, v, v')), uv))$ *(Reduce the graph so it contains a K_3 on the vertices u, v and uv)*

So $H^4 \leq_c G$ and $H^4 \in Complete(TF_4)$.

$|W| = 3$ Recall that at least two of the significant articulation points, without loss of generality say w_1 and w_2, must be adjacent in $Outer(C)$. If w_3 is adjacent in $Outer(C)$, C is a K_3 and a partitioning is always possible. So for C to be a violating component either w_1 or w_2 is non-adjacent to w_3. Family γ requires that $w_1 = x_1$ and $w_2 = y_1$ and $w_3 \in \{x_2, y_2\}$. As in the previous case if there exist two edges (u_1, v_1) and (u_2, v_2) such that u_1, v_1, v_2 and u_2 occur in that order around $Outer(C)$ and some w_i, $1 \leq i \leq 3$ occurs in $V(Span(C, u_1, u_2,))$ then for w_j, w_k where $i \neq j \neq k$ let $G_0' = Prune(G, C, w_j)$ and $G_1' = Prune(G_0', C, w_k)$ and the procedure given for the case where $|W| = 1$ applies to G_1'. If there is an edge $(u, v) \in E(Outer(C))$ and $\{u, v\} \not\subseteq \{w_1, w_2, w_3\}$ and for some w_i, $1 \leq 2$ $(u, v) \in E(Span(C, w_i, w_3, v'))$ for some $v' \in V(C)$ and $v' \notin \{w_1, w_2, w_3$. If we let $G' = Prune(G, C, w_j)$ for some $1 \leq j \leq$, $j \neq i$ then the operations given in the case where $|W| = 2$ apply to G'. Otherwise in order for C to be a violating component there must be the edges (w_1, w_3) and (w_2, w_3) in $E(Inner(C))$. We can then apply the following:

- Let $H_0^1 = Replace_{K_2}(G, C, w_1)$, $H_1^1 = Replace_{K_2}(H_0^1, C, w_2)$ and $H_2^1 = Replace_{K_2}(H_1^1, C, w_3)$. *(Reduce the paths to single edges)*
- Let $H_0^2 = Contract(H_2^1, V(Span(C, w_1, w_2, w_3)), w_1')$ and $H_1^2 = Contract(H_0^2, V(Span(C, w_1, w_3, w_2)), w_2')$ *(Reduce the graph so it contains two K_3s on the vertices w_1, w_2, $w_1'andw_1$, w_3, w_2')*

By the above $H_1^2 \leq G$ and $H_1^2 \in Complete(TW_1)$.

$V(C)$ contains one significant articulation point, s_1, that is strong *w.r.t.* C

Let $W = V(C) \cap NFSA(G)$. If W is non-empty then we label the vertices of $w_1, \ldots, w_|W|$. There are further restrictions depending on the cardinality of W, which are as follows:

$|W| = 0$ Recall that s_1 must be in $\{x_1, x_2\}$ as an endpoint. Therefore, if we let $G' = Replace_{K_2}(G, C, s_1)$ then G' has the same criteria as if C has one weak significant articulation point. So if C is a violating component then G' must contain a violating component and the operations given above show that one of given graphs is a contraction of G.

$|W| = 1$ Family γ requires that $s_1 \in \{x_1, x_2\}$, $w_1 \in \{y_1, y_2\}$ or that $V(P_x) = \{s_1, w_1\}$. If $G' = Replace_{K_2}(G, C, s_1)$ then the case where C contains two significant articulation points that are weak with respect to C applies to G'. So if C is violating component in G then G' contains a violating component and the above gives the operations to obtain one of the given graphs.

$|W| = 2$ If C is a violating component and w_1 and w_2 are adjacent and both are non-adjacent to s_1 in $Outer(C)$ then it has the same conditions in $Inner(C)$ as a violating component with three weak articulation points. Again, if we let $G'_1 = Replace_{K2}(G, C, a_1)$ then if C is a violating component then G' will have a violating component and the above steps detail the given graph that will be derived. Otherwise, as the case with three non-adjacent significant articulation points has been covered above, without loss of generality, say s_1 and w_1 are adjacent in O_{C_i}, there are four cases:

w_2 is non-adjacent to both s_1 and w_1 in $Outer(C)$ Recall that the restrictions state that $s_1 \in \{x_1, x_2\}$ and $w_1 \in \{y_1, y_2\}$. As above, the conditions of in $Inner(C)$ that would make C a violating component are identical to the case where $V(C)$ contains three significant articulation points that are week with respect to C. Once again, let $G'_1 = Replace_{K2}(G, C, a_1)$, and if C is a violating component then G' will have a violating component and one of the graphs given in the statement of the theorem will be a contraction of G.

w_2 is adjacent to both s_1 and w_1 in $Outer(C)$ The adjacencies imply that C is a K_3, so it is impossible for C to be a violating component.

w_2 is adjacent to s_1, but not adjacent to w_1 in $Outer(C)$

Family γ requires that $V(P_x) = \{s_1, w_1\}$ or $V(P_x) = \{s_1, w_2\}$. If C is a violating component then there is an edge $(u, v) \in E(Inner(C))$ such that $\{u, v\} \subseteq Span(C, w_1, w_2, s_1)$. Without loss of generality, we can assume that $v \notin Span_E(C_i, w_1, u, s_1)$. We can then apply the following steps:

- Let $H^1 = Prune(G, C, s_1)$. *(Remove the subgraph that follows s_1)*

- Let $H_0^2 = Replace_{K_2}(H^1, C, w_1)$ and $H_1^2 = Replace_{K_2}(H_0^2, C, w_2)$ *(Reduce the paths to single edges)*
- Let $H^3 = Contract(H^2, V(Span(u, v, w_1)), uv)$. *(Reduce the graph so it contains a K_3 on the vertices u, v and uv)*
- Let $H_0^4 = Contract(H^3, V(Span, w_1, u, w_2)) \cup \{u\}, u)$ and

$$H_1^4 = Contract(H_0^4, V(Span, w_2, v, w_1)) \cup \{v\}, v)$$

. *(Make w_1 adjacent to u and w_2 adjacent v)*

Thus $H_1^4 \leq G$ and $H_1^4 \in Complete(TF_3)$.

w_2 is adjacent to w_1, but not adjacent to s_1 in $Outer(C)$ Recall that there are two possible partitionings allowed for this case in family γ; however, if C is a violating component there is only one case that will forbid either partitions. If C is a violating component then there exists an edge $(u, v) \in E(Inner(C))$ such that $\{u, v\} \subseteq V(Span(C, s_1, w_2, w_1))$. Without loss of generality assume that $v \notin Span(C, s_1, v_1, w_2)$. The following operations can then be applied:

- If s_1 is split with respect to C then let $H^1 = Replace_{P_2}(G, C, s_1)$. Otherwise, let $H^1 = Replace_{K_2}(G, C, s_1)$. *(Reduce the subgraph that follows s_1)*
- Let $H_0^2 = Replace_{K_2}(H^1, C, w_1)$ and $H_1^2 = Replace_{K_2}(H_0^2, C, w_2)$ *(Reduce the paths to single edges)*
- Let $H^3 = Contract(H^2, V(Span(u, v, w_1)), uv)$. *(Reduce the graph so it contains a K_3 on the vertices u, v and uv)*
- Let $H_0^4 = Contract(H^3, V(Span, w_1, u, w_2)) \cup \{u\}, u)$ and

$$H_1^4 = Contract(H_0^4, V(Span, w_2, v, w_1)) \cup \{v\}, v)$$

. *(Make w_1 adjacent to u and w_2 adjacent v)*

- Let $H^5 = Contract(H^3, \{w_1, w_2\}, ww)$. *(Bring the two single edges together so they share an endpoint)*

Thus $H^5 \leq G$ and $H_1^4 \in Complete(WA_2)$.

$\{s_1, s_2\} \subseteq V(C) \cap DSA(G, C)$ such that s_1 and s_2 are both strong *w.r.t.* C

Let $W = V(C) \cap NFSA(G)$. If W is non-empty then we label the vertices of

$w_1, \ldots, w_{|W|}$. There are further restrictions depending on the cardinality of W, which are as follows:

$|W| = 0$ If s_1 and s_2 are non-adjacent in $Outer(C)$ the conditions in $Inner(C)$ that make C a violating component are identical to the case where C is a violating component where $V(G)$ contains exactly with two weak articulation points. Therefore, if we let $G'_0 = Replace_{K_2}(G, C, s_1)$ and $G'_1 = Replace_{K_2}(G'_1, C, s_2)$ then $G' \leq G$ and by the above there exists a graph given in the statement of the theorem that is a contraction of G' and thus G. Otherwise if s_1 and s_2 are adjacent in $Outer(C)$ and if C is a violating component and the partitioning such that $P_x = \{s_1, s_2\}$ then there must exit an edge $(u_v) \in E(Inner(C))$ such that no $\{s_1, s_2\} \cap \{u, v\} = \emptyset$. Assume with out loss of generality that $v \notin V(Span(C, s_1, u, s_2))$. Then we can apply the following:

- If s_1 is split with respect to C then let $H_0^1 = Replace_{P_2}(G, C, s_1)$ and otherwise $H_0^1 = Replace_{K_3}(G, C, s_1)$. Similarly, If s_2 is split with respect to C then let $H_1^1 = Replace_{P_2}(H_0^1, C, s_1)$ and $H_1^1 = Replace_{K_3}(H_0^1, C, s_1)$ otherwise. *(Reduce the graphs that follow s_1 and s_2 from C)*
- Let $H^2 = Contract(H_1^1, V(Span(C, u, v, s_1)), uv)$. *(Reduce the graph so it contains a K_3 on the vertices u, v and uv)*
- Let $H_0^3 = Contract(H^2,\ V(Span(C, s_1, u, s_2)) \cup \{u\}, u)$ and

$$H_1^3 = Contract(H_0^3, V(Span(C, s_2, v, s_1)) \cup \{v\}, v)$$

. *(Contract so that s_1 is adjacent to u and s_1 is adjacent to s_2)*

By the above $H_1^3 \leq_c G$ and $H_1^3 \in Complete(WA_2)$

$|W| = 1$ In the definition of family γ it is stated that w_1 must be adjacent in $Outer(C)$ to one of s_1 or s_2. The condition where all three significant articulation points are non-adjacent has been addressed above so one pair of the significant points must be adjacent if C is still a violating component. If s_1, s_2 are adjacent in $Outer(C)$ and w_1 is non-adjacent to both of them then the following operations apply:

- If s_1 is split with respect to C then let $H_0^1 = Replace_{P_2}(G, C, s_1)$ and otherwise $H_0^1 = Replace_{K_3}(G, C, s_1)$. Similarly, If s_2 is split with respect to C

then let $H_1^1 = Replace_{P_2}(H_0^1, C, s_1)$ and $H_1^1 = Replace_{K_3}(H_0^1, C, s_1)$ otherwise. Let $H_2^1 = Replace_{K_2}(H_1^1, C, w_1)$ *(Reduce the graphs that follow s_1 and s_2 from C and reduce the path that follows C to a single vertex)*

- Let $H_0^2 = Contract(H_2^1, V(Span(C, s_1, w_1, s_2)), u)$ and $H_1^2 = Contract(H_0^2, V(Span(C, s_2, w_1, s_1)), v)$. *(Reduce the graph so there is a pair of vertices u and v such that u is adjacent to both s_1 and w_1 and v is adjacent to s_2 and w_1)*

Thus $H_1^2 \leq_c G$ and $H_1^2 \in Complete(WA_1)$. If w_1 is adjacent to both s_1 and s_2 and s_1 and s_2 are adjacent then C is a K_3 and the given partitioning is always possible and so C cannot be a violating component. If C is a violating component that does not have any of the criteria already covered, it must be the case that w_1 is adjacent to one of the significant articulation points in $V(C)$ that is strong with respect to C, without loss of generality say s_1, and not adjacent to s_2.

If s_2 is non-adjacent to both w_1 and s_1 in $Outer(C)$ then the restrictions given in the definition of family dictate that $s_1 = x_1$, $s_1 = y_1$ and $s_2 \in \{x_2, y_2\}$. If C is a violating component then $Inner(C)$ must be have the same configuration as the case above where $V(C)$ contains three significant articulation points that are weak with respect to C and C is a violating component. Thus if $G_1' = Replace_{K_2}(G, C, s_1)$ and $G_2' = Rep_{K_2}(G_1', C_i, s_2)$ then $G' \leq_c G$, and if G contains component then G' must also contain a violating component, and the above shows that one of the given graphs will be a contraction of G.

If s_2 is adjacent to s_1 and not s_1 in $Outer(C)$, recall that P_x must be $\{s_1, s_2\}$ if C_i is a violating component then the same conditions hold for $Inner(C)$ as the case where C has only two significant articulation points that are both strong with respect to C. Therefore, if C is a violating component, and if $G' = Prune(G, C, w_1)$ then G' must contain a violating component, and the steps above show that there is one of the given graphs that is a contraction of G'. Since $G' \leq_c G$ then the same graph is a contraction of G.

Finally, if s_2 is adjacent to w_1, but not to s_1 in $Outer(C)$ family γ requires that

$P_x = \{s_2, w_1\}$. If this partitioning is impossible for C then there must exist an edge $(u, v) \in E(Inner(C))$ such that $\{u, v\} \subseteq Span(C, s_1, s_2, w_1)$. Without loss of generality, say that $u \in Span_E(C_i, s_1, v, s_2)$. We then apply the following:

- Let $H_0^1 = Prune(G, C, w_2)$, $H_1^1 = Replace_{K_2}(H_0^1, C, s_1)$ and additionally let $H_2^1 = Replace_{K_2}(H_1^1, C, s_2)$. *(Remove the path that follows w_1 from C and reduce the graphs that follow the other two significant articulation points to single edges)*
- Let $H^2 = Contract(H_2^1, V(Span(C, u, v, s_1)), u, v)$. *(Reduce the graph so it contains a K_3 on the vertices u, v and uv)*
- Let $H_0^3 = Contract(H^2,\ V(Span(C, s_1, u, s_2)) \cup \{u\}, u)$ and

$$H_1^3 = Contract(H_0^3, V(Span(C, s_2, v, s_1)) \cup \{v\}, v)$$

. *(Contract so that s_1 is adjacent to u and s_1 is adjacent to s_2)*

So, $H_1^3 \leq G$ and $H_1^3 \in Complete(TF_3)$.

So by examining every configuration of the significant articulation points of C, a violating component in G, we can conclude that if $G \in \beta$ and $G \notin \gamma$, it must have one of the given graphs as contraction. □

We now reach the final family, δ, where we will make use of the definitions of *Entrance* and *Exit* defined in family γ in order to weed out the last of the graphs unsearchable by two searchers.

5.7 Family Delta

We define $\delta \subseteq \gamma$ to be the set of graphs such that $G \in \gamma$ if $G \in \delta$ and for each distinct C and C' in $Demarcate(G)$ then:

(1) If $Exit(C)$ is defined and $Exit(C')$ is defined then either C is not a subgraph of $Follow(G, C', Exit(C'))$, or C' is not a subgraph of $Follow(G, C, Exit(C))$.

(2) If $Entrance(C)$ and $Entrance(C')$ are both defined then either C is not a subgraph of $Follow(G, C', Entrance(C'))$, or C' is not a subgraph of $Follow(G, C, Entrance(C))$.

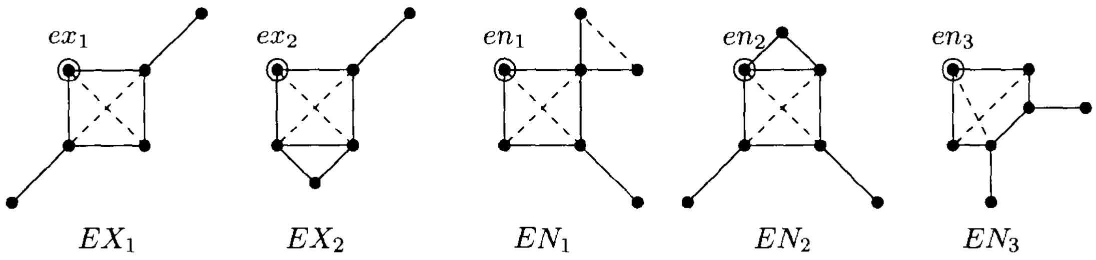


Figure 5.6: Plans used in the construction of the contraction obstructions for family δ

The plans EX_1 and EX_2 are given in figure 5.6. Let $EX_i^\star = Complete(EX_i)$, $1 \leq i \leq 2$. We define the family $\mathbb{EX}$ as all the unique graphs produced by selecting two, not necessarily non-isomorphic, graphs G_1, G_2 from $EX_1^\star \cup EX_2^\star$ and if $G_i \in F_j^\star$ then $a_i = ex_j$ and performing the operation $Join(G_1, G_2, a_1, a_2)$. Similarly, the plans EN_1, EN_2 and EN_3 are also given in figure 5.4. Let $EN_i^\star = Complete(EN_i)$, $1 \leq i \leq 3$. We define the family $\mathbb{EN}$ as all the unique graphs produced by selecting three, not necessarily non-isomorphic, graphs G_1, G_2, G_3 from $EN_1^\star \cup EN_2^\star \cup EN_3^\star$, and if $G_i \in F_j^\star$ then $a_i = en_j$, and by performing the operation $Join(G_1, Join(G_2, G_3, a_2, a_3, a_3)a_1, a_3)$.

Lemma 5.7.1. *For any graph $G \in \gamma$, $G \notin \delta$ implies that there exists an $O \in \mathbb{EN} \cup \mathbb{EN}$ such that $O \leq_c G$.*

Proof. As in the proof for family β we show the for if for a graph G with a demarcated component C such that:

- If $Exit(C)$ is defined and $a = Exit(C)$ then if $G' = Prune(G, C, a)$ there exists an $H \in EX_i^\star$ for some $1 \leq i \leq 2$ such that $H \leq_c^{(ex_i \to a)} G$.
- If $Entrance(C)$ is defined and $a = Entrance(C)$, then if $G' = Prune(G, C, a)$ there exists an $H \in EN_i^\star$ for some $1 \leq i \leq 3$ such that $H \leq_c^{(en_i \to a)} G$.

The following components from family γ have $Exit(C)$ or $Entrance(C)$ defined:

$V(C)$ contains a significant articulation point s_1, which is strong *w.r.t.* C

Let $W = V(C) \cap NFSA(C)$. By the nature of family γ W is non-empty and we label the vertices of $w_1, \ldots, w_{|W|}$. There are further restrictions depending on the cardinality of W, which are as follows:

$|W| = 1$ If w_1 and s_1 are adjacent in $Outer(C)$ then $Exit(C)$ is defined only if the partitioning where $P_1 = \{s_1, w_2\}$ is impossible. In this case $Exit(C)$ is defined to be s_1. The impossibility of the partitioning implies that there exists an edge $(u, v) \in E(Inner(C))$ such that $\{u, v\} \cap \{s_1, w_1 = \emptyset$. Assume without loss of generality that $v \notin V(Span(C, s_1, u, w_1))$. We can then apply the following:

- Let $H^1 = Replace_{K_2}(G, C, w_1)$. *(Replace the path with an edge)*
- Let $H^2 = Prune(H^1, C, s_1)$. *(Remove that graph that follows s_1 from C)*
- Let $H^3 = Contract(H^2, V(Span(C, u, v, s_1)), uv)$. *(Contract so there is K_3s with the vertices u, v and uv)*
- Let $H_0^4 = Contract(H^3, V(Span(C, s_1, u, w_1)) \cup \{u\}, u)$ and

$$H_1^4 = Contract(H_0^4, V(Span(C, w_1, v, s_1)) \cup \{v\}, v)$$

.*(Contract so that s_1 and u are adjacent and w_1 and v are adjacent)*

Then $H_1^4 \leq_c^{(s_1 \to s_1)} G$ and $H_1^4 \in Complete(EX_2)$.

$|W = 2|$ At least one pair of the articulation points must be adjacent in $Outer(C)$. If w_1 and w_2 are adjacent and both are non-adjacent to s_1 in $Outer(C)$ then $Entrance(C)$ is defined to be s_1. In this case the following apply:

- Let $H_0^1 = Replace_{K_2}(G, C, w_1)$ and $H_1^1 = Replace_{K_2}(H_0^1, C, w_1)$. *(Replace the paths with single edges)*
- Let $H^2 = Prune(H_1^1, C, s_1)$. *(Remove that graph that follows s_1 from C)*
- Let $H_0^3 = Contract(H^2, V(Span(C, w_1, s_1, w_2)), ws_1)$ and

$$H^3 = Contract(H_0^3, V(Span(C, w_2, s_1, w_1)), ws_2)$$

. *(Contract so that there are vertices ws_1 and ws_2 so that ws_1 is adjacent in $Outer(C)$ to both w_1 and s_1 and ws_2 is adjacent in $Outer(C)$ to both w_2 and s_1)*

By the above $H_1^3 \leq_c^{(s_1 \to s_1)} G$ and $H_1^3 \in Complete(EN_3)$.

Otherwise, without loss of generality say s_1 and w_1 are adjacent in $Outer(C)$. If w_1 is non-adjacent to w_2 in $Outer(C)$ then $Exit(C) = s_1$ whether w_2 is adjacent to s_1 or not. The operations to show the desired contraction are as follows:

- Let $H_0^1 = Replace_{K_2}(G, C, w_1)$ and $H_1^1 = Replace_{K_2}(H_0^1, C, w_1)$. *(Replace the paths with single edges)*
- Let $H^2 = Prune(H_1^1, C, s_1)$. *(Remove that graph that follows s_1 from C)*
- Let $H^3 = H^2$ if w_1 is adjacent to s_2 in $Outer(C)$, otherwise

$$H^2 = Contract(H^2, V(Span(C, w_1, s_1, w_2)) \cup \{w_1\}, w_1)$$

. *(Ensure that w_1 and s_1 are adjacent in $Outer(C)$)*

- and $H^4 = Contract(H^3, V(Span(C, w_1, w_2, s_1)), ww)$. *(Contract so that there is a vertex ww such that ws is adjacent in $Outer(C)$ to both w_1 and w_2)*

By the above $H^4 \leq_c^{(s_1 \to s_1)} G$ and $H^4 \in Complete(EX_1)$.

Finally, we consider the case where w_2 is adjacent to w_1, but not adjacent in $Outer(C)$. Recall that $Entrance(C)$ is defined only if the chords force the partitioning on C such that $P_x = \{s_1, w_1\}$. In this case $Entrance(C) = s_1$. So if $Entrance(C)$, is defined this implies that there is an edge (v, s_1) in $Inner(C)$ such that $v \in V(Span(C, s_1, w_2, w_1))$. In this case we can apply the following:

- Let $H_0^1 = Replace_{K_2}(G, C, w_1)$ and $H_1^1 = Replace_{K_2}(H_0^1, C, w_1)$. *(Replace the paths with single edges)*
- Let $H^2 = Prune(H_1^1, C, s_1)$. *(Remove that graph that follows s_1 from C)*
- Let $H^3 = Contract(H^2, V(Span(C, v, s_1, w_1)), vs)$. *(Contract so there is K_3s with the vertices v, s_1 and vs)*
- Let $H^4 = Contract(H^3, V(Span(C, v, w_2, w_1)) \cup \{v\}, v)$. *(Make v and w_2 adjacent)*

By the above $H^4 \leq_c^{(s_1 \to s_1)} G$ and $H^4 \in Complete(EN_2)$.

Otherwise, if P_x is forced to be $\{w_1, w_2\}$ then $Exit(C) = w_1$. The following then apply:

- Let $H^1 = Prune(G, C, w_1)$. *(Remove that graph that follows w_1)*
- Let $H_0^2 = Replace_{K_2}(H^1, C, w_2)$ and $H_1^2 = Replace_{K_2}(H_0^2, C, s_1)$. *(Reduce the graphs that follow w_2 and s_1 to single edges)*

- Let $H^3 = Contract(H^3, V(Span(C, w_2, s_1, w_1)), ws)$. *(Contract so there is a vertex ws such that ws is adjacent to both w_2 and s_1)*

So $H^3 \leq_c^{(w_1 \to w_1)} G$ and $H^3 \in Complete(EX_1)$.

$V(C)$ contains s_1, s_2, which are strong significant articulation points *w.r.t.* C

Since every member of family δ is a member of α then s_1 and s_2 can be the only vertices in $V(C)$ that are strong with respect to $V(C)$. In this case, if *Entrance* or *Exit* are defined, $V(C)$ must also contain a significant articulation point w_1 such that w_1 is weak with respect to C. Since G is in family γ we know that w_1 must be adjacent in $Outer(C)$ to one of the significant articulation points that is strong, without loss of generality say s_1.

If s_2 is non-adjacent to w_1 and s_1 in $Outer(C)$ then only $Exit(C)$ is defined and $Exit(C) = s_1$. The follow will produce one of the desired graphs:

- Let $H^1 = Prune(G, C, s_1)$. *(Remove that graph that follows s_1 from C)*
- Let $H_0^2 = Replace_{K_2}(H^1, C, w_1)$ and $H_1^2 = Replace_{K_2}(H_0^2, C, s_2)$. *(Reduce the remaining graphs that follow from significant articulation points to single edges)*
- Let $H^3 = Contract(H_1^2, V(Span(C, s_2, w_1, s_1)), sw)$. *(Contract so there is a vertex sw such that sw is adjacent to s_2 and w_1 in $Outer(C)$)*
- Let $H^4 = Contract(H^3, V(Span(C, s_1, s_2, w_1)) \cup \{s_2\}, s_2)$. *(Make s_1 and s_2 adjacent)*

So $H^4 \leq_c^{(s_1 \to s_1)} G$ and $H^4 \in Complete(EX_1)$.

If s_2 is adjacent to s_1, but not w_2, both *Entrance* and *Exit* are defined. Two sets of operations will be needed to show what is desired. In this case $Exit(C) = s_1$ so we can do the following:

- Let $H^1 = Prune(G, C, s_1)$. *(Remove that graph that follows s_1 from C)*
- Let $H_0^2 = Replace_{K_2}(H^1, C, w_1)$ and $H_1^2 = Replace_{K_2}(H_0^2, C, s_2)$. *(Reduce the remaining graphs that follow from significant articulation points to single edges)*

- Let $H^3 = Contract(H_1^2, V(Span(C, s_2, w_1, s_1)), sw)$. *(Contract so there is a vertex sw such that sw is adjacent to s_2 and w_1 in Outer(C))*

So $H^3 \leq_c^{(s_1 \to s_1)} G$ and $H^3 \in Complete(EX_1)$.

As well in this case $Entrance(C) = s_2$ so we can apply the following:

- Let $H^1 = Prune(G, C, s_2)$. *(Remove that graph that follows s_2 from C)*
- Let $H_0^2 = Replace_{K_2}(H^1, C, w_1)$ and $H_1^2 = Replace_{K_3}(H_0^2, C, s_2)$. *(Reduce the remaining graphs that follow from significant articulation points)*
- Let $H^3 = Contract(H_1^2, V(Span(C, s_2, w_1, s_1)), sw)$. *(Contract so there is a vertex sw such that sw is adjacent to s_2 and w_1 in Outer(C))*

So $H^3 \leq_c^{(s_2 \to s_2)} G$ and $H^3 \in Complete(EN_1)$.

Finally if s_2 is adjacent to w_1 and not to s_1 in $Outer(C)$. Recall that the definition of $Entrance(C)$ and $Exit(C)$ are dependent on whether one of two partitionings are selected; however, the two cases where one partitioning is forced are symmetric so we need only consider the case where $P_x = \{s_1, w_1\}$ and so $Entrance(C) = s_1$ and $Exit(C) = s_2$. The other case follows identical operations except that s_1 and s_2 are exchanged. In this case P_x is forced to be $\{s_1, w_1\}$ which implies that there must be an edge (v, s_1) in $E(Inner(C))$ such that $v \in Span(C, s_1, s_2, w_1)$.

Since $Entrance(C) = s_1$ we can apply the following:

- Let $H^1 = Prune(G, C, s_1)$. *(Remove that graph that follows s_1 from C)*
- Let $H_0^2 = Replace_{K_2}(H^1, C, w_1)$ and $H_1^2 = Replace_{K_2}(H_0^2, C, s_2)$. *(Reduce the remaining graphs that follow from significant articulation points to single edges)*
- Let $H^3 = Contract(H_1^2, V(Span(C, v, s_1, w_1)), vs)$. *(Contract so there is K_3s with the vertices v, s_1 and vs)*
- Let $H^4 = Contract(H^3, V(Span(C, v, s_2, s_1)) \cup \{v\}, v)$. *(Contract so v and s_2 are adjacent in Outer(C))*

So $H^4 \leq_c^{(s_1 \to s_1)} G$ and $H^4 \in Complete(EN_2)$.

Additionally, $Exit(C) = s_2$ so we apply the following:

- Let $H_0^1 = Prune(G, C, s_2)$ and $H_1^1 = Prune(G, C, s_1)$. *(Remove the graphs that follow s_1 and s_2 from C)*
- Let $H^2 = Replace_{K_2}(H_1^1, C, w_1)$. *(Reduce the remaining path with a single edges)*
- Let $H^3 = Contract(H^2, V(Span(C, v, s_1, w_1)), vs)$. *(Contract so there is K_3s with the vertices v, s_1 and vs)*
- Let $H^4 = Contract(H^3, V(Span(C, v, s_2, s_1)) \cup \{v\}, v)$. *(Contract so v and s_2 are adjacent in $Outer(C)$)*

So $H^4 \leq_c^{(s_2 \to s_2)} G$ and $H^4 \in Complete(EX_2)$.

With the above we can easily prove the lemma. We need to consider the two cases where a graph from family γ violates the conditions of family beta. The steps are identical in both cases, only the graphs involve changes.

Let G be a member a family γ with two demarcated components C and C' such that they violate the conditions given form family δ. If $Exit(C)$ and $Exit(C')$ are defined, let G_0' be the graph in $Split(G, Exit(C))$ such that C is a subgraph of G_0', and G_1' be graph in $Split(G, Exit(C'))$ such that C' is a subgraph of G_1'. Let $a = Exit(C)$ and $a' = Exit(C')$. By the above there must exist a H_0^1 such that $H_0^1 \in EX_i^\star$, for some $1 \leq i \leq 2$ and $H_0^1 \leq_c^{(ex_i \to a)} G_0'$, and an H_1^1 such that $H_1^1 \in EX_j^\star$, for some $1 \leq j \leq 2$ and $H_0^1 \leq_c^{(ex_j \to a')} G_1'$. Let $r_0 = ex_i$ and $r_1 = ex_j$.

Otherwise $Entrance(C)$ and $Entrance(C')$ must be defined. Let $a = Entrance(C)$ and $a' = Entrance(C')$. In which case we let G_0' be graph in $Split(G, Entrance(C))$ such that C is a subgraph of G_0' and G_1' be graph in $Split(G, Entrance(C'))$ such that C' is a subgraph of G_1'. By the above there must exist a H_0^1 such that $H_0^1 \in EN_i^\star$, for some $1 \leq i \leq 3$ and $H_0^1 \leq_c^{(en_i \to a)} G_0'$ and an H_1^1 such that $H_1^1 \in EN_j^\star$, for some $1 \leq j \leq 3$ and $H_0^1 \leq_c^{(ex_j \to a')} G_1'$. Let $r_0 = en_i$ and $r_1 = en_j$

We can then apply the following:

- Let Q be the set of vertices

$$(V(G)\backslash(V(G_0' \cup V(G_1')))) \cup \{a\} \cup \{a'\}\}$$

. Q induces a connected subraph in G. Let $H^2 = Contract(G, Q, a'')$. *(Contract the graph such that it consists of only* $G_0'andG_1'$ *and* a, a' *have been merged into the same vertex)*

- Let $H_0^2 = Prune(H^1, C, a'')$ and $H_1^2 = Join(H_0^2, H_0^1, a'', r_1)$.*(Reduce* G_1' *to the one of the given graphs)*

- *LetG'* be the graph induced by the vertices $(V(H_0^2)\backslash V(G_0')) \cup \{a''\}$

- Let $H_0^3 = Prune(H_0^2, G', a'')$ and $H_1^2 = Join(H_0^3, H_1^1, a'', r_0)$. *(Reduce* G_0' *to the other given graph)*

Thus $H_1^3 \leq G$ and $H_1^3 \in \mathbb{EX} \cup \mathbb{EN}$ proving the lemma.

□

5.8 Searching family Delta

Lemma 5.8.1. *Every member of family* δ *has* $cms \leq 2$

Proof. To prove this fact we will give a procedure for constructing a search strategy for every member of family δ. For members with multiple bipartite components we will show a general strategy for each and demonstrate that these can be used in concert without violating the conditions of connected mixed search.

Consider a graph $G \in \delta$. If G is a path then by Theorem 4.0.4 $cms(G) = 1$, so we assume that G is a non-path.

First we define a sub procedure that augments a search strategy with a new move.

Algorithm 1: Add(G,S,B)

Input: G: graph to be searched,

$S = (A_0, Z_0), \ldots, (A_n, Z_n)$: incomplete valid search strategy for the graph G

B: $B \subset V(G)$

▷We assume that $n \geq 1$ and $A_0 = \emptyset$ and $Z_0 = \emptyset$.

$Z_{n+1} \leftarrow B$

$A_{n+1} \leftarrow A_n$

foreach $b \in B$ **do**

  if *there exits an* $a \in Z_n$ *such that* $(a, b) \in E(G)$ **then**

    $A_{n+1} = A_{n+1} \cup (a, b)$

    ▷A searcher has 'slid' across an edge

  if *there exists a* $b' \in B$ *such that* $b \neq b'$ *and* $(b, b') \in E(G)$ **then**

    $A_{n+1} = A_{n+1} \cup (b, b')$

    ▷Two searchers have occupied the end points of an edge

return $(A_0, Z_0), \ldots, (A_n, Z_n), (A_{n+1}, Z_{n+1})$

Key to the following is searching the biconnected components that do not belong to fans. In these cases the definitions of P_x and P_y from family γ prove essential. The following is a procedure to produce a partial search strategy of these components.

By examining the definitions in family γ, it can be seen that each significant articulation point occurs as an endpoint of P_x or P_y, so searching the areas of the graph that follow those points can be considered a separate concern. So we then define **Bicon** to augment a search strategy with the necessary valid elements for a given demarcated biconnected component.

Algorithm 2: *Bicon*(G, C, P_x, P_y, p, S)

Input: G: graph to be searched,
C: $C \in Demarcate(G)$
P_x, P_y: as defined for family γ corresponding to C
$p \in \{x_1, x_2, y_1, y_2\}$
$S = (A_0, Z_0), \ldots, (A_n, Z_n)$: incomplete valid search strategy for the graph G

$p'_x \leftarrow p$ and $p'_y \leftarrow p$; ▷Both searchers start at the same vertex
$S \leftarrow$ **Add**(S, p'_x, p'_y)
if $p \in \{x_1, y_1\}$ **then** $p'_x \leftarrow x_1$ and $p'_y \leftarrow y_1$ **else** $p'_x \leftarrow x_2$ and $p'_y \leftarrow y_2$
$S \leftarrow$ **Add**(S, p'_x, p'_y)
$U_V = V(Outer(C)) \backslash \{p'_x, p'_y\}$
$U_{IE} = E(Inner(C))$, $U_{OE} = E(Outer(C))$ ▷Define the unsearched edges and vertices
while $U_V \neq \emptyset$ **do** ▷Until the component is completely searched
  if *there does not exist an edge* $(p'_x, b_2) \in U_{IE}$ *and* $E(P_x) \cap U_{OE} \neq \emptyset$ **then**
    ▷Moving p'_x will not cause an unsearched vertex to be adjacent to a searched, unoccupied vertex
    $p'_x \leftarrow p''$ where $(p'_x, p'') \in U_{OE}$
    $U_{OE} \leftarrow U_{OE} \backslash \{(p'_x, p'')\}$ and $U_V \leftarrow U_V \backslash \{p''\}$
    if $(p'_x, p'_y) \in U_{IE}$ **then** $U_{IE} \leftarrow U_{IE} \backslash \{(p'_x, p'_y)\}$
  if *there does not exist an edge* $(p'_y, b_1) \in U_{IE}$ *and* $E(P_y) \cap U_{OE} \neq \emptyset$ **then**
    ▷Moving p'_y will not cause an unsearched vertex to be adjacent to a searched, unoccupied vertex
    $p'_y \leftarrow p''$ where $(p'_y, p'') \in U_{OE}$; $U_{OE} \leftarrow U_{OE} \backslash \{(p'_y, p'')\}$ and $U_V \leftarrow U_V \backslash \{p''\}$
    if $(p'_x, p'_y) \in U_{IE}$ **then** $U_{IE} \leftarrow U_{IE} \backslash \{(p'_x, p'_y)\}$
  $S \leftarrow$ **Add**$(S, \{p'_x, p'_y\})$
return S

Observe that the outerplanar nature of C and the bipartite nature of P_x and P_y ensure that during each iteration at least one of p'_x or p'_y must be able to move as otherwise it would imply an edge crossing. The conditions in the **If** statements will always produce a series of moves that does not violate the conditions of *cms*, independent of any graphs that may follow C via significant articulation points, which will addressed below. Note that the final position of p'_x and p'_y are always two adjacent endpoints of P_x and P_y. It is guaranteed that will always produce a valid partial search strategy for all the vertices $V(C) \subseteq V(G)$.

Similarly, we define a method for producing partial search strategies for paths and fans, since by the definition a path is a fan and so can be searched in an identical manner. Recall that the definition of a fan implies that there is a subgraph P of the fan that is a path where $V(P)$ is all the vertices of the fan.

Algorithm 3: **Fan**(G, F, e, q, S)

Input: G: graph to be searched,
F: subgraph of G such that F is a fan
e: endpoint of path P such that P is a subgraph of F such that $V(P) = V(F)$
q: $q \in V(F)$

let $p_1 = e$
for $2 \leq i \leq |V(P)|$ let p_i be the vertex such that $p_i \in V(P)$ and $(p_{i-1}, p_i) \in E(P)$
for $j \leftarrow 1$ **to** $|V(P)|$ **do** $S \leftarrow$ **Add**$(S, \{p_j, q\})$
return S

Recall from the definition of a fan that all edges not in $E(P)$ are of the form (e', v_i) where e' is an endpoint of P so if $e = e'$ and $q = e'$, if S is valid before the searching of F then **Fan**(G, F, e', e') will also be valid and the entirety of F will be searched. Notice that **Fan** always terminates such that the endpoint of P that is not e will have a searcher occupying it.

Before dealing with the general searching, that is to say searching those members of δ with multiple biconnected components, we will address two special cases. We address those cases where there is no $C \in Demarcate(G)$ such that $V(C)$ contains a vertex that is strong with respect to C.

If G consists entirely of fans, there is either one fan or there are multiple fans that meet at a single significant articulation point. If the former, we let a be the endpoint of the path P that is a subgraph of G, as described previously, such that there is an edge $(a, v) \in E(G)$ such that $(a, v) \notin E(P)$. If there is no such edge, we arbitrarily choose a as either endpoint. Otherwise, where there are multiple fans, we let a be the significant articulation point that joins all the fan.

- Let $S'_0 = (A_0, Z_0)$ where $A_0 = \emptyset$ and $Z_0 = \emptyset$.
- Let $n = |Split(G, a)|$.
- For each $F \in Split(G, a)$ let $S'_i = \mathbf{Fan}(G, F, a, a)$ for $1 \leq i \leq n$.

Thus S'_n will be a valid connected mixed search strategy for G using two searchers.

We consider the case where G contains only a single non-fan biconnected component with P_x and P_y values derived from the definition of family γ such that every graph that follows from a significant articulation point in C is be a fan. Let $C \in Demarcate(G)$ be the single non-fan biconnected component and Let $S_0 = (A_0, Z_0)$ where $A_0 = \emptyset$ and $Z_0 = \emptyset$.

If $SA(G) = \emptyset$, recall there are no restriction on P_x or P_y, and for any valid P_x and P_y if $S = Bicon(G, C, P_x, P_y, x_1, S')$ then S will be a valid connected mixed search strategy for G.

If $|SA(G)| = 1$ then let $\{w_1\} = SA(G)$. Family γ dictates that there exist P_x, P_y as defined previously for C such that $w_1 \in \{x_1, x_2\}$. Let $W_1 = Follow(G, C, w_1)$. Then $Bicon(G, C, P_x, P_y, w_1, Fan(G, W_1, w_1, w_1, S_0))$ a valid connected mixed search strategy for G.

If $|SA(G)| = 2$ then let $\{w_1, w_2\}$. Recall that $w_1 \in \{x_1, x_2\}$ and $w_2 \in \{y_1, y_2\}$. Let $W_1 = Follow(G, C, w_1)$ and $W_2 = Follow(C, w_2)$. If w_1 and w_2 are adjacent then let $q = x_2$ if $w_1 = x_1$ and $q = x_1$ otherwise. It follows that

$$Fan(G, W_2, w_2, w_2, Fan(G, W_1, w_2, w_1, Bicon(G, C, P_x, P_y, q, S_0)))$$

is a valid connected mixed search strategy for G. If w_1 and w_2 are non-adjacent then

$$Fan(G, W_2, w_2, w_2, Bicon(G, C, P_x, P_y, w_1, Fan(G, W_1, w_1, w_1, S_0)))$$

is a valid connected mixed search strategy for G.

Finally if $|SA(G)| = 3$ then let $\{w_1, w_2, w_3\}$. From the description in family γ we know that without loss of generality w_1 and w_2 are adjacent and $x_1 = w_1$, $y_1 = w_2$ and $w_3 \in \{x_2, y_2\}$. Let $W_i = Follow(G, C, w_i)$ for $1 \leq i \leq 3$. Therefore,

$$Fan(G, W_2, w_2, w_2, Fan(G, W_1, w_2, w_1, Bicon(G, C, P_x, P_y, q, Fan(G, W_3, w_3, w_3, S_0))))$$

is a valid connected mixed search strategy for G.

We now consider the final set of cases where there at least one $C \in Demarcate(G)$ such that $V(C)$ contains a vertex that is strong with respect to C. As was discussed in section 5.5 there are two vertices b_1 and b_2, both in $NFSA(G)$, such that for $1 \leq i \leq 2$ then there is a $C' \in Demarcate(G)$ where $b_i \in V(C')$, b_i is strong with respect to C' and every vertex in $V(C')\backslash\{b_i\}$ is either not a significant articulation point or is weak with respect to C'. One of these two vertices will determine our 'initial vertex', where the search begins.

Note that if the graph consists of exactly two biconnected components that share a significant articulation point then $b_1 = b_2$. The restrictions of family δ dictate that at least one of b_i, $1 \leq b_i \leq 2$ is such that for all $C'' \in Demarcate(G)$ if $Entrance(C'')$ is defined $b_i \in V(Follow(G, C'', Entrance(C'')))$ and if $Exit(C'')$ is defined $b_i \notin V(Follow(G, C'', Exit(C'')))$. Let b' be one of b_1 or b_2 such that b' meets these requirement. Let C''' be the demarcated component such that b' is the only significant articulation point in $V(C''')$ that is strong. If there are two such components, choose one arbitrarily. If $V(C''')$ contains no significant articulation point that is weak with respect to C''' we choose a, the initial vertex, to be b'. If there is a single $w_1 \in V(C''')$ such that w_1 is weak with respect to C''' then if w_1 is not adjacent in $Outer(C''')$ to b', we select w_1 as a. If not, a is b'. If there are two vertices $\{w_1, w_2\} \subseteq V(C''')$ such that both are weak with respect to C''', at most one of these vertices can be non-adjacent to b'. If w_i, $1 \leq i \leq 2$ is non-adjacent to b' then select w_i to be a. If both are adjacent to b' then we arbitrarily pick one to be a.

Finally we need to determine the starting state of the partial search strategy S_0. Let $S_0 = (A_0, Z_0)$ where $A_0 = \emptyset$ and $Z_0 = \emptyset$. If b' is the only significant articulation point in $V(C''')$, the restrictions of family γ state that there must be a valid P_x, P_y as defined before where $a = x_1$. In this case we let $S_1 = \textbf{Bicon}(G, C''', P_x, P_y, x_2, S_0)$ and we define $G' = \mathit{Follow}(G, C''', a)$. If there is exactly one $w_1 \in V(C''')$ such that w_1 is weak with respect to C''' and w_1 is adjacent to b', recall that there are multiple possible partitionings of $\mathit{Outer}(C''')$ into P_x and P_y. If the partitioning such that $P_x = \{w_1, b'\}$ is possible then let $S_1 = \textbf{Bicon}(G, C''', P_x, P_y, w_1, \textbf{Fan}(G, \mathit{Follow}(G, C''', w_1), w_1, w_1, S_0))$ otherwise let $S_1 = \textbf{Fan}(G, \mathit{Follow}(C''', w_1), w_1, b', \textbf{Bicon}(G, C''', P_x, P_y, x_2, S_0))$. In both of the preceding cases $G' = \mathit{Follow}(G, C''', a)$.

Otherwise, if none of these special cases are present then $S_1 = S_0$ and $G' = G$.

We can now define **Process**, the center of strategy generating procedure. **Process** takes the remainder of the graph to be searched, G', the starting vertex a and the partial search strategy S.

Algorithm 4: Process(G', a, S)

Input: G': unsearched subgraph of original graph G
  a: $a \in SA(G)$
  $S = (A_0, Z_0), \ldots, (A_n, Z_n)$: incomplete valid search strategy for the graph G

if *there exists a* $G'' \in \mathit{Split}(G', a)$ *where* G'' *is not a fan* **then**
    ▷Family γ and the initial conditions guarantees there will be only one non-fan
  $F_1, \ldots, F_n \leftarrow \mathit{Split}(G', a) \backslash \{G''\}$
  for $i \leftarrow 1$ **to** n **do** $S \leftarrow \textbf{Fan}(G', F_i, a, a, S_1)$;   ▷First search the fans
  Let $C \in \mathit{Demarcate}(G')$ be such that C is a subgraph of G'' and $a \in V(C)$
  if C *is a path* **then**
    ProcessPathStart(G', a, S)
  else
    ProcessNonPathStart(G', a, S)
else
  $F_1, \ldots, F_n \leftarrow \mathit{Split}(G', a)$
  for $i \leftarrow 1$ **to** n **do** $S \leftarrow \textbf{Fan}(G', F_i, a, a, S)$
return S

Algorithm 5: **ProcessPathStart**(G', a, S)

Input: G': unsearched subgraph of original graph G
a: $a \in SA(G)$
$S = (A_0, Z_0), \ldots, (A_n, Z_n)$: incomplete valid search strategy for the graph G

$e \leftarrow (V(C) \cap NFSA(G')) \backslash \{a\}$; ▷e will be the other endpoint of the path
$S \leftarrow$ **Fan**(G', C, a, a, S) ; ▷Search the path
$S \leftarrow$ **Process**$(Follow(G', C, e), e, S)$; ▷Recursively search the rest of the graph
return S

Algorithm 6: **ProcessNonPathStart**(G', a, S)

Input: G': unsearched subgraph of original graph G
a: $a \in SA(G)$
$S = (A_0, Z_0), \ldots, (A_n, Z_n)$: incomplete valid search strategy for the graph G

Let $W \subseteq V(C) \cap NFSA(G')$ such that $w \in W$ if and only if w is weak *w.r.t.* C
Let $R \subseteq V(C) \cap NFSA(G')$ such that $r \in R$ if and only if r is weak *w.r.t.* C
$S \leftarrow$ **Bicon**$'_{(|W|,|R|)}(G', C, a, S)$
return S

Notice that the above **ProcessNonPathStart** depends on the as yet undefined sub-procedure **Bicon**$'_{(|W|,|R|)}(G', C, a, S_1)$. Since by the restrictions on the graphs in family δ there are only a limited number of possibilities for $|W|$ and $|R|$, which we enumerate below. If $|W| > 0$ we refer to the vertices of W as $w_1, \ldots, w_{|W|}$ and similarly if $|R| > 0$ we refer to the vertices of R as $s_1, \ldots, s_{|R|}$. Note that the cases where $|R| = 0$ were all handled in the 'special cases' above. If $|W| = 0$ and $|R| = 1$ then family γ dictates that that P_x and P_y will be defined such that $x_1 = s_1$ and so with the input to **Bicon**$'$, a must be s_1. Since this component has only one significant articulation point, which must be the one that the searchers have enter on, this will be the last component searched.

Algorithm 7: **Bicon**$'_{(1,0)}(G', C, a, S)$

Input: G': unsearched subgraph of original graph G
C: $C \in Demarcate(G')$ a: $a \in SA(G) cup V(G')$ and $a \notin SA(G')$
$S = (A_0, Z_0), \ldots, (A_n, Z_n)$: incomplete valid search strategy for the graph G

return Bicon$(G'', C, P_x, P_y, a, S_1)$

If $|W| = 0$ and $|R| = 2$ then the conditions from family γ do not define *Entrance* or *Exit* and so a as in input to **Bicon**$'$ could be either s_1 or s_2.

Algorithm 8: $\mathbf{Bicon}'_{(0,2)}(G', C, a, S)$

Input: G': unsearched subgraph of original graph G

C: $C \in \mathit{Demarcate}(G')$ a: $a \in SA(G) cup V(G')$ and $a \notin SA(G')$

$S = (A_0, Z_0), \ldots, (A_n, Z_n)$: incomplete valid search strategy for the graph G

$S \leftarrow \mathbf{Bicon}(G', C, P_x, P_y, a, S)$ **return** $\mathbf{Process}(\mathit{Follow}(G', C, s'), s', S)$ *where* $s' \in \{s_1, s_2\}$ *and* $s' \neq a$

If $|W| = 1$ and $|R| = 1$ recall that if w_1 and s_1 are non-adjacent in *Outer*(C) then $s_1 \in \{x_1, x_2\}$, $w_1 \in \{y_1, y_2\}$, *Entrance*(C) and *Exit*(C) are undefined. Otherwise either $V(P_x) = \{s_1, w_2\}$ and both *Entrance*(C) and *Exit*(C) are undefined or $w_1 \in \{x_1, x_2\}$, $s_1 \in \{y_1, y_2\}$, *Exit*$(G, C) = s_1$ and *Entrance*(C) is undefined. So if C cannot be partitioned such that $V(P_x) = \{s_1, w_2\}$, a must be w_1 in the input to $\mathbf{Bicon}'_{(1,1)}$.

Algorithm 9: $\mathbf{Bicon}'_{(1,1)}(G', C, a, S)$

Input: G': unsearched subgraph of original graph G

C: $C \in \mathit{Demarcate}(G')$ a: $a \in SA(G) cup V(G')$ and $a \notin SA(G')$

$S = (A_0, Z_0), \ldots, (A_n, Z_n)$: incomplete valid search strategy for the graph G

```
if w1 and s1 are non-adjacent in Outer(C) or V(Px) ≠ {s1, w2} then
    if a = w1 then
        S ← Bicon(G', C, Px, Py, w1, S1)
        S ← Process(Follow(G', C, s1), s1, S1)
    else
        S ← Bicon(G', C, Px, Py, s1, S1)
        S ← Process(Follow(G', C, w1), w1, S1)
else
    S ← Bicon(G', C, Px, Py, w1, S1)
    S ← Process(Follow(G', C, w1), s1, S1)
return S
```

If $|W| = 2$ and $|R| = 1$ the restrictions inherited by family δ are such that at least one pair of the significant articulation points in $V(C)$ must be adjacent in $Outer(C)$. If $(w_1, w_2) \in E(Outer(C))$ and both w_1 and w_2 are non-adjacent to s_1 in $Outer(C)$ then $\{x_1, x_2\} = \{w_1, s_1\}$, $w_2 \in \{y_1, y_2\}$, $Entrance(C) = s_1$ and $Exit(C)$ is undefined. In this case a must be s_1 in the input to **Bicon′**. Otherwise, without loss of generality, say s_1 and w_1 are adjacent in $Outer(C)$ and there are four further cases:

w_2 is non-adjacent to both s_1 and w_1 in $Outer(C)$ The restrictions are that $s_1 = x_1$, $w_1 = y_1$, $w_2 \in \{x_2, y_2\}$, $Exit(C) = s_1$ and $Entrance(C) = w_2$. So a must be w_1 in the input to **Bicon′**.

w_2 is adjacent to both s_1 and w_1 in $Outer(C)$ In this case C is a K_3 and $V(P_x) = \{s_1, w_1\}$, $V(P_2) = \{w_1\}$ and both $Entrance(C)$ and $Exit(C)$ are undefined.

w_2 is adjacent to s_1 in $Outer(C)$ Either $V(P_x) = \{s_1, w_1\}$ or $V(P_x) = \{s_1, w_2\}$, $Exit(C) = s_1$ and $Entrance(C)$ is undefined. So a must be w_1 or w_2 in the input to **Bicon′**

w_2 is adjacent to w_1 in $Outer(C)$ Either $V(P_x) = \{s_1, w_1\}$, $Entrance(G, C) = s_1$ and $Exit(G, C)$ is undefined or $V(P_x) = \{w_1, w_2\}$, $Exit(G, C) = s_1$ and $Entrance(G, C)$ is undefined. However, if either choice of $V(P_x)$ leads to a valid partitioning, both $Entrance(C)$ and $Exit(C)$ are undefined. Concerning the value of a as an input to **Bicon′** in the former case the constraints dictate that $a = s_1$. In the later case a the constraints dictate that a could be either w_1 or w_2. However, since C contains a vertex that is strong with respect C then C must be the first biconnected component searched and so by the initial conditions of the search w_2 would have been selected as w_1 is weak and adjacent to a strong vertex.

Algorithm 10: $\textbf{Bicon}'_{(2,1)}(G', C, a, S)$

Input: G': unsearched subgraph of original graph G
C: $C \in Demarcate(G')$ a: $a \in SA(G)cupV(G')$ and $a \notin SA(G')$
$S = (A_0, Z_0), \ldots, (A_n, Z_n)$: incomplete valid search strategy for the graph G

if *w_1 is adjacent to w_2 and both are non-adjacent to s_1 in $Outer(C)$* **then**
  $S \leftarrow \textbf{Bicon}(G'', C, P_x, P_y, s_1, S_1)$
  $S \leftarrow \textbf{Fan}(G', Follow(C, w_1), w_1, w_2, S_1)$
  $S \leftarrow \textbf{Fan}(G', Follow(C, w_2), w_2, w_2, S_1)$
else if *s_1 is adjacent to w_1 and both are non adjacent in $Outer(C)$* **then**
  $S \leftarrow \textbf{Bicon}(G'', C, P_x, P_y, w_2, S_1)$
  $S \leftarrow \textbf{Fan}(G', Follow(C, w_1), w_1, s_2, S_1)$
  $S \leftarrow \textbf{Process}(Follow(G', C, s_1), s_1, S_1)$
else if *s_1 is adjacent to w_1 and w_1 is adjacent to w_2 in $Outer(C)$* **then**
  if *w_2 adjacent to s_1 in $Outer(C)$* **then**
    if $a = s_1$ **then**
      $S \leftarrow \textbf{Fan}(G', Follow(C, w_1), w_1, w_2, S_1)$
      $S \leftarrow \textbf{Fan}(G', Follow(C, w_2), w_2, w_2, S_1)$
    else
      $S \leftarrow \textbf{Fan}(G', Follow(C, w'), w', s_1, S_1)$ where $w' \in \{w_1, w_2\}$ and $w' \neq a$
      $S \leftarrow \textbf{Process}(Follow(G'', C, s_1), s_1, S_1)$
  else
    if $a = s_1$ **then**
      $S \leftarrow \textbf{Bicon}(G'', C, P_x, P_y, s_1, S_1)$ $S \leftarrow \textbf{Fan}(G', Follow(C, w_2), w_2, w_1, S_1)$
      $S \leftarrow \textbf{Fan}(G', Follow(C, w_1), w_1, w_1, S_1)$
    else ▷a must be w_2

      $S \leftarrow \textbf{Bicon}(G'', C, P_x, P_y, w_2, S_1)$ $S \leftarrow \textbf{Fan}(G', Follow(C, w_1), w_1, s_2, S_1)$
      $S \leftarrow \textbf{Process}(Follow(G'', C, s_1), s_1, S_1)$
else ▷w_1 is adjacent to s_1, s_1 adjacent to w_2, w_2 non-adjacent to s_1 in $Outer(C)$
  $S_1 \leftarrow \textbf{Bicon}(G'', C, P_x, P_y, a, S_1)$
  $S_1 \leftarrow \textbf{Fan}(G', Follow(C, w'), w', s_1, S_1)$ where $w' \in \{w_1, w_2\}$ and $w' \neq a$
  $S_1 \leftarrow \textbf{Process}(Follow(G'', C, s_1), s_1, S_1)$
return S

Finally, If $|W| = 1$ and $|R| = 2$ recall that there are several conditions. In all cases

w_1 must be adjacent to either s_1 or s_2 in $Outer(C)$, without loss of generality say that w_1 is adjacent to s_1. The conditions for $Exit$ and $Entrance$ depend on the positioning in $Outer(C)$ of s_2:

s_2 is non-adjacent to both w_1 and s_1 in $Outer(C)$ In this case $s_1 = x_1$, $w_1 = y_1$, $s_2 \in \{x_2, y_2\}$. $Exit(C) = s_1$ and $Entrance(C)$ is undefined.

s_2 is adjacent to both w_1 and s_1 in $Outer(C)$ Once again this implies that C is a K_3 and the restrictions are that $V(P_x) = \{s_1, s_2\}$, $V(P_y) = \{w_1\}$ and both $Entrance(C)$ and $Exit(C)$ are undefined.

s_2 is adjacent to s_1, but not to w_1 In this case $V(P_x) = \{s_1, s_2\}$, $Entrance(C) = s_2$ and $Exit(C) = s_1$.

s_2 is adjacent to w_1, but not to s_1 Either $V(P_x) = \{s_1, w_1\}$, $Entrance(C) = s_1$ and $Exit(C) = s_2$ or $V(P_x) = \{s_2, w_1\}$, $Entrance(C) = s_2$ and $Exit(C) = s_1$. However if either choice of $V(P_x)$ leads to a valid partitioning then $Exit(C) = w_1$ and $Entrance(C)$ is undefined.

Considering a as an input to **Bicon′**, w_1 is weak with respect to C, and since $V(C)$ contains two significant articulation points that are strong with respect to C, w_1 could have not been selected as the initial vertex for searching. So a is either s_1 or s_2, and by all the cases above we know that w_1 is adjacent in $Outer(C)$ to the vertex s' such that $s' \in \{s_1, s_2\}$ and $s' \neq a$.

Algorithm 11: $\mathbf{Bicon}'_{(1,2)}(G', C, a, S)$

Input: G': unsearched subgraph of original graph G
C: $C \in Demarcate(G')$ a: $a \in SA(G) cup V(G')$ and $a \notin SA(G')$
$S = (A_0, Z_0), \ldots, (A_n, Z_n)$: incomplete valid search strategy for the graph G

$S \leftarrow$ **Bicon**(G', C, P_x, P_y, a, S)
$S \leftarrow$ **Fan**$(G', Follow(C, w_1), w_1, a', S)$
return Process$(Follow(G'', C, a'), a', S)$

By examining the various sub-procedures defined above it can be seen that the algorithm will always produce a valid connected mixed search strategy. As was described above the **Bicon** procedure always produces a valid partial strategy for a particular biconnected piece due to the bipartite nature of the P_x, P_y partitioning. Inspection will show that no fan is searched without the other searcher remaining stationary on the border of the unsorted area. The various instances of the *Bicon′* cover every possible combination of significant articulation point, not covered by the special cases, ensure that by the end of their execution the searchers are in a position to search the rest of the graph given that they begin searching the components in the appropriate direction. This last condition is ensured by initial conditions that respect the *Entrance* and *Exit* vertices. So we can conclude if $G \in \gamma$ then $cms(G) \leq 2$. □

5.9 Final results

We define $\mathbb{O}_2$ to be $Complete(OP_1) \cup Complete(OP_2) \cup Complete(D) \cup Complete(S) \cup Complete(TF_1) \cup Complete(TF_2) \cup Complete(TF_3) \cup Complete(TF_4) \cup Complete(W_1) \cup Complete(H_1) \cup Complete(H_2) \cup \mathbb{FN} \cup \mathbb{EN} \cup \mathbb{EX}$.

Theorem 5.9.1. *$\mathbb{O}_2$ is the complete set of obstructions in the contraction order for the family of graph with $cms \leq 2$.*

Proof. Lemma 5.8.1 shows each member of family δ has $cms \leq 2$. Lemmas 5.1.1 to 5.6.1 collectively show that if a graph G is not in family δ then it has some $O \in \mathbb{O}_{\not\delta}$ such that $O \leq_c G$. Since $cms(O) = 3$ by the construction of $\mathbb{O}_{\not\delta}$ so by Theorem 3.1.1 $cms(G) \geq 3$. Therefore every graph not in family δ has $cms > 2$. Also by Theorem 3.1.1 we see that if there was a graph $G' \in \delta$ such that if there was an $O' \leq_c G'$, for some $O' \in \mathbb{O}_2$ that $cms(G) > 2$ which is contradicted by Lemma 5.8.1. Finally, if there an obstruction O'' such that $O'' \notin \mathbb{O}_2$ then $O'' \notin \delta$ and so, by Lemmas 5.1.1 to 5.6.1, there exists a O''' such that $O''' \leq_c O''$. So the theorem is proved. □

Chapter 6

Conclusion

In this work we have shown the obstructions in the contraction order for $cms \leq 2$. The techniques and the results presented open avenues for further inquiry. While it was shown that for $cms \leq 2$ there were a finite number of obstructions in the contraction order is this true for all $cms \leq k$, $k > 2$?. If not, what is the largest value of k for which this is true. If so is there a generalization that would allow the construction of these obstruction sets, or at the very least a bound on their number.

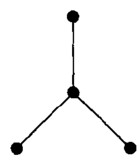
It seems likely that, even if finite, that the number of obstructions would grow rapidly as k increased, very soon surpassing the limits of human beings to generate and check them. If there were to be the case then a bound on the size of the largest obstruction, in terms of vertices, then it would be feasible to machine generate the obstruction sets for much larger k.

While this work concentrated on connected mixed search the techniques given would extend easily to node search, edge search and certain variations on the 'cops and robbers' and other search games.

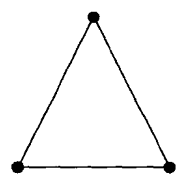
Finally, are there other non-trivial graph families that are not minor closed, but are contraction closed and have finite obstruction sets?

Appendix A

Obstructions for $cms \leq 1$ ($\mathbb{O}_1$)



Y



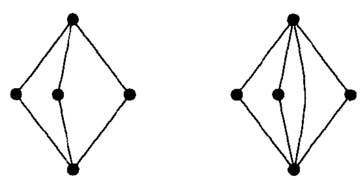
D

Appendix B

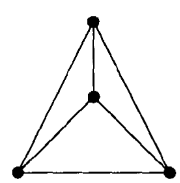
Contraction Obstructions for $cms \leq 2$ ($\mathbb{O}_2$)

B.1 Obstructions for Outerplanar Graphs

B.1.1 OP_1

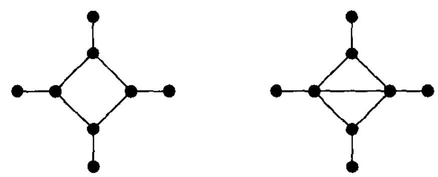


B.1.2 OP_2

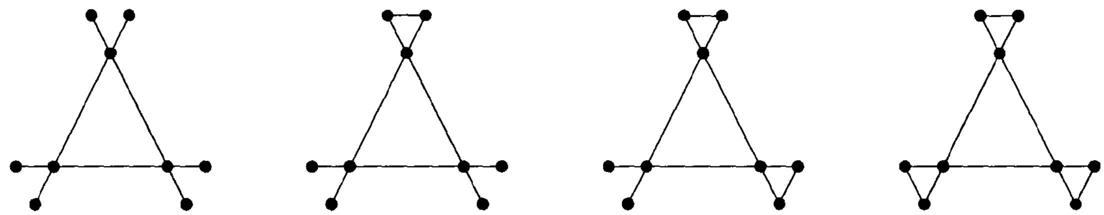


B.2 Obstructions for Family Alpha

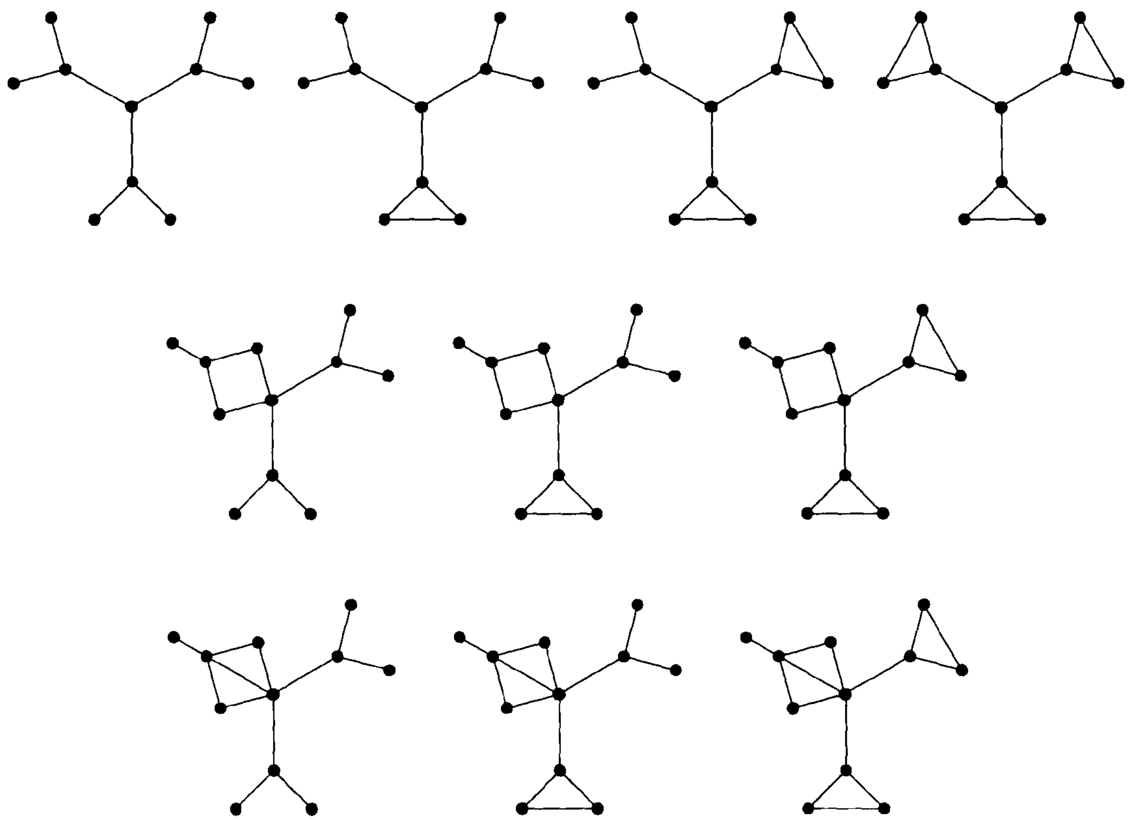
B.2.1 *FW*

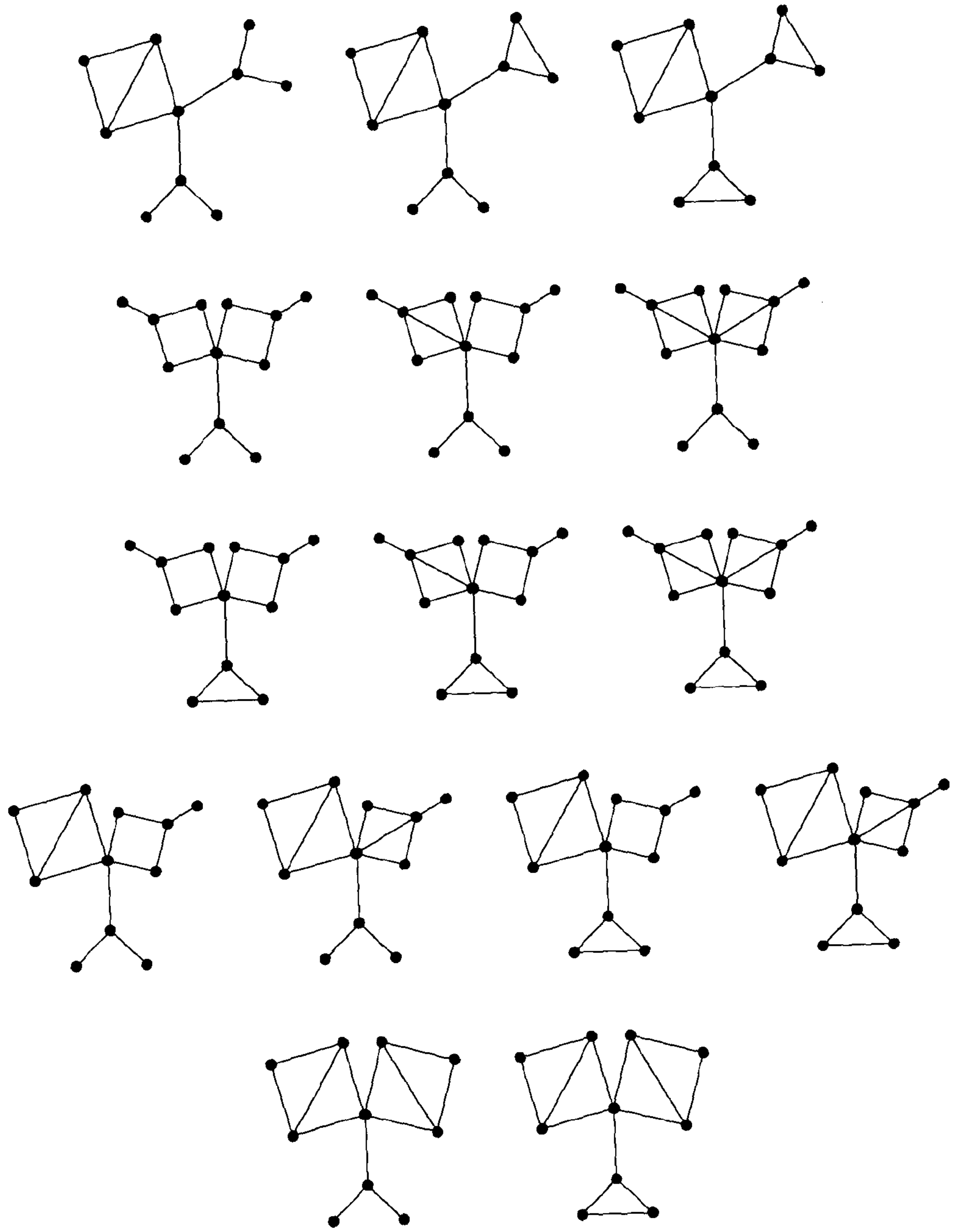


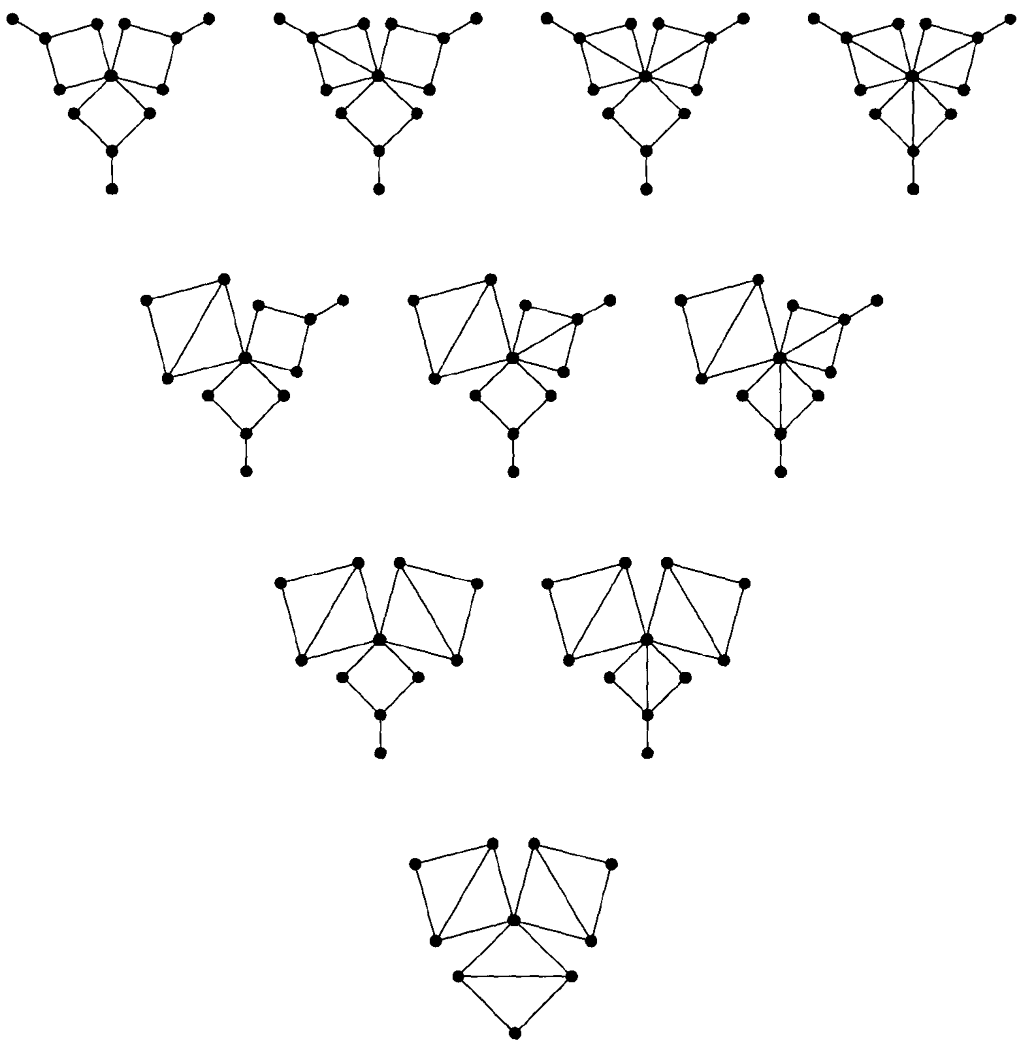
B.2.2 *TS*



B.3 Obstructions for Family Beta

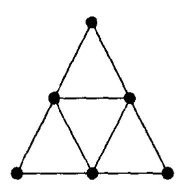




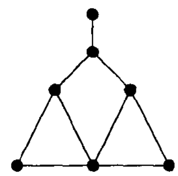


B.4 Obstructions for Family Gamma

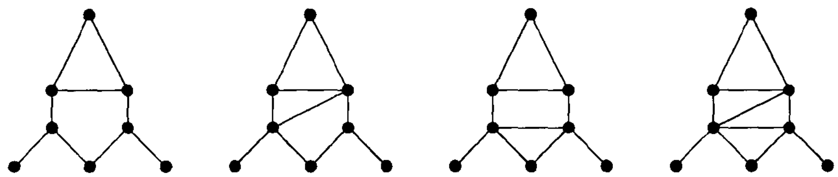
B.4.1 TF_1



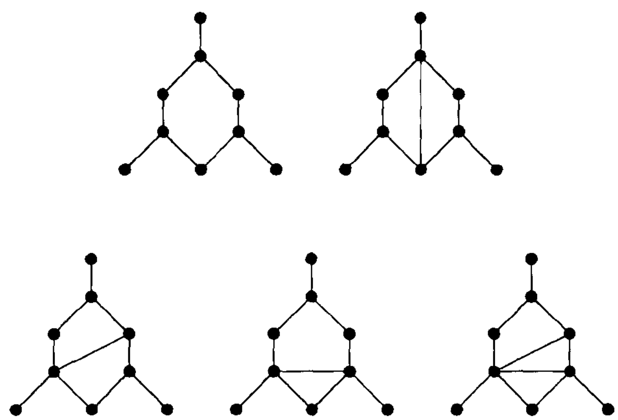
B.4.2 TF_2



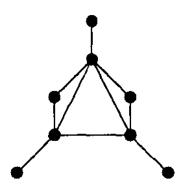
B.4.3 TF_3



B.4.4 TF_4

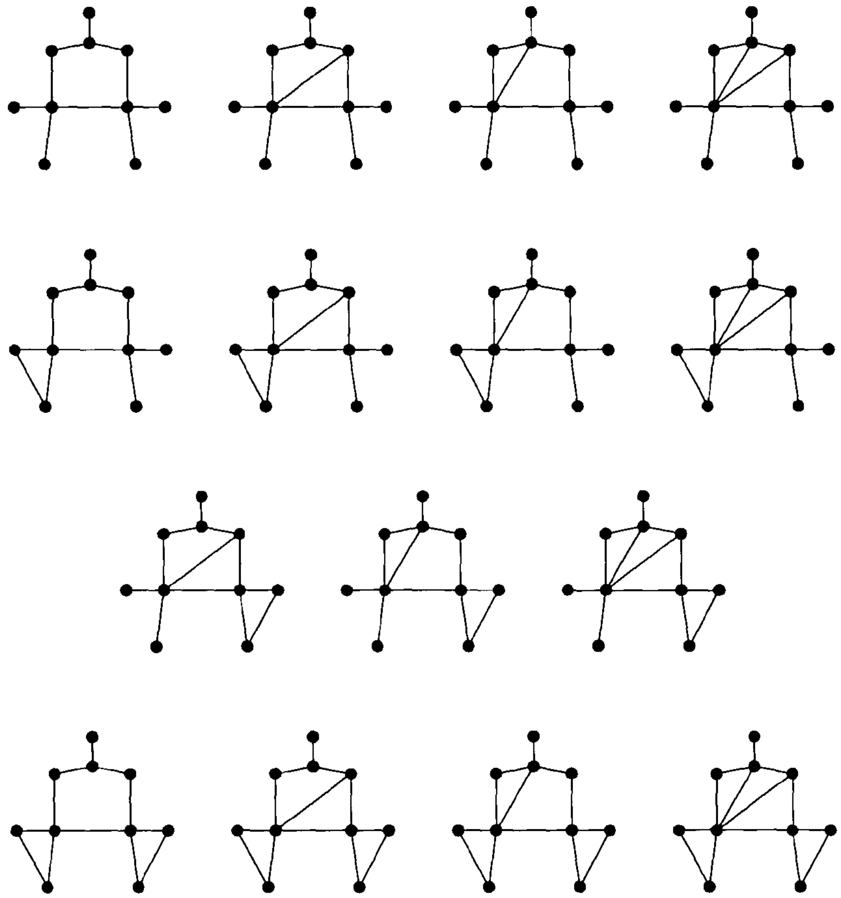


B.4.5 TW_1

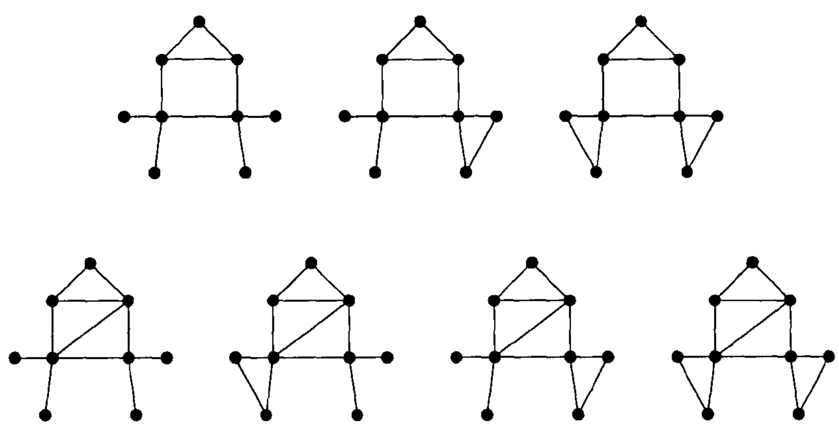


TW_1

B.4.6 WA_1

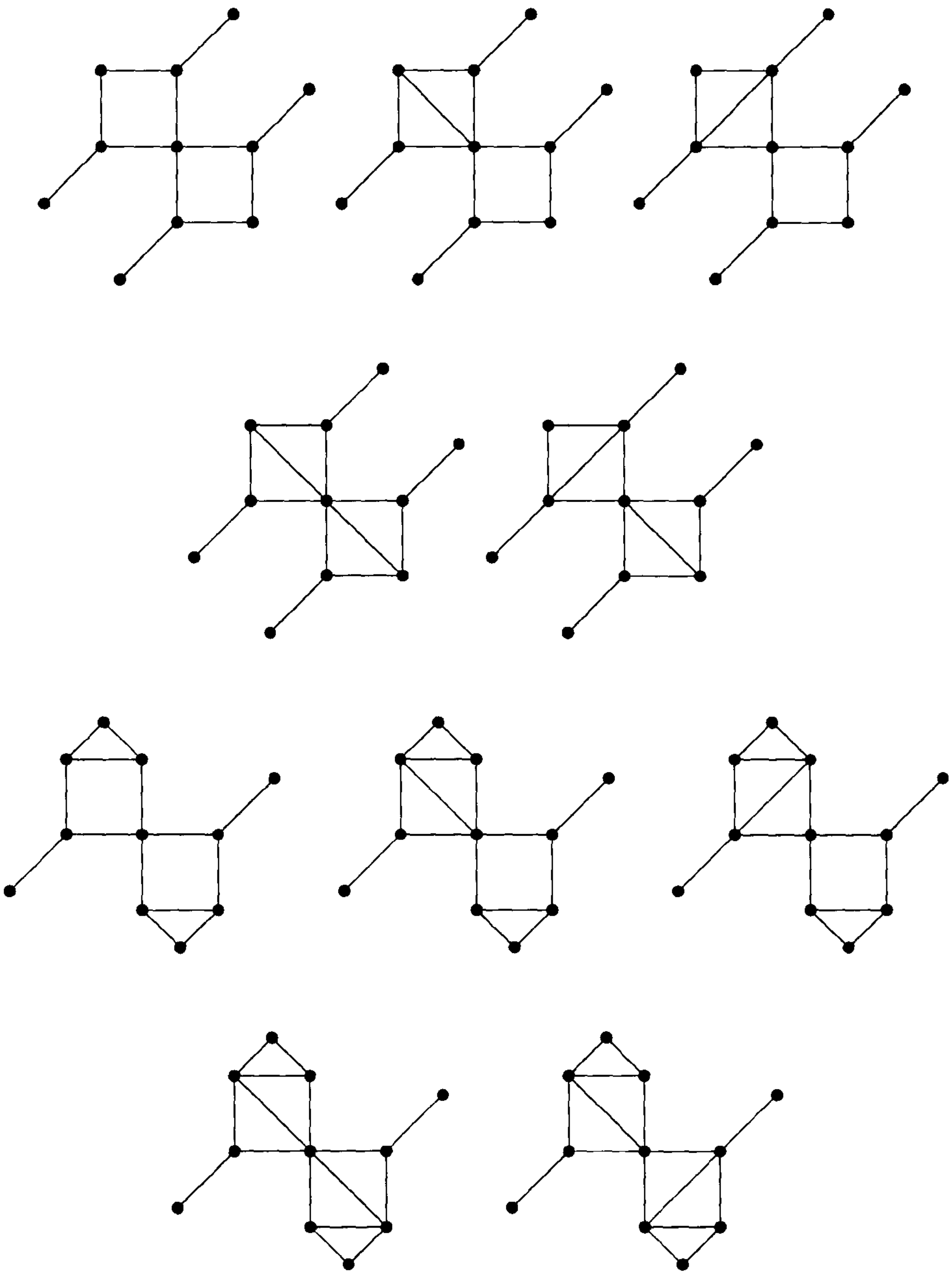


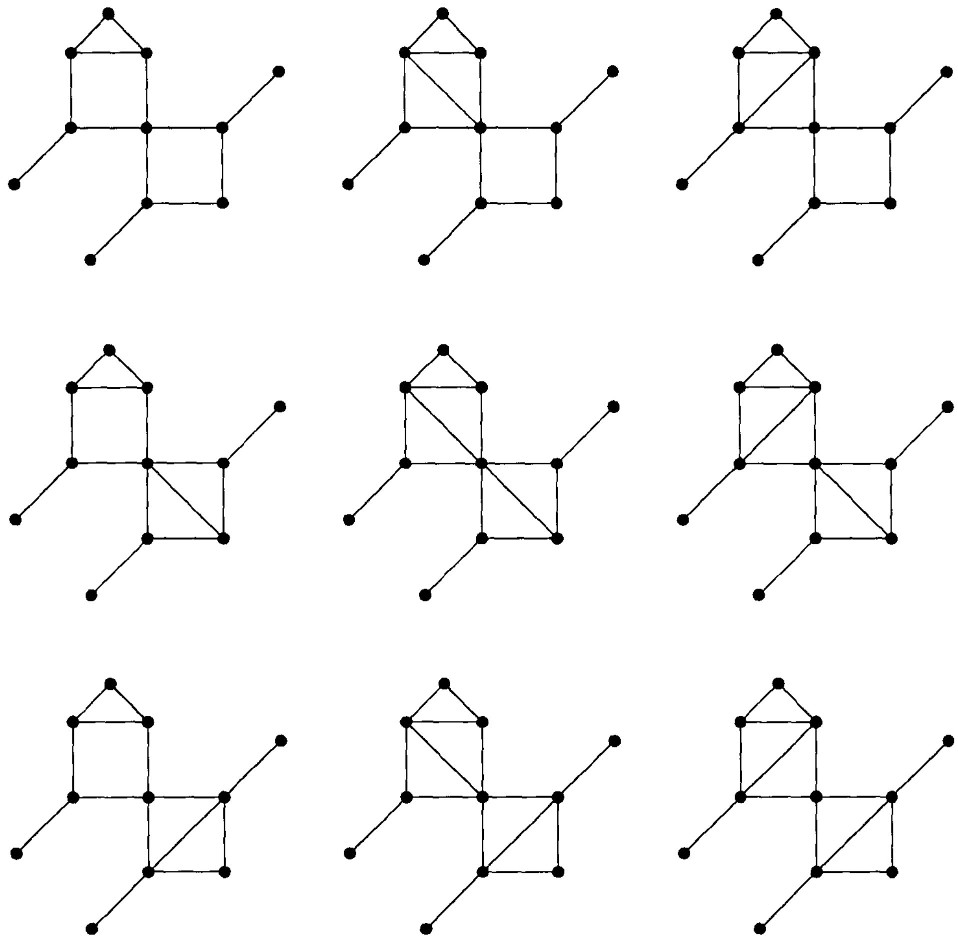
B.4.7 WA_2



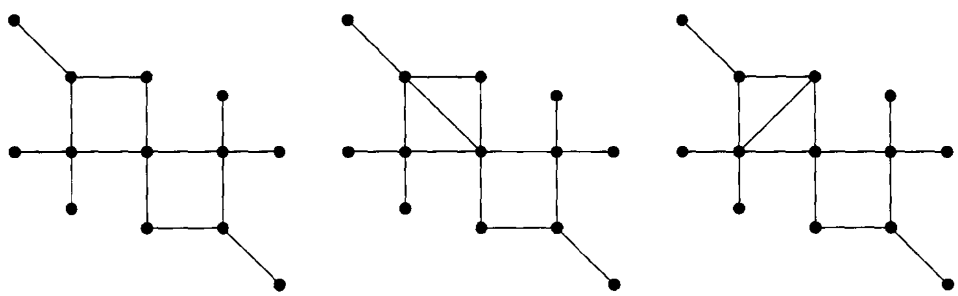
B.5 Obstructions for Family Gamma

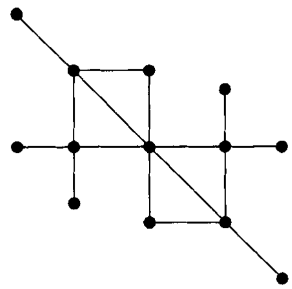
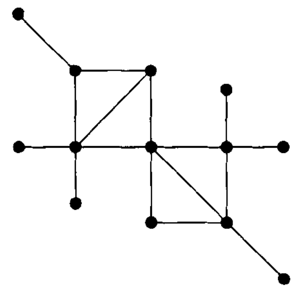
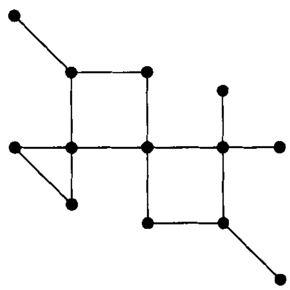
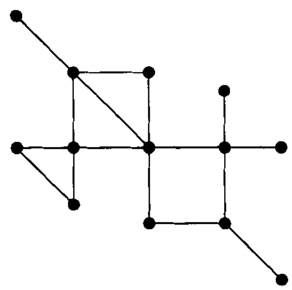
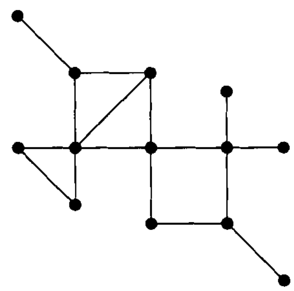
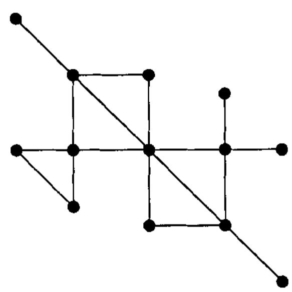
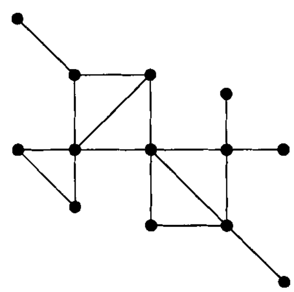
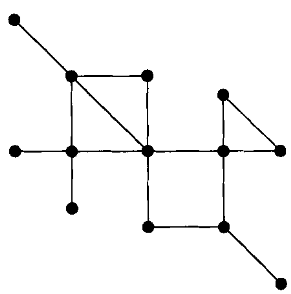
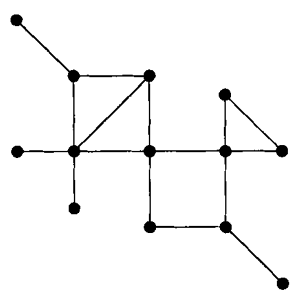
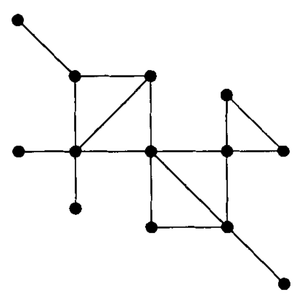
B.5.1 $\mathbb{EX}$

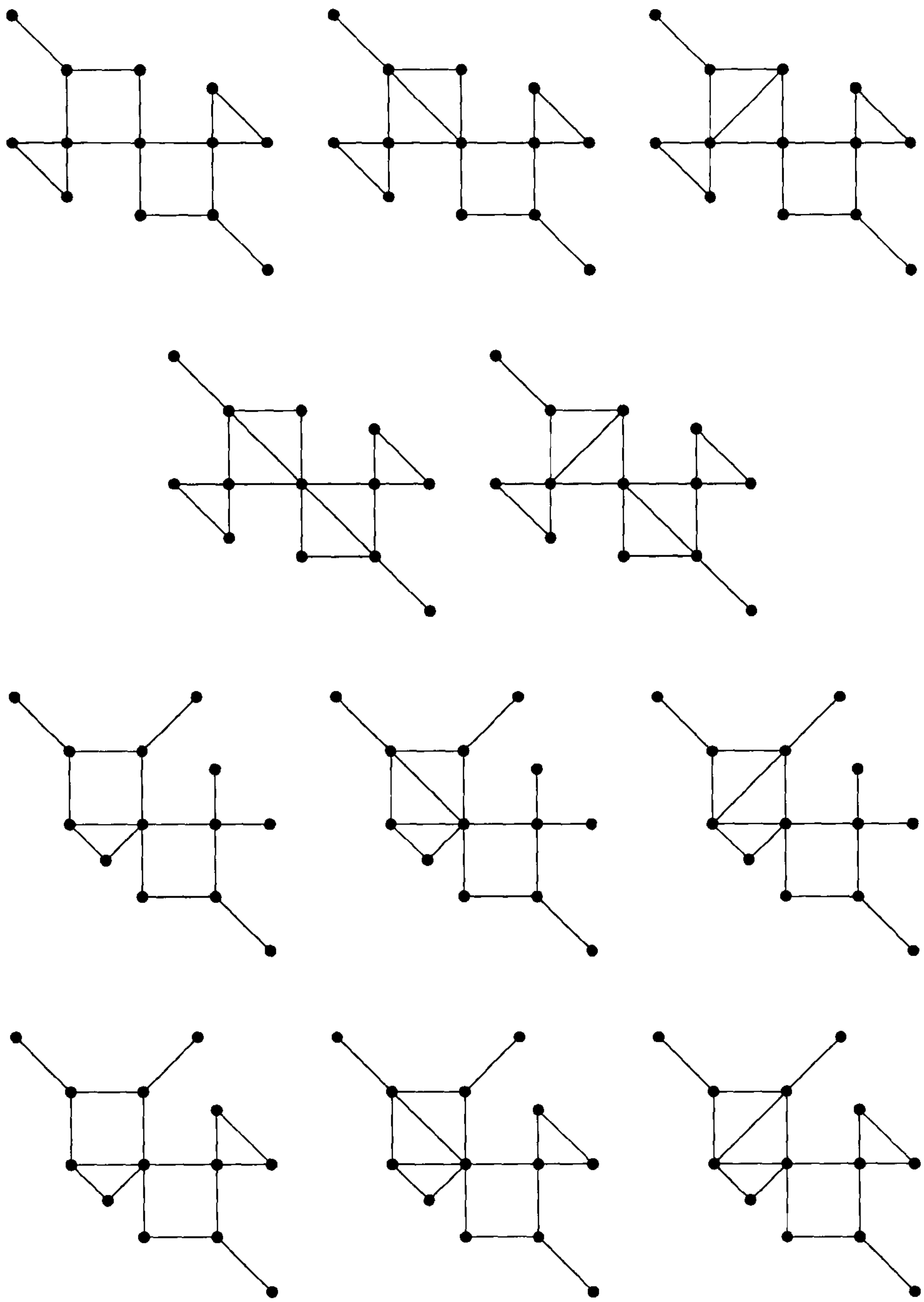


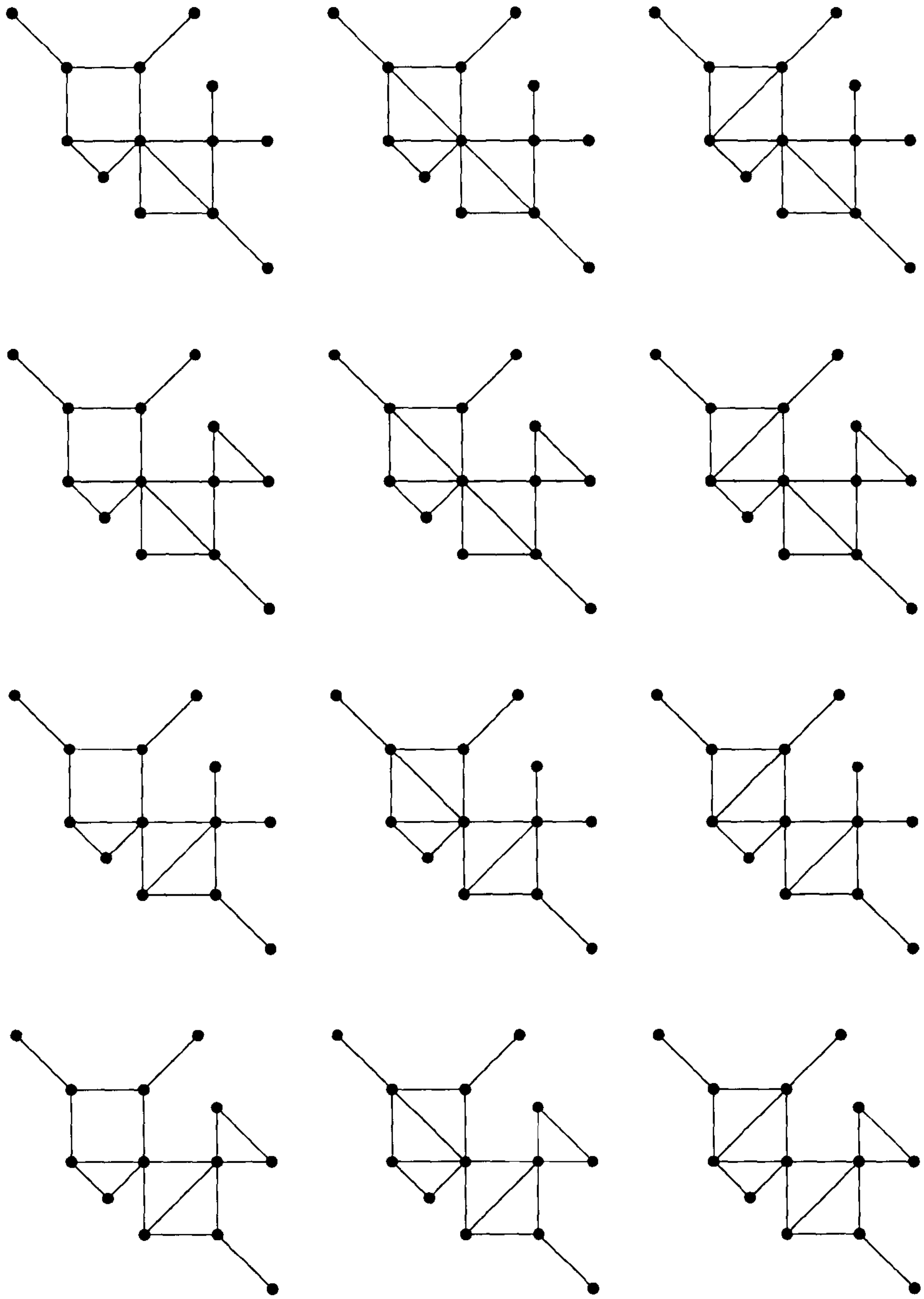


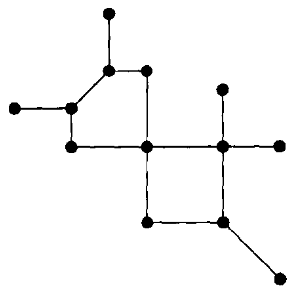
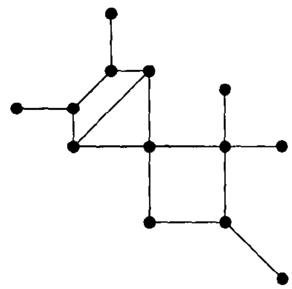
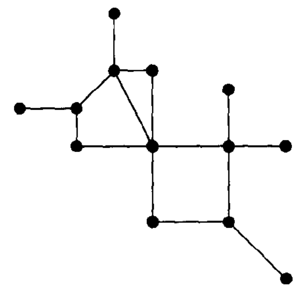
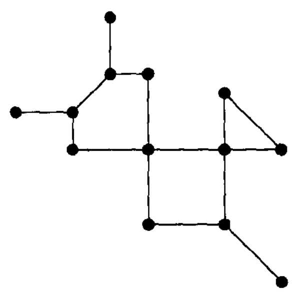
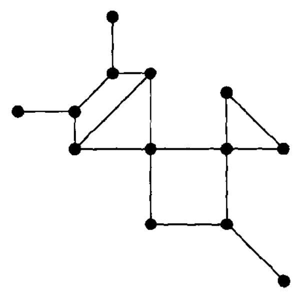
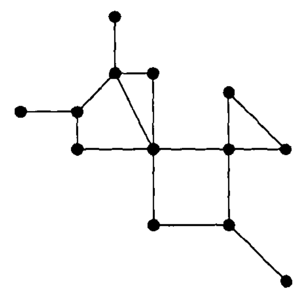
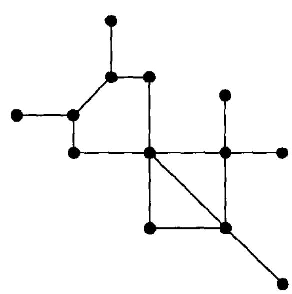
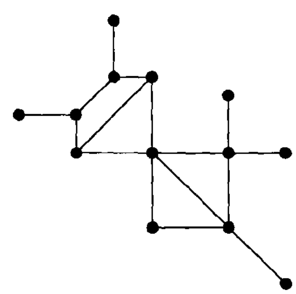
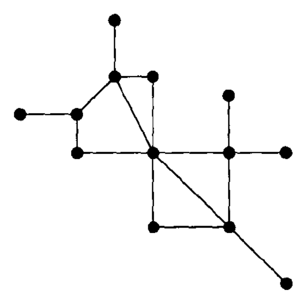
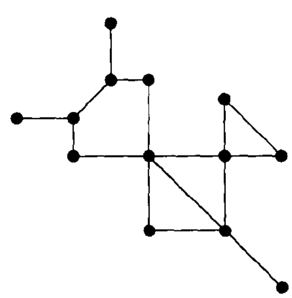
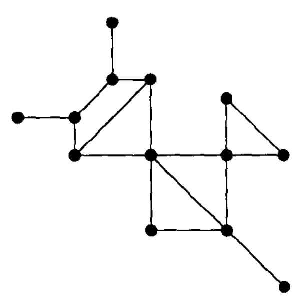
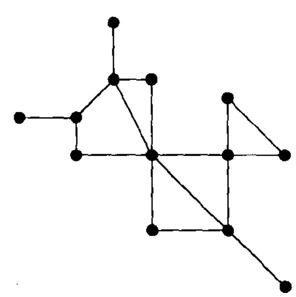
B.5.2 $\mathbb{EN}$

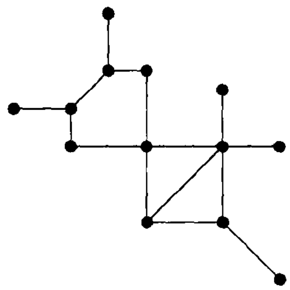
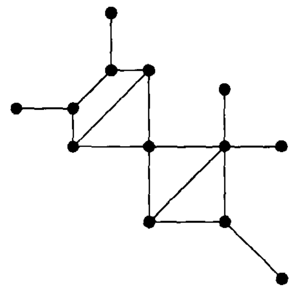
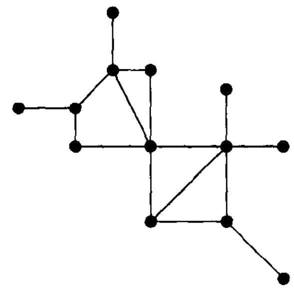
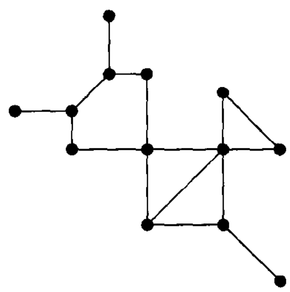
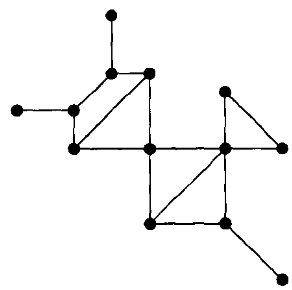
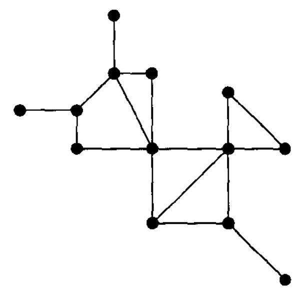
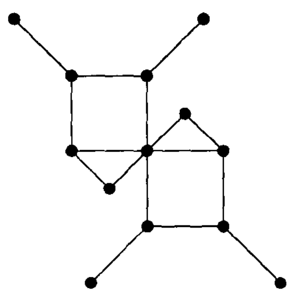
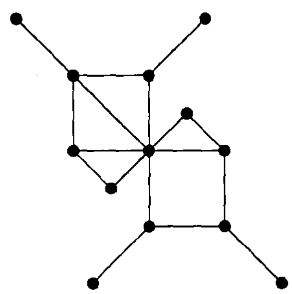
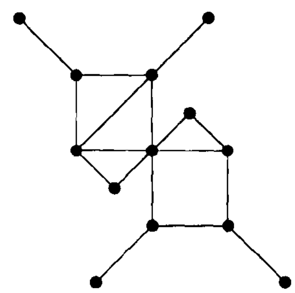
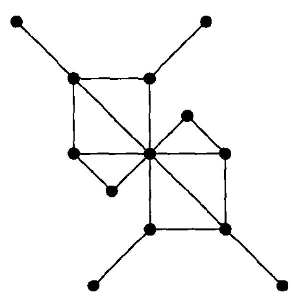
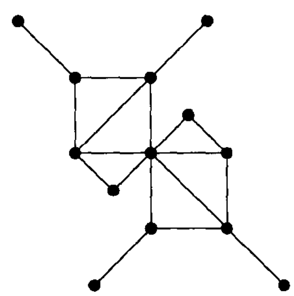


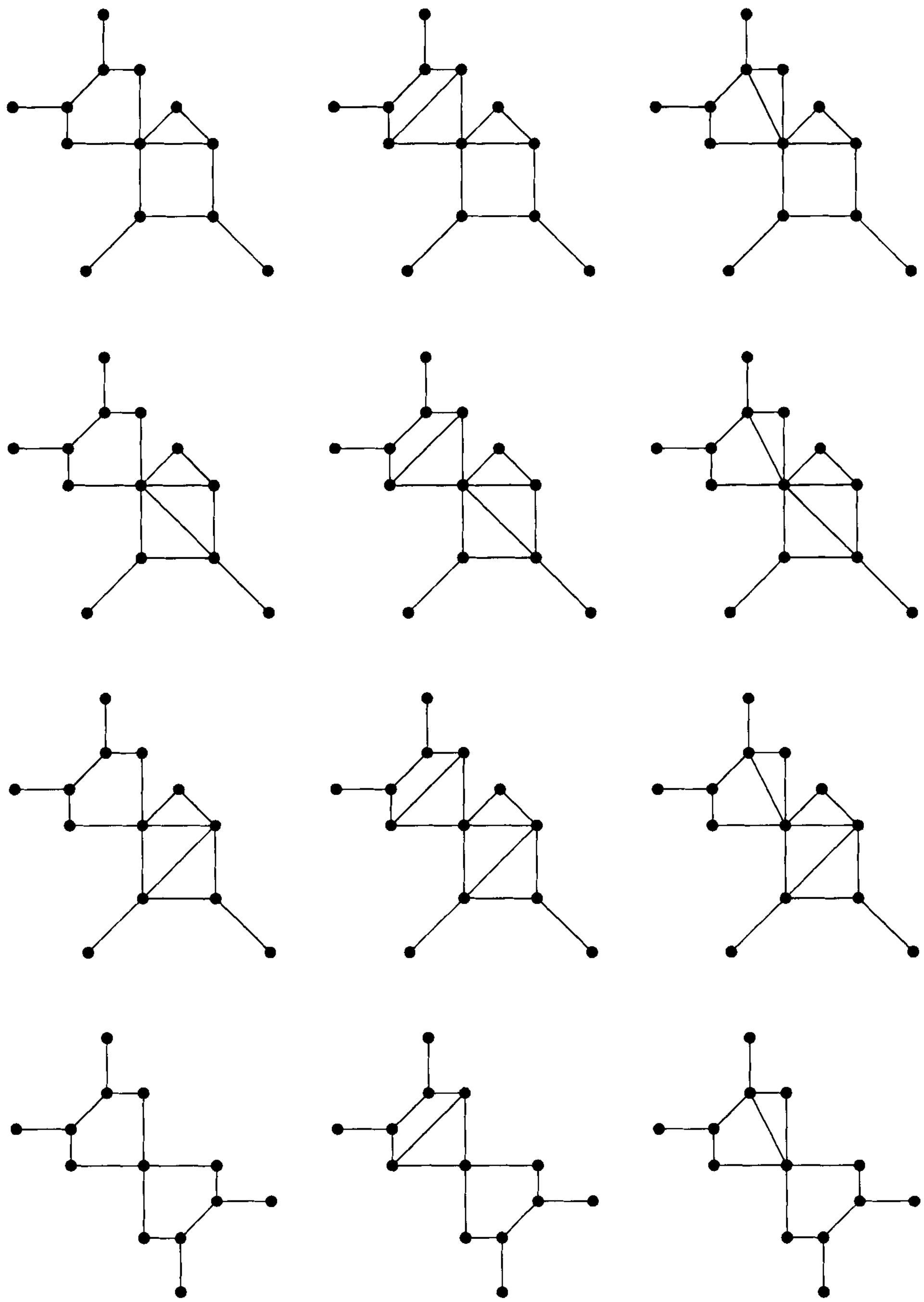











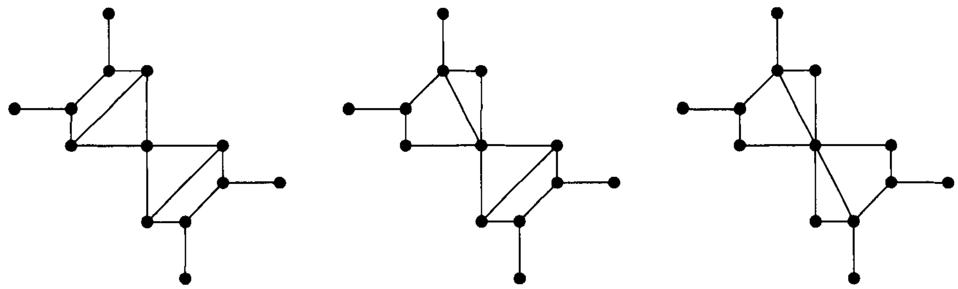








Bibliography

[1] Yoji Kajitani Atsushi Takahashi, Shuichi Ueno. Mixed searching and proper-path-width. *Theoretical Computer Science*, 137(2):253–268, 1995.

[2] Lali Barriere, Pierre Fraigniaud, Nicola Santoro, and Dimitrios M. Thilikos. Connected and internal graph searching.

[3] D. Bienstock and Paul Seymour. Monotonicity in graph searching. *J. Algorithms*, 12(2):239–245, 1991.

[4] Daniel Bienstock. Graph searching, path-width, tree-width and related problems (a survey). *DIMACS Ser. in Discrete Mathematics and Theoretical Computer Science*, 5:33–49, 1991.

[5] Reinhard Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, Heidelberg, third edition edition, July 2005.

[6] L.M. Kirousis and C.H. Papadimitriou. Interval graphs and searching. *Discrete Mathematics*, 55(1):181–184, 1985.

[7] Andrea S. LaPaugh. Recontamination does not help to search a graph. *Journal of the ACM*, 40(2):224–245, 1993.

[8] C. H. Papadimitriou M. Kirousis. Searching and pebbling. *Theoretical Computer Science*, 47(2):205–218, 1986.

[9] N. Megiddo, S. L. Hakimi, M. R. Garey, D. S. Johnson, and C. H. Papadimitriou. The complexity of searching a graph. *J. ACM*, 35(1):18–44, 1988.

[10] T. D. Parsons. Pursuit-evasion in a graph. In Y. Alavi and D. R. Lick, editors, *Theory and Application of Graphs*, pages 426–441. Springer-Verlag, Berlin, 1976.