UNIVERSITY OF OTTAWA

MASTERS THESIS

Transforming Plane Triangulations by Simultaneous Diagonal Flips

---

M Tanvir KAYKOBAD

*A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science*

in the

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

© M Tanvir Kaykobad, Ottawa, Canada, 2020

Declaration of Authorship

This thesis titled, “Transforming Plane Triangulations by Simultaneous Diagonal Flips” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Author: M Tanvir KAYKOBAD

UNIVERSITY OF OTTAWA

Abstract

Faculty of Engineering
School of Electrical Engineering and Computer Science

Master of Computer Science

Transforming Plane Triangulations by Simultaneous Diagonal Flips

by M Tanvir KAYKOBAD

We explore the problem of transforming plane triangulations using simultaneous diagonal flips. Wagner showed that any n -vertex plane triangulation can be transformed to any other plane triangulation on equal number of vertices using a finite sequence of diagonal flips. Later on it has been established that $\mathcal{O}(n)$ individual flips suffice to complete this transformation. Bose et al. showed that the transformation can also be done in $4 \times (\frac{2}{\log \frac{54}{53}} + \frac{2}{\log \frac{6}{5}}) \log n + 2 \approx 327.1 \log n$ simultaneous flips. This bound is asymptotically tight.

We present two algorithms to improve the leading coefficient of this bound for transforming any plane triangulation into any other. The first of the two algorithms lowers this bound down to $4 \times (\frac{2}{\log \frac{12}{11}} + \frac{2}{\log \frac{9}{7}}) \log n + 2 \approx 85.8 \log n$. By processing and preprocessing the interior and exterior of the triangulation's Hamiltonian cycle parallelly in an interlaced fashion, we make further improvement of the algorithm from $\approx 327.1 \log n$ down to $\frac{12}{\log \frac{6}{5}} \log n + 2 \approx 45.6 \log n$.

Acknowledgements

This gives me immense pleasure to acknowledge the inspiring guidance of Dr. Jean-Lou De Carufel. He has dedicated an enormous amount of time in familiarizing me with the field of computational geometry. Our interactions have given me an appreciation towards the deep insights required in postulating, and the rigorousness required in proving the said postulates in this field. I thank Dr. Anil Maheshwari and Dr. Vida Dujmović for carefully examining my thesis, and Dr. Paola Flocchini for chairing my thesis defence.

I would like to thank my father Dr. Mohammad Kaykobad who has always been a passionate follower of my works, living in the diametrically opposite part of the globe. He has been a constant source of inspiration whenever I needed it. My mother, Saleha Sultana Kaykobad, who has always been a great hand of support for me, accepted my isolation from her for many years only for me to have a rewarding and fruitful life. I thank my only brother Kayes Kaykobad, his wife Tanzina Khan, and my only beloved nephew Ervin Kaykobad for accepting my inability of spending adequate time with them and for the understanding and support they provided for me.

Contents

Declaration of Authorship	ii
Abstract	iii
Acknowledgements	iv
1 Introduction	1
1.1 Triangulations, Flips and Simultaneous Flips	1
1.2 Literature Review	1
1.3 Summary of Known Bounds on Flips	3
1.4 Scope and Objectives of the Thesis	3
1.5 Highlighted Results of the Thesis	4
1.6 Organization of the Thesis	5
2 Existing Results	6
2.1 Preliminaries	6
2.2 Bounds Related to Sequence of Flips	10
2.2.1 Wagner's Results	11
2.2.2 Mori et al.'s Bound	12
2.2.3 Other Bounds	15
2.3 Simultaneous Flips Preliminaries	16
2.4 Transforming Triangulations in $\mathcal{O}(\log n)$ Simultaneous Flips	20
2.4.1 Necessity of $\Omega(\log n)$ Simultaneous Flips	20
2.4.2 Related Definitions	21
2.4.3 Related Lemmas	22
2.4.4 Overview of the Algorithm	26
2.4.5 $G_1 \rightarrow G_H$: Transforming Into Hamiltonian Triangulation	27
2.4.6 $G_H \rightarrow G_\delta$: Introducing the First Dominant Vertex	30
2.4.7 $G_\delta \rightarrow \Delta_n$: Introducing the Second Dominant Vertex	30
2.4.8 $\Delta_n \rightarrow G_2$: Do the Reverse to Transform Δ_n to Target Triangulation	30
2.4.9 Transforming Any Maximal Outerplane Graph to Any Other of Equal Order	31
2.4.10 Transforming Any Triangulation to Any Other of Equal Order	31
3 A Modification to Bose et al.'s Algorithm	32
3.1 Improvement of Constants c_1 and c_2	32
3.2 Optimal Degree Restriction for Independent Set to Introduce a Dominant Vertex in a Maximal Outerplane Graph	40
4 Interlaced Algorithm	43
4.1 An Overview of the Algorithm	43
4.2 $G_H \rightarrow G'_H$: Transformation to Logarithmic Diameter	44

4.2.1	Overview of the Process	44
4.2.2	Definition	46
4.3	Creating Ears in G_{ear}^i	48
4.3.1	Properties of Preparation Method	49
4.3.2	Identifying the Earing Vertex Set I_{ear}^i	50
4.3.3	Identifying I_{avoid}^i by Partially Computing S_{ear}^i	52
	Steps for Partially Computing I_{avoid}^i	53
	Steps for Completing The Computation of I_{avoid}^i	54
4.3.4	Computing S_{prep}^i Using the Preparation Method	55
4.3.5	Completing the Computation of S_{ear}^i —Obtaining S^i	56
	Steps for Completing The Computation of S_{ear}^i	58
4.3.6	Flippability of S^i	59
4.3.7	Initiating the 1st Iteration	62
4.4	The Preparation Method	63
4.4.1	Properties of Preparation Method	63
4.4.2	Preliminaries	63
4.4.3	Requiring Avoiding Vertices to Have at Least One Internal Chord	64
4.4.4	Maximal Outerplane Graph Partitioning Algorithm	64
4.4.5	Colouring of the Vertices and the Overall Goal	66
4.4.6	Procedure GENERATESUBGRAPHS(G, I_{avoid}) Generates Partitioned Subgraphs	66
4.4.7	Flipping All Provisioning Chords of a Partitioned Subgraph	71
4.4.8	Categorization of the Red Vertices	74
	Case 1: Non-priority BRB vertex	77
	Case 2: Non-priority BRG vertex	77
	Case 3: Non-priority GRG vertex	77
	Case 4: Priority BRB vertex	77
	Case 5: Priority BRG vertex	78
	Case 6: Priority GRG vertex	78
4.4.9	Provisioning Chord Sufficiency	78
4.5	$G'_H \rightarrow \Delta_n$: Introducing Two Dominant Vertices	82
4.6	Complexity Analysis for the Interlaced Algorithm	83
	$G_1 \rightarrow G_H$: Transforming Into Hamiltonian Triangulation	83
	$G_H \rightarrow G'_H$: Transformation to Logarithmic Diameter	84
	$G'_H \rightarrow \Delta_n$: Introducing Two Dominant Vertices	84
	$\Delta_n \rightarrow G_2$: Do the Reverse to Transform Δ_n to Target Triangulation	84
5	Conclusion	86
5.1	Open Problems	87
	Bibliography	88
	Alphabetical Index	90

List of Figures

2.1	Example of graphs	7
2.2	Diagonal flip	9
2.3	Canonical triangulation	10
2.4	Introducing dominant vertex via Wagner's method	11
2.5	Transforming Hamiltonian triangulation to Δ_n via sequential flips	13
2.6	Removing separating triangles via flips	14
2.7	Simultaneous flip	16
2.8	Consecutivity	17
2.9	Bad pair	17
2.10	Blocking edge	17
2.11	Blocking edge is flippable	19
2.12	An edge's corresponding bad edge and blocked edge are consecutive	19
2.13	Necessity of $\Omega(\log n)$ simultaneous flips	20
2.14	Dual tree of maximal outerplane graph	22
2.15	Introducing a dominant vertex	24
2.16	Requirement for Hamiltonian triangulation via simultaneous flip	28
2.17	Transforming into Hamiltonian Triangulation	29
3.1	No two consecutive chords can both be blocked	33
3.2	No three blocking chords can share a face	34
3.3	Vertices of I_5 must have 2 non-consecutive chords	37
4.1	Overview of the Interlaced Algorithm	45
4.2	Avoiding vertex set I_{avoid}^i	47
4.3	Example of Chord-path	47
4.4	Preliminaries on the order of flipping Consecutive earing chords	49
4.5	Earing vertex's last chord is flippable iff the vertex is not avoiding	50
4.6	Both earing chords cannot block provisioning chords	53
4.7	Vertices of I_{avoid}^i have at least 1 local chord each in G_{prep}^i	56
4.8	Neither earing chords are 2-chords	57
4.9	If a blocked earing chord produces foreign 2-chord of C_{prep}^i , its consecutive earing chord is flippable	59
4.10	Provisioning chord cannot form bad pair with earing chord	61
4.11	Avoiding vertices must have at least 1 internal chord	65
4.12	Partitioning a maximal outerplane graph into partitioned subgraphs	68
4.13	Every partitioned subgraph must contain at least one green vertex	70
4.14	Partitioned Subgraph BRB adjacency	70
4.15	Provisioning chords sharing a face	72
4.16	GRG vertex has multiple provisioning chords or it dominates a green vertex	75
4.17	Processing priority BRG vertices	78
4.18	Meeting singly provisioning chord requirement	82

List of Tables

1.1	Summary of bounds on flips	3
3.1	Optimal degree for ear decomposition	41
4.1	Types of red vertices in partitioned subgraphs	76

To Mom and Dad

Chapter 1

Introduction

Computational geometry is a branch of computer science that studies algorithms for solving geometrical problems arising in various fields including robotics, computer graphics, CAD/CAM and computational visualization. Other important applications of computational geometry include geographic information system, integrated circuit design, computer-aided engineering, and computer vision. Computational geometry deals with development of efficient data structures, and algorithms for solving problems defined in terms of geometrical objects like points, lines, circles and polygons. Some of the classical problems addressed in computational geometry are closest pair, convex hull, Voronoi diagram, Delaunay triangulation of a set of points in static or dynamic setting.

1.1 Triangulations, Flips and Simultaneous Flips

An *edge flip* is an operation that replaces an edge of a graph with a new edge, keeping the graph in the same class as the original graph. For graphs of equal vertices and edges, the number of edge flips required to transform one into another gives a sense of similarity between the two graphs. A *plane* graph is a graph that is embedded in a plane without any of its edges intersecting except at their vertices. A graph is *planar* if it admits a plane embedding. A *simple* graph does not contain any *loops* or *parallel edges*. For the class of simple plane graphs, if an edge lies on two triangular faces, the two faces form a quadrilateral of which that edge is a diagonal. By *flipping* of an edge we mean replacing this diagonal by the other diagonal of quadrilateral provided that after the flip the graph remains plane and simple. If a set of edge flip operations are done simultaneously with the resulting graph still staying in the same class, we call it a *simultaneous flip*. A *triangulation* is a simple plane graph in which every face is a triangle.

1.2 Literature Review

The problem of transforming one triangulation into another via diagonal flips has been well addressed in the literature (see Whitney (1931), Wagner (1936), Negami and Nakamoto (1993), Nakamoto (1996), Brunet, Nakamoto, and Negami (1996), Komuro (1997), Hurtado, Noy, and Urrutia (1999), Gao, Urrutia, and Wang (2001), Cortés et al. (2002), Mori, Nakamoto, and Ota (2003), Bose et al. (2007), Bose and Hurtado (2009), Bose and Verdonechot (2011), Souvaine, Tóth, and Winslow (2011), Bose et al. (2014), Frati (2015), Lubiw and Pathak (2015), Cardinal et al. (2018)). Wagner (1936) proved that any two triangulations can be transformed into each other,

up to homeomorphism, by a finite sequence of diagonal flips. Nakamoto (1996) further proved that any two bipartite quadrangulations of any closed surface are transformed into each other by two kinds of transformations, called the diagonal slide and the diagonal rotation, up to homeomorphism, if they have the same and sufficiently large number of vertices. Hurtado, Noy, and Urrutia (1999) proved that every triangulation on n vertices has at least $\frac{n-4}{2}$ flippable edges, and this bound cannot be improved in general. Cortés et al. (2002) proved that any two outer-triangulations on the same closed surface can be transformed into each other by a sequence of diagonal flips, up to isotopy, if they have sufficiently large and equal number of vertices. Galtier et al. (2003) proved that triangulation of any convex polygon can be transformed into another using $\mathcal{O}(\log n)$ simultaneous flips, and that this bound is tight. Bose et al. (2007) devised an $\mathcal{O}(\log n)$ algorithm to transform any triangulation to any other triangulation on the same number of vertices by simultaneous flips.

Souvaine, Tóth, and Winslow (2011) proved that every straight-line triangulation on n vertices contains at least $\frac{(n-4)}{5}$ simultaneously flippable edges. This bound is the best possible, and resolves an open problem posed by Galtier et al. (2003).

Diagonal flipping has received wide attention in the literature. Bose and Verdonschot (2011) discussed interesting history of diagonal flipping in *combinatorial triangulations*. Bose and Hurtado (2009) cited over 100 papers related to the topic. The first paper on the topic came out in 1936, where Wagner proved that any triangulation can be transformed to any other by flipping of diagonals. He also found that this process can be standardized by transforming through the so-called *canonical triangulation*. Construction of canonical triangulation can be facilitated if there is a *Hamiltonian cycle*. Chiba and Nishizeki (1989) presented an algorithm for finding a Hamiltonian cycle in linear time and storage in 4-connected planar graphs while the previously best algorithm given by Beauchamps (1978) uses $\mathcal{O}(n^3)$ time and space, where n is the number of vertices in a graph.

There have been some extremal results as to the maximum and minimum number of edges that can always be simultaneously flipped in an n -vertex triangulation. Bose et al. (2007) showed that at most $n - 2$ edges can ever be flipped simultaneously, that every triangulation has at least $\frac{(n-2)}{3}$ edges that can be flipped simultaneously and that there exist triangulations where at most $\frac{6(n-2)}{7}$ edges can be flipped simultaneously. They further proved that every combinatorial triangulation of at least six vertices has a simultaneous flip into a 4-connected triangulation, and that it can be computed in linear time. It follows that every triangulation has a simultaneous flip into a Hamiltonian triangulation. This result is used to prove that for any two n -vertex triangulations, there exists a sequence of $\mathcal{O}(\log n)$ simultaneous flips to transform one into the other. The total number of edges flipped in this sequence is $\mathcal{O}(n)$. They also showed that for every triangulation on n vertices there is a simultaneous flip of less than $\frac{2n}{3}$ edges to a 4-connected triangulation. The bound on the number of edges is tight, up to an additive constant.

Bose et al. (2007) found an upper bound on the number of simultaneous flips by using similar arguments as in Mori, Nakamoto, and Ota (2003) made in their paper.

Souvaine, Tóth, and Winslow (2011) showed that every triangulation on $n \geq 6$ vertices can be flipped into a Hamiltonian triangulation using a sequence of less than $\frac{n}{2}$ combinatorial edge flips. The previously best upper bound uses 4-connectivity as a means to establish Hamiltonicity. But in general, about $\frac{3n}{5}$ flips are necessary to reach a 4-connected triangulation. Their result improves the upper bound on the diameter of the flip graph of combinatorial triangulations on n vertices from $5.2n - 33.6$ to $5n - 23$.

1.3 Summary of Known Bounds on Flips

In Table 1.1, we have summarized results available in the literature both in respect of individual flips and simultaneous flips, and their lower and upper bounds for transforming combinatorial and geometric triangulations.

Some of the terms used in this table are defined in Chapter 2.

TABLE 1.1: Summary of bounds on flips

Problem	Combinatorial		Geometric	
	lower	upper	lower	upper
Sequential Flips on Triangulations	$\Omega(n)$	$6n - 30$ Mori, Nakamoto, and Ota (2003)	$\Omega(n^2)$ Hurtado, Noy, and Urrutia (1999)	$\mathcal{O}(n^2)$
Simultaneous Flips on triangulations/general polygons	$\Omega(\log n)$	$4 \times (\frac{2}{\log \frac{24}{13}} + \frac{2}{\log \frac{5}{3}}) \log n + 2$ Bose et al. (2007)	$\Omega(n)$ Hurtado, Noy, and Urrutia (1999)	$\mathcal{O}(n)$ Gao, Urrutia, and Wang (2001)
Simultaneous Flips maximal outerplane graphs	$\Omega(\log n)$	$\mathcal{O}(\log n)$ Bose et al. (2007)	$\Omega(\log n)$	$\mathcal{O}(\log n)$ Galtier et al. (2003)
Maximum number of flippable edges	$n - 2$ Gao, Urrutia, and Wang (2001)		$\lceil (n-4)/2 \rceil$ Hurtado, Noy, and Urrutia (1999)	
Size of a Maximum Flippable Edge set	$(n-2)/3$	$n - 2$ Bose et al. (2007)	$(n-4)/6$	$(n-4)/5$ Galtier et al. (2003)

1.4 Scope and Objectives of the Thesis

In this thesis we study simple and plane triangulation—in particular, how to transform an n -vertex triangulation G_1 into another n -vertex triangulation G_2 via simultaneous diagonal flips. The available literature is studied to identify opportunities for improvement.

Properties of *bad pairs*, *blocked edges* and *blocking edges* will be further studied and exploited under the context of *empty cycles* to devise a more efficient algorithm for transforming an arbitrary triangulation to the one with logarithmic diameter.

In fact we have devised two algorithms, the first one in Chapter 3 and the second one in Chapter 4 that have resulted in improvement of leading coefficients of Bose et al. (2007) algorithm. The first one ensures transformation of any triangulation into another by $85.8 \log n$ simultaneous flips, whereas the second one does the job in $45.6 \log n$ simultaneous flips compared to $327.1 \log n$ of Bose et al. (2007) algorithm.

1.5 Highlighted Results of the Thesis

This thesis has the following important results that led to the improvement of the algorithm for transforming a triangulation into another by simultaneous flips.

Lemma 3.3. *Given a triangulation G with a cycle C , no two consecutive chords of C can both be blocked.*

Lemma 3.11. *Let G be a triangulation with an empty cycle C , with a set S of blocking external chords of C . Then at least half of the chords in S can be simultaneously flipped.*

The above results lead to the following theorem that reduces the complexity of transforming a maximum outerplane graph into one diameter of whose dual tree is $\mathcal{O}(\log n)$.

Theorem 3.13. *Let G be a maximal outerplane graph on n vertices. Then G can be transformed by a sequence of at most $c_1 \log n$ simultaneous flips into a maximal outerplane graph X such that the diameter of its dual tree X^* is at most $c_1 \log n$, where $c_1 = \frac{2}{\log \frac{9}{7}} \approx 5.51$.*

Lemma 3.14. *Let G be a triangulation and C be an empty cycle of G with n vertices. (G may have more than n vertices.) Then G can be transformed by a sequence of at most $c_2 \log n$ simultaneous flips into a triangulation X in which C is an empty cycle and the diameter of the dual tree of $X \setminus C$ is at most $c_2 \log n$. Here $c_2 = 2 / \log \frac{12}{11} \approx 15.93$.*

The above results of Chapter 3 have been embodied to obtain a better algorithm for transformation of a triangulation into one, for which the dual tree has $\mathcal{O}(\log n)$ diameter using reduced number of simultaneous flips.

Theorem 3.15. *Let G_1 and G_2 be triangulations on n vertices. There is a sequence of at most $2 + 4(c_1 + c_2) \log n \approx 85.8 \log n$ simultaneous flips to transform G_1 into G_2 , where $c_1 = \frac{2}{\log \frac{9}{7}} \approx 5.52$ and $c_2 = \frac{2}{\log \frac{12}{11}} \approx 15.93$.*

The next lemma gives the intuition that given a Hamiltonian triangulation, it may be possible to make the diameter of the dual tree of its two maximal outerplane graphs in an interlaced fashion.

Lemma 4.24. *No provisioning chord can form a bad pair with any earing chord.*

For a Hamiltonian triangulation, the next lemma introduces a method to preprocess the other maximal outerplane graph while creating ears in one.

Lemma 4.45. *Given a maximal outerplane graph G and a set on non-consecutive vertices I_{avoid} on its maximal cycle C , an ordered set of two simultaneous flip sets $S_{\text{prep}} := (S_{\text{prep}_1}, S_{\text{prep}_2})$ can be computed, such that the graph $G \langle S_{\text{prep}} \rangle$ satisfies the following properties:*

- 1 *The simultaneous flip sets in S_{prep_1} and S_{prep_2} consist of only provisioning chords of G and $G' := G \langle S_{\text{prep}_1} \rangle$ respectively.*
- 2 *Every vertex in I_{avoid} either has at least 3 internal chords in $G \langle S_{\text{prep}} \rangle$, or it dominates a vertex of $G \langle S_{\text{prep}} \rangle$ not in I_{avoid} .*
- 3 *No two singly provisioning chords of the same vertex of I_{avoid} is in S_{prep_1} .*

The above results of Chapter 4 led to a further improvement to the algorithm for transformation of a triangulations.

Theorem 4.48. *Let G_1 and G_2 be two triangulations on n vertices. There is a sequence of no more than $c \log n + 2$ simultaneous flips to transform G_1 into G_2 , where $c = \frac{12}{\log \frac{6}{5}} \approx 45.6$.*

1.6 Organization of the Thesis

In Chapter 1 we give an introduction to our problem and briefly review the available literature on the topic, discuss scope and objectives of the thesis and its organization. In Chapter 2 we discuss some of the important results available in the literature on diagonal flips, provide a brief explanation, and sketch proofs of these results for ready reference and completeness of the exposition.

We present two new algorithms in this thesis. The first of these algorithms is a direct modification of the algorithm introduced by Bose et al. (2007) and presented in Chapter 3. Here we have chosen vertices of the independent set I of degree not exceeding 5 for transforming a larger fraction of vertices into ears. We have also exploited some properties of blocking and blocked chords to reduce loss incurred in Bose et al. (2007) algorithm. In Chapter 4 we present our second algorithm which further reduces the number of simultaneous flips required to transform any triangulation into any other with the same number of vertices. In Chapter 5 we present conclusions and provide a list of open problems that can be pursued for future research in the area. The thesis ends with a bibliography and an index.

In this thesis, all logarithms used are to the base 2. So we use $\log n$ to mean $\log_2 n$. We also use abbreviation WLOG for the phrase "without loss of generality" to reduce the text size. Similarly, the phrase 'if and only if' is abbreviated as IFF.

Chapter 2

Existing Results

There have been a number of works on diagonal flips in *(planar) triangulation*. Wagner showed that the process of transformation of a triangulation into another by diagonal flips can be standardized by transforming each of them into a *canonical triangulation*. In this chapter we discuss some important results together with their proofs. We start with some important definitions.

2.1 Preliminaries

Let us first start with some definitions.

Definition 2.1 (Graph). A graph G is an ordered pair $G = (V, E)$, where V is a set of vertices and E the set of edges. Each edge is a pair $\{u, v\}$, where $u \in V, v \in V$ and $u \neq v$ and is expressed by $uv \in E$.

The *order* of a graph is the cardinality of its vertex set and is denoted by n . The number of edges incident to a vertex v in a graph G is called its *degree*, and is written as $\deg_G(v)$ or just $\deg(v)$ when there is no ambiguity. If two vertices of a graph are connected by an edge then they are called *adjacent*, and otherwise *non-adjacent*. Notice that according to Definition 2.1, a graph can neither have multiple edges connecting the same pair of vertices (called *parallel edges*), nor have any edge that connects a vertex to itself (called *loop*). Graphs without loops and parallel edges are called *simple graphs*. **In this thesis unless specifically mentioned, all graphs are simple**, and thus fall into our definition of graph. Figures 2.1a, 2.1b, and 2.1c are examples of graphs. Figure 2.1d is not a simple graph as it contains parallel edges and therefore in the scope of this thesis is not considered a graph.

Definition 2.2 (Subgraph). Given two graphs $G = (V, E)$ and $G' = (V', E')$, if $V' \subseteq V$ and $E' \subseteq E \cap (V' \times V')$, then G' is called a *subgraph* of G . And G is called the *supergraph* of G' .

Definition 2.3 (Induced Subgraph). Given a graph $G = (V, E)$ and its subgraph $G' = (V', E')$, G' is called a *vertex induced subgraph* or in short an *induced subgraph* of G , if there exists a vertex set $S \subseteq V$ such that for all edges $v_i v_j \in E$, the vertices $v_i, v_j \in S$ if and only if $v_i v_j \in E'$.

Definition 2.4 (Edge Induced Subgraph). Given a graph $G = (V, E)$ and its subgraph $G' = (V', E')$, if for all vertex $v \in V$, $v \in V'$ if and only if v is incident to an edge in E' , then G' is called an *edge induced subgraph* of G by E' .

Definition 2.5 (Complete Graph). A graph of order n , where every pair of vertices is connected by an edge, is a *complete graph* and is represented as K_n .

FIGURE 2.1: Example of Graphs

Definition 2.6 (Complete Bipartite Graph). A complete bipartite graph is a graph $G = (V, E)$ where every vertex $v \in V$ can be assigned to one of two sets S_1 and S_2 , such that no pair of vertices in the same set are connected and every pair of vertices from different sets are. If $|S_1| = m$, $|S_2| = n$, then the complete bipartite graph is referred as $K_{m,n}$.

Definition 2.7 (Edge Subdivision). Given a graph $G = (V, E)$, the edge subdivision operation for an edge $uv \in E$ is the deletion of the edge uv and adding two edges uw and vw such that w is a new vertex in the graph. The obtained new graph after the operation is $G' := (V' := \{V \cup \{w\} \mid w \notin V\}, E' := \{(E \setminus \{uv\}) \cup \{uw, vw\}\})$.

Definition 2.8 (Graph Subdivision). Given a graph $G = (V, E)$, any graph is called a subdivision of G , if it can be generated by a sequence of edge subdivision operations on G .

Definition 2.9 (Planar Graph). A graph is said to be planar if it can be embedded in the plane with edges intersecting only at their endpoints.

Definition 2.10 (Plane Graph). If an embedding in the plane has been provided for a planar graph such that no two edges in the given embedding intersect except at their endpoints, then the embedded graph is called a plane graph.

Figure 2.1b does not correspond to a planar graph. Although K_4 is planar, the embedding of K_4 given in Figure 2.1c is not a plane graph. Figure 2.1a and 2.1d are examples of plane graphs. Kuratowski (1930) proved that a graph is planar IFF it does not contain any subdivision of K_5 or $K_{3,3}$.

Definition 2.11 (Path). Given a graph $G = (V, E)$, a path in G is a finite sequence $(v_0, v_1, \dots, v_k)$ of vertices such that for all $0 \leq i \leq k-1$, $v_i v_{i+1} \in E$ and $\forall i, j \{ 0 \leq i, j \leq k-1 \mid v_i = v_j \iff i = j \}$.

Path can also be denoted by the sequence of edges $(v_0 v_1, v_1 v_2, \dots, v_{k-1} v_k)$. The length of a path is the number of edges in it. In this case it is k . If there is a path between every pair of vertices of a graph then the graph is called *connected*.

Definition 2.12 (Cycle). Given a graph $G = (V, E)$, a cycle or k -cycle is a sequence of vertices $(v_0, v_1, \dots, v_{k-1}, v_0)$, where $(v_0, v_1, \dots, v_{k-1})$ form a $(k - 1)$ -path, and $v_{k-1}v_0 \in E$.

In this thesis every 3-cycle is called a *triangle*.

Definition 2.13 (Hamiltonian Cycle). Let $G = (V, E)$ be a graph of order n . A Hamiltonian cycle in G is an n -cycle.

Definition 2.14 (Chord). Given a graph $G = (V, E)$ with a cycle C , any edge whose both endpoints connect two non-adjacent vertices of C is called a chord of C . If G is a plane graph, a chord is called an internal chord if it passes through the interior of its cycle. Otherwise, it is called an external chord.

Definition 2.15 (k -Vertex-Connected). A graph $G = (V, E)$ of order $> k$ is a k -vertex-connected or in short k -connected graph if it cannot be made disconnected by deletion of fewer than k vertices.

Definition 2.16 (Tree). A graph $G = (V, E)$ is a tree if there is a unique path between every pair of its vertices.

A tree of n vertices has exactly $n - 1$ edges and no cycles.

Definition 2.17 (Triangulation). A triangulation is a plane graph in which every face is a triangle.

Definition 2.18 (Combinatorial Embedding). An embedded planar graph uniquely defines cyclic order of edges incident to the same vertex. The set of all these cyclic orders is called a rotation system and often combinatorial embedding. Embeddings following the same rotation system are considered to be equivalent and the corresponding equivalence class of embeddings is called combinatorial embedding.

Definition 2.19 (Region, Face). Given a plane graph $G = (V, E)$, every cycle of G separates the plane into two distinct regions. If a region does not contain another region, then it is called a face.

Given a combinatorial embedding of a planar graph, all plane graphs corresponding to the combinatorial embedding are said to be in an *equivalence class*. They all have exactly one external or unbounded face, with all the remaining faces being bounded by cycles. To be able to talk about the interior and exterior of cycles we nominate one face of the graph to be the *outerface*. In every figure in this thesis, the outerface is intentionally drawn as the unbounded face of the plane graphs—making the external face the outerface. Once the outerface has been defined, for any cycle C in the graph we can differentiate between the two regions it separates—one as the *internal* denoted by $int(C)$ and the other the *external* region of the cycle denoted by $ext(C)$. Neither $int(C)$ nor the $ext(C)$ contains the vertices of C .

Definition 2.20 (Outerplane Graph). Given a plane graph $G = (V, E)$, if its outerface contains all its vertices, then G is an outerplane graph.

Definition 2.21 (Outerplanar Graph). If a planar graph $G = (V, E)$ can be embedded in a plane such that it would be an outerplane graph, then G is an outerplanar graph.

Chartrand and Harary (1967) proved that any graph that does not contain a subdivision of K_4 or $K_{2,3}$ is outerplanar. Given an outerplanar graph, its Hamiltonian cycle is called the *maximal cycle*. Note that the maximal cycle of an outerplane graph separates its outerface from all the other faces.

FIGURE 2.2: Diagonal Flip

Definition 2.22 (Maximal Outerplane Graph). A maximal outerplane graph is an outerplane graph which cannot have any additional edges keeping it an outerplane graph.

Definition 2.23 (Maximal Outerplanar Graph). If a planar graph $G = (V, E)$ can be embedded in a plane such that it would be a maximal outerplane graph, then G is a maximal outerplanar graph.

As a consequence of having maximal number of edges, all faces excepting the outerface of maximal outerplanar graphs are triangles. Thus a maximal outerplanar graph can also be called an *outerplanar triangulation*.

Definition 2.24 (Isomorphism). Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if there are bijections $\theta : V_1 \rightarrow V_2$, and $\psi : E_1 \rightarrow E_2$ such that $uv \in E_1 \leftrightarrow \theta(u)\theta(v) = \psi(uv) \in E_2$.

Definition 2.25 (Seeing). Let $G = (V, E)$ be a graph and u, v, w form a 3-cycle face in G . We say that u sees the edge vw .

In Figure 2.2(a) left, the blue edge vw is seen by vertex x on the face vwx and vertex y on the face vwy . Observe that in triangulation with $n \geq 4$ vertices, every edge is seen by two distinct vertices.

Definition 2.26 (Flippable Edge). Let $G = (V, E)$ be a plane graph with an edge uv incident to two triangular faces xuv and yuv . Then the two triangles form a quadrilateral $xuyv$, where uv is a diagonal. If $xy \notin E$, then we say uv is flippable.

Definition 2.27 (Diagonal Flip). Let G be a plane graph with an edge uv incident to two triangular faces xuv and yuv . Then the two triangles form a quadrilateral $xuyv$, where uv is a diagonal. If uv is flippable, then replacing this diagonal uv with the other diagonal xy in the same region of the quadrilateral is called a diagonal flip or in short flip.

Definition 2.28 (Sequence of Flips). Given a graph $G_0 = (V, E)$, let S be a sequence of flip operations $(e_1, e_2, \dots, e_k)$ such that e_i is flippable in the graph G_{i-1} and it produces the graph G_i , then this sequence of flip operations is called a sequence of flips.

Note that for a valid diagonal flip the resulting graph must remain plain and simple. In Figure 2.2, vw is an edge of a triangulation (which by Definition 2.17 is both plane and simple), and (v, w, x) and (v, w, y) faces are incident to it, then x and y are distinct vertices unless $G = K_3$. Let G' be obtained from G by deleting edge vw from G and inserting edge xy between xv and xw (yv and yw). If G' is also a triangulation then vw is flippable, and G is flipped into G' by vw . Also notice that if vw is flipped in G' , then the new edge is inserted in the same region of the quadrilateral (x, w, y, v) . Thus flips are reversible operations. For any sequence of flips

FIGURE 2.3: The canonical triangulation has two dominant vertices (in blue)

$S = (e_1, e_2, e_3, \dots, e_i)$ that transforms a graph G_1 into another graph G_2 , there exists another sequence of flips $S' = (e'_i, e'_{i-1}, e'_{i-2}, \dots, e'_1)$ where e'_k is the corresponding edge obtained by flipping the edge e_k to transform G_2 back into G_1 .

Definition 2.29 (Dominant Vertex). Given a graph $G = (V, E)$, a vertex $v \in V$ is said to be dominant if it is adjacent to all other vertices in V .

Definition 2.30 (Canonical Triangulation). A canonical triangulation on $n \geq 3$ vertices, also known as standard triangulation, is a triangulation with two dominant vertices. A canonical triangulation on $n \geq 3$ vertices is denoted by Δ_n . (e.g. Figure 2.3).

All canonical triangulation on $n \geq 3$ vertices have the same combinatorial embedding, and thus for any given value of $n \geq 3$ the canonical triangulation is unique.

The following three lemmas are direct result of the Jordan curve Theorem. They are commonly used in finding properties related to flips in graphs.

Lemma 2.31. Given a plane graph $G = (V, E)$ with a cycle C and two vertices u and v such that u is in the interior of C and v is in the exterior of C , then uv cannot be an edge in G .

Lemma 2.32. Given a plane graph $G = (V, E)$ with a cycle C , if an edge e is seen by two vertices u and v such that u is in the interior of C and v is in the exterior of C , then e must be on C .

Lemma 2.33. Given a plane graph $G = (V, E)$ with a cycle C , if a face f of G is in the interior of C , then no vertex on f can be in the exterior of C .

Proof. Since f is in the interior of C , every edge e on f must either be on or in the interior of C , as otherwise f would overlap between the two regions of C . If a vertex v on f is in the exterior of C , then the edges of f incident to v must also be in the exterior of C . Thus no vertex of f can be in the exterior of C . $\square$

Wagner (1936) first introduced the idea of canonical triangulation, as an intermediate triangulation to transform to, when transforming one triangulation to another. All Δ_n triangulations are isomorphic, and as a result if any triangulation of n vertices can be transformed to Δ_n , then we can easily transform any triangulation of n vertices to any other on n vertices by first transforming the first triangulation to Δ_n and then follow the transformation of the other to Δ_n in reverse order.

2.2 Bounds Related to Sequence of Flips

In this section, we present some important results on sequence of flips, and for completeness sketch their proofs. The techniques and ideas involved are closely related to our work in Chapter 3 and Chapter 4.

FIGURE 2.4: Introducing dominant vertex via Wagner's method. Shaded region may be further triangulated. Zigzagged curves are paths, or their end-vertices are the same vertex.

2.2.1 Wagner's Results

Wagner (1936) was the first to study sequence of diagonal flips for transforming one triangulation into another. The following two lemmas are due to Wagner.

Lemma 2.34 (Wagner (1936), Theorem 4). *Any triangulation on n vertices can be transformed into a canonical triangulation Δ_n by a sequence of at most $n^2 - 7n + 12$ flips.*

Proof. If $n = 3$, we already have our canonical triangulation. So let us assume $n \geq 4$. To transform any triangulation G into a canonical triangulation we first fix a face which we call an *outerface* and select two arbitrary vertices a and b on the outerface to be made dominant (Figure 2.4). If a is not already a dominant vertex in G then there exists a face uvw in G such that two of its vertices u and v are adjacent to a while w is not. w can be either inside the triangle auv or outside it. Let uvx be the other face adjacent to uv . If w is inside auv then x must be outside, and to preserve planarity wx cannot be an edge of G . On the other hand, if w is outside auv then x must be inside it, and again wx cannot be an edge of G . Since wx is the edge we get by flipping uv , uv is flippable. As a result of this flip the degree of w increases by 1. If w gets connected to a then we are done. Otherwise, we can repeat this process and since the graph is finite, w will eventually get connected to a .

Since G is a triangulation, the degree of w must be at least 3, so this will require no more than $n - 4$ flips for w to be connected to a . Observe that during the process of connecting w to a , none of the edges flipped are incident to a . Therefore, the next vertex will not require more than $n - 5$ flips to be adjacent to a and no edges incident to a will be lost in the process. In this way the number of flips that suffice for a to be dominant is

$$\sum_{i=1}^{n-4} i = \frac{(n-4)(n-3)}{2} = \frac{n^2 - 7n + 12}{2}$$

Not more than the same number of flips will suffice to make b also dominant. So the result follows. $\square$

By using the above result twice Wagner solves the problem of transforming any triangulation T_1 into any other T_2 .

Theorem 2.35 (Wagner (1936), Theorem 4). *Any pair of triangulations T_1 and T_2 on n vertices can be transformed into each other by a sequence of at most $2n^2 - 14n + 24$ flips.*

The idea is to transform T_1 into Δ_n using $n^2 - 7n + 12$ flips. Since we can transform T_2 into Δ_n using the same number of flips, we need to execute these flips in reverse order to get T_2 from Δ_n .

Negami and Nakamoto (1993) prove a better quadratic upper bound on the number of flips for the above problem. To do so instead of increasing the degree of the would-be-dominant vertices a and b directly, they reduce the degree of the third vertex c adjacent to a and b down to 3 on the outerface of G . Once the degree of c becomes 3 they recursively repeat the process on the remaining subgraph $G := G \setminus \{c\}$ until the subgraph becomes a triangle; rendering the original graph into the canonical triangulation.

Knowing that any triangulation can be transformed into any other via a sequence of flips, one may ask what the minimum number of flips required is for this transformation. To address problems of this nature the concept of flip graphs is useful.

Definition 2.36 (Flip Graph). *The flip graph is the graph whose vertices correspond to non-isomorphic combinatorial triangulations and whose edges connect pairs of triangulations that can be obtained from one another by flipping a single edge.*

The diameter of the flip graph is the maximum number of flips required to transform any triangulation to any other of equal order.

Wagner showed that for any triangulation of n vertices the diameter of the flip graph is in $\mathcal{O}(n^2)$. Komuro (1997) lowered this bound to $8n - 54$ for $n \geq 13$ and $8n - 48$ for $n \geq 7$ confirming that the diameter of flip graph is rather $\mathcal{O}(n)$.

2.2.2 Mori et al.'s Bound

Mori, Nakamoto, and Ota (2003) also studied the problem of transforming a triangulation into any other. They proved that given a maximal outerplane graph a dominant vertex can be constructed in linear time, and as a consequence found a better upper bound for transforming any *Hamiltonian triangulation* into the canonical triangulation.

Definition 2.37 (Hamiltonian Triangulation). *Let $G = (V, E)$ be a triangulation with a Hamiltonian Cycle, then G is called a Hamiltonian triangulation.*

Lemma 2.38. *Every internal edge of a maximal outerplane graph is flippable.*

Proof. Let G be a maximal outerplane graph with an internal chord ab blocked by another edge xy . Then the induced subgraph $G[\{a, b, x, y\}]$ must form K_4 . But Chartrand and Harary (1967) proved that no maximal outerplane graph can contain K_4 . Thus every internal chord of such a graph must be flippable. □

Lemma 2.39 (Mori, Nakamoto, and Ota (2003), Lemma 8). *Any vertex v in a maximal outerplane graph on n vertices can be made dominant by $n - 1 - \deg(v)$ flips.*

Proof. If v is not dominant in a maximal outerplane graph, then there exists a triangle vxy where xy is not an edge of the outerface. By virtue of Lemma 2.38 xy must be flippable, and flipping xy would increase the degree of v . Since each of the flips made this way would increase the degree of v by 1, the total number of flips necessary to make v dominant would be at most $n - 1 - \deg(v)$. □

All triangulations of order up to 5 have at least two dominant vertices and are isomorphic to any other triangulations of equal order. So while Theorem 2.40 and 2.45 require the order of the graph to be at least 6, their results hold for any order when the upper bound is accepted as 0 whenever it is negative.

FIGURE 2.5: Transforming Hamiltonian triangulation G to Δ_8 via sequential flips. Hamiltonian cycle drawn in blue.

Theorem 2.40 (Mori, Nakamoto, and Ota (2003), Proposition 9). *Any Hamiltonian triangulation on $n \geq 6$ vertices can be transformed into the canonical triangulation by at most $2n - 10$ flips, preserving the existence of Hamiltonian cycle.*

Proof. If a graph G is Hamiltonian, then there exists a Hamiltonian cycle C that visits every vertex of the graph and thus it can be decomposed into two maximal outerplane subgraphs G_1 and G_2 as illustrated in Figure 2.5. The first subgraph G_1 contains all the vertices and edges of the cycle and all the edges in the interior of the cycle (called *internal chords*) and the other subgraph G_2 contains all the vertices and edges on the cycle and all the edges that are in the exterior of the cycle (called *external chords*). We now induce two dominant vertices—one on G_1 and the other on G_2 . To do so, we select one vertex, b in G_1 that has degree 2 on G_2 . This ensures that there are no chords in G_2 adjacent to b . Since every triangulation of order $n \geq 4$ is 3-connected, G must be 3-connected and since b has no additional neighbours in G_2 , the degree of b in G_1 is at least 3. Thus by Lemma 2.39, we can make b dominant by at most $n - 4$ flips. These flips will all be valid since b does not have any neighbours in G_2 except for its two adjacent vertices on the cycle C .

Note that since G_1 has b as a dominant vertex, all chords in G_1 are connected to b . Thus b cannot have any incident chords in the maximal outerplane graph G_2 . Thus b must be an ear, and therefore, $G'_2 = G_2 \setminus \{b\}$ is also a maximal outerplane graph containing $n - 1$ vertices and $2n - 5$ edges. Since the order of G'_2 is ≥ 5 , it must contain a vertex of degree at least 4. By applying Lemma 2.39 again we can create another dominant vertex in G'_2 as well with at most $n - 6$ flips. Finally, We superimpose the two resulting maximal outerplane graphs back into a Hamiltonian triangulation such that the Hamiltonian cycle stays intact, and planarity is maintained. So $2n - 10$ flips are sufficient to transform any non-canonical Hamiltonian triangulation into a canonical triangulation. $\square$

We now have a method to transform any Hamiltonian triangulation to a canonical triangulation. If we can convert any triangulation to a Hamiltonian triangulation with a small number of flips, we can transform any triangulation to any other of equal order in an efficient manner.

(a) z has not yet been placed on the plane (b) z must be the same as c .

FIGURE 2.6: Flipping ab (red) of the separating triangle abc introduces the xy (blue), creating a new separating triangle xyz . Shaded region may be further triangulated, dashed edges may not be distinct.

Definition 2.41 (Separating Triangle). Given a connected graph $G = (V, E)$ with a triangle T in G such that the deletion of vertices of T results in a disconnected graph then T is called a separating triangle.

In Figure 2.2(a) left wvz is a separating triangle since removing its three vertices disconnects the graph.

Lemma 2.42 (Bose et al. (2007) Lemma 2.2). An edge in a separating triangle of a triangulation is flippable.

Proof. Let vw be an edge in a separating triangle vwk , seen by the two vertices p and q . This makes pvw and qvw faces and thus neither p nor q can be the same as k as then vwk would no longer be a separating triangle. Thus one of them must be in the interior of T , and the other in the exterior. Hence to preserve planarity, pq cannot be an edge of T . So vw is flippable. $\square$

Lemma 2.43 (Mori, Nakamoto, and Ota (2003), Lemma 11). Given a triangulation with $n \geq 6$ vertices, flipping any edge of a separating triangle $D = abc$ will remove that separating triangle. This never introduces a new separating triangle, provided that at least one of the following two conditions is satisfied:

- The selected edge belongs to multiple separating triangles.
- None of the edges of D belong to multiple separating triangles.

Proof. Let $D = abc$ be a separating triangle. Due to Lemma 2.42 every edge of D is flippable. We prove this lemma by contrapositive. Let ab be the edge of D that when flipped produces the edge xy creating another separating triangle $D' = xyz$ as illustrated in Figure 2.6a. Since the flip removes an edge of D , it is no longer a separating triangle. If z is in the interior or exterior of D then one of x and y cannot be connected to z . Thus the only way for $D' = xyz$ to be a triangle is for z to be on the 3-cycle D . Since xya and xyb are faces of G , and since a separating triangle must contain vertices in both its interior and exterior, $D' \notin \{xya, xyb\}$. Thus D' must be xyz making $z = c$ as illustrated in Figure 2.6b.

Since $n \geq 6$ and our construction so far uses only 5 vertices, one of the faces ayc, byc, axc or bxc must be a separating triangle as well. This would result in either the edge ac or the edge bc belonging to multiple separating triangles. Furthermore,

because a is outside the cycle (x, y, c) while b is inside, they cannot both be neighbours of any other vertices apart from x, y and c . Since abx and aby are faces, and $abc = D$, ab cannot be part of any other separating triangle while at least one of bc and ca is—which is the negation of the premise. $\square$

Lemma 2.44 (Mori, Nakamoto, and Ota (2003), Lemma 10). *Any triangulation on n vertices can be made 4-connected by at most $n - 4$ flips.*

Proof. The proof is done by induction on n . The base case can be obtained for $n = 4$ giving us a K_4 graph which contains no separating triangles, and we do not need any flips at all. For the induction step, let us assume that our graph G with a fixed outerface has a separating triangle T that divides the graph into two induced subgraphs $G_1 := G \setminus \text{int}(T)$ and $G_2 := G \setminus \text{ext}(T)$.¹ By induction, both G_1 and G_2 have at most $n_1 - 4$ and $n_2 - 4$ separating triangles respectively where n_1 is the number of vertices in G_1 and n_2 is the number of vertices in G_2 . Since the separating triangle that divides the graph into G_1 and G_2 does not contribute to any separating triangles in either of them, G can have at most $n_1 - 4 + n_2 - 4 + 1 = (n_1 + n_2 - 3) - 4 = n - 4$ separating triangles. So $n - 4$ flips are sufficient to break all separating triangles. Since any triangulation without separating triangles is at least 4-connected the result follows by using Lemma 2.43. $\square$

Theorem 2.45 (Mori, Nakamoto, and Ota (2003), Theorem 4). *Any two triangulations on $n \geq 6$ vertices can be transformed into each other by at most $6n - 30$ flips.*

Proof. Since every face of a triangulation is a 3-cycle, no triangulation can be made disconnected by deleting less than 3 vertices—making all triangulations 3-connected. A triangulation of order ≥ 4 with no separating triangles cannot be made disconnected by deleting 3 vertices and thus any such triangulation is 4-connected. Whitney (1931) proves that a 4-connected triangulation is Hamiltonian. So we can use Theorem 2.40 to transform this Hamiltonian triangulation G into the canonical triangulation using $2n - 10$ flips. Theorem 2.40 begins by applying Lemma 2.39 to introduce a dominant vertex v using $n - 1 - \deg(v)$ flips. Note that G is not only Hamiltonian, it is also 4-connected. Thus every vertex of G must have a degree of at least 4 (instead of 3). Thus Lemma 2.39 would require at most $n - 5$ flips (instead of $n - 4$ as counted in Theorem 2.40) to introduce the first dominant vertex. Thus using Theorem 2.40 we would only need at most $2n - 11$ flips to convert G into Δ_n . And the total number of flips required to transform any triangulation with $n \geq 6$ vertices to the Hamiltonian triangulation and then the canonical triangulation is $(n - 4) + (2n - 11) = 3n - 15$ flips.

Finally, to transform any triangulation of $n \geq 6$ vertices to any other triangulation of equal order, we first follow the sequence of flips to transform the first triangulation to Δ_n . Then we transform Δ_n to the second triangulation, by applying the reverse sequence of the flips required to transform the second triangulation to Δ_n . The total number of flips required for the transformation is at most $(3n - 15) \times 2 = 6n - 30$. $\square$

2.2.3 Other Bounds

Bose et al. (2014) prove that bound on the number of flips provided by Mori, Nakamoto, and Ota (2003) is not tight. They give a tighter bound stating that $\frac{(3n-6)}{5}$ flips are

¹Vertices of T are not present in either of G_1 and G_2 .

FIGURE 2.7: $S = \{vw, xy\}$ is a valid simultaneously flippable set

sufficient and in some cases necessary to make a triangulation 4-connected. By combining this result with the bound for 4-connected triangulations by Mori, Nakamoto, and Ota (2003), they also prove that any pair of triangulations on n vertices can be transformed into each other by at most $5.2n - 24.4$ flips.

Sleator, Tarjan, and Thurston (1988) proved that any maximal outerplane graph can be transformed to any other of equal order by no more than $2n - 10$ diagonal flips for $n > 12$. This bound on the number of sequential flips for maximal outerplane graphs is tight.

2.3 Simultaneous Flips Preliminaries

Researchers also considered the problem of transforming one triangulation into another of equal order via simultaneous flips. Mori, Nakamoto, and Ota (2003) proved that to do so $6n - 30$ simultaneous flips are sufficient. Their proof uses a two-step approach. First, it converts the triangulation into a Hamiltonian triangulation. In the second step, they introduce a dominant vertex in each of the two maximal outerplane graphs defined by a Hamiltonian cycle in the graph.

In this section, we present some concepts related to the results by Bose et al. (2007) concerning transformation of one triangulation into another via simultaneous flips. The algorithm presented in the aforementioned paper is what this thesis builds upon. Let us first present a few definitions.

Definition 2.46 (Flippable Set). *Given a plane graph $G = (V, E)$, a (simultaneously) flippable set is a set S of edges in G that when simultaneously flipped, the resulting graph denoted by $G\langle S \rangle$ stays plane and simple.*

Like sequential flips, simultaneous flip operations are reversible. Let S be a flippable set in a triangulation G such that $G' := G\langle S \rangle$. Furthermore, Let S' be the set consisting of the edges that are obtained by flipping edges of S . Because in any plane embedding of G , every edge $e' \in S'$ and its corresponding edge $e \in S$ is in the same region of the quadrilateral they are chords (diagonals) of—flipping e' in G' produces e . So flipping every edge in S' in G' would produce all the edges in S transforming G' back to G —making the operation of simultaneous flip is reversible.

Definition 2.47 (Consecutive Edges). *If two edges of a triangulation are incident to a common face then they are called consecutive edges.*

Definition 2.48 (Bad Pair). *Let G be a triangulation with two distinct edges e_1 and e_2 , both seen by two distinct vertices v_1 and v_2 . Then e_1 and e_2 are called bad pair.*

In Figure 2.9 the edges xy and vw are both seen by the two vertices p and q . Thus flipping xy and vw simultaneously produces parallel edges. Here xy and pq form a bad pair.

FIGURE 2.8: Because xw and vy share the face $xvyw$, flipping them simultaneously no longer maintains planarity

FIGURE 2.9: xy and vw form a bad pair as they can both be seen by the vertices p and q

Definition 2.49 (Blocked Edge, Blocking Edge). Let $G = (V, E)$ be a triangulation with two edges $uv, xy \in E$ such that uv is seen by the vertices x and y . Then uv is called a blocked edge and xy its blocking edge.

In Figure 2.10a the edge ax is seen by b and c where bc is already an edge of the graph so flipping ax produces parallel edges. Here ax is blocked by bc .

Note that even if S is flippable, S may contain edges in G that are not flippable. E.g. in Figure 2.7 $S = \{vw, xy\}$ is a flippable set, despite the fact that neither of the edges vw and xy is flippable. Bose et al. (2007) prove that there are 3 necessary and sufficient conditions for a set S to be flippable. These conditions are provided below.

FIGURE 2.10: For every edge e in S , either vw is flippable or the edge that blocks vw is also in S

Lemma 2.50 (Bose et al. (2007) Lemma 2.1). *A set of edges S in a triangulation $G \neq K_3$ is flippable if and only if:*

1. *no two edges in S are consecutive,*
2. *no two edges in S form a bad pair, and*
3. *for every edge $vw \in S$ either vw is flippable, or the edge that blocks vw is also in S .*

Proof. $\Leftarrow$ Every edge in a triangulation lies on two faces, which form a quadrilateral. When an edge is flipped, the new edge is placed in between the edges of the quadrilateral in their corresponding connecting vertices' cyclic order. As illustrated in Figure 2.8, when two consecutive edges xw and vw seen by $\{q, v\}$ and $\{x, p\}$ respectively are flipped we get vq and xp —each passing through the interior of the cycle (x, q, w, p, v) . Notice that the new edge vq separates the two points x and p in the interior of the cycle (x, q, w, p, v) , denying the edge (internal chord) xp in the interior of the cycle. This is why when consecutive edges are flipped, their corresponding flipped edges (vq and xp) intersect destroying the planarity of the new graph.

When two edges forming a bad pair are flipped simultaneously, it produces parallel edges—which is not allowed in simple graphs. E.g. in Figure 2.9 the edges xy and vw form a bad pair because they can both be seen by p and q , resulting in both their flipped edges to be pq .

Finally, if an edge of the set S is blocked by another edge in the graph, then to avoid parallel edges in the new graph obtained by flipping the edges of the set, the blocking edge must be removed by inserting it also in the set. In Figure 2.10a we see that the edge ax is blocked by the edge bc . Clearly for ax to be flipped we must also flip bc , as shown in Figure 2.10b. Thus all three conditions listed in the lemma are necessary.

Proof. $\Rightarrow$ We now show that given a triangulation G , these conditions are sufficient. Since no two edges in S are consecutive, $G\langle S \rangle$ is a graph embedded in the plane and every face is a triangle. Suppose two edges e_1 and e_2 are parallel in $G\langle S \rangle$. Since G does not have parallel edges, not both of e_1 and e_2 are in G . If exactly one of e_1 and e_2 is in G , then Condition 3 fails. If neither of e_1 and e_2 is in G , then the edges in S that are flipped to e_1 and e_2 form a bad pair. So these conditions are also sufficient for S to be flippable. $\square$

Lemma 2.51 (Bose et al. (2007) Lemma 2.4). *A blocking edge is flippable in a triangulation $G \neq K_4$.*

Proof. Let G be a triangulation with two edges pq and uv such that pq is blocking uv as shown in Figure 2.11. Then puv and quv must be faces of G . WLOG, let the vertex u be inside the triangle pqv . Let pqx and pqy be the two faces the edge pq is incident to. Then one of them must be *on* or *in* the interior of the cycle pqv while the other *on* or *in* the exterior of pqv . WLOG let y be *on* or *inside* pqv and x be *on* or *outside* pqv . x and y may not be distinct from other vertices of G . Because pqy is a face, y must be *on* or *inside* the cycle pqu .

Since G has more than 4 vertices, there must be at least one other vertex z in the graph located in the exterior of pqv or interior of pqu . If z is on the exterior of pqv then pqv cannot be a face while pqx is and hence in this case $x \neq v$. Then x must be strictly *in* the exterior of pqv while y is *on* or *in* the interior of pqu and hence in the interior of pqv . Thus in this case xy cannot be an edge of G . The same argument

FIGURE 2.11: Blocking edge is flippable. Shaded region may be further triangulated, dashed edges may not be distinct.

applies if z is inside pqu , as then $y \neq u$ and thus must be in the interior of pqu while x is in the exterior of pqu and thus xy again is not an edge of G . Thus xy is never an edge of G and so pq must be flippable in G . $\square$

Lemma 2.52 (Bose et al. (2007) Lemma 2.5). *Suppose that uv and xy are a bad pair in a triangulation both seen by vertices p and q . Suppose that uv blocks some edge ab . Then xy and ab are consecutive, and vw and xy are in a common triangle.*

(a) uv and xy have no vertex in common

(b) uv and xy have one vertex in common

(c) uv blocks ab and forms bad pair with xy

FIGURE 2.12: An edge's corresponding bad edge and blocked edge are consecutive. Shaded region may be further triangulated, dashed edges may not be distinct. (a) and (b) show the two possible cases. (c) depicts the final configuration

Proof. Let uv and xy be bad pairs seen by p and q in a triangulation G . Furthermore, let uv block an edge ab . In Figure 2.12a u, v, x, y are distinct vertices, where as in Figure 2.12b v and y are the same vertex. Since uv and xy are distinct edges, this covers all cases without loss of generality. Since uv blocks ab , uab and vab must be faces. Similarly, uv and xy are seen by p and q so puv, quv, pxy and qxy must be faces.

In Figure 2.12a neither a nor b can be strictly in the interior of (p, x, q, v) as then a or b would be in the interior of the cycle while u is in the exterior, resulting in ua or ub not being an edge. Similarly in Figure 2.12a and 2.12b neither a nor b can be in the exterior of (p, y, q, u) as then they would be in the exterior of the cycle while v is inside—disallowing the edge av or bv . Then ab must be on the cycle (p, y, q, u) or

(p, x, q, v) . Because u and v can see ab , neither of the vertices a and b can be the same as u or v . Then $\{a, b\} \subset \{p, q, x, y\}$.

Let us assume that $ab = pq$. If pq exists in G , it must either be in interior of the cycle (p, x, q, v) (Figure 2.12a), or in the exterior of the cycle (p, y, q, v) (Figure 2.12a and 2.12b). If pq lies in the interior of (p, x, q, v) then uab cannot be a face as u lies on the exterior of the cycle. Similarly if pq lies on the exterior of (p, y, q, v) then vab cannot be a face as v lies in the interior of the cycle. Thus $ab \neq pq$ and as a result at least one of the vertices between a and b must be distinct from p and q .

WLOG let a be the vertex distinct from p and q . Then $a \in \{x, y\}$.

In Figure 2.12a, if $a = x$ then au cannot be an edge as a and u would be separated by the cycle (p, y, q, v) . Similarly $a \neq y$ as the cycle (p, x, q, u) would separate a from v denying the edge av . Thus this case is impossible and uv and xy must have a common vertex.

In Figure 2.12b the only remaining candidate for a is x . Let $a = x$. Then $ua = ux$ must be an edge. WLOG let b be q . Then $qx = ab$ is blocked by uv . Finally, since xvq is a face, ab is consecutive to $vx = xy$ as illustrated in Figure 2.12c. $\square$

2.4 Transforming Triangulations in $\mathcal{O}(\log n)$ Simultaneous Flips

In this section, we explain the algorithm of Bose et al. (2007) for transforming any triangulation to any other using logarithmic number of simultaneous flips. First, in Section 2.4.1 we show that $\Omega(\log n)$ is necessary. Then we start with some definitions and a few important lemmas and conclude the chapter with a detailed explanation of the algorithm.

2.4.1 Necessity of $\Omega(\log n)$ Simultaneous Flips

Lemma 2.50 Condition 1 asserts that no two edges in a simultaneous flippable set can belong to a common face. This restricts reducing degree of any vertex by at most half in each simultaneous flip. In Figure 2.13, we have a canonical triangulation Δ_n with the two dominant vertices v_1 and v_2 having a degree $n - 1$ each. We want to transform this triangulation into one whose vertex degree is at most 6. Since in every simultaneous flip only nonconsecutive chords can be flipped, degree of a vertex can at most be halved. So this will necessitate at least $\lceil \log \frac{n-1}{6} \rceil$ simultaneous flips to bring the degree of v_1, v_2 down to 6 from $n - 1$.

FIGURE 2.13: Necessity of $\Omega(\log n)$ simultaneous flips. Structure can be arbitrarily repeated in the shaded region.

2.4.2 Related Definitions

Definition 2.53 (Empty Cycle). *An empty region is a region that does not contain any vertex strictly in its interior. A cycle C of a plane graph G is an empty cycle if one of its two regions is empty and is assigned as its local region. The other region is referred to as the foreign region of the empty cycle.*

If an empty cycle neither contains any vertices in its interior nor in its exterior, then it must be a Hamiltonian cycle passing through all the vertices. Thus all Hamiltonian cycles define empty cycles. In fact, a Hamiltonian cycle defines two empty cycles—one is the cycle and its empty interior region as its local region, and the other is the cycle and its empty exterior region as its local region. In Chapters 2 and 3 no figure requires defining more than one empty cycle. So in all the figures the embedding is chosen such that the local region of the empty cycle is always in the interior of the cycle.

Definition 2.54 (k -regular). *A graph $G = (V, E)$ is called a k -regular graph if for all vertices $v \in V$, $\deg(v) = k$.*

Definition 2.55 (Connected Components). *Given a graph $G = (V, E)$, a connected component or in short component, of G is a subgraph of G in which any two vertices are connected by paths, and which is connected to no additional vertices in the supergraph.*

Definition 2.56 (Bridge). *Given a graph $G = (V, E)$, an edge $e \in E$, deletion of which increases the number of components in G , is called a bridge.*

Definition 2.57 (Dual Graph). *Given a graph $G = (V, E)$ with its set of faces F , the dual graph $G^* = (V^*, E^*)$ of G is also a graph with a bijection between F and V^* , and another bijection between E and E^* , such that for every edge $e \in E$ on two faces $f_i, f_j \in F$ and its corresponding edge $e^* \in E^*$, $e^* = v_i^* v_j^*$ where $v_i^*, v_j^* \in V^*$ are the corresponding vertices of the faces f_i and f_j .*

The dual graph connects two vertices IFF their corresponding faces in the primal graph share a region. Thus given a plane embedding of the primal graph, if we place the vertices of the dual graph in the interior of their corresponding faces of the primal graph, no edges of the dual graph would intersect each other except at their endpoints—making the dual graph of any planar graph planar.

Also, note that a dual graph of any planar graph is not necessarily simple as it is possible for a plane graph to have two faces having multiple edges in common, producing parallel edges in the dual graph. Or if a planar graph contains a bridge, it would produce a loop in its dual graph. However, since a triangulation cannot have a bridge, its dual graph cannot have loops. For a dual graph of a triangulation to have parallel edges, the primal triangulation must have two faces sharing at least two edges. For two triangles T_1 and T_2 to share two edges uv and vw , both the triangles must be described by the cycle (u, v, w) . Since a triangulation can have only one (u, v, w) cycle and since the only case (u, v, w) is a face in its interior as well as in its exterior is when the triangulation has exactly 3 vertices, dual graph of any triangulation of order > 3 cannot have parallel edges. Since we do not refer to the dual graph of a triangle, every graph in this thesis is simple.

Definition 2.58 (Dual Tree). *The dual tree G^* of a maximal outerplane graph G is the dual graph of G without the vertex corresponding to the outerface of G .*

A maximal outerplane graph has $n - 3$ chords and $n - 2$ faces excluding its outerface. Since every face excepting the outerface is a triangle, its dual tree must be

FIGURE 2.14: The dual tree of the maximal outerplane graph is drawn in blue.

connected, have $n - 3$ edges and $n - 2$ vertices. Thus the dual tree must be a tree—which is also a simple graph. An example of a dual tree is illustrated in Figure 2.14. Any vertex of degree 1 in a tree is called a *leaf*.

Definition 2.59 (Matching). Given a connected graph $G = (V, E)$, an edge set S forms a matching in G if no two edges in S are connected to the same vertex in V .

It may be noted that Condition 1 of Lemma 2.50 is equivalent to stating that the edges of the dual tree G^* (see Definition 2.58) that correspond to the edges of the flippable set S form a matching.

Definition 2.60 (Perfect Matching). Given a connected graph $G = (V, E)$ and a matching S , S is a perfect matching if every vertex in V is connected to an edge in S .

Definition 2.61 (Independent Set). Given a connected graph $G = (V, E)$ an independent set is a subset of pairwise non-adjacent vertices of V in G .

Definition 2.62 (Ear). Given an outerplane graph $G = (V, E)$, an ear of G is a 3-cycle face with 2 of its edges on the maximal cycle of G .

2.4.3 Related Lemmas

The result of the following set of lemmas leads to the correctness of the algorithm by Bose et al. (2007).

The next lemma establishes a strong relationship between a flip set of a maximal outerplane graph and its corresponding edge set in the graph's dual tree.

Lemma 2.63 (Bose et al. (2007) Lemma 4.2). A set S of internal edges in a maximal outerplane graph G is flippable if and only if the corresponding dual edges S^* form a matching in the dual tree G^* .

Proof. $\Leftarrow$ Let S^* form a matching in the dual tree G^* . Then no two edges of S^* can have a common vertex. Thus no two edges of S can have a common face in the primal outerplane graph G —making edges of S pairwise non-consecutive. Since G is a maximal outerplane graph, due to Lemma 2.38 no edges in S can be blocked. Furthermore, for two edges $ab, xy \in S$ to form a bad pair, they must be seen by the vertices p and q . Then the subgraph induced by the vertices a, b, x, y, p, q must form a subdivision of K_4 . Since an outerplane graph can contain neither K_4 nor its subdivision, this cannot happen. Thus due to Lemma 2.50 S must be flippable.

Proof. $\Rightarrow$ Let S be a simultaneously flippable set in G . Then due to Lemma 2.50, no two edges of S can share the same face. Then in the corresponding flip set S^* in the dual tree G^* , no two edges can share the same vertex—making S^* a matching in G^* . $\square$

The next lemma shows that any set of edges S of a triangulation can be 3-coloured. Given any set S of a triangulation G , the lemma provides us a way to select at least a third of the edges in S such that they are pairwise non-consecutive.

Lemma 2.64 (Bose et al. (2007) Lemma 3.6). *Every n -vertex planar graph G has an edge 3-colouring that can be computed in $\mathcal{O}(n^2)$ time, such that every triangle is trichromatic (edges coloured with 3 different colours).*

Proof. Robertson et al. (1997) proved that the vertices of G can be coloured with no more than 4-colours such that no two neighbouring vertices have the same colour. For such a solution, let the 4 colours be $\{1, 2, 3, 4\}$. Now for each triangle in G , we colour each edge e in G in the following manner:

- Colours of the vertices e connects to are $\{1, 2\}$, or $\{3, 4\} \Rightarrow$ colour e as red.
- Colours of the vertices e connects to are $\{1, 3\}$, or $\{2, 4\} \Rightarrow$ colour e as green.
- Colours of the vertices e connects to are $\{1, 4\}$, or $\{2, 3\} \Rightarrow$ colour e as blue.

Since no two vertices of the same colour can be neighbours, the only way two edges uv and xy on the same triangle can have the same colour is when all u, v, x and y are coloured differently. But since uv and xy are on a common triangle, they must have a vertex in common—disallowing them from having 4 distinct colours. Hence the edges of each triangle must be trichromatic. $\square$

The next lemma provides an upper bound on the number of simultaneous flips required to make any vertex dominant in a maximal outerplane graph. Notice that the concept of diagonal flips and flippable sets stay valid in outerplane graphs so long as we are flipping internal chords and not edges incident to the outerplane. An outerplane graph on n vertices with a dominant vertex is referred as D_n . Since all outerplane graphs of equal order with a dominant vertex are isomorphic, once we can transform any outerplane graph to D_n using simultaneous flips, we can transform any outerplane graph to another of equal order using the same technique. Observe that a flip in an outerplane graph is equivalent to a rotation in its dual tree (see Chen, Chang, and Wang (2005), Pallo (2000), Pallo (1987), Sleator, Tarjan, and Thurston (1988)).

Lemma 2.65 (Bose et al. (2007) Lemma 4.7). *Let G be a maximal outerplane graph on n vertices. Suppose that its dual tree G^* has diameter k . Let v be a fixed vertex of G . Then G can be transformed by at most k simultaneous flips into a maximal outerplane graph X in which v is dominant.*

Proof. In Figure 2.15, let P be the set of internal faces of G incident to the vertex v . Then in the dual tree G^* the corresponding dual vertices of the faces in P would form a path P^* . We define the distance of a vertex of G^* to P^* as its distance from its nearest vertex on P^* . Let L^* be the set of vertices in G^* that are at distance 1 from P^* . Since every vertex in L^* has a distance 1 from P^* , each of these vertices must have a unique path of length 1 (an edge) to P^* . Let the set of all these edges connecting P^* to L^* be S^* , and its corresponding edge set in the primal graph be S (drawn in red).

FIGURE 2.15: Introducing a dominant vertex. Zigzagged arcs represent a set of edge(s). Shaded region may be further triangulated. Dashed edges may be a chord or an edge of the empty cycle. Red edges when flipped produce dotted blue edges. Dual tree is drawn in green.

Each vertex of L^* must be connected to P^* by exactly one edge, as otherwise, it would introduce a cycle with a subinterval of P^* which is not possible as G^* is a tree. Finally, because G is a triangulation, the degree of vertices in G^* can be at most 3. Also note that the first and last vertex of P^* can have a degree of at most 2 since they correspond to faces that have at least one edge on the outerface of G which is not represented by a vertex in G^* . Every other vertex of P^* has two edges connecting vertices of P^* . Due to this, every vertex on P^* can be connected to at most one vertex that is not on P^* . Thus every edge of S^* is incident to two unique vertices and as a result, S^* must be a matching in G^* . As a result due to Lemma 2.63 S must be a flippable set in G .

Let $G' := G \setminus S$. Then in the dual tree G'^* of G' , the distance of every vertex that is not adjacent to P^* has been reduced by 1. Thus by repeating the process at most k (the diameter of the graph) times, G can be transformed to a maximal outerplane graph where the vertex v is adjacent to every internal face of the graph—making v a dominant vertex. $\square$

The next theorem has been used in transforming any outerplane triangulation into one where the diameter of its dual tree is logarithmic in number of vertices n . The same technique has been used in the transformation of a Hamiltonian triangulation to another. For any outerplane graph OP_n of order n , $\alpha_d(OP_n)$ is the largest cardinality of an independent set where for any vertex v of the set $\deg_{OP_n}(v) \leq d$. Theorem 2.66 has been used in proving the following two important results.

Theorem 2.66 (Bose, Dujmović, and Wood (2005) Theorem 8). *For all $d \geq 4$ and $n \geq 5$,*

$$\alpha_d(OP_n) \geq \left(\frac{d-3}{3d-6}\right)n + \frac{2}{d-2} \quad (2.1)$$

Conversely, for all even $d \geq 6$ and for infinitely many values of n ,

$$\alpha_d(OP_n) \leq \left(\frac{d-4}{3d-10}\right)(n-6) + 3 \quad (2.2)$$

The next two lemmas are similar in nature. The first one shows how using $\mathcal{O}(\log n)$ simultaneous flips the diameter of the dual tree of the interior of an empty cycle of a triangulation can be bounded by $\mathcal{O}(\log n)$. The second lemma proves

the same statement for maximal outerplane graphs. Given a plane graph G and an empty cycle C , $G\{C\}$ denotes the subgraph of G whose vertices are the vertices of C and whose edges are the edges of C along with the internal chords of C .

Lemma 2.67 (Bose et al. (2007) Lemma 5.5). *Let H be a Hamiltonian cycle of a triangulation G with n vertices. Then G can be transformed by a sequence of at most $c_2 \log n$ simultaneous flips into a triangulation X in which H is a Hamiltonian cycle and the diameter of the dual tree of $X\{C\}$ is at most $c_2 \log n$. During the process no edge of H is flipped. Here $c_2 = 2 / \log(\frac{54}{53}) \approx 74.2$.*

Proof. This result can be established from Equation 2.1 of Theorem 2.66 that asserts that in any n -vertex outerplane subgraph $G\{C\}$ there is an independent set I of at least $\frac{n}{6}$ vertices each of degree no more than 4. We build our first simultaneous flip set S by choosing one arbitrary chord from each vertex of I of degree at least 3. None of these chords can be consecutive since I is an independent set. For two edges to form a bad pair, or an edge to block another, the graph must contain a subdivision of K_4 . Since an outerplane subgraph cannot contain subdivisions of K_4 , none of these chords in $G\{C\}$ can be blocked, or form bad pairs with each other. However, they could still be blocked by external chords of C that are not in $G\{C\}$. So to maximize the number of chords we have selected for flipping we must insert as many of these blocking chords in S as well. An internal chord of C cannot be consecutive to an external chord of C . However, these blocking chords can be consecutive with each other. To avoid consecutivity using Lemma 2.64 we 3-colour the external chords and insert the largest set of external chords of same colour to S . This ensures that at least one-third of the blocking external chords can be inserted to S such that none of them are pairwise consecutive. Notice that the number of vertices whose degree can be reduced to 2 depends on how many of the blocked external chords are consecutive to each other. Thus for the first simultaneous flip, we assume the worst-case where all of the $\frac{n}{6}$ internal chords are blocked by external chords. Then at least one-third of the $\frac{n}{6}$ internal chords, together with appropriate external chords can be simultaneously flipped to obtain at least $\frac{n}{18}$ vertices having a degree of at most 3. Thus flipping S on G we obtain $G' = G\langle S \rangle$, where at least one-third of the vertices in I has a degree of at most 3 in $G'\{C\}$.

Repeating the process in another simultaneous flip at least one-third of the degree 3 vertices in I , that is, $\frac{n}{54}$ vertices can be guaranteed to have a degree 2 in $G''\{C\}$. Notice that each of these vertices corresponds to an ear in $G''\{C\}$. Then there must be at least $\frac{n}{54}$ ears in $G''\{C\}$. Each ear corresponds to a leaf node in the dual tree T^* of $G''\{C\}$. By removing all the leaf nodes from T^* , we get a smaller dual tree $T^{*'}.$ Notice that the primal graph of $T^{*'}$ is still a subgraph of $G''\{C\}$. Let the maximal cycle of this primal maximal outerplane subgraph be C' in $G''\{C\}$. Notice that C' contains all the vertices of C excepting the vertices having degree 2 in $G''\{C\}$. Thus $G''\{C'\}$ is a maximal outerplane graph containing at most $\frac{53n}{54}$ vertices. Furthermore since its dual tree $T^{*'}$ is missing all the leaf nodes of the dual tree T^* , the diameter of $G''\{C'\}^*$ is exactly 2 less than the diameter of T^* . Finally, notice that for any internal chord of C to be blocked, its blocking edge must be an external chord of C . Hence none of the blocking chord we are flipping can be present in $G\{C\}$ or $G'\{C\}$ and by following the process, each newly introduced maximal outerplane subgraph's dual tree is exactly 2 less in diameter than the previous one's dual tree.

Thus on each iteration, a new outerplane subgraph is introduced, consisting of at most $\frac{53n}{54}$ vertices from the previous subgraph, such that the dual tree of the new subgraph is exactly 2 less than the previous one's dual tree. The process continues

until the size of the graph is smaller than 6. Therefore, the recurrence relation² we get over the diameter gives us $2 + c_2 \log \frac{53n}{54} = c_2 \log n$. Similarly, in every iteration, we require 2 simultaneous flips and the recurrence relation over the number of simultaneous flips is the same as well. Thus with no more than $c_2 \log n$ simultaneous flips we can transform G into X such that the diameter of the dual tree of $X\{C\}$ is no more than $c_2 \log n$, where $c_2 = \frac{2}{\log \frac{54}{53}}$. $\square$

Lemma 2.68 (Bose et al. (2007) Lemma 4.6). *Let G be a maximal outerplane graph on n vertices. Then G can be transformed by a sequence of at most $c_1 \log n$ simultaneous flips into a maximal outerplane graph X such that the diameter of the dual tree X^* is at most $c_1 \log n$. During the process the edges of the maximal cycle of G is preserved. [$c_1 = \frac{2}{\log \frac{6}{5}} \approx 7.6 \log n$]*

Proof. This lemma is an outcome of Lemma 2.67. The difference being that in Lemma 2.67, the internal chords selected could be blocked by external chords but in a maximal outerplane graph, the maximal cycle does not contain any external chords.

Like before using Equation 2.1 of Theorem 2.66 we select a set of independent vertices I of size at least $n/6$ such that for all vertex $v \in I \mid \deg_G(v) \leq 4$ in G . We build our first simultaneous flip set S by choosing one arbitrary chord from each vertex of I of degree at least 3. Because a maximal outerplane graph cannot contain any subdivision of K_4 , none of the edges in S can be blocked or can form bad pairs. Finally, because I is an independent set, no two chords in S can be consecutive. Thus due to Lemma 2.50 S is a flippable set.

Let $G' := G\langle S \rangle$. Then just as shown in the proof of Lemma 2.67 I is still an independent set in G' , and $\forall v \in I \mid \deg_{G'}(v) \leq 3$. Repeating the process in another simultaneous flip we obtain the graph G'' where the degree of all vertices in I is 2. Notice that for every vertex v in G' such that $\deg(v) = 2$, the vertex is part of a unique ear in G'' . Let the cycle C be the maximal cycle of G'' without the vertices of degree 2. Note that C does not contain any external chords in G'' . Thus they cannot be part of a K_4 with any internal chord of C . As a result, no edge in the exterior of C can neither block nor form bad pair with an internal chord of C and we can repeat the process until C becomes a triangle.

Notice that every leaf node of the dual tree of a maximal outerplane graph corresponds to an ear of the primal graph. Thus removing all the ears of a maximal outerplane graph is equivalent to removing all the leaves of its dual tree, and doing so would reduce the diameter of the tree by exactly 2. Furthermore, in the successive iterations, we do not flip any edges on or in the exterior to the newly introduced maximal outerplane graph, ensuring that all the previously introduced ears continue to be present. Thus the diameter of the dual tree stays exactly 2 less than the one generated previously. From the recurrence relationship between the dual tree of the maximal outerplane graphs G'' and $G''\{C\}$ we get: $2 + c_1 \log \frac{5n}{6} = c_1 \log n$.

On each iteration we require 2 simultaneous flips, again giving us the same recurrence relation $2 + c_1 \log \frac{5n}{6} = c_1 \log n$.

This gives us $c_1 = \frac{2}{\log \frac{6}{5}} \approx 7.6 \log n$ $\square$

2.4.4 Overview of the Algorithm

As we have seen in the results of Section 2.2, if we can transform any triangulation into the canonical triangulation via simultaneous flips, then we can transform any

²Lemma 3.12 in Chapter 3 provides an in-depth explanation of the recurrence relation

triangulation into another of equal order via simultaneous flips. This is done by transforming the first triangulation into the canonical triangulation, and then continuing the transformation by following the reverse sequence of the inverse of the simultaneous flip sets required to transform the second triangulation into the canonical triangulation.

Bose et al. (2007) in their algorithm first transforms a triangulation to a Hamiltonian triangulation using a single simultaneous flip. Then in $\mathcal{O}(\log n)$ simultaneous flips they transform the Hamiltonian triangulation into a canonical triangulation. In Section 2.4.5 we explain how to transform any triangulation into a Hamiltonian triangulation. Then in Section 2.4.7 we explain how to introduce a dominant vertex in a maximal outerplane graph. Finally in Section 2.4.6 we show how to transform any Hamiltonian triangulation to a canonical triangulation.

1. $G_1 \rightarrow G_H$: Transforming into Hamiltonian triangulation.
2. $G_H \rightarrow G_\delta$: Introduce the first dominant vertex.
3. $G_\delta \rightarrow \Delta_n$: Introduce the second dominant vertex.
4. $\Delta_n \rightarrow G_2$: Do the reverse to transform Δ_n to target triangulation.

2.4.5 $G_1 \rightarrow G_H$: Transforming Into Hamiltonian Triangulation

Whitney (1931) proves that any 4-connected triangulation is Hamiltonian. Bose et al. (2007) show a method to make any triangulation 4-connected to obtain a Hamiltonian triangulation with just one simultaneous flip. To do so the authors introduce the next lemma made in the spirit of Lemma 2.44 but for simultaneous flips.

Lemma 2.69 (Bose et al. (2007) Lemma 3.2). *Let G be a triangulation with $n \geq 6$ vertices. Let S be a set of edges of G such that no two edges in S are in a common triangle, every edge in S is in a separating triangle, and every separating triangle contains an edge in S . Then S is flippable and $G\langle S \rangle$ is 4-connected.*

Proof. Note that the condition $n \geq 6$ in Lemma 2.69 does not put any additional restrictions as triangulations of order up to 5 must have a vertex of degree 3 and thus cannot be 4-connected.

Due to Lemma 2.42 every edge of S is (individually) flippable. Thus none of the edges in S are blocked. Because no two edges in S are from a common triangle, the edges in S are pairwise non-consecutive. Finally, for two edges $e_1, e_2 \in S$ to form a bad pair, they must be seen by two distinct vertices p and q . Let e_1 be from separating triangle T_1 and e_2 be from the separating triangle T_2 . Then WLOG p must be inside and q outside the triangles T_1 and T_2 . But that would mean $T_1 = T_2$ resulting in both e_1 and e_2 being in the same triangle. Thus there cannot be any bad pair and S must be a flippable set.

Let the new set of edges introduced in $G\langle S \rangle$ that are not present in G be S' . Then $|S| = |S'|$. Since one edge of each separating triangle of G is being flipped, and none of these edges are blocking, none of these separating triangles would be present in $G\langle S \rangle$. Thus for $G\langle S \rangle$ to contain a new separating triangle T , at least one of the edges of T must be in S' . Then the authors investigated the three possible cases as illustrated in figure 2.16 and complete the proof by contradiction as below:

- Case 1 $|T \cap S'| = 1$: Let xyz be a separating triangle whose edge xy when flipped, results in vw introducing the separating triangle uvw . As shown in the proof of Lemma 2.43 this requires $u = z$, and at least one of the triangles xuv, xuw, yuv ,

FIGURE 2.16: Dotted blue edges are flipped to create red separating triangle. Non-shaded regions are faces.

yuw must be a separating triangle. Thus at least one of the edges $\{ux, uy, vx, vy, wx, wy, uv, uw\}$ must also be in S . But since no two edges of S can be in a common triangle, and since $xy \in S$, none of $\{ux, uy, vx, vy, wx, wy\}$ can be in S . Then either of uv or uw must be in S , but since they are edges of the separating triangle uvw , we would not end up with this triangle in $G\langle S \rangle$. Thus this case is not possible.

Case 2 $|T \cap S'| = 2$: Let the edges xy and rs be edges of the separating triangle xyz and rst respectively, such that $\{xy, rs\} \subseteq S$. We obtain uv by flipping xy and vw by flipping rs such that uvw is a separating triangle. Then similar to the proof of Lemma 2.43, flipping uv of the separating triangle uvw results in the separating triangle xyw making $w = z$. Similarly flipping vw of the separating triangle uvw results in the separating triangle rst making $u = t$. But then the subgraph induced by $\{u, v, w, x, y, r, s\}$ is not planar or if $x = r$ or $y = s$ not simple.

Case 3 $|T \cap S'| = 3$: Let the edges xy, rs and ab be edges of the separating triangle xyz, rst , and abc respectively, such that $\{xy, rs, ab\} \subseteq S$. We obtain uv by flipping xy, vw by flipping rs and uw by flipping ab such that uvw is a separating triangle. Then as shown in the proof of Lemma 2.43 and the previous two cases, $z = w, t = u$ and $c = v$. But then the subgraph induced by these vertices is not planar, or if $x = r = a$ or $y = s = b$ not simple.

□

Now the question is how to obtain such a set of edges S so that every separating triangle has exactly one edge in S and every edge in S is from at least one separating triangle. To achieve this the notion of perfect matching is useful. Petersen (1891) proved a theorem which in modern terminologies implies that every 3-regular bridgeless graph has a perfect matching. Biedl et al. (2001) showed how such a perfect matching that contains a prespecified edge can be computed in $\mathcal{O}(n)$ time. Since every face of a triangulation G is a triangle, its dual graph G^* is 3-regular, and because G does not contain any loops, G^* cannot contain any bridges. As a result, by finding such a perfect matching S^* in G^* , its corresponding edge set S in G would be pairwise non-consecutive and contain an edge from every face in G .

Given a triangulation G , let G_1 be the triangulation we get by removing the interior of all the outermost separating triangles of G . Each of those removed separating

FIGURE 2.17: Transforming G into a 4-connected triangulation G' with 1 simultaneous flip. The 3 separating triangles of G are $T_1 = (a, b, d)$, $T_2 = (b, d, i)$ and $T_3 = (e, h, f)$. Red denotes edges of perfect matching. Blue denotes obtained new edges.

triangles now defines a face in G_1 . Let G_1^* be the dual graph of G_1 . By using Biedl et al. (2001) we can compute a perfect matching in G_1^* whose corresponding edge set S_{temp} in the primal graph G_1 is pairwise non-consecutive. Furthermore, every face of G_1 has exactly one edge in S_{temp} . We now insert every edge of S_{temp} into S that is an edge of a face that used to be a separating triangle in G . Notice that for every outermost separating triangle of G , there is exactly one edge S that is on the separating triangle. So, we can repeat the process for each of those outermost separating triangles that we removed from G to obtain G_1 . Except now for each of those subgraphs, there is exactly one edge that has been selected in S . Thus using Petersen (1891)'s algorithm, we can repeat the process but now we have a prespecified edge that would be present in the matching. Thus at the end of this process once all the subgraphs without separating triangles have been perfectly matched, S would be a set of edges meeting the requirement of Lemma 2.69.

An example of the process is shown in figure 2.17. The triangulation G contains 3 separating triangles— $T_1 = (a, b, d)$, $T_2 = (b, d, i)$ and $T_3 = (e, h, f)$. Notice that the outermost separating triangles of G are T_1 and T_2 . So we compute $G_1 := G \setminus (\text{int}(T_1) \cup \text{int}(T_2))$, $G_2 := \text{int}(T_1)$ and $G_3 := \text{int}(T_2)$. We then compute a perfect matching of G_1 by using Biedl et al. (2001) algorithm on the dual graph G_1^* . Notice that an edge (bd) is matched that is contained in G_2 and G_3 . Thus we can compute a perfect matching matching of G_2 and G_3 as well using the same method, but this time we prespecify the selected edge (bd) for each subgraph. G_3 contains T_3 so the same process is repeated and we get $G_4 := G_3 \setminus \text{int}(T_3)$ and $G_5 := \text{int}(T_3)$. Once we have a perfect matching of G , we define S as the set of matched edges that are on a separating triangle. Finally we get our 4-connected triangulation $G' := G \langle S \rangle$. As Whitney (1931) proved all 4 connected triangulations are Hamiltonian, G' must be Hamiltonian as well.

Bose et al. (2007) also provide an algorithm to compute a flip set that meets the requirement of Lemma 2.69 in $\mathcal{O}(n)$ time. This is done by first computing, in $\mathcal{O}(n)$ time, a sequence consisting of all the separating triangles of the triangulation such that no separating triangle is in the interior of any of the previous separating triangles in the sequence. Once we have such a sequence, S can be computed in another

$\mathcal{O}(n)$ time. The details for this algorithm is not explained in this thesis as we do not make any improvement in the process of creating Hamiltonian triangulation.

2.4.6 $G_H \rightarrow G_\delta$: Introducing the First Dominant Vertex

Given a Hamiltonian triangulation G_H with a Hamiltonian cycle H , using Lemma 2.67, G_H can be transformed into another triangulation G'_H , where the cycle H is preserved and the diameter of the dual tree of the subgraph $G'_H \setminus \{H\}$ is at most $c_2 \log n$. This requires no more than $c_2 \log n$ simultaneous flips. Then a vertex v in G'_H is selected such that v is not incident to any external chord of cycle H . By using Lemma 2.65 with no more than another $c_2 \log n$ simultaneous flips we transform G'_H into G_δ where the cycle H is preserved and v is a dominant vertex in $G_\delta \setminus \{H\}$. Thus by no more than $2c_2 \log n$ simultaneous flips the first dominant vertex can be introduced in a Hamiltonian triangulation.

2.4.7 $G_\delta \rightarrow \Delta_n$: Introducing the Second Dominant Vertex

Once we have a triangulation G_δ of order n with a dominant vertex v , notice that $G := G_\delta \setminus \{v\}$ has $n - 1$ vertices such that all the vertices are sharing a face. Thus G is a maximal outerplane graph. We take an embedding of G such that the outerface of G is in the exterior of its maximal cycle. Since v is not a part of this maximal outerplane subgraph, the edges incident to v in the original graph cannot block any internal chords of this maximal outerplane graph. Thus by using the next lemma, we introduce the second dominant vertex in the triangulation using another $2c_1 \log n$ simultaneous flips completing its transformation into Δ_n .

Lemma 2.70 (Bose et al. (2007) Lemma 4.5). *For every maximal outerplane graph G on n vertices and for every vertex v of G , there is a sequence of $2c_1 \log n$ simultaneous flips to transform G into a maximal outerplane graph in which v is dominant. [$c_1 = \frac{2}{\log \frac{6}{5}} \approx 7.6 \log n$]*

Proof. The proof of this lemma proceeds in two parts. Using Lemma 2.68 we reduce the diameter of the dual tree to $c_1 \log n$ with at most $c_1 \log n$ simultaneous flips. Then using Lemma 2.65 a dominant vertex is introduced with at most a further $c_1 \log n$ simultaneous flips. Thus the total number of simultaneous flips required to transform G into D_n is $2c_1 \log n$. $\square$

2.4.8 $\Delta_n \rightarrow G_2$: Do the Reverse to Transform Δ_n to Target Triangulation

In Section 2.4.5, 2.4.6 and 2.4.7 we described how to transform any triangulation to a canonical triangulation. As explained in Definition 2.46, a simultaneous flip is an invertible operation. Thus once we know the sequence of simultaneous flips $S = (S_1, S_2, S_3, \dots, S_k)$ required to transform $G_2 \rightarrow \Delta_n$, we can follow the sequence of simultaneous flips $S^{-1} = (S_k^{-1}, S_{k-1}^{-1}, S_{k-2}^{-1}, \dots, S_1^{-1})$ on Δ_n where for all $1 \leq k \leq i$, S_k^{-1} is the inverse flip set of S_k —which is simply the set of new edges obtained by flipping S_k .

Since $|S^{-1}| = |S|$, to complete the transformation we need the same number of simultaneous flips as we need to transform G_2 into Δ_n . Thus we need at most 1 simultaneous flip to introduce Hamiltonicity, $2c_2 \log n$ simultaneous flips to introduce the first dominant vertex, and another $2c_1 \log n$ simultaneous flips to introduce the second dominant vertex—requiring a total of $1 + 2(c_1 + c_2) \log n$ simultaneous flips.

2.4.9 Transforming Any Maximal Outerplane Graph to Any Other of Equal Order

Now that we have a method to transform any maximal outerplane graph to another maximal outerplane graph δ_n with a dominant vertex via simultaneous flips, we have the following theorem for transforming any maximal outerplane graph to any other of equal order via simultaneous flips.

Theorem 2.71 (Bose et al. (2007) Theorem 4.4). *Let G_1 and G_2 be maximal outerplane graphs on n vertices. There is a sequence of at most $4c_1 \log n$ simultaneous flips to transform G_1 into G_2 , where $c_1 = \frac{2}{\log \frac{6}{5}} \approx 7.6$.*

Proof. Using Lemma 2.68 convert G_1 to a maximal outerplane graph with a dominant vertex D_n . Then transform D_n to the second triangulation, by applying the reverse sequence of the flip sets required to transform G_2 to D_n . The total number of simultaneous flips required for the process is $2 \times 2c_1 \log n = 4c_1 \log n$ where $c_1 = \frac{2}{\log \frac{6}{5}} \approx 7.6$. $\square$

2.4.10 Transforming Any Triangulation to Any Other of Equal Order

Finally Bose et al. (2007) obtains the following result.

Theorem 2.72 (Bose et al. (2007) Theorem 5.1). *Let G_1 and G_2 be triangulations on n vertices. There is a sequence of $2 + 4(c_1 + c_2) \log n$ simultaneous flips to transform G_1 into G_2 .*

Here G_1 and G_2 need one simultaneous flip each for transformation into Hamiltonian triangulation, and a further $2(c_1 + c_2) \log n$ flips to transform each of them into a canonical triangulation. Thus with an equal number of simultaneous flips we can transform G_1 to Δ_n and then to G_2 . So by virtue of the above results we have an algorithm that transforms any triangulation G_1 into another triangulation G_2 in $2 + 4(c_1 + c_2) \log n \approx 327 \log n$ simultaneous flips.

Chapter 3

A Modification to Bose et al.'s Algorithm

Since Bose et al. (2007) algorithm is of optimal order any improved algorithm can only reduce the leading coefficient. In fact in this chapter we devise an algorithm for which the leading coefficient of complexity function, that is of the term $\log n$ is lower than that algorithm. . The modification leads to improvement in the constants c_1 and c_2 present in Theorem 2.71 and 2.72. This lowers the complexity of the algorithm for transforming a triangulation to another of equal order from $\approx 327 \log n$ to $\approx 86 \log n$.

3.1 Improvement of Constants c_1 and c_2

We start with a definition that is used in the following lemmas.

Definition 3.1 (Arc). *Given a plane graph $G = (V, E)$ and a cycle C in G , an arc is a path on C . For two vertices u and v on C , the notation $\widehat{uv}$ is used to describe the arc from vertex u to vertex v in clockwise direction on the cycle C .*

Note that this notation is always used with respect to a planar embedding of the graph. We also use braces for defining the sequence of vertices on the arc. By $[\widehat{pq}]$ we denote the sequence of vertices from vertex p to vertex q on the arc inclusively whereas $(\widehat{pq})$ is the sequence of vertices on the arc from p to q excluding both p and q . If a chord is a blocked edge, a blocking edge or a bad edge it is referred to as a *blocked chord*, *blocking chord* or *bad chord* respectively.

The following lemma is a consequence of the Jordan curve Theorem.

Lemma 3.2. *Given a plane graph $G = (V, E)$ with a cycle C , any chord pq of C separates the vertices in $(\widehat{pq})$ from the vertices in $(\widehat{qp})$, such that for all $x \in (\widehat{pq})$ and for all $y \in (\widehat{qp})$, if xy is an edge then xy and pq belong to different regions of C .*

The following lemmas are applied to achieve an improvement of the Bose et al. (2007) algorithm. Notice that the cycle C referred in Lemma 3.3 is not necessarily an empty cycle.

Lemma 3.3. *Given a triangulation G with a cycle C , no two consecutive chords of C can both be blocked.*

Proof. We prove this lemma by contradiction. Let vx and vy be two consecutive chords sharing the face vxy in G . WLOG let us pick an embedding such that these chords are internal as illustrated in Figure 3.1. Since vx and vy are in the interior of C , there must be other edges va and vb that are not in the exterior of C such that va is

FIGURE 3.1: No two consecutive chords can both be blocked. Cycle drawn in blue, shaded region may be further triangulated, dashed edges may be edges on the cycle.

consecutive to vx and vb consecutive to vy . Since the graph is triangulated, ax and by must be edges in G . Then vx is seen by $\{a, y\}$ and vy is seen by $\{b, x\}$. Since vx and vy are blocked edges, their corresponding blocking edges ay and bx must also exist in G . Since in the interior of C , vx separates a from y and vy separates b from x , due to Lemma 3.2 ay and bx must be in the exterior of C . Thus a and b cannot be in the interior of C . Similarly, since va and vb are not in the exterior of C , neither of a and b can be in the exterior of C . Thus a and b must be on C . Furthermore, (ay, yv, va) is a cycle where x is in its interior and b is in its exterior. Then due to Lemma 2.31 bx cannot be an edge of G . Thus no two consecutive chords in a triangulation can both be blocked. $\square$

We start by proving a property between a blocked internal chord and its blocking chord in Lemma 3.4. By using this property in Lemma 3.5 we prove that no three blocking external chords can share a face.

Lemma 3.4. *Given a plane graph G and a cycle C in G , if an internal chord xx' of C of a plane graph G is blocked by another chord ab of C , then not both x and x' can be on the arc $\widehat{ab}$.*

Proof. Let xx' be an internal chord of C , blocked by another chord ab in a graph G as illustrated in Figure 3.2a. Then both a and b must see xx' . Thus ax, ax', bx and bx' must be edges on or in the interior of C . WLOG let x and x' be on $\widehat{ab}$ in the order a, x, x', b . $G \setminus \{C\}$ is a maximal outerplane graph. But it contains the paths $\widehat{ax}, \widehat{xx'}, \widehat{x'b}, \widehat{ba}$ and edge set $\{ax', bx\}$ creating a subdivision of K_4 . But Chartrand and Harary (1967) proved that no outerplane graph can contain a subdivision of K_4 . Thus not both x and x' can be on the arc $\widehat{ab}$ of C . $\square$

Lemma 3.5. *Given a triangulation G with an empty cycle C , no more than 2 blocking external chords of C can share the same face.*

Proof. We prove by contradiction. As illustrated in Figure 3.2b let ab, bc and ac be 3 blocking external chords of C that share the face abc in G . Also, let the internal chord xx' of C be the edge blocked by ab . Then a and b must see xx' , and the 4 edges xx' must be consecutive to are ax, ax', bx, bx' . Thus, none of these edges can be external

FIGURE 3.2: No three blocking chords can share a face. Cycle drawn in blue. Dashed edges may not be distinct, zigzagged arcs are paths.

chords. Due to Lemma 3.4, both ends of xx' cannot be on $\widehat{ab}$. Thus WLOG let x be on $\widehat{ab}$ and x' on $\widehat{ba}$.

Since none of ax, ax', bx, bx' can be external chords, and since ac and bc are external chords, the vertex c must be distinct from x and x' . Let us first assume c is on $\widehat{xx'}$. Then xx' separates a and c in the interior of C . However, for ac to be an external blocking chord of an internal chord zz' of C , azz' and czz' must be faces. Then az, az', bz, bz' must be edges of $G\{C\}$. Since xx' separates a from c in the interior of C , for z and z' to be adjacent to both a and b , $\{z, z'\}$ must be $\{x, x'\}$. But since the edge xx' is seen by the vertices a and b , if $\{z, z'\} = \{x, x'\}$ then zz' cannot be seen by c . Thus ac cannot be a blocking external chord.

Finally, if c is not on $\widehat{xx'}$, then c must be on $\widehat{x'x}$. But then xx' separates c from b in the interior of C . Thus due to Lemma 3.2 internal chord yy' cannot be seen by both b and c . So using the same argument we show that bc cannot be a blocking external chord, completing the proof. $\square$

The next two lemmas investigate when chords of a cycle can form bad pairs. The result is useful in identifying whether a set of chords is flippable.

Lemma 3.6. *Given a plane graph G with a cycle C , no two chords of C located in the same region of C can form a bad pair.*

Proof. Let us take an embedding of G such that both of the two distinct chords uv and xy are in the interior of C . To prove by contradiction, let uv and xy be seen by two distinct vertices p and q . Since $\{puv, quv, pxy, qxy\}$ are faces in the interior of C , due to Lemma 2.33 neither p nor q can be in the exterior of C . Similarly since uvp and uvq are faces in the interior of C , due to Lemma 2.33 the edges $\{up, uq, vp, vq\}$ must be on or in the interior of C as well. But the edge xy separates p from q in the interior of C . Thus for $\{up, uq, vp, vq\}$ to be on or in the interior of C , $\{u, v\}$ must be the same as $\{x, y\}$. But then uv and xy are no longer distinct chords of C . $\square$

Lemma 3.7. *Given a plane graph G with an empty cycle C , an internal chord of C can only be blocked by an external chord of C .*

Proof. Let uv be an internal chord of C , blocked by another edge xy . Then the edge uv must be seen by the vertices x and y —making xuv and yuv faces in the interior of C . Since uv is in the internal region of C , due to Lemma 2.33 for xuv and yuv to be faces neither x nor y can be in the exterior of C . Thus x and y must be vertices in $G\{C\}$. However, the blocking edge xy cannot be in $G\{C\}$ because then the subgraph of $G\{C\}$ induced by the vertices $\{u, v, x, y\}$ would create a K_4 . But Chartrand and Harary (1967) proved that no maximal outerplane graph can contain K_4 . Thus xy must be an external chord of C not present in $G\{C\}$. $\square$

Lemma 3.8. *Given a triangulation G with an empty cycle C , any set S of blocking non-consecutive external chords of C can be simultaneously flipped.*

Proof. Since the edges in S are blocking, due to Lemma 2.51 they are (individually) flippable. Since these edges are in the same region of C , Lemma 3.6 asserts that no two of these edges can form a bad pair. Finally, since these edges are pairwise non-consecutive, due to Lemma 2.50 S must be simultaneously flippable. $\square$

For the next lemma the notion of *forest* and *unary tree* is useful.

Definition 3.9 (Forest). *A graph $G = (V, E)$ is a forest if there is at most one unique path between every pair of its vertices.*

Definition 3.10 (Unary Tree). *A unary tree is a tree no vertex of which has degree more than 2.*

Notice that a unary tree is equivalent to a path.

Lemma 3.11. *Let G be a triangulation with an empty cycle C , with a set S of blocking external chords of C . Then at least half of the chords in S can be simultaneously flipped.*

Proof. Let G^* be the dual graph of G . Since the smallest cycle that can have chords in a graph is a quadrilateral with 4 vertices, for the proof we assume that the number of vertices in C and hence G is at least 4. Since the dual graph of any triangulation on > 3 vertices is simple, the dual graph $G^* = (V^*, E^*)$ of G must also be simple. Let S^* be the set of corresponding edges of S in G^* . Define $G^{*'} = (V^*, S^*)$. Notice that only the dual edge of every edge in S is still present in $G^{*'}$. Thus two edges are consecutive in G and are both in S if and only if their corresponding edges are present and are incident to a common vertex in $G^{*'}$.

Let $uv \in S$. Then $\{\widehat{uv} \cup uv\}$ creates a cycle C_{uv} . Since none of the edges of C are in S , uv must be the only edge of C_{uv} which is in S . Then the only edge in $G^{*'}$ that connects any vertex corresponding to a face in the interior of C_{uv} to another vertex corresponding to another face in the exterior of C_{uv} must be u^*v^* . Thus the corresponding dual edge of every external blocking chord in S is a bridge in $G^{*'}$ —making each edge of $G^{*'}$ a bridge. Then $G^{*'}$ must be a forest. Due to Lemma 3.5 no more than 2 blocking chords can share a face and thus no vertex in G^* can have a degree more than 2—making $G^{*'}$ a forest of unary trees.

Let there be i unary trees in $G^{*'}$ with the k -th tree containing p_k edges. Since there is a bijection between the edges in $G^{*'}$ and S , $\sum_{k=1}^i p_k = |S|$. Every unary tree is a path where, by choosing every other edge starting from the first edge in the sequence of a path of length p , a maximum matching of $\lceil p/2 \rceil$ edges can be found in the path. Aggregating the maximum matching of all unary trees of the forest $G^{*'}$ we can select a matching M^* of size:

$$\left\lceil \frac{p_1}{2} \right\rceil + \left\lceil \frac{p_2}{2} \right\rceil + \left\lceil \frac{p_3}{2} \right\rceil + \cdots + \left\lceil \frac{p_i}{2} \right\rceil \geq \lceil (p_1 + p_2 + p_3 + \cdots + p_i)/2 \rceil = \lceil |S|/2 \rceil$$

Let the corresponding edges of M^* in the primal graph be M . Because M^* is a matching, the edges of M must be non-consecutive. Finally, all the chords in M are external chords of C and thus due to Lemma 3.8, M must be simultaneously flippable. $\square$

The next lemma explains the recurrence relation that is found when we recursively obtain smaller maximal outerplane subgraphs with a fraction of the vertices of the original graph, while having its dual tree's diameter a constant amount less than the previous ones. Notice that a key factor of this process is that all the produced ears in each iteration is preserved in the successive iterations. This ensures that constant difference between the dual tree diameter between the maximal outerplane graphs produced on each iteration is preserved in the subsequent iterations.

Lemma 3.12. *Let G_0 be a plane graph with an empty n -cycle C_0 . If there is a recursive process with a constant $0 < \gamma < 1$ with the following properties*

1. *Each iteration consists of k_1 simultaneous flips.*
2. *The i -th iteration generates a graph G_i preserving all empty cycles C_j for $0 \leq j < i$ and introduces another empty cycle C_i such that*
  - (a) *The cycle C_i is in the interior of the cycle C_{i-1} in G^i .*
  - (b) *Length of C_i in G_i is at most $(1 - \gamma)$ times the length of C_{i-1} .*
  - (c) *The diameter of the dual tree of $G_i \setminus \{C_i\}$ is exactly k_2 less than the diameter of the dual tree $G_i \setminus \{C_{i-1}\}$*
3. *On the final iteration f the recursion creates a 3-cycle C_f in G_f .*

Then the process consists of no more than $\frac{k_1 \log n}{\log \frac{1}{1-\gamma}}$ simultaneous flips, and the diameter of the dual tree of $G_f \setminus \{C_0\}$ is at most $\frac{k_2 \log n}{\log \frac{1}{1-\gamma}}$.

Proof. The length of the cycle reduces at least by a fraction of $(1 - \gamma)$ in each iteration. So after f iterations, the cycle is at most of length $(1 - \gamma)^f$. Thus for the final cycle to have a length no more than 3, the length of the final cycle C^f :

$$\begin{aligned} (1 - \gamma)^f n &\leq 3 \implies f \log(1 - \gamma) \leq \log\left(\frac{3}{n}\right) \\ \implies f &\geq \frac{\log 3 - \log n}{\log(1 - \gamma)} \geq -\frac{\log n}{\log(1 - \gamma)} = \frac{\log n}{\log \frac{1}{1-\gamma}} \end{aligned}$$

Then the total number of simultaneous flips required is $k_1 \times f = \frac{k_1 \log n}{\log \frac{1}{1-\gamma}}$. Moreover, notice that on each iteration i , all the previously generated empty cycles $\{C_0, C_1, \dots, C_{i-1}\}$ are preserved. This ensures that the region interior to C_0 and exterior to C_i (all generated ears) is preserved during i -th and each subsequent iterations. Hence, the dual tree of $G^i \setminus \{C_i\}$ is a subtree of the dual tree of $G^i \setminus \{C_{i-1}\}$ with diameter exactly k_2 less. Thus the diameter of the dual tree of $G_f \setminus \{C_0\}$ is k_2 times the number of iterations plus the diameter of the dual tree of $G_f \setminus \{C_f\}$. The diameter of the dual tree of a maximal outerplane graph on 3 vertices is 0, making the diameter of the dual tree of $G_f \setminus \{C_0\}$ to be at most $\frac{k_2 \log n}{\log \frac{1}{1-\gamma}}$. $\square$

The next proof is done in the same spirit as Lemma 2.68. It formulates a process to transform a maximal outerplane graph into one whose diameter of the dual tree is logarithmic to its order.

Proof. The proof here is in the same spirit as in Bose et al. (2007). We proceed by induction on n . The induction hypothesis is true for the base case $n = 3$ as the diameter of a triangle's dual tree is 0. Let G be a maximal outerplane graph on n vertices. Then using Equation 2.1 of Theorem 2.66, G has an independent set of vertices I of degree no more than 5, where $|I| \geq \frac{2}{9}n$. Since every vertex of a maximal outerplane graph is on a cycle, the degree of such vertices is not less than 2. Let for $d \in \{2, 3, 4, 5\}$, I_d be the set of vertices of I having degree exactly d .

Let S be the set of edges that shall be flipped on the first simultaneous flip. For every vertex $v \in I_3 \cup I_4$ add one internal chord incident to v in S . For each vertex $v \in I_5$, add the 2 corner chords of v in S .

To prove non-consecutivity in S , assume on the contrary that $xu, xv \in S$ are consecutive. If $x \in I_5$, xu and xv must be corner chords of x and hence they cannot be consecutive to each other. x cannot be a member of $I_3 \cup I_4$ either since only one edge from each vertex of those sets has been included in S . Thus $x \notin I$. So $u, v \in I_3 \cup I_4 \cup I_5$. Since G is triangulated, for xu and xv to be consecutive, uv must be an edge. But that contradicts u and v being members of the independent set I , and therefore cannot be adjacent. Thus the edges in S must be pairwise non-consecutive.

Let S^* be the corresponding edges of S in the dual tree G^* of G . Then S^* must form a matching in G^* and due to Lemma 2.63, S must be simultaneously flippable in G .

Let $G' = G \setminus S$. Let x be a vertex of I and xy be an edge in S that when flipped produced uv in G' . Then both u and v must be adjacent to x in G , and hence neither u nor v can be in I . Thus no newly introduced edge in G' can be incident to any vertex in I . Thus the set of vertices I must be independent in G' as well. Furthermore, since flipping S reduced the degree of vertices in $I_3 \cup I_4$ by one and vertices in I_5 by two, in G' every vertex $v \in I_2 \cup I_3$ has a degree of 2, and every vertex $v \in I_4 \cup I_5$ has a degree of 3. Since $I_4, I_5 \subseteq I$ and I is still an independent set in G' , $I_4 \cup I_5$ must be an independent set of G' . Let S' be the set of internal edges of G' incident to vertices in $I_4 \cup I_5$. Thus $|S'| = |I_4| + |I_5|$, and by the same argument used for S , S' must be simultaneously flippable in G' .

Let $G'' = G \setminus S'$. Every vertex $v \in I$ has degree 2 in G'' . Thus G has been transformed by two simultaneous flips into a maximal outerplane graph G'' containing at least $\frac{2}{9}n$ vertices of degree 2. Let v be a vertex with degree 2 in G'' . Then v has no internal chords in G'' . Let the two vertices consecutive to v on the maximal cycle be u and w . Because every face in the interior of G'' is a triangle, uw must be an edge in G'' creating an ear (u, v, w) in G'' . Furthermore, because $v \in I$, $u, w \notin I$. Let C be the maximal cycle of G'' . We define a new cycle C' consisting of vertices of C of degree > 2 in $G'' \setminus \{C\}$. Notice that we are only removing all the ears of $G'' \setminus \{C\}$ to define the maximal outerplane graph $G''' := G'' \setminus \{C'\}$. Since every ear of G''' corresponds to a unique leaf of the dual tree of G''' , the diameter of this dual tree must be exactly 2 less than the diameter of the dual tree of $G'' \setminus \{C\}$.

Finally notice that every edge of G'' that is not present in G''' is incident to a vertex in I . Since G''' contains no vertex in I , none of these edges can be chords of the maximal cycle of G''' . Hence, due to Lemma 3.7 they cannot block any internal chord of G''' or any other successive subgraphs obtained by repeating the process. Hence we can repeat the process until we obtain a maximal outerplane graph X where the final cycle C_f is a triangle. Notice that in each successive iterations, we do not flip any edges on or in the exterior of the newly introduced maximal outerplane subgraph, ensuring that all the previously introduced maximal cycles (and ears) continue to be present. Thus the diameter of the dual tree of each maximal outerplane subgraph stays exactly 2 less than the dual tree of the one generated previously. Thus Lemma 3.12 applies on this recursive process where $k_1 = k_2 = 2$ and $\lambda = 2/9$. As a result with $\frac{2 \log n}{\log \frac{1}{1-\frac{2}{9}}} = \frac{2 \log n}{\log \frac{7}{9}}$ simultaneous flips we can transform

G into another maximal outerplane graph X whose diameter of the dual tree is at most $\frac{2 \log n}{\log \frac{7}{9}}$, making $c_1 = \frac{2}{\log \frac{7}{9}} \approx 5.51$. $\square$

Lemma 3.14. *Let G be a triangulation and C be an empty cycle of G with n vertices. (G may have more than n vertices.) Then G can be transformed by a sequence of at most $c_2 \log n$ simultaneous flips into a triangulation X in which C is an empty cycle and the diameter of the dual tree of $X \setminus \{C\}$ is at most $c_2 \log n$. During the process no edge of H is flipped. Here $c_2 = 2 / \log \frac{12}{11} \approx 15.93$.*

Proof. By Theorem 2.66 the maximal outerplane subgraph $G \setminus \{C\}$ has an independent set I of at least $\frac{n}{6}$ vertices such that $\forall v \in I \mid \deg_{G \setminus \{C\}}(v) \leq 4$. Let I_d be the set of vertices $v \in I$ with $\deg_{G \setminus \{C\}}(v) = d$. Since all the vertices are on the cycle C , $\deg_{G \setminus \{C\}}(v) \geq 2$. Since every vertex in I_4 has 2 internal chords of C , by virtue of Lemma 3.3 at least one of the chords of every vertex in I_4 must be unblocked. So we make our first simultaneous flip set S consisting of one unblocked chord incident to

each vertex $v \in I_4$. All the edges in S are internal chords of the maximal outerplane graph $G\{C\}$. Then none of the chords in S can be blocked. As shown in the proof of Theorem 3.13, the edges of S are pairwise non-consecutive in the maximal outerplane graph $G\{C\}$. Thus in the dual tree $G^*\{C\}$, the corresponding edges S^* of S form a matching. Thus due to Lemma 2.63 S must be simultaneously flippable in $G\{C\}$ and hence also in G . Notice that we could also insert all the unblocked chords of I_3 in S , however, it does not contribute to our analysis of the worse case as I_3 could be empty.

Let $G' := G\langle S \rangle$. Using the same argument as in the proof of Theorem 3.13, I is still an independent set in $G'\{C\}$ and since we reduced the degree of all vertices in I_4 , $\forall v \in I \mid \deg_{G'\{C\}}(v) \leq 3$. For every vertex $v \in I_3$, insert its remaining internal chord to S' . Let B' be the set of all external chords that block chords of S' . Using Lemma 3.11 we can compute flippable set $M' \subseteq B'$ such that $|M'| \geq \frac{1}{2}|B'|$. Let S'' be the set consisting of all the unblocked chords of S' , and all the blocked chords of S' whose corresponding blocking chord is in M' . Then $|S''| \geq \frac{1}{2}|S'|$. Neither S'' nor M' contains pairwise consecutive chords, and since edges of S'' and M' are in different regions of C , edges of $S'' \cup M'$ must be pairwise non-consecutive as well. Due to Lemma 3.6 neither two chords of S'' nor two chords of M' can be bad pairs. Furthermore, a chord e of M' cannot form bad pair with another chord e' of S'' because then due to Lemma 2.52 e' would be consecutive to a chord of S' . Thus there is no bad pair in $S'' \cup M'$. Finally, every blocked chord of S'' has its corresponding blocking chord in M' , and because every chord of M' is blocking, due to Lemma 2.51 they cannot be blocked. Thus due to Lemma 2.50 $S'' \cup M'$ is a flippable set.

Let $G'' = G'\langle S'' \cup M' \rangle$. As explained in the proof of Theorem 3.13, none of the new edges of $G''\{C\}$ can increase the degree of a vertex in I as every new chord connects two vertices that are consecutive to a vertex in the independent set I . Since we reduced at least half of the vertices in I to degree 2, there are least $n/12$ vertices in $G''\{C\}$ with a degree 2, introducing one ear in $G''\{C\}$ each.

The remaining part of the proof follows exactly as in the proof of Theorem 3.13 but now we define a cycle C' in G'' removing at least $n/12$ vertices of degree 2 in $G''\{C\}$ from C . We also ensure that all of the previously introduced ears are preserved in the successive iterations. Thus Lemma 3.12 applies on this recursive process where $k_1 = k_2 = 2$ and $\gamma = 1/12$. As a result with $\frac{2 \log n}{\log \frac{1}{1-\frac{1}{12}}} = \frac{2 \log n}{\log \frac{12}{11}}$ simultaneous flips we can transform G into another maximal outerplane graph X whose diameter of the dual tree is at most $\frac{2 \log n}{\log \frac{12}{11}}$, making $c_2 = \frac{2}{\log \frac{12}{11}}$. $\square$

So the above results lead to the following theorem.

Theorem 3.15 (In the spirit of Bose et al. (2007) Theorem 5.1). *Let G_1 and G_2 be triangulations on n vertices. There is a sequence of at most $2 + 4(c_1 + c_2) \log n \approx 85.8 \log n$ simultaneous flips to transform G_1 into G_2 , where $c_1 = \frac{2}{\log 7} \approx 5.52$ and $c_2 = \frac{2}{\log \frac{12}{11}} \approx 15.93$.*

Proof. As explained in Section 2.4.5, Bose et al. (2007) proved that with 1 simultaneous flip transform G_1 into a Hamiltonian triangulation G_H with a Hamiltonian cycle H . To convert G_H to a canonical triangulation, first transform it into another Hamiltonian triangulation X using Lemma 3.14 where the dual tree of $X\{H\}$ has a diameter of at most $c_2 \log n$. This takes at most $c_2 \log n$ simultaneous flips. Then choose a vertex v that is not incident to any external chords of H in X . By using Lemma 2.65 with no more than another $c_2 \log n$ simultaneous flips, transform X into X_δ where v is the first dominant vertex in $X_\delta\{H\}$. Since v does not have any external chords in X_δ , none of the edges flipped during Lemma 2.65 can be blocked.

Furthermore, for the two vertices u and w consecutive to v in H , uw must be an external chord in X_Δ . So we define a new empty cycle C replaces the edge sequence (uv, vw) with uw from H . Now define $X_{op} := X_\delta \setminus \{v\}$. Notice that X_{op} is a maximal outerplane graph with its maximal cycle being C .

By first using Theorem 3.13 and then Lemma 2.65 on the resulting graph, with no more than $2c_1 \log(n-1)$ simultaneous flips, where $c_1 = \frac{2}{\log \frac{9}{7}}$ we can transform X_{op} into a maximal outerplane graph of $n-1$ vertices δ_{n-1} with a dominant vertex. Furthermore, every edge of X_δ that is missing in X_{op} is incident to the vertex v , which is not present in X_{op} either. Thus, none of these edges are chords of C in X_δ . Therefore, due to Lemma 3.7, none of these chords can block any internal chords of C . Thus the sequence of simultaneous flips made to transform X_{op} into δ_{n-1} is also valid in X_δ transforming the graph into a canonical triangulation Δ_n .

Now that we have transformed G_1 into Δ_n with no more than $1 + 2(c_1 + c_2) \log n$ simultaneous flips, using the exact same method with no more than the the same number of simultaneous flips we can transform G_2 into Δ_n . Therefore, to transform G_1 into G_2 , first transform G_1 into Δ_n , and then by following the reverse sequence of simultaneous flips transform G_2 into Δ_n , complete the transformation of G_1 into G_2 . The process requires no more than $2 + 4(c_1 + c_2) \log n$ simultaneous flips to complete. $\square$

3.2 Optimal Degree Restriction for Independent Set to Introduce a Dominant Vertex in a Maximal Outerplane Graph

In this section we show what the optimal size of the independent set should be for each iteration to introduce a dominant vertex in a maximal outerplane graph using the algorithm described in Theorem 2.71. We optimize our algorithm by selecting right upper bounds on the degree of vertices of the independent sets considering the following two points.

1. The number of ears introduced on each iteration.
2. The number of simultaneous flips required per iteration.

Once the diameter of the dual tree of the maximal outerplane graph is made logarithmic, by using Lemma 2.65 a dominant vertex is introduced in G with no more than the number of simultaneous flips equaling the diameter of its dual tree.

In the following, we show that $d = 5$ indeed ensures the minimum upper bound for the number of flips required for our algorithm to introduce a dominant vertex in any arbitrary maximal outerplane graph G of order n .

Due to Lemma 2.63 the chords of outerplane graphs are not blocked and do not form bad pairs. Hence to flip them simultaneously, all we have to ensure is that they are not consecutive. For any vertex $v \in I$, with degree d , v has $d-2$ internal chords. By selecting every other chord according to their cyclic order from the vertex v we can select at least $\lceil \frac{d-2}{2} \rceil$ internal chords incident to v , such that they are a pairwise non-consecutive set of edges. Flipping these chords ensures a degree reduction of v by the number of incident chords flipped. So the number $\beta(d)$ of simultaneous flips sufficient to reduce any member of I_d into an ear is:

$$\beta(d) = \begin{cases} d-2 & \text{if } 2 < d < 5 \\ 1 + \beta(\lfloor \frac{d+2}{2} \rfloor) & \text{for } d \geq 5 \end{cases}$$

Let us further assume that after $\beta(d)$ simultaneous flips of maximum possible fraction γ (*lower bound*) of vertices are transformed into ears. Now we investigate how the value of c_1 in Theorem 3.13 changes with respect to d to obtain the following result. We know that removing all the ears of a maximal outerplane graph reduces the diameter of its dual tree by exactly 2. Thus using Lemma 3.12 for $k_1 = \beta(d)$, $k_2 = 2$, we get $c_1 = \frac{\beta(d)}{\log \frac{1}{1-\gamma}}$.

The following table has been computed to show the values of c_1 for $4 \leq d \leq 10$.

TABLE 3.1: Optimal degree for ear decomposition

d	β	$\gamma \geq \frac{d-3}{3d-6}$	$c_1 = \frac{\beta(d)}{\log \frac{1}{1-\gamma}}$
5	2	2/9	5.52
6	3	1/4	7.23
7	3	4/15	6.71
8	3	5/18	6.39
9	3	2/7	6.18
10	4	7/24	8.04

In Table 3.1 we present the values of d and corresponding values of c_1 as determined by Lemma 3.12. From the table it is clear that the best value of the coefficient $c_1 = 5.52$ comes for $d = 5$ when we choose independent set of degree not exceeding 5, and $|I| = \frac{2n}{9}$. Note that value of β increases at $d - 2$, being power of 2, by 1, and is a nondecreasing function of d , whereas γ is an increasing function of d . In fact $\beta = \lfloor \log(d - 1) \rfloor$. We note here that as per Equation 2.1 of Theorem 2.66 value of γ approaches $\frac{1}{3}$ as $n \rightarrow \infty$. In fact for $d \geq 4$, $n \geq 5$

$$\gamma \geq \frac{(d-3)n}{(3d-6)n} + \frac{2}{(d-2)n}$$

whereas for infinitely many values of n and even d , $d \geq 6$

$$\gamma \leq \frac{(d-4)n}{(3d-10)n} + \frac{3}{n} = \frac{1}{3} - \frac{2/3}{3d-10} + \frac{3}{n} \leq \frac{1}{3}$$

Thus Equation 2.1 does not guarantee γ to be larger than $1/3$ and so c_1 cannot have a smaller value than when $\frac{1}{1-\gamma}$ is minimum, which happens when $\gamma = 1/3$. Then

$$c_1 \approx \frac{\log \beta(d)}{\log \frac{1}{1-\gamma}} \geq \frac{\log \beta(d)}{\log \frac{3}{2}}$$

Using Equation 2.1 of Theorem 2.66 for any n vertex outerplane graph irrespective of its chord configuration, the maximal order of the independent vertex set we can always ensure is $n/3$ for $\beta(d) \geq 10$. Let us assume that c_1 is minimum for $\beta(d) \geq 10$. Since there can be vertices of degree ≥ 10 containing at least 8 internal chords, and since no two consecutive chords can both be flipped in the same simultaneous flip set, we need at least 4 simultaneous flips to introduce ears for all vertices.

Then $c_1 = \frac{\beta(d)}{\log \frac{1}{1-\gamma}}$ and to minimize c_1 we have to decrease the size of $\beta(d)$ and increase the size of γ . The largest possible value of $\gamma = \frac{d-3}{3d-6} = \frac{d-2}{3d-6} - \frac{1}{3d-6} = \frac{1}{3}$ as $d \rightarrow \infty$, and the smallest possible value of $\beta(d)$ for $d \geq 10$ is 4. Thus using those

values we get $\forall d \geq 10 \mid c_1 \geq \frac{4}{\log \frac{3}{2}} \approx 6.83$. Note that in fact the value of c_1 must be much more since value of $\beta(d)$ will also go up as d increases.

Finally, we can see from Table 3.1, the optimal value of c_1 for $4 \leq d \leq 10$ is ≈ 5.52 for $d = 5$ which gives better result than for values of d exceeding 10. Thus c_1 is minimum for $d = 5$.

Chapter 4

Interlaced Algorithm

In this chapter, we make yet another modification of the Bose et al. (2007) algorithm. This time we alternately process both the maximal outerplane graphs defined by the Hamiltonian cycle in a way that guarantees a larger fraction of vertices to be processed into ears in each iteration.

4.1 An Overview of the Algorithm

In Chapter 2 we have seen how to transform any triangulation into a Hamiltonian triangulation. The Hamiltonian cycle of the triangulation divides it into two maximal outerplane graphs. Instead of processing these two maximal outerplane graphs one at a time, in this chapter we use an interlaced method to process both the interior and the exterior of the Hamiltonian cycle. This requires a logarithmic number of simultaneous flips to ensure that the dual tree of both the maximal outerplane graphs is logarithmic in diameter to the order of the graph. In the next step, we simultaneously introduce a dominant vertex inside each of the two maximal outerplane subgraphs introduced by the Hamiltonian cycle—transforming the graph into a canonical triangulation. The interlaced algorithm consists of the following steps.

1. $G_1 \rightarrow G_H$: Transforming any triangulation into Hamiltonian triangulation with a Hamiltonian cycle H .
2. $G_H \rightarrow G'_H$: Transform G_H into G'_H preserving H such that the diameter of the dual tree of the subgraphs $G'\{H\}_{int}$ and $G'\{H\}_{ext}$ are both logarithmic.
3. $G'_H \rightarrow \Delta_n$: Transform G'_H into Δ_n by simultaneously introducing a dominant vertex in $G'\{H\}_{int}$ and another in $G'\{H\}_{ext}$.
4. $\Delta_n \rightarrow G_2$: Do the reverse to transform Δ_n to target triangulation G_2 .

Step 1 and Step 4 of the interlaced algorithm are identical to Step 1 and Step 4 of Bose et al.'s algorithm in Section 2.4.4. Step 2 is explained in Sections 4.1–4.4 and Step 3 in Section 4.5. Finally the chapter ends with complexity analysis in Section 4.6.

4.2 $G_H \rightarrow G'_H$: Transformation to Logarithmic Diameter

In Chapter 3 we have seen how Bose et al.'s algorithm can be directly improved by using Lemma 3.3, which asserts that no two consecutive chords can both be blocked. The lemma allows us to reduce the degree of all the vertices in the independent set I from degree ≤ 4 down to ≤ 3 on the first simultaneous flip of the iteration. However, in the resulting graph there can still be vertices in I with one internal chord such that this internal chord is blocked by an external chord. Lemma 3.11 guarantees that at least half of these vertices can be turned into ears in the second simultaneous flip. In the interlaced algorithm introduced in this chapter, we present a method that reduces the degree of all the vertices of the independent set down to 2 in the second simultaneous flip. This is done by identifying a set of vertices called *avoiding vertex set* (See Definition 4.6) on each iteration. These are the vertices whose internal chords cannot be reduced to zero to produce an ear. So by avoiding these vertices from being inserted to the independent set of the next iteration for producing ears, we can ensure that none of these remaining chords are blocked. As a result, we are able to create an ear for each vertex in the independent set.

4.2.1 Overview of the Process

Let G be a triangulation with a Hamiltonian cycle H . Each iteration consists of 2 simultaneous flips. The graph obtained at the beginning of i -th iteration is denoted by G^i . In $G^1 = G$, the Hamiltonian cycle H defines two empty cycles—one is the cycle and its interior as the empty region, the other is the cycle and its exterior as the empty region. Every graph G^i has two empty cycles named C_{ear}^i and C_{prep}^i , each defined with a distinct empty region. For the first iteration, G_{ear}^1 and G_{prep}^1 are the two maximal outerplane subgraphs defined by H in G . The maximal outerplane subgraph defined by C_{ear}^i and its empty region is referred to as G_{ear}^i . Similarly, the maximal outerplane subgraph defined by C_{prep}^i and its empty region is referred to as G_{prep}^i . On the i -th iteration we create ears in G_{ear}^i while preparing G_{prep}^i for the next iteration to create ears in. Furthermore, the two cycles C_{ear}^i and C_{prep}^i are non-overlapping—meaning none of the chords in the empty (*local*) region of C_{prep}^i are present in G_{ear}^i and none of the chords in the empty (*local*) region of C_{ear}^i are present in G_{prep}^i .

On the i -th iteration a set of independent vertices I_{ear}^i (called *earring vertex set*) is identified in the subgraph G_{ear}^i , whose degree is brought down to 2 to create ears in G_{ear}^i (see Section 4.3.2). The process of creating ears in G_{ear}^i provides us with an ordered set of simultaneous flip sets $S_{ear}^i = (S_{ear_1}^i, S_{ear_2}^i)$. Meanwhile on the same iteration a set of pairwise non-consecutive vertices I_{prep}^i (called *avoiding vertex set*) on the cycle C_{prep}^i is identified (see Section 4.3.3). These are the vertices that cannot be turned into ears in G_{ear}^{i+1} on the next iteration. As described in Section 4.3.4, we use a method we named the *Preparation Method* (see Section 4.4) to process G_{prep}^i for the next iteration. This method returns another ordered simultaneous flip set $S_{prep}^i = (S_{prep_1}^i, S_{prep_2}^i)$. When creating ears on the next iteration (see Section 4.3.3 and 4.3.5), without compromising the cardinality of the independent set, the vertices of avoiding vertex set is excluded from it. Notice that in the interlaced algorithm, each iteration i for $i > 1$ depends on having a previous iteration where G_{prep}^{i-1} was prepared by using the Preparation Method. Since we cannot do this for the first iteration, the way to execute the first iteration is explained in Section 4.3.7.

FIGURE 4.1: C_{ear} and its local region is drawn in blue. C_{prep} and its local region is drawn in red. White regions denote produced ears. Green regions are foreign to both empty cycles and have been processed. All regions are triangulated.

In Section 4.3.6 we prove that both $S_1^i = S_{ear_1}^i \cup S_{prep_1}^i$ and $S_2^i = S_{ear_2}^i \cup S_{prep_2}^i$ are valid simultaneous flip sets in their respective graphs. Let $G^{i'} = G^i \langle S_1^i \rangle$ and $G^{i''} = G^{i+1} = G^{i'} \langle S_2^i \rangle$. Once we obtain G^{i+1} , we define G_{ear}^{i+1} as the same subgraph as $G_{prep}^{i''}$ present in G^{i+1} . The cycle C_{prep}^{i+1} is defined as the empty cycle obtained after excluding the degree 2 vertices (ears) of $G_{ear}^{i''}$ from the empty cycle $C_{ear}^{i''}$ present in G^{i+1} . The subgraph G_{prep}^{i+1} now denotes the maximal outerplane graph defined by the empty cycle C_{prep}^{i+1} . In other words the corresponding dual tree of G_{prep}^{i+1} is the same as corresponding dual tree of $G_{ear}^{i''}$ with all its leaves removed, and the empty cycle C_{prep}^i in G^i is the same as the empty cycle C_{ear}^{i+1} in G^{i+1} .

An example of the process is illustrated in Figure 4.1. Figure 4.1a corresponds to the graph G^1 having no processed subgraph that is denoted by green colour. It has a Hamiltonian cycle $H = (a, b, c, d, \dots, o, p)$. Note that the Hamiltonian cycle of G^i is always a subgraph of G^{i+1} . In fact, every cycle C_{ear}^k for $1 \leq k \leq i$ (and hence also C_{prep}^k) is preserved in G^{i+1} . Figure 4.1b corresponds to graph G^i at the beginning of iteration i . The green subgraph is the processed part which will continue to increase starting with an empty subgraph in G^1 . The red subgraph corresponds to G_{prep}^i , whereas the blue subgraph corresponds to G_{ear}^i . Figure 4.1c is the graph $G^{i''}$ obtained after two simultaneous flips. The ears prepared in the i -th iteration are drawn in white. In Figure 4.1d G^{i+1} is the same as $G^{i''}$ excepting that white portion is now green, denoting the fact that it has been processed. Regions corresponding to blue and red swap colours. After the final iteration the whole graph G^k has been processed and therefore is coloured green in Figure 4.1e. Now for both the maximal outerplane graphs $G^k \{H\}_{int}$ and $G^k \{H\}_{ext}$ (see Definition 4.7), their corresponding dual trees have a diameter of $\mathcal{O}(\log n)$.

4.2.2 Definition

For ease of exposition, we introduce the following definitions to formally define structures forming the skeleton of our algorithm.

Definition 4.1 (Ordered Set of Flippable Sets). *Given a plane graph $G_0 = (V, E)$ and an ordered set of edge sets $S = (S_1, S_2, \dots, S_k)$, such that S_i is simultaneously flippable in G_{i-1} , and for $1 \leq i \leq k$, $G_i := G_{i-1} \langle S_i \rangle$, then S is called an ordered set of flippable sets. The resulting graph obtained by sequentially flipping S in G is written as $G \langle S \rangle$.*

Given two ordered sets of flippable sets $S_1 = (S_{1,1}, S_{1,2}, \dots, S_{1,m})$ and $S_2 = (S_{2,1}, S_{2,2}, \dots, S_{2,n})$ where $m \leq n$, we define their union as following.

$$S_1 \cup S_2 := (S_{1,1} \cup S_{2,1}, S_{1,2} \cup S_{2,2}, \dots, S_{1,m} \cup S_{2,m}, S_{2,m+1}, S_{2,m+2}, \dots, S_{2,n})$$

Definition 4.2 (Non-overlapping Maximal Outerplane Subgraphs). *Let G be a triangulation with two empty cycles C_1 and C_2 . Furthermore, let G_1 and G_2 be the maximal outerplane subgraphs defined by C_1 and C_2 along with their corresponding local regions respectively. If no local chord of C_1 is an edge in G_2 and no local chords of C_2 is an edge in G_1 , then C_1 and C_2 are non-overlapping empty cycles, and G_1 and G_2 are non-overlapping maximal outerplane subgraphs in G .*

Note that two non-overlapping empty cycles/maximal outerplane subgraphs may have vertices and edges of the cycles in common. In the interlaced algorithm, for any iteration i , G_{ear}^i and G_{prep}^i are non-overlapping maximal outerplane subgraphs of G^i .

Definition 4.3 (Local Chord, Foreign Chord). *Given a plane graph $G = (V, E)$ with an empty cycle C , a chord of C is a local chord if it passes through C 's local region. Otherwise it is a foreign chord of C .*

In Chapters 2 and 3 all the empty cycles we considered have their local regions located in the interior of the cycle. Thus in all previous lemmas, internal chords are local chords, and external chords are foreign chords of their respective empty cycles. Because C_{ear}^{i+1} and C_{prep}^{i+1} point to different regions than C_{ear}^i and C_{prep}^i respectively, for clarity when dealing with chords of empty cycles we shall be using the terms local chord and foreign chord for the rest of this chapter.

Definition 4.4 (k -chord). *Given a triangulation $G = (V, E)$ with an empty cycle C let p_0 and p_k be two vertices on C connected by the chord $p_0 p_k$ such that there is a path $P = (p_0, p_1, \dots, p_k)$ of length k on the cycle C . Then $p_0 p_k$ is called a k -chord of C .*

A k -chord of an n -cycle is also an $(n - k)$ -chord.

Definition 4.5 (Avoiding Vertex). *Given a triangulation $G = (V, E)$ with an empty cycle C let pq be a foreign 2-chord of C creating a 3-cycle (p, q, r) with another vertex r of C such that pr and qr are edges of C . Then r is called pq 's avoiding vertex, and pq is referred to as r 's corresponding 2-chord.*

Note that these vertices p, q must be non-consecutive on the cycle C , as otherwise G is not simple.

Definition 4.6 (Avoiding Vertex Set, I_{avoid}). *Given the setup described in Section 4.2.1, the avoiding vertex set, referred to as I_{avoid}^i is the set of all avoiding vertices of C_{ear}^{i+1} in G^{i+1} .*

FIGURE 4.2: Empty cycle $C_{prep}^i = C_{ear}^{i+1}$ is drawn in blue in the triangulation G^{i+1} . Vertices of I_{avoid}^i are drawn in red. Remaining vertices of C_{ear}^{i+1} are drawn in green. Vertices outside C_{ear}^{i+1} are drawn in black.

FIGURE 4.3: (a, e, f, c) is the only chord-path of the empty cycle (a, b, c, d) .

An example of avoiding vertex set is provided in Figure 4.2. Here triangulation G^{i+1} is illustrated with an empty cycle $C_{prep}^i = C_{ear}^{i+1}$ drawn in blue. The avoiding vertex set $I_{avoid}^i = \{a, c, g, k, m\}$ is drawn in red. Each of these vertices have a corresponding foreign 2-chord on C_{prep}^i . Notice that i is not an avoiding vertex because mk is a local 2-chord of i in C_{ear}^{i+1} . It may seem that to know I_{avoid}^i we need to know G^{i+1} even before computing S^i . We shall show in Section 4.3.3 how to compute I_{avoid}^i without knowing G^{i+1} .

Definition 4.7 (Internal/External Subgraph of Cycle). Given a plane graph G with a cycle C , the internal subgraph of C in G , denoted by $G\{C\}_{int}$ or $G\{C\}$ is the edge induced subgraph of the edges of C and the edges in the interior of C in G . Similarly the external subgraph of C in G , denoted by $G\{C\}_{ext}$ is the edge induced subgraph of the edges of C and the edges in the exterior of C in G .

Definition 4.8 (Chord-Path). Given a graph G with an empty cycle C if there exists a path $P = (v_0 v_1, v_1 v_2, \dots, v_{k-1} v_k)$ such that only the first and the last vertices of P is on C , then P is called a chord-path of C . For the purpose of this work, by chord-path we mean non-adjacent foreign chord-path where v_0 and v_k are non-adjacent on C and the in-between vertices of the chord-path must lie on the foreign region of the empty cycle.

In figure 4.3 (a, e, f, c) is the only chord-path of the empty cycle $C = (a, b, c, d)$. The path (a, e, b) is a chord-path of C because a and b are consecutive on C . The path (a, e, b, f, c) is not a chord-path of C because b is an in-between vertex of the path that also lies on C .

Definition 4.9 (Earing Vertex, Earing Chord). *Given the setup described in Section 4.2.1, on the i -th iteration, The set of earing vertices referred by I_{ear}^i is the set of independent vertices in G_{ear}^i whose degree is to be reduced to 2 to produce ears in $G_{ear}^{i''}$. Each vertex of I_{ear}^i is called an earing vertex. Any local chord of C_{ear}^i incident to an earing vertex of C_{ear}^i is called an earing chord.*

Definition 4.10 (Provisioning Chord). *Given the setup described in Section 4.2.1, on the i -th iteration, a provisioning chord e is a local chord of the empty cycle C_{prep}^i that is seen by a vertex v in I_{avoid}^i of the cycle. So when e is flipped, the degree of v (the avoiding vertex of e) increases. If flipping e increases the degree of exactly one vertex in I_{avoid}^i , then e is called a singly provisioning chord and otherwise it must be increasing the degree of two vertices of I_{avoid}^i and thus is called a doubly provisioning chord.¹*

Definition 4.11 (Neighbourhood). *Given a graph $G = (V, E)$ and a vertex $v \in V$, the neighbourhood of v (denoted by $N_G(v)$) is the set of vertices v is connected to. The closed neighbourhood of v (denoted by $N_G[v]$) is the neighbourhood of v together with v . $N_G[v] = N_G(v) \cup \{v\}$.*

Definition 4.12 (Dominating Vertex, Corner Vertex). *Given a graph $G = (V, E)$, for two vertices $u, v \in V$, if $N_G[u] \subseteq N_G[v]$ then u is called a corner vertex of v and v is called a dominating vertex to u .*

4.3 Creating Ears in G_{ear}^i

As illustrated in Figure 4.4, on the i -th iteration, let u be a vertex in I_{ear}^i with 2 local chords $e_1 = uv$ and $e_2 = uw$ in G_{ear}^i (subgraph of G^i). Let p, q be the two vertices adjacent to u on C_{ear}^i such that uv and uw are respectively seen by p and q . To reduce the degree of u to 2 in G_{ear}^i , these two chords can be flipped in two different orders— $(\{e_1\}, \{e_2\})$, and $(\{e_2\}, \{e_1\})$. Let e'_1 be the chord obtained by flipping e_1 individually in G^i and let e'_2 be the chord obtained by flipping e_2 individually in G^i . Regardless of their order, the flipping of the second chord would produce the chord $e' = pq$ introducing the ear pqu in $G_{ear}^{i''}$.

Notice that e' and one of e'_1 and e'_2 may end up being foreign 2-chords of $G_{prep}^{i''}$, and thus the order in which e_1 and e_2 are flipped is important as the set of avoiding vertices I_{avoid}^i depends upon whether $G^{i''}$ contains $\{e'_1, e'\}$ or $\{e'_2, e'\}$. On the other hand, without knowing I_{avoid}^i we cannot compute $S_{prep_1}^i$ and $S_{prep_2}^i$. Furthermore, it may be possible that one or both of e_1 and e_2 are blocking provisioning chords, complicating the process of computing the provisioning chords and the blocking chords as we do not yet know the vertices of I_{avoid}^i , which demands knowing the order in which the earing chords are flipped.

¹While the definition of provisioning chord is accurate for the purpose of this section, a more complete definition is provided in Definition 4.28.

FIGURE 4.4: G^i_{ear} is an externally maximal outerplane triangulation with its maximal empty cycle C^i_{ear} drawn in blue. Alternating curved lines represent a path consisting one or more edges. Dotted edges are not part of the graph. Blue shaded region can be further triangulated.

In this section we show how to compute the ordered set of flippable sets $S^i := (S^i_{ear_1} \cup S^i_{prep_1}, S^i_{ear_2} \cup S^i_{prep_2})$, and prove its flippability. This is done using the following steps that are explained in details.

1. Identifying the earing vertex set I^i_{ear} (Section 4.3.2)
2. Identifying I^i_{avoid} by partially computing S^i_{ear} (Section 4.3.3)
3. Computing S^i_{prep} using the *Preparation Method* (Section 4.3.4)
4. Completing the computation of S^i_{ear} —Obtaining S^i (Section 4.3.5)

This section will describe how an earing chord may be blocking a provisioning chord, how to compute the ordering in which the earing chords would be flipped without yet knowing which provisioning chords $S^i_{prep_1}$ and $S^i_{prep_2}$ will consist of. But before that, we need to familiarize ourselves with some properties of the Preparation Method.

4.3.1 Properties of Preparation Method

As described in Section 4.2.1, to preprocess G^i_{prep} for creating ears in the next iteration, the interlaced algorithm relies on the Preparation Method. The presentation of this procedure is long and technical. We present full details of this method in Section 4.4. For now, we use it as a black box which satisfies the following properties.

1. Input parameters of Preparation Method (See Section 4.4 Input Parameter 1a–1c):
  - (a) A maximal outerplane graph G^i_{prep} with its maximal cycle C^i_{prep} .
  - (b) A set of pairwise non-consecutive vertices I^i_{avoid} on C^i_{prep} .
  - (c) Each vertex of I^i_{avoid} has at least one local chord in G^i_{prep} .
2. Output parameters of Preparation Method:
  - (a) An ordered set of two simultaneous flip sets $S^i_{prep} := (S^i_{prep_1}, S^i_{prep_2})$ on G consisting of only provisioning chords. (See Lemma 4.45 Property 1)

- (b) In $G_{prep}^{i''} := G_{prep}^i \langle S_{prep} \rangle$, for every vertex $v \in I_{avoid}^i$ at least one of the following is true (See Lemma 4.45 Property 2):
- i. v has at least 3 local chords in $G_{prep}^{i''}$.
  - ii. v dominates a vertex u in $G_{prep}^{i''}$ where $u \notin I_{avoid}^i$.
- (c) No more than one singly provisioning chord of any vertex $v \in I_{avoid}^i$ is in $S_{prep_1}^i$ (See Lemma 4.45 Property 3)

4.3.2 Identifying the Earing Vertex Set I_{ear}^i

G_{ear}^i is a maximal outerplane graph. So by Equation 2.1 of Theorem 2.66 there is a set of independent vertices I on G_{ear}^i of cardinality $|I| \geq n/6$, where n is the order of G_{ear}^i such that for all $v \in I$, $\deg_{G_{ear}^i}(v) \leq 4$. Using Lemma 3.3 we can compute $S_{ear_1}^i$ such that no vertices of I has more than 1 local chord in $G_{ear}^{i'}$. However, if we want to create an ear for each vertex in I , we must ensure that all the remaining chords of I on C_{ear}^i are unblocked in $G^{i'}$ —and thus can be inserted in $S_{ear_2}^i$. The next lemma proves that any vertex whose only local chord is blocked must be an avoiding vertex with a foreign 2-chord.

Lemma 4.13. *Given a triangulation G and an empty cycle C , for any vertex v on the cycle C with exactly one local chord uv in C , uv is blocked IFF v is an avoiding vertex of a foreign 2-chord of C in G .*

FIGURE 4.5: Empty cycle C is drawn in blue. Zigzagged arcs represent a set of edge(s). Dashed edges may be a chord or an edge of the empty cycle they are in. Shaded region may be further triangulated.

Proof. In Figure 4.5 let G be a triangulation with an empty cycle C (drawn in blue) with v having exactly one local chord uv on the empty cycle C . Since every face of G is a triangle, let uv be seen by the two vertices p and q then vp and vq must be edges of G . Since uv is the only local chord of v in C , vp and vq must be edges of the cycle C making p and q adjacent to v on C .

$\Rightarrow$ If uv is a blocked chord then pq must be its blocking chord. Since uv separates p and q in the local region of C , due to Lemma 3.2 pq cannot be a local chord of C . Furthermore, since pv and vq are edges on C , pq must be a foreign 2-chord of C with v as its avoiding vertex.

$\Leftarrow$ Since p and q are consecutive to v on the cycle C , v 's corresponding 2-chord must be pq . Thus if the foreign 2-chord pq of the avoiding vertex v is in G , uv is a blocked edge in G . $\square$

As a result of Lemma 4.13, to ensure that we can flip all the earing chords, we are interested in having an independent set of vertices where none of the vertices have a foreign 2-chord on C_{ear}^i . This is where the Preparation Method explained in Section 4.4 becomes useful. For any iteration $i > 1$ we processed the maximal outer-plane subgraph G_{ear}^i in the previous iteration ensuring that G_{ear}^i has the Property 2b described in Section 4.3.1. Notice, that on the $(i - 1)$ -th iteration, vertex set I_{avoid}^{i-1} denotes to the set of vertices each of which has a foreign 2-chord of the cycle C_{ear}^i in G^i .

We first obtain a set of independent vertices I of degree at most 4 on G_{ear}^i using Equation 2.1 of Theorem 2.66 such that $|I| \geq n/6$. We know that some of these vertices can be avoiding vertices. However, for any iteration $i > 1$, due to Property 2b of the Preparation Method, any vertex v in I that is an avoiding vertex would dominate another vertex u in G_{ear}^i that is not an avoiding vertex, so $\deg_{G_{ear}^i}(u) \leq \deg_{G_{ear}^i}(v)$. Thus we can replace each vertex of I that is an avoiding vertex, with a non-avoiding vertex and call the new set I_{ear}^i . Now we have to ensure that I_{ear}^i is still a set of independent vertices, and the order of I_{ear}^i is the same as the order of I . The next lemma indeed confirms independence and the required order of I_{ear}^i .

Lemma 4.14. *Given a graph $G = (V, E)$, an independent set of vertices $I \subseteq V$, a vertex $v \in I$ and another vertex $u \in V$ such that v dominates u , then $I' = (I \setminus \{v\}) \cup \{u\}$ is an independent vertex set of the same cardinality as I .*

Proof. Since v dominates u and $v \in I$, $u \notin I$. Furthermore, since every vertex u is connected to, v is also connected to and none of the other vertices of I is connected to v , none of the vertices in $I \setminus \{v\}$ can be connected to u . Thus $I' = I \setminus \{v\} \cup \{u\}$ is an independent set. Because we get I' by replacing one vertex of I with another vertex, $|I'| = |I|$. $\square$

As we can see in Lemma 4.14, every time we replace an avoiding vertex v from our independent set with another vertex u that v dominates, the new set is still an independent set with the same cardinality. Thus repeating the process until there are no avoiding vertices in the set will produce our preferred independent set I_{ear}^i with $|I_{ear}^i| \geq n/6$, where n is the order of G_{ear}^i such that no vertex of I_{ear}^i has a foreign 2-chord in G^i .

Finally, it is important to establish that no vertex in I_{ear}^i can end up as an avoiding vertex of $C_{ear}^{i'}$ after the first simultaneous flip. Since we only flip earing and provisioning chords, to introduce a corresponding foreign 2-chord of a vertex of I_{ear}^i in $C_{ear}^{i'}$ or $C_{ear}^{i''}$, this foreign 2-chord must have been introduced by the flipping of a provisioning chord. The next lemma establishes this fact. ²

Lemma 4.15. *Under the setup described in Section 4.2.1, flipping a provisioning chord of a vertex of I_{prep}^i does not make any vertex of I_{ear}^i an avoiding vertex of C_{ear}^i .*

Proof. Let u be a vertex of I_{ear}^i consecutive to the vertices p and q on C_{ear}^i in G^i . Observe that $S_{ear_1}^i$ and $S_{ear_2}^i$ are flippable in G_{ear}^i and $G_{ear}^{i'}$ since they are obtained following the same method as in Chapter 3—except by using the set of independent

²While we know that no vertex in I_{ear}^i can have a corresponding foreign 2-chord of C_{ear}^i in G^i , we have to also ensure that none of the vertices in I_{ear}^i can end up with a foreign 2-chord of $C_{ear}^{i'}$ in $G^{i'}$ either. This is because if a vertex v in I_{ear}^i indeed receives a corresponding foreign 2-chord of $C_{ear}^{i'}$ in $G^{i'}$, then flipping S_{ear}^i would introduce a corresponding local 2-chord of v on $C_{ear}^{i''}$ in $G_{ear}^{i''}$ —introducing parallel edges. Hence $G^{i''}$ will no longer be simple. In this case, S^i will no longer be a sequence of valid simultaneous flip sets. Theorem 4.25 ensures the flippability of S^i in G^i .

vertices I_{ear}^i . Moreover, $S_{prep_1}^i$ and $S_{prep_2}^i$ are flippable by Property 2a of the Preparation Method in Section 4.3.1. However, we do not know yet if $S_{ear_1}^i \cup S_{prep_1}^i$ and $S_{ear_2}^i \cup S_{prep_2}^i$ are flippable in G^i and $G^{i'}$ respectively. Assume we compute $G^{i''}$ anyway. Since u is in I_{ear}^i , the edge pq will be created as a local chord in $C_{ear}^{i''}$. For pq to be created as a foreign chord of $C_{ear}^{i''}$, it is sufficient to show that pq cannot be created as a local chord $C_{prep}^{i'}$ or $C_{prep}^{i''}$.

If pq is created in C_{prep}^i , then u is an avoiding vertex of C_{ear}^i and pq is a foreign 2-chord of C_{ear}^i . Since $u \in I_{ear}^i$, pq must have been introduced by flipping an edge that is in the foreign region of C_{ear}^i . Since we only flip earing and provisioning chords, this can only happen by the flipping of a provisioning chord of C_{prep}^i .

Let xy be such a provisioning chord that when flipped introduces the edge pq . Since xy is a chord, x and y are distinct from each other. Thus, u may be the same vertex as x or y but not both. Then xyp and xyq must be faces. Because C_{prep}^i is an empty cycle and because xy is a local chord of it, both p and q must be on the cycle C_{prep}^i . Since xy is a provisioning chord, at least one of p and q must be in I_{avoid}^i . Notice that pq shall be the local chord of $C_{ear}^{i''}$ —making pq a foreign 2-chord of $C_{prep}^{i''}$. Since both p and q have at least one foreign chord of $C_{prep}^{i''} = C_{ear}^{i+1}$, neither of these two vertices can be in I_{avoid}^i . Hence xy cannot be a provisioning chord of C_{prep}^i . $\square$

4.3.3 Identifying I_{avoid}^i by Partially Computing S_{ear}^i

The Preparation Method takes G_{prep}^i as its maximal outerplane graph, and a set of pairwise non-consecutive vertices I_{avoid}^i on the maximal empty cycle of G_{prep}^i . It returns an ordered set of flippable sets S_{prep}^i consisting two simultaneous flip sets, such that in $G_{prep}^{i''} := G_{prep}^i \langle S_{prep}^i \rangle$ each of those vertices of I_{avoid}^i has at least 3 local chords, or it dominates another vertex of the empty cycle $G_{prep}^{i''}$ that is not in I_{avoid}^i . Then for the next iteration $C_{ear}^{i+1} := C_{prep}^{i''}$. So if we can set I_{avoid}^i as the set of all the vertices that have a corresponding foreign 2-chord on the cycle $C_{prep}^{i''}$, due to Lemma 4.13, by avoiding these vertices we would be able to select a set of independent vertices I_{ear}^{i+1} on the next iteration where none of the vertices' last earing chord can be blocked. So our goal is to set I_{avoid}^i as the set of all vertices in G^i that will become avoiding vertices of the cycle C_{ear}^{i+1} in G^{i+1} .

Although we do not yet know what the graph G^{i+1} shall look like, notice that only the foreign chords of C_{ear}^{i+1} can play a role in creating avoiding vertices in C_{ear}^{i+1} . On the i -th iteration we only flip local chords of C_{ear}^i and C_{prep}^i of G^i . Thus if a foreign 2-chord of C_{ear}^{i+1} is not present in G^i , it must have been produced by flipping an earing chord of C_{ear}^i . Thus once we have computed I_{ear}^i , we iterate through all the foreign chords of C_{prep}^i . If such a chord e is a foreign 2-chord of the cycle and is not an earing chord of C_{ear}^i , we insert its corresponding avoiding vertex in I_{avoid}^i . At the end of the i -th iteration, no vertex in I_{ear}^i is incident to a local chord in $G_{ear}^{i''}$. So, for any three consecutive vertices p, u, q on the cycle C_{ear}^i such that $u \in I_{ear}^i$, the face puq will be an ear in $G_{ear}^{i''}$. Thus if the newly introduced chord pq is a foreign 2-chord of $G_{prep}^{i''}$, we insert its corresponding avoiding vertex in I_{avoid}^i .

FIGURE 4.6: Zigzagged arcs are paths and may overlap. Blue and red denote C_{ear}^i and C_{prep}^i respectively. Green region is foreign to both empty cycles and has been processed. Flipping uv results in pw . Non-shaded regions are faces, dashed edges may not be distinct. Dotted edge is not part of G^i . Shaded region may be further triangulated.

Steps for Partially Computing I_{avoid}^i

1. For every foreign 2-chord e of C_{prep}^i in G_{prep}^i that is not an earing chord of I_{ear}^i , insert the corresponding avoiding vertex of e on C_{prep}^i to I_{avoid}^i .
2. For every three consecutive vertices p, u, q on C_{ear}^i where $u \in I_{ear}^i$, and pq forms a foreign 2-chord of C_{prep}^i , insert the corresponding avoiding vertex of pq for C_{prep}^i in I_{avoid}^i .

Notice that we are still not done preparing I_{avoid}^i since any earing vertex with two internal chords produces two new chords either of which could be foreign 2-chords of $G_{prep}^{i'}$, where we are yet to learn whether the chord that was flipped first among the two would produce a foreign 2-chord of $C_{prep}^{i'}$ in $G^{i'}$. The next lemma is used to resolve this issue.

Lemma 4.16. *Under the setup described in Section 4.2.1, let u be an earing vertex of G_{ear}^i with two earing chords e_1 and e_2 . Furthermore, flipping e_1 in G_{ear}^i produces e'_1 , which is a foreign 2-chord of $G_{prep}^{i'}$. Then at most one of e_1 and e_2 can be blocking a local chord of G_{prep}^i .*

Proof. As illustrated in Figure 4.4, let us assume the contrapositive that both $e_1 = uv$ and $e_2 = uw$ are blocking local chords of G_{prep}^i . By the hypothesis, e'_1 (but not necessarily e'_2) is a foreign 2-chord of $G_{prep}^{i'}$. WLOG let e_1 and e_2 be seen by p and q respectively, and thus vp and wq must be edges of G_{ear}^i . Notice that each of the arcs $\widehat{pv}$, $\widehat{vw}$ and $\widehat{wq}$ may be single edges.

To complete the picture, let us also consider the other maximal outerplane subgraph G_{prep}^i in G^i illustrated in Figure 4.6a. In the illustration an embedding is chosen where C_{prep}^i is in the interior of C_{ear}^i . So G_{prep}^i (drawn in red) is a maximal outerplane graph with its local region in the interior of the maximal cycle C_{ear}^i (drawn in blue). Let lm and no be local chords of C_{prep}^i that are blocked by $e_1 = uv$ and $e_2 = uw$ respectively. Then lm is seen by both u and v and no is seen by u and w . WLOG let l, m, n, o be on $(\widehat{uv})$, $(\widehat{vu})$, $(\widehat{uw})$, $(\widehat{wu})$ respectively on the cycle C_{prep}^i . As a result

lm, mv, nu, vw must be distinct faces, making l, m, u, v distinct vertices from each other, and n, o, w, u distinct vertices from each other.

Because $e'_1 = pw$ would form a foreign 2-chord of C_{prep}^i , there must be exactly 1 vertex x between p and w on C_{prep}^i . Since $(\widehat{wp})$ on C_{prep}^i has the distinct vertices w, o, u and p , $(\widehat{pw})$ cannot contain any other vertices apart from p, v and w —making $x = v$. Taking this into consideration we obtain the graph illustrated in Figure 4.6b where pv and vw must both be edges of both C_{ear}^i and C_{prep}^i . But since vw is an edge of C_{prep}^i , there cannot be any other vertices on $(\widehat{vw})$ in C_{prep}^i . Since m is distinct from u and v , m must be in $(\widehat{wu})$. Similarly n must be on $(\widehat{uw})$, and o must be on $(\widehat{wu})$. Observe the following two cases.

1. **Case 1 ($m \neq w$):** Then $m \in (\widehat{wu})$. In this case lm separates u from w , and hence the two vertices n and o cannot both be connected to u and w in G_{prep}^i while ensuring lm and no are distinct edges.
2. **Case 2 ($m = w$):** Since um is an edge, uw must be a local chord of G_{prep}^i separating $(\widehat{uw})$ from $(\widehat{wu})$. Then due to Lemma 3.2 no cannot be a local chord in G_{prep}^i .

Thus when e'_1 is a foreign 2-chord of C_{prep}^i , at most 1 of e_1 and e_2 can be a blocking chord to provisioning chords of G_{prep}^i . Same way we can prove the statement for when e'_2 is a foreign 2-chord of C_{prep}^i , completing this proof. $\square$

Using Lemma 4.16 we conclude that if at least one chord (say e_1) of an earing vertex forms a foreign 2-chord when flipped then at most one of e_1 and e_2 can be blocking a provisioning chord. So, for every vertex $v \in I_{ear}^i$ with two earing chords e_1 and e_2 and their individually flipped chord e'_1 and e'_2 , if one of $\{e'_1, e'_2\}$ is a foreign 2-chord of C_{prep}^i , we insert whichever of $\{e_1, e_2\}$ blocking any local chord of C_{prep}^i in $S_{ear_1}^i$ and the other in $S_{ear_2}^i$. Due to Lemma 2.51, the chord inserted in $S_{ear_1}^i$ must be flippable. If neither of them are blocking any local chord of C_{prep}^i , then due to Lemma 3.3 at least one of them must be unblocked and hence flippable. We put this unblocked chord in $S_{ear_1}^i$ and the other in $S_{ear_2}^i$. Since v cannot have a foreign 2-chord of C_{ear}^i in G^i , due to Lemma 4.13 the chords in $S_{ear_2}^i$ must also be flippable in G^i .

So for all earing vertex $u \in I_{ear}^i$, whose at least one earing chord produces a foreign 2-chord of C_{prep}^i when individually flipped, we do the following.

Steps for Completing The Computation of I_{avoid}^i

1. If u has two earing chords e_1, e_2 where e_2 is blocked and e'_2 is a foreign 2-chord of C_{prep}^i then insert e_1 in $S_{ear_1}^i$ and e_2 in $S_{ear_2}^i$.³
2. If an earing chord e of u is blocking any local chord of G_{prep}^i then due to Lemma 2.51 the blocking chord is flippable, so insert e in $S_{ear_1}^i$, and any remaining earing chord of u in $S_{ear_2}^i$.

³The reason behind having Step 1 is explained in Lemma 4.21 in Section 4.3.6. Notice that since e'_2 is a foreign 2-chord of C_{prep}^i and since e_2 is blocked, the corresponding avoiding vertex of e'_2 in G_{prep}^i has no local incident chords in C_{prep}^i . This would violate Requirement 1c of the Preparation Method in 4.3.1. Thus by not flipping e_2 we ensure that we do not introduce an avoiding vertex with no incident local chords.

3. If u has two earing chords and neither of them are blocking any local chord of I_{avoid}^i on C_{prep}^i then due to Lemma 3.3 at least one of them (e) is flippable so insert such an unblocked edge e in $S_{ear_1}^i$ and the other in $S_{ear_2}^i$.
4. If u has only one earing chord e , and e does not block any local chord of I_{avoid}^i on C_{prep}^i then since u cannot have a foreign 2 chord in $C_{ear}^{i''}$, it must be flippable in $G^{i'}$ so insert e in $S_{ear_2}^i$.
5. In Steps 1–4, if any new edge is a foreign 2-chord of $C_{prep}^{i''}$ in $G^{i''}$ then insert its corresponding avoiding vertex of $C_{prep}^{i''}$ in I_{avoid}^i .

In Section 4.3.3 we partially computed the avoiding vertex set I_{avoid}^i based on all the edges of G^i that were preserved in G^{i+1} . By deciding the order in which the earing chords that produce foreign 2-chord of C_{prep}^i are flipped, we have now completed the computation of I_{avoid}^i . Also, we have partially computed $S_{ear_1}^i$ and $S_{ear_2}^i$ such that every edge in these two sets is (individually) flippable in their respective graphs. To complete the computation of S_{ear}^i , we still need to decide on the flipping order of the two earing chords of every vertex $u \in I_{ear}^i$ where neither of the earing chords produce foreign 2-chords of C_{prep}^i when flipped. We do this once we have computed S_{prep}^i by using the Preparation Method.

4.3.4 Computing S_{prep}^i Using the Preparation Method

Now that we have fully computed I_{avoid}^i , we can use the Preparation Method referenced in Section 4.3.1 to compute $S_{prep}^i := (S_{prep_1}^i, S_{prep_2}^i)$. Notice that the Preparation Method requires that the vertices of I_{avoid}^i must be non-consecutive on the cycle C_{prep}^i . The next lemma proves that this requirement is satisfied.

Lemma 4.17. *Under the setup described in Section 4.2.1, the vertices of I_{avoid}^i are pairwise non-consecutive in C_{prep}^i .*

Proof. By definition vertices of I_{avoid}^i are on the cycle C_{ear}^{i+1} , each having a corresponding foreign 2-chord in G^{i+1} . We proceed by contradiction. Let $u, v \in I_{avoid}^i$ be two consecutive vertices on the cycle C_{ear}^{i+1} . Furthermore, let p be the other vertex consecutive to u on C_{ear}^{i+1} , and q be other the vertex consecutive to v on C_{ear}^{i+1} . Then pv and uq are the corresponding foreign 2-chords of the vertices u and v respectively on C_{ear}^{i+1} in G^{i+1} . But then pv separates u from all the other vertices of C_{ear}^{i+1} on the foreign region of C_{ear}^{i+1} . Thus due to Lemma 3.2 uq cannot be foreign 2-chord of C_{ear}^{i+1} . Notice that since on the i -th iteration, we only flip local chords of the two non-overlapping empty cycles, the edges of the two cycles C_{prep}^i and C_{ear}^i are preserved. And at the end of the i -th iteration, the cycle $C_{prep}^{i''}$ is assigned as the C_{ear}^{i+1} for the next iteration, $C_{ear}^{i+1} = C_{prep}^{i''} = C_{prep}^{i'} = C_{prep}^{i''}$. Thus the vertices of I_{avoid}^i must be pairwise non-consecutive in all these empty cycles. $\square$

Thus I_{avoid}^i is pairwise non-consecutive on the cycle $C_{prep}^i = C_{ear}^{i+1}$.

FIGURE 4.7: Vertices of I_{avoid}^i have at least 1 local chord each in G_{prep}^i . G_{prep}^i and its empty region is drawn in red. G_{ear}^i and its empty region is drawn in blue. Green denotes processed region. Lightly shaded regions are faces. Zigzagged lines represent paths consisting one or more edges. Dashed edges may not be distinct. Shaded region may be further triangulated.

Lemma 4.18. *Under the setup described in Section 4.2.1, every vertex $v \in I_{avoid}^i$ must have at least one local chord in G_{prep}^i .*

Proof. We proceed by contradiction. Assume a vertex $v \in I_{avoid}^i$ has no local chord in C_{prep}^i (refer to Figure 4.7). Let p and q be the two vertices consecutive to v on C_{prep}^i . Since the graph is triangulated and the empty cycle C_{prep}^i has > 3 vertices and since v has no local chord in G_{prep}^i , pq must be a local chord of G_{prep}^i creating the ear $p v q$.

Since $v \in I_{avoid}^i$, v has a corresponding 2-chord of the cycle C_{ear}^{i+1} in G^{i+1} . Since pq is a local chord of C_{prep}^i in G^i , pq cannot be a foreign chord of C_{prep}^i in G^i . Then the foreign chord pq must have been introduced to C_{ear}^i by flipping an earing chord st in S_{ear}^i . Then pst and qst were faces before this flip, and one of s and t must be an earing vertex in I_{ear}^i .

WLOG, let $s \in I_{ear}^i$. If st is the last earing chord of s then s would be an avoiding vertex of G^i with its corresponding foreign 2-chord in C_{ear}^i being pq —making $s \in I_{avoid}^{i-1}$. Thus st cannot be the last earing chord of s that was flipped on the i -th iteration. Then st must be the first of the two earing chords of s . In this case, since st is blocked by pq , due to Step 1 in Section 4.3.3 st could not have been the first of the two earing chords of s to be flipped, making it impossible for v to be in I_{avoid}^i . $\square$

Recall that G_{prep}^i is a maximal outerplane graph. Due to Lemma 4.17 and since $C_{prep}^i = C_{prep}'^i = C_{prep}''^i = C_{ear}^{i+1}$, I_{avoid}^i must also be pairwise non-consecutive in all these empty cycles. Finally, due to Lemma 4.18 every vertex of I_{avoid}^i has at least one local chord in G_{prep}^i . Thus G_{prep}^i and I_{avoid}^i fulfill all the requirements for the Preparation Method mentioned in Section 4.3.1. Thus using G_{prep}^i and I_{avoid}^i in the Preparation Method we obtain $S_{prep}^i = (S_{prep_1}^i, S_{prep_2}^i)$.

4.3.5 Completing the Computation of S_{ear}^i —Obtaining S^i

We may still have earing vertices of C_{ear}^i in G^i with 2 earing chords that have not yet been inserted in S_{ear}^i . When individually flipped, these earing chords do not produce foreign 2-chords of C_{prep}^i . The next lemma is used to decide the order in which they are to be flipped without blocking the edges in S_{prep}^i .

FIGURE 4.8: C_{ear} is drawn in blue, C_{prep} drawn in red, zigzagged arcs are paths and may overlap, shaded region may be further triangulated. Dashed chords of C_{prep} may be edges of the cycle C_{prep} . Blue, red and green shades are parts of G_{ear} , G_{prep} and the processed region in between respectively. l and m and o are vertices on C_{prep} such that lm is a provisioning chord of v

Lemma 4.19. *Under the setup described in Section 4.2.1, let u be an earing vertex of G_{ear}^i with two earing chords e_1 and e_2 . Furthermore, flipping e_1 and e_2 individually in G_{ear}^i produces e'_1 and e'_2 respectively—neither of which is a foreign 2-chords of C_{prep}^i . Then if either of e_1 and e_2 blocks a provisioning chord lm of C_{prep}^i , then lm must be a singly provisioning chord of the avoiding vertex u of C_{prep}^i .*

Proof. As illustrated in Figure 4.8a, u (earing vertex) is in I_{ear}^i with two local chords $e_1 = uv$ and $e_2 = uw$ of C_{ear}^i (drawn in blue) where neither of their corresponding flipped chord $e'_1 = pw$ $e'_2 = qv$ are foreign 2-chords of $C_{prep}^{i''}$ (drawn in red). Since $\deg_{G_{ear}^i}(u) = 4$, up and uq must be edges of C_{ear}^i . Because u is a vertex of the independent set I_{ear}^i in G_{ear}^i and since uv, up, uq and uw are also edges of G_{ear}^i , $v, p, q, w \notin I_{ear}^i$. Since we only flip chords incident to earing vertices, none of the edges in $\{vp, pu, qu, qw, vw\}$ can be in S_{ear}^i . As a result these 5 edges of G_{ear}^i is also present in $G_{ear}^{i''}$.

We are ready to prove the lemma by contradiction. WLOG, assume $e_1 = uv$ blocks a doubly provisioning chord lm of C_{prep}^i . Then both u and v are avoiding vertices of $C_{prep}^i = C_{ear}^{i+1}$ in G^{i+1} . Let x and y be the consecutive vertices of u on C_{ear}^i such that $x \in (\hat{u}v)$ and $y \in (\hat{v}u)$ of C_{ear}^i . Then xy must be a foreign 2-chord of $C_{prep}^{i''}$. Notice that u and v cannot be consecutive on the cycle $C_{prep}^{i''}$.

We need the following two properties:

1. **Property 1 (y and w must be the same vertex):** Notice that if w is not a vertex on $C_{prep}^{i''}$, then (vw, wq, qu) forms a chord-path separating the vertices in $(\hat{u}v)$ from the vertices in $(\hat{v}u)$ of the cycle $C_{prep}^{i''}$ in its foreign region. If w is a vertex on $C_{prep}^{i''}$, and vw is a chord of $C_{prep}^{i''}$, then vw separates the vertices in $(\hat{w}v)$ from the vertices in $(\hat{v}w)$. But then due to Lemma 3.2 xy cannot be a foreign 2-chord of v since $x \in (\hat{w}v)$ and $y \in (\hat{v}w)$. Thus for xy to be a foreign chord of $C_{prep}^{i''}$, y and w must be the same vertex.

2. **Property 2 (x and p must be the same vertex):** Similar to the proof of Property 1, we can show that if p is not in $C_{prep}^{i''}$ then (vp, pu) forms a chord-path separating x from y in the foreign region of $C_{prep}^{i''}$. If p is on $C_{prep}^{i''}$ then if vp is a chord then vp separates x from y in the foreign region of $C_{prep}^{i''}$. Thus due to Lemma 3.2 for xy to be a foreign chord of $C_{prep}^{i''}$, x and p must be the same vertex.

By setting $y = w$ and $x = p$ we get the graph illustrated in Figure 4.8b. Notice that now $e'_1 = pw$ forms a foreign 2-chord of $C_{prep}^i = C_{prep}^{i''}$. This is a contradiction. $\square$

Due to Lemma 4.19, when two earing chords e_1 and e_2 are consecutive such that neither of them produce a foreign 2-chord of C_{prep}^i then at most one of them can be blocking a provisioning chord. Due to Lemma 2.51 this blocking earing chord (say e_1) must be flippable. So we insert e_1 in $S_{ear_1}^i$ and insert e_2 in $S_{ear_2}^i$. Notice that in Section 4.3.1 Property 2c guarantees that no more than one singly provisioning chord of an avoiding vertex (in this case u) is in $S_{prep_1}^i$. This ensures that no provisioning chord that is in $S_{prep_1}^i$ is being blocked by e_2 . In the other possible case, if neither of e_1 or e_2 blocks any local chord of C_{prep}^i , then due to Lemma 3.3 one of them must be flippable. We insert that chord in $S_{ear_1}^i$ and the other in $S_{ear_2}^i$.

Steps for Completing The Computation of S_{ear}^i

For all earing vertices whose earing chords do not produce foreign 2-chords of C_{prep}^i when individually flipped, we do the following to complete the computation of $S_{ear}^i = (S_{ear_1}^i, S_{ear_2}^i)$:

1. If neither e_1 or e_2 is blocking any chords of $S_{prep_1}^i$ then due to Lemma 3.3 at least one of $\{e_1, e_2\}$ is flippable, so insert that chord in $S_{ear_1}^i$ and the remaining chord in $S_{ear_2}^i$.
2. If one of e_1 or e_2 is blocking a chords of $S_{prep_1}^i$ then due to Lemma 2.51 the blocking chord is flippable, so insert that chord in $S_{ear_1}^i$ and the remaining chord in $S_{ear_2}^i$.

This completes the computation of $S_{ear}^i = (S_{ear_1}^i, S_{ear_2}^i)$. The next lemma proves the flippability of S_{ear}^i in G_{ear}^i .

Lemma 4.20. *Under the setup described in Section 4.2.1, S_{ear}^i is an ordered set of simultaneously flippable sets in G_{ear}^i .*

Proof. Since I_{ear}^i is a set of independent vertices in G and since every edge in $S_{ear_1}^i$ is incident to a unique vertex of I_{ear}^i , the edges of $S_{ear_1}^i$ must be pairwise non-consecutive in G_{ear}^i . Thus the corresponding set of edges $S_{ear_1}^{i*}$ of the edge set $S_{ear_1}^i$ in the dual tree G_{ear}^{i*} of G_{ear}^i forms a matching. Thus due to Lemma 2.63 $S_{ear_1}^i$ is flippable in G_{ear}^i . Let $G_{ear}' = G_{ear}^i \setminus S_{ear_1}^i$. Since the vertices that can see the earing chords cannot be in the independent set I_{ear}^i , none of the new edges introduced by flipping $S_{ear_1}^i$ can be incident to any earing vertex in I_{ear}^i . Thus the set of edges I_{ear}^i is still independent in G_{ear}' . Thus using the exact same argument as before, $S_{ear_2}^i$ is pairwise non-consecutive in G_{ear}' . Thus due to Lemma 2.63 again, $S_{ear_2}^i$ must be flippable in G_{ear}' . Thus S_{ear}^i is flippable in G_{ear}^i . $\square$

Since we have already computed S_{prep}^i , we have the complete ordered set of simultaneous flip set $S^i = (S_{prep_1}^i \cup S_{ear_1}^i, S_{prep_2}^i \cup S_{ear_2}^i)$.

FIGURE 4.9: If a blocked earing chord produces foreign 2-chord of C_{prep}^i , its consecutive earing chord is flippable. G_{prep}^i and its empty region is drawn in red. G_{ear}^i and its empty region is drawn in blue. Green denotes processed region. Lightly shaded regions are faces. Zigzagged lines represent paths consisting one or more edges. Dashed edges may not be distinct. Shaded region may be further triangulated.

4.3.6 Flippability of S^i

Due to Property 2a of the Preparation Method in Section 4.3.1, S_{prep}^i is flippable in G_{prep}^i . Due to Lemma 4.20, S_{ear}^i is flippable in G_{ear}^i . Since C_{ear}^i and C_{prep}^i are non-overlapping empty cycles, none of the chords of C_{ear}^i can be consecutive to any chords in C_{prep}^i —ensuring non-consecutivity among the edges of S^i . Thus to show that S^i is an ordered set of flippable sets in G^i , due to Lemma 2.50 we only have to prove that the following two requirements are met.

1. Any edge in the sets of S^i that is blocked, its blocking edge is also in the respective set.
2. For each of these sets, none of the edges form bad pairs.

The next Lemma provides a proof that in Step 1 of Section 4.3.3 the earing chord that is flipped in the $S_{ear_2}^i$ cannot be blocking any provisioning chord.

Lemma 4.21. *Under the setup described in Section 4.2.1, assume we execute Step 1 of Section 4.3.3, where $i > 1$. Hence, let e_1 and e_2 be the two local chords of an earing vertex such that e_2 is blocked in G^i and flipping e_2 individually results in a foreign 2-chord of C_{prep}^i . Then e_2 cannot block any provisioning chord in S_{prep}^i .*

Proof. As illustrated in Figure 4.9, let $e_2 = st$ be seen by the edges p and q . Since flipping e_2 individually introduces a foreign 2-chord pq of C_{prep}^i , there must be exactly one vertex v between p and q on an arc of C_{prep}^i . To prove by contradiction, assume e_2 is blocked by a provisioning chord of G_{prep}^i . Then pq is a local 2-chord of G_{prep}^i introducing the ear pqv in G_{prep}^i .

Since $i > 1$, we removed all the ears of $G_{ear}^{(i-1)''}$ to get G_{prep}^i , so pqv cannot be an ear of $G_{ear}^{(i-1)''}$. Then pq cannot be a local 2-chord of $C_{ear}^{(i-1)''}$ and therefore there must be other vertices on the arc $\widehat{qp}$ of the cycle $C_{ear}^{(i-1)''}$, which was removed from $C_{ear}^{(i-1)''}$ while removing all the ears of $G_{ear}^{(i-1)''}$. WLOG let vuq be such an ear making vq a local chord of $C_{ear}^{(i-1)''}$. Then uv and uq edges must be present in the foreign region of

C_{prep}^i in G^i . Since vq is a chord of $G_{ear}^{(i-1)''}$, and because no chord in $G_{ear}^{(i-1)''}$ can be an edge in $G_{prep}^{(i-1)''}$, vq cannot be an edge of $G_{prep}^{(i-1)''} = G_{ear}^i$. Since sq must be an edge in G_{ear}^i , $s \neq v$. Then s must be in the foreign region of C_{prep}^i and hence no local chord of C_{prep}^i when flipped can be incident to s . Thus $e_2 = st$ cannot block any provisioning chords in S_{prep}^i . $\square$

The next two lemmas prove that if an edge in S^i is a blocked chord, then the edge blocking it is also in the corresponding edge set in S^i .

Lemma 4.22. *Under the setup described in Section 4.2.1, if any provisioning chord in S^i is blocked, then its corresponding blocking edge must also be present in the respective edge set of S^i .*

Proof. Let pq be the provisioning chord of the vertex $v \in I_{avoid}^i$ that is blocked by uv . We prove this lemma in two steps. In Case 4.3.6 we prove that if $pq \in S_{prep_1}^i$ then $uv \in S_{ear_1}^i$. In Case 4.3.6 we prove that if $pq \in S_{prep_2}^i$ then $uv \in S_{ear_2}^i$.

Case 1 ($pq \in S_{ear_1}^i \implies uv \in S_{prep_1}^i$): Due to Property 2c of the Preparation Method in Section 4.3.1, no two earing chords of the same earing vertex can both be blocking provisioning chords in $S_{prep_1}^i$. Due to Lemma 4.21, in Step 1 of Section 4.3.3, the earing chord inserted in $S_{ear_1}^i$ cannot block any provisioning chords in $S_{prep_1}^i$. Observe that in every other step listed in Section 4.3.3 and Section 4.3.5, when selecting the order in which the earing chords of an earing vertex are flipped, we have explicitly ensured that any earing chord of the earing vertex that is blocking a provisioning chord in $S_{prep_1}^i$ is always inserted in $S_{ear_1}^i$. Thus if $pq \in S_{prep_1}^i$, uv must be in $S_{ear_1}^i$.

Case 2 ($pq \in S_{prep_2}^i \implies uv \in S_{ear_2}^i$): Since v must have a corresponding foreign 2-chord in $C_{ear}^{i+1} = C_{prep}^{i+1}$ in G^{i+1} , v is separated from u in the foreign region of C_{ear}^{i+1} . Thus uv cannot be an edge in G^{i+1} . Therefore uv must have been flipped in G^i which means $uv \in S_{ear_2}^i$.

Since $S^i := (\{S_{ear_1}^i\} \cup \{S_{prep_1}^i\}, \{S_{ear_2}^i\} \cup \{S_{prep_2}^i\})$, the proof of the lemma is complete by the above two cases. $\square$

Lemma 4.23. *Under the setup described in Section 4.2.1, if any earing chord in S^i is blocked, then its corresponding blocking edge must also be present in the respective edge set of S^i .*

Proof. As listed in Section 4.3.3 and Section 4.3.5, there are four cases in which an earing chord is inserted in S^i . These cases are:

1. **Case 1 (Section 4.3.3 Step 1):** Assume, we have an earing vertex v with two earing chords e_1 and e_2 where e_2 is blocked. Moreover, individually flipping e_2 in G^i produces parallel edges with a foreign 2-chord of C_{prep}^i . In this case e_1 is inserted in $S_{ear_1}^i$. Due to Lemma 2.51, e_1 is flippable in G^i . The proof that e_2 is also flippable is covered in Case 4.
2. **Case 2 (Section 4.3.3 Step 2, Section 4.3.5 Step 1):** Assume, an earing chord e blocks a provisioning chord in $S_{prep_1}^i$, where e is not consecutive to any other earing chords. In this case the earing chord is inserted in $S_{ear_1}^i$. Due to Lemma 2.51, e is flippable in G^i .
3. **Case 3 (Section 4.3.3 Step 3, Section 4.3.5 Step 1):** Assume we have an earing vertex u having two earing chords e_1 and e_2 , neither of which are blocking provisioning chords in $S_{ear_1}^i$. In this case, due to Lemma 3.3, we can assume

that e_1 is flippable in G^i . We inserted e_1 in $S_{ear_1}^i$. e_2 is inserted in $S_{ear_2}^i$. The proof that e_2 is also flippable is covered in Case 4.

4. **Case 4 (Section 4.3.3 Step 4):** An earing chord e of the earing vertex $u \in I_{ear}^i$ is inserted in $S_{ear_2}^i$. The vertex u has only one internal chord and u does not have any corresponding foreign 2-chord in G^i . Due to Lemma 4.15 v cannot have any foreign 2-chord of C_{ear}^i in $G^{i'}$ either. Thus due to Lemma 4.13 e cannot be a blocked edge of S_2^i in $G^{i'}$.

□

Finally, we have to prove that edges of S_{ear}^i cannot form a bad pair with edges of S_{prep}^i in their respective graphs (G^i or $G^{i'}$).

Lemma 4.24. *Under the setup described in Section 4.2.1, no provisioning chord of a vertex in I_{avoid}^i can form a bad pair with any earing chord of a vertex in I_{ear}^i .*

FIGURE 4.10: Provisioning chord of G_{prep}^i cannot form bad pair with earing chord of G_{ear}^i in G^i . C_{ear} is drawn in blue, C_{prep} drawn in red, zigzagged arcs are paths and may overlap, shaded region may be further triangulated. Dotted edge is not part of the graph. Dashed chords of C_{prep} may be edges of the cycle C_{prep} . Blue, red and green shades are parts of G_{ear} , G_{prep} and the processed region in between respectively.

Proof. As illustrated in Figure 4.10, let u be a vertex of I_{prep}^i on G_{prep}^i with a provisioning chord pq seen by the two vertices u and v . For pq to form a bad pair with another edge st , st must also be seen by the same two vertices u and v . Let the two adjacent vertices of u on C_{prep}^i be m and n (note that m and n may not be distinct from p and q respectively). Now let us assume st is an earing chord, forming a bad pair with pq . This implies either s is in I_{ear}^i or t is. Furthermore, since flipping a local chord of an empty cycle produces another local chord of the empty cycle, u and v must be on C_{ear}^i . But since u is an avoiding vertex of C_{prep}^i , mn must be a foreign 2-chord of C_{ear}^{i+1} in G_{ear}^{i+1} . But mn intersects with at least one edge of the path (us, sv) and the path (ut, tv) each. That means on the i -th iteration at least one edge of (us, sv) and one edge of (ut, tv) must be flipped. This can only happen if at least one of $\{u, v\}$ is in I_{ear}^i , or if both $\{s, t\}$ is in I_{ear}^i . But since the vertices of I_{ear}^i are non-consecutive on C_{ear}^i , s and t cannot both be in I_{ear}^i . Then at least one of $\{u, v\}$ must be in I_{ear}^i . But then st cannot be an earing chord, completing the proof by contradiction. □

Thus S^i is an ordered set of flippable sets in G^i and we get the next Theorem.

Theorem 4.25. *Under the setup described in Section 4.2.1, S^i is an ordered set of flippable sets on G^i .*

4.3.7 Initiating the 1st Iteration

A Hamiltonian cycle in a plane graph has two empty regions, one on each side. Thus it defines two distinct empty cycles- each having the same Hamiltonian cycle, but with a distinct empty region of the cycle. Once we have obtained a triangulation G^1 with a Hamiltonian cycle H , we name these two empty cycles C_{ear}^1 and C_{prep}^1 . In the interlaced algorithm, for the i -th iteration, we need I_{avoid}^{i-1} to compute I_{ear}^i . But for the first iteration, we do not have I_{avoid}^0 . The next lemma shows how to compute I_{ear}^1 such that none of the vertices in I_{ear}^1 has a corresponding 2-chord of C_{ear}^1 in G^1 . However, we can only guarantee that $|I_{ear}^1| \geq n/12$ (instead of $n/6$).

Lemma 4.26. *On the first iteration, we can find a set of independent vertices I_{ear}^1 of degree at most 4 in G_{ear}^1 such that $|I_{ear}^1| \geq n/12$. Furthermore, none of the vertices in I_{ear}^1 is an avoiding vertex in G^1 .*

Proof. In our 1st iteration, we shall create ears in G_{ear}^1 and we shall prepare G_{prep}^1 for the next iteration. Due to Theorem 2.66 we can compute an independent set of vertices I of size $n/6$ in G_{ear}^1 such that every vertex has at most degree 4 in G_{ear}^1 . If at least half of these vertices are non-avoiding vertices in C_{ear}^1 , then the proof is complete. But if more than half of these vertices are avoiding vertices, each of them has a corresponding foreign 2-chord on C_{ear}^1 . Since the two empty cycles are the same cycles but with different empty regions, this means G_{prep}^1 currently has at least $n/12$ ears. Since the graph by Definition 2.1 is simple, none of these vertices can have a foreign 2-chord in G_{prep}^1 . Thus none of these vertices have a foreign 2-chord in G_{prep}^1 . So we relabel G_{prep}^1 as G_{ear}^1 and G_{ear}^1 as G_{prep}^1 , and use these vertices as the set of I_{ear}^1 for our first iteration. $\square$

Once we have obtained I_{ear}^1 we can continue the 1st iteration as explained for the i -th iteration.

4.4 The Preparation Method

The Preparation Method takes a maximal outerplane graph $G = (V, E)$ with its maximal cycle C and a set of pairwise non-consecutive vertices I_{avoid} on C , where each vertex in I_{avoid} has at least one chord in G . Vertices of I_{avoid} are called *avoiding vertices* as these are the vertices that we would like to avoid from creating ears with for the next iteration. Vertices in V that are not in I_{avoid} are called *safe vertices*. The Preparation Method returns an ordered set of flippable sets $S_{\text{prep}} = (S_{\text{prep}_1}, S_{\text{prep}_2})$ on G . Let $G'' = G \langle S_{\text{prep}} \rangle$. In G'' , for all vertices $v \in I_{\text{avoid}}$, either v has at least 3 internal chords, or v dominates another vertex u where $u \notin I_{\text{avoid}}$.

4.4.1 Properties of Preparation Method

1. Input parameters of Preparation Method:
  - (a) A maximal outerplane graph $G = (V, E)$ with the maximal empty cycle denoted by C .
  - (b) A set of pairwise non-consecutive vertices I_{avoid} on C .
  - (c) Each vertex of I_{avoid} has at least one chord in G .
2. Output parameters of Preparation Method:
  - (a) Returns an ordered set of two simultaneous flip sets $S_{\text{prep}} := (S_{\text{prep}_1}, S_{\text{prep}_2})$ on G consisting of only *provisioning chords*. (See Lemma 4.45 Property 1)
  - (b) In $G''_{\text{prep}} := G_{\text{prep}} \langle S_{\text{prep}} \rangle$, for every vertex $v \in I_{\text{avoid}}$ at least one of the following is true (See Lemma 4.45 Property 2):
    - i. v has at least 3 local chords in G''_{ear} .
    - ii. v dominates a vertex u where $u \notin I_{\text{avoid}}$.
  - (c) No more than one *singly provisioning chord* of any vertex $v \in I_{\text{avoid}}$ is in S_{prep_1} . (See Lemma 4.45 Property 3)

The Preparation Method achieves this in two steps:

1. Divide G into a set of smaller subgraphs and assign each of the vertex in I_{avoid} to one *partitioned subgraph*.
2. Insert provisioning chords in each partitioned subgraph to S_{prep} so that the assigned vertices of each subgraph either dominate a vertex not in I_{avoid} , or they have at least 3 internal chords in G'' .

4.4.2 Preliminaries

We start with some definitions that are used in this section.

Definition 4.27 (Partitioned Subgraph). Given a maximal outerplane graph $G = (V, E)$, its maximal cycle C and a set of pairwise non-consecutive vertices I_{avoid} on C , where each of the vertices in I_{avoid} has at least 1 local chord in G . The graph G is split into a set of k subgraphs $G_p := \{g_1, g_2, \dots, g_k\}$ where for all $1 \leq i \leq k$, $g_i = (V_i, E_i)$ and each subgraph g_i is called a *partitioned subgraph* having the following properties:

1. The partitioned subgraphs in G_p are pairwise non-overlapping maximal outerplane subgraphs of G .

2. Every vertex and edge of G is present in at least one partitioned subgraph.
3. Every vertex $v \in I_{\text{avoid}}$, is colored red in exactly one partitioned subgraph in G_p . In all the other partitioned subgraphs where v is present, v is colored black. Vertices that are not in I_{avoid} are coloured green.
4. No two red vertices are neighbours in a partitioned subgraph.
5. No green vertex appears in more than one partitioned subgraph of G_p .
6. Every partitioned subgraph contains at least one green vertex.
7. No black vertex is consecutive to two red vertices on the maximal empty cycle of any partitioned subgraph.

We now extend the definition of provisioning chord to the context of partitioned subgraphs.

Definition 4.28 (Provisioning chord). Let $G = (V, E)$ be a maximal outerplane graph with the maximal cycle C , and a set of pairwise non-consecutive vertices $I_{\text{avoid}} \subseteq V$ on C . Let G_p be the set of partitioned subgraphs of G . Then for a partitioned subgraph $g_i = (V_i, E_i) \in G_p$, let pq be a local chord of g_i and u be a red vertex in g_i such that pq is seen by u in the local region of g_i . Then pq is a provisioning chord of u . Furthermore u is called the corresponding avoiding vertex of the provisioning chord pq . If a provisioning chord is seen by two distinct red vertices in its respective partitioned subgraph, then it is called a doubly provisioning chord. Otherwise it is called a singly provisioning chord.

4.4.3 Requiring Avoiding Vertices to Have at Least One Internal Chord

One may ask why the Preparation Method Property 1c in Section 4.4.1 is necessary. Unless every vertex in I_{avoid} has at least 1 internal chord in G , there may be some vertices whose degree must be increased by 3 in the Preparation Method. At first glance, it may seem possible by increasing the degree by 1 in the first simultaneous flip and then adding another 2 to the degree in the second simultaneous flip. In Figure 4.11(a) G has 3 avoiding vertices u, v and w , and 3 provisioning chords am, fg and gh . On the first simultaneous flip by flipping all 3 of the provisioning chords, we get G' illustrated in Figure 4.11(b). Notice that in G' each of the vertices u, v and w require 2 more chords to meet their requirement of having at least 3 internal chords. The provisioning chords in G' are $\{cf, cg, ag, gm, gk, hk\}$. However, since cg and ag are consecutive, and gk and gm are consecutive, due to Lemma 2.50 Condition 1 they cannot be flipped together in the second simultaneous flips. This makes it impossible to ensure that u, v and w each would have at least 3 internal chords by only flipping provisioning chords using 2 simultaneous flips.

Furthermore, regardless of which of the provisioning chords $\{cf, cg, ag, gm, gk, hk\}$ are flipped in G' on the second simultaneous flip, neither of u, v or w dominates any of the green vertices in the resulting graph. Thus Preparation Method Property 1c in Section 4.4.1 is necessary.

4.4.4 Maximal Outerplane Graph Partitioning Algorithm

The Algorithm **GENERATESUBGRAPHS**(G, I_{avoid}) takes a maximal outerplane graph G and a set of non-consecutive vertices I_{avoid} on its maximal cycle. Then every vertex of I_{avoid} is coloured red and the remaining vertices are coloured green in G . Next, G is inserted into the stack of unprocessed graphs S .

FIGURE 4.11: Avoiding vertices must have at least 1 internal chord. Edges of the maximal cycle are drawn in blue. Vertices in I_{avoid} are drawn in red. Vertices not in I_{avoid} are drawn in green. Zigzagged arcs are paths. Shaded region may be further triangulated. Red edges are provisioning chords that when flipped the degree of a red vertex increases.

A graph g is popped from the stack S and processed until S is empty. To process g , the entire graph is traversed by running *BFS* on its dual tree g^* . The *BFS* starts from the last visited node. If no node is yet visited we start from a leaf node of G^* which corresponds to an ear of g .

Every time a new node n_{new} of g^* is traversed, let the predecessor node be n_{pred} and let pq be the chord of g that is on the two faces whose corresponding nodes are n_{new} and n_{pred} in g^* . If both p and q are red vertices in g , then g is split along the chord pq into two smaller maximal outerplane graph g_1 and g_2 , where both g_1 and g_2 contains the edge pq on their maximal cycle. WLOG let g_1 be the subgraph that contains the explored part of g . If one of p and q contains an internal chord in g_1 , it is coloured red in g_1 and black in g_2 . In the other graph g_2 p and q are coloured in reversed order. Notice that only when g_1 is a triangle, is it possible for neither p nor q to have any internal chords in g_1 . In this case one of $\{p, q\}$ is arbitrarily coloured red in g_1 , and the other is coloured red in g_2 . Next we set $g := g_1$, insert g_2 in the set of unprocessed graphs S and continue the traversal of g via *BFS*. Once g has been fully traversed via *BFS*, insert g in the set of all partitioned subgraphs G_p , and continue on by processing the next graph in S until S is empty.

An example of how Procedure **GENERATESUBGRAPHS**(G, I_{avoid}) works is shown in Figure 4.12. In the maximal outerplane graph G , we start traversing from face $(h, i, j) \rightarrow (g, h, j) \rightarrow (g, j, k) \rightarrow (g, k, m)$. We now cross the edge gk where both g and k are red vertices. We split G along the chord gk into two smaller maximal outerplane graphs g_1 and G' where $g_1 := (g, h, i, j, k)$ and $G' := (k, l, m, n, o, p, a, b, c, d, e, f, g)$. Notice that g_1 contains the explored region of G where the faces of G' are still unexplored. Since g has an internal chord in the subgraph g_1 , g is coloured red in g_1 and k is coloured black in G' . Since g_1 is now entirely traversed, it is now a partitioned subgraph and is inserted in G_p .

The process continues with G' where we traverse $(g, k, m) \rightarrow (k, l, m)$ crossing the edge km where both k and m are red vertices. G' is split along km , generating the two graphs G'' and g_2 , where G'' contains the explored region of G' . k has no internal chord in G' so m and k are coloured red and black respectively in the G'' . They are coloured in opposite colours in g_2 .

We continue the traversal in G'' starting from the last visited face $(g, k, m) \rightarrow (a, g, m) \rightarrow (a, m, n)$ crossing the next edge (am) incident to two red vertices. G'' is partitioned into g_3 and g_4 where g_3 contains the explored region of G'' . Since both

a and m has internal chords in G'' , since both a and m has an internal chord in g_3 , a is arbitrarily coloured red in g_3 and m is coloured black. The vertices a and m are coloured in opposite colours in g_4 . BFS continues in g_3 traversing $(a, g, m) \rightarrow (a, f, g) \rightarrow (a, b, f) \rightarrow (b, c, f) \rightarrow (c, d, e)$ without crossing any edge connecting two red vertices. Since g_3 is fully explored, it is now a partitioned subgraph and is inserted in G_p . Similarly using BFS g_2 and g_4 are also traversed but since these graphs do not contain any chord incident to two red vertices, these graphs are not split any further. g_2 and g_4 also get inserted to G_p . At the end of the algorithm the computed set of all partitioned subgraphs $G_p = \{g_1, g_2, g_3, g_4\}$.

4.4.5 Colouring of the Vertices and the Overall Goal

In each of the partitioned subgraph $g_i \in G_p$, the vertices are divided into three colours—green, red and black. All vertices that are not in I_{avoid} are coloured green. Each vertex $v \in I_{avoid}$ is coloured red in exactly one partitioned subgraph in G_p . In all other partitioned subgraphs that contain v , v is coloured black. In each partitioned subgraph g_i we compute an ordered set of two simultaneous flip sets $s_i := (s_{i_1}, s_{i_2})$ consisting of only provisioning chords of the red vertices in g_i . Once we have computed s_i for all partitioned subgraphs g_i where $1 \leq i \leq k$, we compute $S_{prep} := \bigcup_{i=1}^k s_i$.

The goal of these two simultaneous flips is to obtain $G'' := G\langle S_{prep} \rangle$ where every red vertex of g_i meets Property 2b in Section 4.4.1 in G'' . This is done by ensuring that for any red vertex in g_i , it either dominates a green vertex in G'' , or it has at least 3 chords in G'' .

One may ask why we split the graph into a set of partitioned subgraphs and assign a subset of vertices of I_{avoid} as red vertices to each partitioned subgraph. This is because as a result of no two red vertices being adjacent in a partitioned subgraph, flipping a provisioning chord of a red vertex does not reduce the degree of any other red vertex in the partitioned subgraph. Thus by this method of only flipping provisioning chords, while meeting the degree requirement of one red vertex, we do not risk reducing the degree of some other red vertex in the graph.

4.4.6 Procedure GENERATESUBGRAPHS(G, I_{avoid}) Generates Partitioned Subgraphs

The next few Lemmas are used to prove that the partitioned subgraphs in G_p generated by the Procedure GENERATESUBGRAPHS(G, I_{avoid}) indeed have all the properties of Partitioned Subgraphs. Using these lemmas Theorem 4.35 proves that G_p consists of partitioned subgraphs of G .

Lemma 4.29. *The partitioned subgraphs in G_p generated by the Procedure GENERATESUBGRAPHS(G, I_{avoid}) meet the following properties:*

1. *The partitioned subgraphs in G_p are pairwise non-overlapping maximal outerplane subgraphs of G .*
2. *Every vertex and edge of G is in at least one partitioned subgraph in G_p .*

Proof. The original graph G is a maximal outerplane graph. We run BFS on the dual tree G^* of G , and every time we cross an edge whose corresponding edge in the primal graph connects two red vertices, we split the tree into two smaller subtrees by removing that edge. Thus, each of our resulting partitioned subgraphs is a primal maximal outerplane graph to some dual tree that is a subtree of G^* . Furthermore,

Algorithm 1 Computing Set of Partitioned Subgraphs of G

```

procedure GENERATESUBGRAPHS( $G, I_{avoid}$ )
2:   Initialize stack  $S$ 
    $S \leftarrow$  the dual tree  $G^*$  of  $G$ 
4:   Mark a leaf node of  $G^*$  as visited
   Colour vertices in  $I_{avoid}$  red and the remaining vertices green in  $G$ .
6:   Initialize empty set of partitioned subgraphs  $G_p$ 
   while  $g^* := S.pop()$  do
8:     Initialize queue  $Q$  ▷  $Q$  is used to run BFS on  $g^*$ 
      $Q \leftarrow$  the visited node of  $g^*$ 
10:    while  $n := Q.pop()$  do
      Mark  $n$  as visited in  $g^*$ 
12:      for all  $t$  that is an unvisited neighbour of  $n$  in  $g^*$  do
        Let  $xpq$  be the corresponding face of  $n$  in  $G$ 
14:        Let  $ypq$  be the corresponding face of  $t$  in  $G$ 
▷  $pq$  is corresponding edge of  $nt$  in primal graph
16:        if  $p$  and  $q$  are coloured red in  $G$  then ▷ Partition graph
          Partition  $g^*$  along  $nt$  obtaining  $g^*$  and  $g_{new}^*$ 
          such that  $g^*$  contains the visited region
           $S \leftarrow g_{new}^*$ 
20:           $g^* := g^* \setminus g_{new}^*$ 
▷ Assign which vertex gets processed in which subgraph
22:          if  $xp$  is not a chord of primal graph  $g$  of  $g^*$  then
            Colour  $p$  black,  $q$  red in  $g$ 
24:            Colour  $p, q$  with opposite colours in  $g_{new}$ 
          else
26:            Colour  $p$  red,  $q$  black in  $g$ 
            Colour  $p, q$  with opposite colours in  $g_{new}$ 
28:          end if
          Mark  $t$  as visited in  $g_{new}$ 
30:        else
           $Q \leftarrow t$ 
32:        end if
      end for
34:    for all vertex  $v$  in primal graph  $g$  of  $g^*$  do
      if  $v \notin I_{avoid}$  then
36:        Colour  $v$  green in  $g$  ▷ Colour all safe vertices green
      else if  $v$  is not coloured then
38:        Colour  $v$  red ▷ Colour remaining  $I_{avoid}$  vertices red
      end if
40:    end for
     $G_p \leftarrow g$ 
42:  end while
end while
44: end procedure

```

FIGURE 4.12: Example of partitioning a maximal outer-plane graph into partitioned subgraphs

every partitioned subgraph is a primal maximal outerplane graph of a subtree of the dual tree G^* of G . None of these subtrees have any edge of G^* in common. Thus G_p must be consisting of pairwise non-overlapping maximal outerplane subgraphs of G , proving **Property 1**.

Finally, the edge of the dual tree that gets cut, its corresponding primal edge is kept in both the primal maximal outerplane graphs of the dual subtrees. Thus at no step of the Procedure GENERATESUBGRAPHS(G, I_{avoid}) is any vertex or edge removed from both the primal maximal outerplane graphs of the dual subtree g^* and g_{new}^* . Thus every vertex and edge of G must be present in at least one partitioned subgraph of G_p proving **Property 2**. $\square$

Lemma 4.30. *In Procedure GENERATESUBGRAPHS(G, I_{avoid}), every vertex in I_{avoid} of G is coloured red in exactly one partitioned subgraph.*

Proof. Initially every avoiding vertex is coloured red. Then during BFS, every time a red-to-red chord pq is encountered, Procedure GENERATESUBGRAPHS(G, I_{avoid}) separates the graph into two smaller graphs. Thus only vertices whose incident edge is used to split the graph into two are the vertices that can be present in multiple partitioned subgraphs. In one of the two subgraphs, the vertex p is coloured red and q is coloured black. In the other, p is coloured black and q is coloured red. Thus every avoiding vertex ends up being a red vertex in exactly one partitioned subgraph of G . $\square$

Lemma 4.31. *No two red vertices are adjacent in a partitioned subgraph generated by Procedure GENERATESUBGRAPHS(G, I_{avoid}).*

Proof. In the original graph G , no two vertices of I_{avoid} are consecutive on its maximal cycle C . Thus none of the edges of the maximal cycle connects two red vertices.

The BFS traverses the entire dual tree crossing every chord of G , and every time it encounters a chord uv connecting two red vertices u and v , it generates two subgraphs—in one of which u is coloured red and v black while, in the other u is coloured black and v red. This ensures that the edge uv is no longer connecting two red vertices. As a result, no two red vertices can be adjacent in a partitioned subgraph. $\square$

Lemma 4.32. *Every green vertex is present in exactly one partitioned subgraph. All neighbours of a green vertex in G must also be present in the same partitioned subgraph.*

Proof. Procedure GENERATESUBGRAPHS(G, I_{avoid}) splits the graph along chords connecting two red vertices. This excludes any green vertex to be incident to such chords. Thus for all green vertices, none of its incident chords can be a common edge between two partitioned subgraphs. Hence green vertices can only appear in exactly one partitioned subgraph and all its neighbours in G must also be present in this partitioned subgraph. $\square$

Lemma 4.33. *Every partitioned subgraph generated by Procedure GENERATESUBGRAPHS(G, I_{avoid}) must have at least one green vertex on it.*

Proof. To prove by contradiction, given the maximal outerplane graph G and the set of avoiding vertices I_{avoid} , let us assume that the Procedure GENERATESUBGRAPHS(G, I_{avoid}) produced a partitioned subgraph g of G that contains no green vertex. In other words, g only contains avoiding vertices. Let C be the maximal empty cycle of G . As illustrated in Figure 4.13, let the ordering of the vertices of g along its maximal empty cycle be $(v_1, v_2, v_3, \dots, v_k)$. Then g 's maximal empty cycle is $(v_1v_2, v_2v_3, \dots, v_kv_1)$.

FIGURE 4.13: Every partitioned subgraph must contain at least one green vertex. Red and black vertices are in I_{avoid} . green vertices are not in I_{avoid} . Green region must contain at least 1 green vertex. Zigzagged arcs represent a set of edge(s). Black arrow in the graph denotes the path taken by Procedure GENERATESUBGRAPHS(G, I_{avoid})

Due to Preparation Method Property 1b in Section 4.4.1, for $1 \leq i \leq k$, each of these vertices v_i must be consecutive to 2 green vertices on C . Thus none of the $v_i v_{i+1}$ edges can be edges of the cycle C . As a result, every $\widehat{v_i v_{i+1}}$ must contain at least one green vertex between v_i and v_{i+1} in C .

Because the algorithm starts from a leaf node of the dual tree G^* corresponding to an ear in G , the Procedure GENERATESUBGRAPHS(G, I_{avoid}) must have started from outside of g and at some point entered into the subgraph g in G . WLOG let the 1st chord connecting two red vertices of g that is traversed be $v_1 v_2$. When $v_1 v_2$ was traversed by the procedure, it split the graph into two subgraphs, in one of which v_1 was marked red and v_2 was marked black, while in the other they are marked in opposite colours. WLOG let v_1 be marked red and v_2 be marked black in the subgraph that contains g . But now $v_2 v_3$ is no longer a chord connecting two red vertices. Thus g cannot be a partitioned subgraph produced by Procedure GENERATESUBGRAPHS(G, I_{avoid}). So every partitioned subgraph produced must have at least one green vertex in it. $\square$

Lemma 4.34. *In a partitioned subgraph there cannot be any black vertex with its two consecutive vertices on the maximal empty cycle both being red.*

FIGURE 4.14: A black vertex cannot be consecutive to two red vertices on the maximal cycle of a partitioned subgraph. Shaded region may be further triangulated. Zigzagged arcs represent a set of edge(s).

Proof. Let u be a black vertex on a partitioned subgraph g_i of the maximal outerplane graph G . Assume that v and w are two red vertices, both of which are consecutive to u on the maximal empty cycle of g_i (refer to Figure 4.14). Since u is a black vertex in g_i , along with v and w , u must have been an avoiding vertex in I_{avoid} . But then neither uv nor uw can be an edge of G 's maximal empty cycle C as the avoiding vertices are non-adjacent on C . Then at some point, G must have been split along both the edges uv and uw . Since the graph is split along only red-to-red chords and when such an edge is used for splitting, one of its incident vertices gets coloured black and the other coloured red in each of the partitioned subgraphs. So WLOG let u be coloured black when a partition was made along the red-to-red chord uw on g_i . Then vu is no longer a red-to-red chord in g_i and so we shall not be partitioning along vu . Thus there cannot be a red, black and red vertex consecutively placed on the maximal cycle of a partitioned subgraph. $\square$

Theorem 4.35. *The subgraphs in G_p generated by the Procedure GENERATESUBGRAPHS(G, I_{avoid}) are partitioned subgraphs of G as per Definition 4.27.*

Proof. The theorem holds because of the following:

1. Lemma 4.29 Property 1 Satisfies Property 1 of Definition 4.27.
2. Lemma 4.29 Property 2 Satisfies Property 2 of Definition 4.27.
3. Lemma 4.30 Satisfies Property 3 of Definition 4.27.
4. Lemma 4.31 Satisfies Property 4 of Definition 4.27.
5. Lemma 4.32 Satisfies Property 5 of Definition 4.27.
6. Lemma 4.33 Satisfies Property 6 of Definition 4.27.
7. Lemma 4.34 Satisfies Property 7 of Definition 4.27. $\square$

4.4.7 Flipping All Provisioning Chords of a Partitioned Subgraph

We use the next lemma to show that in the cases the Preparation Method fails to increase an avoiding vertex's degree, it ensures that the vertex dominates a non-avoiding vertex.

Lemma 4.36. *In a partitioned subgraph g_i , if a red vertex does not have any provisioning chord, then it dominates a green vertex.*

Proof. Let v be a red vertex of g_i . Since v does not have any provisioning chord in g_i , v must be connected to every vertex of g_i . Due to Lemma 4.33, g_i contains at least one green vertex u . Then v must be dominating u . $\square$

Lemma 4.37. *Given a partitioned subgraph g_i ,*

1. *If all three edges of a face are provisioning chords, then by flipping these chords with two simultaneous flips, their corresponding red vertices' degree can be increased by 3, 1 and 1 respectively.*
2. *If exactly two edges of a face are provisioning chords, then by flipping these chords with two simultaneous flips, their corresponding red vertices' degree can be increased by 2 and 1 respectively.*

3. If exactly one edge of a face is a provisioning chord, then by flipping this chord with one simultaneous flip its corresponding red vertex's degree can be increased by 1.

Furthermore, in Case 1 and Case 2, these flips do not interfere with the flipping of any other provisioning chords present in g_i .

FIGURE 4.15: Provisioning chords sharing a face

Proof. We prove the three cases separately.

Case 1: Assume uv, vw and wu are three provisioning chords sharing the face uvw of a partitioned subgraph g_i (refer to Figure 4.15a). Because provisioning chords cannot be connected to a red vertex, none of $\{u, v, w\}$ can be red vertices. Let the other vertices that can see uv, vw and wu be x, y and z respectively. Then x, y and z must be the corresponding red vertices of uv, vw and uw . Since x, y and z are red vertices and, if any of $\{xu, xv, yv, yw, zu, zw\}$ exist as chord in g_i , it cannot be provisioning chord as otherwise one of $\{x, y, z\}$ would be consecutive to another red vertex. Thus with two simultaneous flips ($\{uv\}, \{vw, uw\}$) all three of these provisioning chords can be flipped. By doing so the degree of x would increase by 3 and degree of y and z would increase by 1 each. Furthermore, the order in which the remaining provisioning chords of g_i are flipped does not affect the flipping of these sets since these chords are non-consecutive to any other provisioning chords of g_i .

Case 2: Assume uv and vw are the consecutive provisioning chords sharing the face uvw of a partitioned subgraph g_i (refer to Figure 4.15b). uw is either an edge of g_i 's maximal empty cycle, or is a chord. If uw is a chord in g_i , let z be the other vertex that can see uw in g_i . If the vertex z exists, it cannot be a red vertex. Using the same reasoning as in Case 1, if any of $\{xu, xv, yv, yw\}$ exists as a chord in g_i , it cannot be provisioning chord as otherwise one of $\{x, y\}$ would be consecutive to another red vertex in g_i . Thus with two simultaneous flips ($\{uv\}, \{vw\}$) both of these provisioning chords can be flipped. By doing so the degree of both x and y would increase by 2 and 1 respectively.

Case 3: Assume uv is the only provisioning chord on the face uvw of a partitioned subgraph g_i (refer to Figure 4.15c). vw and uw may be edges of the maximal cycle of g_i . If vw is a chord in g_i , then let y be the other vertex that can see vw in g_i . Similarly, if uw is a chord in g_i , let z be the other vertex that can see uw in g_i . Since vw and uw are not provisioning chords, if the vertex y or z exists, it cannot be a red vertex. Using the same reasoning as in Case 1, if any of $\{xu, xv\}$ exist as chord in g_i , it cannot be provisioning chord as otherwise x would be consecutive to another red vertex. Thus uv cannot be consecutive to any other provisioning chord in g_i . Thus

uv can be flipped in either the first simultaneous flip s_{i_1} or the second simultaneous flip s_{i_2} . By doing so the degree of x would increase by 1. $\square$

The notion of branches come useful in the next Lemma.

Definition 4.38 (Branch). *Given a tree T with a node u , a branch of u is a maximal subtree in T containing u as a leaf node.*

Given a partitioned subgraph g_i , the next lemma provides us a way to build $s_i := (s_{i_1}, s_{i_2})$ ensuring that every red vertex's degree increases by the number of provisioning chords it has in its respective partitioned subgraph.

Lemma 4.39. *Let g_i be a partitioned subgraph, with a provisioning chord pq being preselected to be in one of the flip sets in $s_i := (s_{i_1}, s_{i_2})$, then:*

1. *All the remaining provisioning chords in g_i can be inserted in one of s_{i_1} and s_{i_2} .*
2. *s_i is an ordered set of flippable sets on g_i .*
3. *The degree of each red vertex in $g_i'' := g_i \langle s_i \rangle$ increases at least by the number of provisioning chords it has in s_i .*

Proof. To complete this proof, first we see how given a face of g_i with at most one preselected provisioning chord inserted in s_i , we can insert the remaining chords in s_i while keeping s_i flippable. We divide the problem into the following 4 cases.

Let e be an edge of a face f of g_i that has already been inserted to s_{i_1} or s_{i_2} :

Case 1: Assume f has 3 provisioning chords.

1. If $e \in s_{i_1}$, then the other two chords on f are inserted in s_{i_2} .
2. If $e \in s_{i_2}$, then insert one of the remaining two provisioning chords on f in s_{i_1} and the other in s_{i_2} .

Due to Lemma 4.37 none of these chords can be consecutive to any other provisioning chords. As a result, they are flippable in this order regardless in the order other provisioning chords are inserted in s_i .

Case 2: Assume f has 2 provisioning chords.

1. If $e \in s_{i_1}$, then the other provisioning chord on f is inserted in s_{i_2} .
2. If $e \in s_{i_2}$, then the other provisioning chord on f is inserted in s_{i_1} .

Due to Lemma 4.37 none of these chords can be consecutive to any other provisioning chords. As a result, they are flippable in this order regardless in the order other provisioning chords are inserted in s_i .

Case 3: Assume f has only one provisioning chord. Since none of the other 2 chords of f are provisioning chords, we do not need to insert any other edges of f in s_i .

Case 4: No provisioning chord of f is preselected. If the face does not have any processed chords, then by following the proof of Lemma 4.37 its provisioning chords can be inserted in s_{i_1} and/or s_{i_2} maintaining the flippability of s_i .

Notice that in each case the degree of the corresponding red vertices of each provisioning chords on f increases by at least 1 in $g_i'' := g_i \langle s_i \rangle$.

Now that we know how to insert the remaining provisioning chords of a face given at most one preselected provisioning chord on the face, next we show how to

insert all the remaining provisioning chords of g_i , given a preselected provisioning chord pq already inserted in s_{i_1} or s_{i_2} .

We take the following steps:

1. Let g_i^* be the dual tree of the partitioned subgraph g_i , with pq being the preselected provisioning chord in g_i .
2. Let f_{start} be a face of g_i that pq is on, and let f_{start}^* be its corresponding node in g_i^* .
3. Insert pq in s_{i_1} or s_{i_2} accordingly and mark it as processed.
4. Run BFS on the tree g_i^* starting from the node f_{start}^*
  - (a) For each traversed node f^* , let its corresponding face in the primal graph be f .
  - (b) By following the 4 cases explained at the beginning of the proof of this lemma, to insert provisioning chords of a given face, insert the unprocessed provisioning chords on f in the flip sets in s_i , keeping s_i flippable.

The BFS method traverses the entire tree g_i^* visiting all the faces in the interior of g_i . Since when a face is processed, all provisioning chords on the face are inserted in s_i , all remaining provisioning chords in g_i would be eventually inserted in s_i . This completes the proof of **Property 1**.

The BFS processes all nodes up to distance δ from f_{start}^* before processing any node at distance greater than δ . Thus while processing the face f^* , at most one of the branches of g_i^* incident to the node f^* can be partially or fully processed. Let this branch be B . Then s_i must be a subset of the edges of g_i whose corresponding edges in g_i^* are on B . Let f be the face in g_i whose corresponding node in g_i^* is f^* . Furthermore, let e_1 be the edge on f whose corresponding edge in g_i^* is on B . Let e_2 and e_3 be the remaining two edges of f . Then notice that the only edge of g_i , whose corresponding edge in g_i^* is on B , that e_2 or e_3 can be consecutive to, is e_1 . Thus when processing f , e_1 can be the only provisioning chord in f that has been inserted in s_i . Furthermore, by following the method explained at the beginning of the proof of this lemma, we build s_i , while keeping the edges in s_{i_1} and s_{i_2} pairwise non-consecutive. Then the corresponding edges in s_{i_1} and s_{i_2} would form a matching in g_i^* and $(g_i \setminus \langle s_{i_1} \rangle)^*$ respectively. Thus due of Lemma 2.63 s_i must still be flippable in g_i after inserting the provisioning chords on f in s_i . This completes the proof of **Property 2**.

Finally due to Lemma 4.31, no two red vertices can be adjacent in g_i . As a result, no red vertex can be incident to a provisioning chord in g_i . Thus the flipping of a provisioning chord cannot reduce the degree of any red vertices. Since flipping a provisioning chord increases the degree of all its corresponding red vertices, the degree of each red vertex in g_i increases at least by the number of provisioning chords it has in g_i . This completes the proof of **Property 3**. $\square$

4.4.8 Categorization of the Red Vertices

By virtue of Lemma 4.31 no two red vertices can be consecutive on a partitioned subgraph. This means in a partitioned subgraph's maximal empty cycle, a red vertex can be consecutive to only green and black vertices. Taking this into consideration we present the following definition.

FIGURE 4.16: A GRG vertex must have at least 2 provisioning chords or it dominates a green vertex. v is a GRG vertex. u, w are green vertices. x, y are not red vertices. Shaded region may be further triangulated. Zigzagged arcs represent a set of edge(s).

Definition 4.40 (BRB, BRG and GRG Vertices). In a partitioned subgraph g_i , a red vertex can be consecutive to two black vertices (BRB standing for Black-Red-Black), or it can be consecutive to a black vertex and a green vertex (BRG standing for Black-Red-Green), or it can be consecutive to two green vertices (GRG standing for Green-Red-Green).

The next Lemma allows us to assume that any GRG vertex must have at least 2 provisioning chords, or it already satisfy Property 2(b)ii of the Preparation Method referred in Section 4.4.1.

Lemma 4.41. Given a partitioned subgraph g_i with a GRG vertex v , v must have at least 2 provisioning chords or v dominates a green vertex in G .

Proof. Let u, v, w be three vertices placed consecutively on the cycle C of g_i , where u and w are green vertices, and v is a red vertex between them. Since both u and w are green vertices, due to Lemma 4.32, u and w do not appear in any other partitioned subgraphs apart from in g_i . If v is to be present in multiple partitioned subgraphs, one of v 's chords must have been used to split G . But to do so, u and w cannot both be on c_i . Thus, v also occurs only in g_i . Since every red vertex has at least 1 local chord in G , v must have a local chord in g_i .

As illustrated in Figure 4.16, let us assume vx is the leftmost chord in g_i seen by u , and vy be the rightmost chord of v seen by w . Since v has at least 1 chord in g_i , x and y are not necessarily distinct. Since x is adjacent to v in g_i , x cannot be a red vertex in g_i . Because the local region of g_i is triangulated, ux must be an edge of g_i . If ux is a chord in g_i , then it is seen by v and thus ux must be a provisioning chord of v . If ux is not a provisioning chord, then it must be an edge of the cycle C . In this case, the only two neighbours of u would be v and x and thus v would dominate u . Using the same argument, if wy is not a provisioning chord, then v would dominate w . Since u and w are distinct vertices, either v has at least 2 provisioning chords or v dominates a green vertex. $\square$

The next Lemma is used to introduce another category for the Red vertices.

Lemma 4.42. In a partitioned subgraph of order > 3 , there can be at most one red vertex that has no local chords.

Proof. As stated in Property 1c of the Preparation Method in Section 4.4.1, every vertex in I_{avoid} has at least one internal chord in the maximal outerplane graph G . For any partitioned subgraph g_i , all its red vertices are vertices in I_{avoid} . Thus if a red

vertex u of g_i does not have any internal chords in g_i then it must be due to splitting the graph along an edge uv where v is also a vertex in I_{avoid} .

Consider the Procedure **GENERATESUBGRAPHS**(G, I_{avoid}). Here g is a subgraph of G , such that g is in the process of being turned into a partitioned subgraph. The procedure has crossed an edge pq connecting two red vertices p and q . Furthermore, x is the vertex that sees pq where x is strictly in the explored part of the subgraph g . Since we are only considering partitioned subgraphs of order > 3 , for g to produce such a partitioned subgraph, g must have a higher order than 3 after split along pq . Then (x, p, q) cannot be the maximal cycle of g after the split. As a result, one of xp and xq must be a chord of g . In this case, notice that in Line 22–28 of the algorithm, it is always ensured that any time the graph is further split into smaller graphs, the red vertex that is kept in the (partially) explored graph g always consists of at least one internal chord in it. Thus the only time a partitioned subgraph g_{new} can have a red vertex that does not contain any internal chord, is when g_{new} was split off along pq from g . Thus there can be at most one red vertex with no internal chords in a partitioned subgraph. $\square$

Taking Lemma 4.42 in consideration we present the following definition.

Definition 4.43 (Priority Vertex). *Given a partitioned subgraph g_i of a maximal outer-plane graph G with the maximal empty cycle C , and a set of non-consecutive vertices I_{avoid} on C , if the order of g_i is larger than 3, then it can have at most one red vertex with no internal chords. Such a red vertex is called a priority vertex of g_i . Any vertex of g_i that is not a priority vertex is called a non-priority vertex.*

Due to Property 1c of the Preparation Method referred in Section 4.4.1, every vertex in I_{avoid} has at least one local chord. Since a priority vertex does not have any local chords in its respective partitioned subgraph, we can conclude that one or more local chords of the priority vertex were used to split the graph into smaller subgraphs and thus any priority vertex must have at least one black vertex consecutive to its partitioned subgraph's maximal empty cycle. Thus a priority vertex can only be of type *BRB* or *BRG*.

TABLE 4.1: Types of red vertices in partitioned subgraphs

Case	Priority	Pattern	A	B	C
1	No	<i>BRB</i>	3	0	1
2	No	<i>BRG</i>	2	1	1
3	No	<i>GRG</i>	1	2	2
4	Yes	<i>BRB</i>	2	1	1
5	Yes	<i>BRG</i>	1	2	1
6	Yes	<i>GRG</i>	N/A	N/A	N/A

A: Minimum number of chords of the red vertex in G that are present as edges in g_i .

B: Maximum number of chords that are required for the red vertex.

C: Minimum number of provisioning chords of the red vertex already present in g_i .

Taking all this into consideration, in Table 4.1 we categorized a complete list of all the possible types of red vertices that can arise in our partitioned subgraphs.⁴ Every avoiding vertex in G must contain at least 1 local chord. Column *A* refers to the minimum number of chords in G of the said red vertex that must be present in

⁴An observant reader may realize that we already know how to deal with Case 1–4. Case 5 requires a bit more work.

the partitioned subgraph g_i as a chord or as an edge of its maximal empty cycle. Column B denotes the maximum number of chords that may be required for a red vertex of the given type to ensure that the vertex has at least 3 internal chords in the resulting graph $G'' := G \langle S_{prep} \rangle$. Finally Column C notes the minimum number of provisioning chords the red vertex may still require, unless it is already dominating a green vertex due to Lemma 4.41 and Lemma 4.36.

Now let us describe each case in detail.

Case 1: Non-priority *BRB* vertex

Let u be a non-priority *BRB* vertex consecutive to two black vertices v and w on the maximal empty cycle of a partitioned subgraph g_i of the maximal outerplane graph G . Because u is a red vertex and because v and w are black vertices, they must all be in I_{avoid} and thus they cannot be consecutive on G 's maximal empty cycle. Thus both uv and uw must be chords of G . Furthermore, because u is not a priority vertex, u must have at least one local chord ux in g_i . This chord ux along with uv and uw give us the three chords u requires in G . Thus no additional chord is needed for u .

Case 2: Non-priority *BRG* vertex

Let u be a non-priority *BRG* vertex consecutive to a black vertex v and a green vertex w on the maximal empty cycle of the partitioned subgraph g_i of the maximal outerplane graph G . Because u is a red vertex and because v is a black vertex, they must both be in I_{avoid} and thus they cannot be consecutive on G 's maximal empty cycle. Thus uv must be a chord of G . Furthermore, because u is not a priority vertex, u must have at least one local chord ux in g_i . This chord ux along with uv are two chords of u in G and by adding one more chord to be added in g_i we can ensure that u 's chord requirement is met.

Case 3: Non-priority *GRG* vertex

Let u be a non-priority *GRG* vertex consecutive to two green vertices v and w on the maximal empty cycle of the partitioned subgraph g_i of the maximal outerplane graph G . Because both v and w are green vertices, neither uv nor uw can be a partitioning chord of G and thus v and w must be consecutive to u in G 's maximal empty cycle. Thus neither uv , nor uw is a chords of G . Furthermore, because u is not a priority vertex, u must have at least one local chord ux in g_i . This chord ux is the only known chord of u in G . Thus u still requires 2 more chords to meet its requirement. From Lemma 4.41 we know that u would have at least 2 provisioning chords in g_i , or u already dominates a green vertex.

Case 4: Priority *BRB* vertex

Let u be a priority *BRB* vertex consecutive to two black vertices v and w on the maximal empty cycle of the partitioned subgraph g_i of the maximal outerplane graph G . Because u is a red vertex and because v and w are black vertices, they must all be in I_{avoid} and thus they cannot be consecutive in G 's maximal empty cycle. Thus both uv and uw are chords of G . Since u is a priority vertex, u does not have any local chords in g_i . Thus uv and uw are the 2 known chords u has in G , both of which are edges on the maximal empty cycle of g_i . Thus by adding one more chord of u in g_i , we can guarantee that u 's degree requirement is met.

Case 5: Priority BRG vertex

Let u be a non-priority BRG vertex consecutive to a black vertex v and a green vertex w on the maximal empty cycle of the partitioned subgraph g_i of the maximal outerplane graph G . Because u is a red vertex and because v is a black vertex, they must both be in I_{avoid} and thus they cannot be consecutive in G 's maximal empty cycle. Thus uv must be a chord of G . Since u is a priority vertex, u does not have any local chord in g_i . Thus uv is the only known chords u has in G , which is an edge on the maximal empty cycle of g_i . Thus by adding two more chords to u in g_i , we guarantee that u 's degree requirement is met.

Case 6: Priority GRG vertex

Let u be a non-priority GRG vertex consecutive to two green vertices v and w on the maximal empty cycle of the partitioned subgraph g_i of the maximal outerplane graph G . Because both v and w are green vertices, neither uv nor uw can be a partitioning chord of G . This makes both v and w consecutive to u on the maximal empty cycle of G . But then u cannot be incident to a chord that was used to split G as both v and w are both present in g_i . Since u must have at least one chord in G , this chord must also be present in g_i . But then u is no longer a priority vertex in g_i . Thus this case is not possible.

4.4.9 Provisioning Chord Sufficiency

In the next lemma we establish that given a partitioned subgraph, there is sufficient number of provisioning chords to meet the requirement of all red vertex in the partitioned subgraph.

Lemma 4.44. *Given a partitioned subgraph g_i of G , by flipping only provisioning chords, with two simultaneous flips g_i can be transformed into g'' , where the degree requirement of every red vertex has been met, or it dominates a green vertex.*

FIGURE 4.17: u is a priority BRG vertex with exactly one provisioning chord pq . Zigzagged arcs represent a set of edge(s). Dashed edges may be a chord or an edge of the empty cycle they are in. Shaded region may be further triangulated.

Proof. **Cases 1–4:** Due to Lemma 4.36, if a red vertex of a partitioned subgraph g_i of a maximal outerplane graph G does not have any provisioning chords, then it must dominate a green vertex. Otherwise, as illustrated in Table 4.1, given a partitioned subgraph g_i of G , every type of red vertex in it of Cases 1–4 must have at least as many provisioning chords as it may require to have 3 internal chords in G . Thus

for these cases using Lemma 4.39 we can compute $s_i := (s_{i_1}, s_{i_2})$ such that each red vertex of $g_i'' := g_i \langle s_i \rangle$ has at least 3 incident internal chords or it dominates a green vertex.

Case 5: A priority BRG vertex has exactly one provisioning chord, and no incident chord in g_i . Let u be a priority BRG vertex with a single provisioning chord pq (refer to Figure 4.17 left). Then pu and qu are edges of the maximal cycle c_i of g_i . WLOG let p be a green vertex and q be a black vertex and let the other vertex that sees pq be v . We now consider two subcases—a. v is a red vertex and b. v is not a red vertex.

Case 5a: Vertex v is red: Since a partitioned subgraph can have only one priority vertex, v can be a red vertex of three types: a non-priority BRB, BRG or GRG vertex. Due to Lemma 4.34 u and v cannot both be consecutive to q on c_i , so qv must be a chord of g_i . Furthermore, since v is a red vertex, neither pv nor qv can be provisioning chords in g_i . Notice that if pv is not a chord in g_i , then flipping pq makes both u and v dominate the green vertex p . Thus if pv is not a chord, simply using Lemma 4.39 without having any preselected edges is sufficient for meeting the requirements for all red vertices in g_i . **So for the following subcases we assume both pv and qv are chords of g_i .**

1. **v is a non-priority BRB vertex:** In this case both the edges on the maximal cycle c_i incident to v are chords in G . Thus if the final graph $g_i \langle s_i \rangle$ contains at least 1 internal chord of v , it would meet the chord requirement of v . The vertex u requires 2 internal chords in g_i , so it would require one more chord after flipping pq . Define the graph $\bar{g}_i = g_i \langle \{pq\} \rangle$ with the vertex v coloured as black (refer to Figure 4.17 right).⁵ Then $\bar{g}_i$ meets all requirements listed in Definition 4.27 to be a partitioned subgraph of $G \langle pq \rangle$. Thus by using Lemma 4.39, we preselect exactly one (say pv) of these provisioning chords to be flipped in the second simultaneous flip set $\bar{s}_{i_2}$. The other of the two chords (say qv) is marked as a non-provisioning chord so it is not inserted in $\bar{s}_i$. Notice that since both pv and qv are singly provisioning chord with u being their only avoiding vertex, not flipping qv does not affect the process of increasing degree of other red vertices in the graph. Then we get an ordered set of simultaneously flip-pable sets $\bar{s}_i := (\bar{s}_{i_1}, \bar{s}_{i_2})$ that flips all the provisioning chords in $\bar{g}_i$ (except qv). Since uv is not a provisioning chord of $\bar{g}_i$, uv must be present in the resulting graph $\bar{g}_i \langle \bar{s}_i \rangle$. Since qv is not flipped, in the resulting graph v must have two chords uv and qv , meeting v 's degree requirement. Furthermore, flipping pq produces an internal chords incident to u , which along with the chord uv ensures that u has at least two chords. Thus the degree requirement of both u and v are met along with the degree requirement of all the red vertices in $\bar{g}_i \langle \bar{s}_i \rangle$.

Now consider in the graph g_i , $s_i := (\bar{s}_{i_1} \cup \{pq\}, \bar{s}_{i_2})$. Notice that neither of pq 's consecutive edges pv and qv are in $\bar{s}_{i_1}$. Thus $\bar{s}_{i_1} \cup \{pq\}$ must be flippable in g_i . Also notice that the resulting graph $g_i \langle \bar{s}_{i_1} \cup \{pq\} \rangle$ is the same as $\bar{g}_i \langle \bar{s}_{i_1} \rangle$. Thus s_i is flippable in g_i , and in the resulting graph $g_i \langle s_i \rangle$ the degree requirement both u and v along with all the remaining red vertices are met or due to Lemma 4.36 the vertex would dominate a green vertex.

⁵Notice that colouring v to black in g_i means that G_p no longer has any partitioned subgraph where v is red, violating Property 4 of Definition 4.27. This property is used to ensure that so long all red vertices meet their degree requirement, or they dominate a vertex not in I_{avoid} , so do all avoiding vertices in I_{avoid} . Since we explicitly show that v meets its degree requirement, the violation of this property is inconsequential.

2. **v is a non-priority BRG vertex:** In this case if in the final graph $g_i\langle s_i \rangle$, the vertices u is incident to 2 and v is incident to 1 internal chord, then their degree requirement would be met. By following the exact same steps as in Case 5b.1 we compute an ordered set of flippable edge sets s_i in g_i . Then notice that in $g_i\langle s_i \rangle$, u and v have at least 2 and 1 chord respectively. Thus in $g_i\langle s_i \rangle$ requirements of all the red vertices (including u and v) are met or due to Lemma 4.36 the red vertex dominates a green vertex.
3. **v is a non-priority GRG vertex:** We see from Table 4.1 that the maximum number internal chords required by both u and v in g_i is 2. Define the graph $\bar{g}_i = g_i\langle \{pq\} \rangle$ with the vertex v (refer to Figure 4.17 left). Since both u and v are red vertices and they are neighbours, $\bar{g}_i$ is not a partitioned subgraph. Now we split $\bar{g}_i$ along uv and get two maximal outerplane graphs $\bar{g}_l$ and $\bar{g}_r$ defined by the cycles $(\hat{u}\hat{v}, vu)$ and $(\hat{v}\hat{u}, uv)$ respectively. We colour u red and v black in $\bar{g}_l$, and colour these two vertices in reverse in $\bar{g}_r$. Finally, we define u as the priority vertex of $\bar{g}_l$. Since qv is a chord in $\bar{g}_r$, $\bar{g}_r$ has no priority vertex. Notice that both the consecutive vertices of v on the maximal cycle of $\bar{g}_i$ are green and as a result both $\bar{g}_l$ and $\bar{g}_r$ must have at least one green vertex each. Furthermore, since g_i is a partitioned subgraph, and $\bar{g}_l$ and $\bar{g}_r$ are created by splitting g_i , these two graphs meet all requirements listed in Definition 4.27 to be a partitioned subgraph of $G\langle pq \rangle$.

In the partitioned subgraph $\bar{g}_l$, preselect the provisioning chord pv to be flipped on the second simultaneous flip. Then using Lemma 4.39, we compute the ordered set of simultaneously flippable edge sets $\bar{s}_l$ and $\bar{s}_r$ of the graphs $\bar{g}_l$ and $\bar{g}_r$ respectively, and define $\bar{s}_i := (\bar{s}_l \cup \bar{s}_r)$.⁶ Then $\bar{s}_i$ flips all the provisioning chords of both $\bar{g}_l$ and $\bar{g}_r$. The chord qv in $\bar{g}_r$ is not a provisioning chord in $\bar{g}_r$ and thus it cannot be in $\bar{s}_i$. So just as Case 5b.1 neither of pq 's consecutive edges pv and qv are in $\bar{s}_i$. Thus $\bar{s}_{i_1} \cup \{pq\}$ must be flippable in g_i . Observe that the resulting graph $g_i\langle \bar{s}_{i_1} \cup \{pq\} \rangle$ is the same as $\bar{g}_i\langle \bar{s}_{i_1} \rangle$. Thus $s_i := (\bar{s}_{i_1} \cup \{pq\}, \bar{s}_{i_2})$ is flippable in g_i , and in the resulting graph $g_i\langle s_i \rangle$ the degree requirement of all the red vertices are met or due to Lemma 4.36 the vertex would dominate a green vertex.

Case 5b: Vertex v is not red: This case is similar to Case 5a1. Except since v is not a red vertex, we do not need to recolour v to black. We define the graph $\bar{g}_i = g_i\langle \{pq\} \rangle$. Using the same argument as before, $\bar{g}_i$ must be a partitioned subgraph of $G\langle pq \rangle$. So we compute $\bar{s}_i$ and using it compute s_i , as shown in Case 5a1. Then s_i is flippable in g_i , and in the resulting graph $g_i\langle s_i \rangle$ the degree requirement of all the red vertices are met or due to Lemma 4.36 the vertex would dominate a green vertex.

Thus every possible cases for computing $s_i := (s_{i_1}, s_{i_2})$ to sufficiently processing the red vertices in g_i has been handled—completing the proof of this lemma. $\square$

Recall that the Preparation Method Property 2c requires that no two single provisioning chord of the same avoiding vertex can both be in S_{prep_1} . The next lemma satisfies all the output properties of the Preparation Method.

Lemma 4.45. *Given a maximal outerplane graph G and a set on non-consecutive vertices I_{avoid} on its maximal cycle C , an ordered set of two simultaneous flip sets $S_{prep} := (S_{prep_1}, S_{prep_2})$ can be computed, such that the graph $G\langle S_{prep} \rangle$ satisfies the following properties:*

⁶The method to compute the union of two ordered set of edge sets is defined in Definition 4.1.

1. The simultaneous flip sets in S_{prep_1} and S_{prep_2} consist of only provisioning chords of G and $G' := G \langle S_{prep_1} \rangle$ respectively.
2. Every vertex in I_{avoid} either has at least 3 internal chords in $G \langle S_{prep} \rangle$, or it dominates a vertex of $G \langle S_{prep} \rangle$ not in I_{avoid} .
3. No two singly provisioning chords of the same vertex of I_{avoid} is in S_{prep_1} .

Proof. Using Procedure **GENERATESUBGRAPHS**(G, I_{avoid}), partition G into a set of partitioned k subgraphs G_p . For every partitioned subgraph g_i using Lemma 4.44 compute an ordered set of two simultaneous flip sets $s_i := (s_{i_1}, s_{i_2})$.

Define $S_{prep} := \bigcup_{i=1}^k s_i$.⁷ Since each s_i is an ordered set of flippable sets. Because they each correspond to different non-overlapping partitioned subgraphs of G , the edges of each of edge set in S_{prep} must be pairwise non-consecutive. Since G is an outerplane graph, it cannot have blocked chords or bad pairs as it requires the graph to have a subdivision of K_4 . Thus S_{prep} must be simultaneously flippable in G . Furthermore, due to Lemma 4.44 every red vertex of every partitioned subgraph g_i has at least 3 internal chords in $G \langle S_{prep} \rangle$ or it dominates a green vertex. Due to Lemma 4.30 every vertex in I_{avoid} is a red vertex in exactly one partitioned subgraph. Thus, every vertex $v \in I_{avoid}$ either has at least 3 internal chords, or v dominates a green vertex u in its respective partitioned subgraph g_i . Due to Lemma 4.32 if v dominates u in g_i , v must also dominate u in G . This completes the proof of **Property 2**.

We must also ensure Property 3, that no two singly provisioning chords of the same avoiding vertex are inserted in S_{prep_1} . Because every avoiding vertex is red in at most one partitioned subgraph, and because we only flip provisioning chords which are per definition seen by at least one red vertex—for a red vertex of a partitioned subgraph g_i to already have a singly provisioning chord inserted in S_{prep_1} , that provisioning chord must be a chord of g_i and not in some other partitioned subgraph of G_p . Thus for each partitioned subgraph g_i , ensuring that for all red vertices in g_i , no two singly provisioning chords of the red vertex are both inserted in s_{i_1} would satisfy Property 3.

We now minimally modify the Lemma 4.44 for computing the ordered set of flip sets s_i of a partitioned subgraph g_i in G_p such that no two singly provisioning chords of the same red vertex is flipped in s_{i_1} . To do so, while computing s_i in g_i , any time a singly provisioning chord e of a red vertex v is inserted in s_{i_1} , we preselect all the remaining singly provisioning chords of v in g_i in s_{i_2} . For the Lemma 4.44 to fail to compute a flippable ordered set of edge sets s_i due to this preselection, during traversing the faces of g_i via *BFS*, we must have encountered a face uvw with two provisioning chords uv and vw that must both be in s_{i_1} with at least one of them being a singly provisioning chord that has been preselected.

Let uv and vw be two provisioning chords sharing the face uvw in a partitioned subgraph g_i with the maximal cycle c_i (refer to Figure 4.18). Then uv is seen by w and vw is seen by u . Let x and y be the other two vertices that see uv and vw respectively. Recall that in a partitioned subgraph, no provisioning chord can be incident to a red vertex, as it would make two red vertices neighbours. Thus none of the vertices u, v or w can be red vertices. Thus x and y must be the corresponding red vertices of uv and vw respectively. Here x is on the maximal outerplane subgraph defined by cycle $(\widehat{uv}, vu)$ and y on maximal outerplane subgraph defined by the

⁷The method to compute the union of two ordered set of edge sets is defined in Definition 4.1.

FIGURE 4.18: uv and uw are two consecutive provisioning chords in a partitioned subgraph g_i . Its empty cycle c_i is drawn in blue. Zigzagged arcs represent a set of edge(s). Dashed edges may be a chord or an edge of the empty cycle they are in. Shaded region may be further triangulated. Dual tree is drawn in blue.

cycle $(\widehat{vw}, vw)$. Notice that the dual trees of these two subgraphs are on different branches of the dual tree of g_i from the node corresponding to the face uvw . WLOG, let uv be a singly provisioning chord preselected to be inserted in s_{i_2} . Then x must have another singly provisioning chord in g_i that has already been inserted in s_{i_1} . Since all of x 's provisioning chords must be in the subgraph $(\widehat{uv}, vu)$, the previous face that was processed via *BFS* before processing uvw must have been xuv . Let the branch of the dual tree g_i^* that corresponds to subgraph $(\widehat{uv}, vu)$ be B . Then so far the *BFS* search must have processed only a subset of faces, whose corresponding vertices in the dual tree of g_i are all in B . Notice that y cannot have any provisioning chord whose corresponding edge in g_i^* is in B . Thus vw cannot be a preselected provisioning chord when the face uvw is processed. This contradiction completes the proof that the presented method ensures **Property 3**.

Finally, since for every partitioned subgraph g_i in G_p , its ordered set of edge sets s_i consists of only provisioning chords of g_i , S_i must consist of only provisioning chords of G meeting the requirement of **Property 1**. $\square$

4.5 $G'_H \rightarrow \Delta_n$: Introducing Two Dominant Vertices

In this section, we use the following well-known result without proof.

Lemma 4.46. *Any tree of order $n \geq 2$ has at least 2 vertices of degree 1.*

Notice that the dual tree of both the maximal outerplane triangulations defined by the Hamiltonian cycle of G'_H has a diameter $\mathcal{O} \log n$. So instead of introducing the two dominant vertices sequentially, we can introduce them parallelly using the next lemma.

Lemma 4.47. *Let G be a triangulation on n vertices with a Hamiltonian cycle H , where the diameter of the dual tree of both $G\{H\}_{int}$ and $G\{H\}_{ext}$ (See Definition 4.7) is at most d . Then with a sequence of at most d simultaneous flips, G can be transformed into a canonical triangulation.*

Proof. Notice that any triangulation of order $n \leq 5$ vertices already have two dominant vertices. So it is sufficient to provide a way to introduce two dominant vertices for triangulations of higher order. Since H is a Hamiltonian cycle, $G\{H\}_{int}$ and

$G\{H\}_{ext}$ must be maximal outerplane graphs. Let their corresponding empty cycles be C_{int} and C_{ext} . Due to Lemma 4.46 the dual tree of both $G\{H\}_{int}$ and $G\{H\}_{ext}$ must each have at least 2 leaf nodes. Thus, both C_{ext} and C_{int} must have at least 2 foreign 2-chords in G , each with an avoiding vertex. Let u be an avoiding vertex of C_{int} and v be an avoiding vertices of C_{ext} . Because the graph is simple, u and v must be distinct.

We define $G_u := G\{H\}_{int}$ and $G_v := G\{H\}_{ext} \setminus \{u\}$ with their corresponding maximal empty cycle being C_u and C_v respectively. Notice that G_u and G_v are both maximal outerplane graphs. Thus using Lemma 2.65 we can compute two ordered sets of simultaneous flip set S_u and S_v which would make u and v dominant vertex in their corresponding maximal outerplane subgraphs G_u and G_v , where both $|S_u|, |S_v| \leq d$. Set $S := S_u \cup S_v$.

We now prove the flippability of S in G .

- **Non-consecutivity:** Every edge of S_u is in the interior of H and every edge of S_v is in the exterior of H . Thus no edge of S_u can be consecutive to an edge of S_v . Thus edges of S must be pairwise non-consecutive.
- **Bad pair:** For two edges to form a bad pair, they must be seen by the same pair of vertices. As we flip each simultaneous flip set in S_u in sequence, each chord of the next simultaneous flip set in S_u is seen by u . Since u is not in C_v , none of the edges in S_v can be seen by u . Thus for any flip set in S , no two of its edges can form a bad pair.
- **No blocked edge:** Due to Lemma 2.38, each chord of G_u and G_v is (individually) flippable in their respective subgraphs. Thus for any edge in S_u or S_v to be blocked, it must be blocked by a foreign chord of C_u and C_v respectively. Furthermore, since every edge of S_u when flipped, produces an edge incident to u , this blocking chord must be a foreign chord of C_u incident to u . Similarly, for any edge of S_v to be blocked, v must be incident to a foreign chord of C_v . Observe that u and v are avoiding vertices of C_{int} and C_{ext} respectively. Then neither u , nor v can have any foreign chords of C_{int} and C_{ext} respectively. Thus when the edge sets in S are flipped in sequence, none of these the edges can be blocked in G .

Thus due to Lemma 2.50 the ordered set of edge sets S must be flippable in G and in the resulting graph Δ_n u and v are two dominant vertices. Finally $|S| = \max(|S_u|, |S_v|) \leq d$. Thus with a sequence of at most d simultaneous flips, G shall be transformed into the canonical triangulation Δ_n . $\square$

4.6 Complexity Analysis for the Interlaced Algorithm

We compute the complexity of the Interlaced Algorithm, by evaluating the complexity of each of the steps mentioned in Section 4.1.

$G_1 \rightarrow G_H$: Transforming Into Hamiltonian Triangulation

Bose et al. (2007) proved that with a single simultaneous flip any triangulation G_1 can be transformed into a Hamiltonian triangulation G_H with a Hamiltonian cycle H .

$G_H \rightarrow G'_H$: Transformation to Logarithmic Diameter

Once we have the Hamiltonian triangulation G_H , we use the Interlaced algorithm on G_H arbitrarily defining $G_{ear}^1 := G\{H\}_{int}$ and $G_{prep}^1 := G\{H\}_{ext}$.

On the first iteration, G^1 is transformed into G^2 where the diameter of the dual tree of G_{prep}^2 is exactly 2 less than that of G_{ear}^1 maintaining the requirements specified in Lemma 3.12 between G_{ear}^1 and G_{prep}^2 . Furthermore, the order of G_{prep}^2 is at most $\frac{11}{12}$ times the order of G_{ear}^1 .

On each subsequent iteration of the algorithm where $i > 1$, with 2 simultaneous flip G^i is transformed into G^{i+1} where the diameter of the dual tree of G_{prep}^{i+1} is exactly 2 less than that of G_{ear}^i . Furthermore, the order of G_{prep}^{i+1} is at most $\frac{5}{6}$ times the order of G_{ear}^i . This process also satisfies the requirements specified in Lemma 3.12 between G_{ear}^i and G_{prep}^{i+1} . Notice that on the even iterations the interlaced algorithm creates ears in $G\{H\}_{int}$, and in the odd iterations the algorithm creates ears in $G\{H\}_{ext}$. Thus using Lemma 3.12, no more than $2 \times \frac{2 \log n}{\log \frac{6}{5}}$ simultaneous flips are required to transform G_H into G'_H , where the diameter of the dual tree of both $G\{H\}_{int}$ and $G\{H\}_{ext}$ are at most $\frac{2 \log n}{\log \frac{6}{5}}$.

Thus the total number of simultaneous flips required including the first iteration is at most $\frac{4 \log n}{\log \frac{6}{5}} + 2$

We now show how to remove the additional +2 constant from this complexity term. Due to Lemma 4.46 any maximal outerplane graph with $n > 3$ vertices must have at least 2 ears. By removing at least 2 ears on each iteration, no more than 5 iterations are required (instead of $\frac{2 \log 12}{\log \frac{6}{5}}$ iterations) to reduce an outerplane graph with $n \leq 12$ vertices to 3 or less vertices. Thus the total number of simultaneous flips (including the first iteration) required would be at most:

$$\frac{4 \log n}{\log \frac{6}{5}} + \max \left\{ 2 + 5 - \frac{2 \log 12}{\log \frac{6}{5}}, 0 \right\} \leq \frac{4}{\log \frac{6}{5}} \log n$$

Here inside the max function, the term 2 is for the first iteration, and the term $(5 - \frac{2 \log 12}{\log \frac{6}{5}})$ is obtained by using Lemma 4.46.

$G'_H \rightarrow \Delta_n$: Introducing Two Dominant Vertices

Since the diameter of the dual tree of G'_H is at most $\frac{4}{\log \frac{6}{5}} \log n$, using Lemma 4.47, with no more than another $\frac{4}{\log \frac{6}{5}} \log n$ simultaneous flips, G'_H can be transformed into a canonical triangulation Δ_n .

$\Delta_n \rightarrow G_2$: Do the Reverse to Transform Δ_n to Target Triangulation

As explained in Section 2.4.8, sequence of simultaneous flips are reversible. Thus we can convert Δ_n to G_2 with the same number of simultaneous flips required to transform G_2 to Δ_n .

Theorem 4.48. *Let G_1 and G_2 be two triangulations on n vertices. There is a sequence of no more than $c \log n + 2$ simultaneous flips to transform G_1 into G_2 , where $c = \frac{12}{\log \frac{6}{5}} \approx 45.6$.*

Proof. The number of simultaneous flips required to do the following is:

1. $G_1 \rightarrow G_H$: A single simultaneous flip is required.

2. $G_H \rightarrow G'_H$: At most $\frac{4}{\log \frac{6}{5}} \log n$ simultaneous flips are required.
3. $G'_H \rightarrow \Delta_n$: At most $\frac{2}{\log \frac{6}{5}} \log n$ simultaneous flips are required.
4. $\Delta_n \rightarrow G_2$: This is the same as the sum-total of the previous 3 steps. Thus no more than $\frac{6}{\log \frac{6}{5}} \log n + 1$ simultaneous flips are required.

Total number of simultaneous flips required:

$$2 \times \left(\frac{6}{\log \frac{6}{5}} \log n + 1 \right) = \frac{12}{\log \frac{6}{5}} \log n + 2$$

Thus with no more than $c \log n$ simultaneous flips G_1 can be transformed into G_2 , where $c = \frac{12}{\log \frac{6}{5}}$. □

Chapter 5

Conclusion

In this thesis, we have considered the problem of transforming any triangulation into any other triangulation of equal order via a sequence of simultaneous flips. Bose et al. (2007) discovered an algorithm for solving the problem requiring no more than $4 \left(\frac{2}{\log \frac{54}{53}} + \frac{2}{\log \frac{6}{5}} \right) \log n + 2 \approx 327.1 \log n + 2$ simultaneous flips. The bottleneck of this algorithm is the transformation of the two maximal outerplane graphs $G\{H\}_{int}$ and $G\{H\}_{ext}$ of a Hamiltonian triangulation G_H with a selected Hamiltonian cycle H , so that for the resulting graph G'_H , diameters of the dual trees of both $G'\{H\}_{int}$ and $G'\{H\}_{ext}$ are $\mathcal{O}(\log n)$, where n is the number of vertices. To achieve this, for each of these maximal outerplane triangulations, the authors broke down the process into a set of iterations consisting of 2 simultaneous flips each. In each iteration, an independent set of vertices I of size at least $n/6$ is selected where each vertex has at most 2 local chords. The more of these vertices can be made into ears in the respective maximal outerplane subgraphs, by flipping their incident local chords, the fewer number of simultaneous flips will be required to make the diameter $\mathcal{O}(\log n)$ of the dual tree of the resulting maximal outerplane graphs. At the worst-case, the algorithm of Bose et al. (2007) ensures only $n/54$ ears per iteration. The major contribution of this thesis is in addressing this bottleneck of the algorithm. In Chapter 3 we ensured that after the first simultaneous flip of each iteration, each vertex in I has a degree at most 3. This, along with a few other improvements lead to guaranteeing at least $n/12$ ears to be produced in each iteration. In Chapter 4 we ensured that, after the second simultaneous flip of each iteration, each vertex in I has a degree at most 2. This guarantees at least $n/6$ ears to be produced in each iteration.

In Chapter 3 we have modified this algorithm to reduce its complexity to $4 \left(\frac{2}{\log \frac{12}{11}} + \frac{2}{\log \frac{9}{7}} \right) \log n + 2 \approx 85.8 \log n + 2$ simultaneous diagonal flips. To achieve this, we have improved the complexity of transforming both the maximal outerplane subgraphs defined by a Hamiltonian cycle of a triangulation into another whose dual tree is of diameter $\mathcal{O}(\log n)$. The following two properties of triangulations, established in this thesis, have been crucial in the improvement of Bose et al. (2007) algorithm.

1. No two consecutive local chords of an empty cycle can be both blocked.
(see Lemma 3.3)
2. No three blocking foreign chords of an empty cycle can share a face.
(see Lemma 3.5)

We also showed that to make the diameter of the dual tree of an outerplane graph $\mathcal{O}(\log n)$ with $\mathcal{O}(\log n)$ simultaneous flips, the independent set I_5 yields better results than the independent set I_4 . Here I_d denotes a set of independent vertices, none of which has a degree greater than d .

In Chapter 4 we introduced a novel algorithm that further improves the upper bound for transforming triangulations via simultaneous flips. In this new approach, while the interlaced algorithm creates ears in one of the maximal outerplane graphs between $G\{H\}_{int}$ and $G\{H\}_{ext}$, it simultaneously preprocesses the other maximal outerplane graph. As a result of this preprocessing, a larger number of ears are guaranteed to be introduced in it on the next iteration. The process is repeated until the diameter of the dual tree of both the maximal outerplane graphs become logarithmic to their order. Next, by introducing the two dominant vertices simultaneously instead of one after the other, we further reduced the upper bound for transforming a triangulation to a canonical triangulation, and hence to any other triangulation. This has resulted in requiring no more than $\frac{12}{\log \frac{6}{5}} \log n + 2 \approx 45.6 \log n + 2$ simultaneous flips to transform any triangulation to any other of equal order.

We must also mention that during modification of the algorithm we have taken the course of transforming one triangulation into another via canonical triangulation, which is not necessarily optimal. There may be shortcuts in transforming one triangulation into another without ever passing through the canonical triangulation.

5.1 Open Problems

During this research we felt that the following problems related to diagonal flipping are worth exploring.

1. A triangulation G is given with an empty cycle C , and a set of non-consecutive vertices I on C with degree at most 4 in the maximal outerplane subgraph $G\{C\}$. Using no more than 2 simultaneous flips, what is the largest ratio of the vertices in I whose degree can be reduced to 2 in $G\{C\}$? This is the inverse of what is done in our preparation method. (See Section 4.4)
2. Can a more elegant method or proof be introduced for the Preparation Method in Section 4.4?
3. Is it possible to improve the complexity of the interlaced algorithm in Chapter 4 by making the diameter of the dual tree of the two maximal outerplane subgraphs logarithmic to the graph's order more efficiently? One possibility is to use a larger number of simultaneous flips per iteration to create ears out of a larger set of independent vertices I_d , where each vertex has a degree at most d .
4. From Section 3.2, we know that I_5 is optimal for transforming a maximal outerplane graph into one whose dual tree is logarithmic in diameter to its size. Does such a Preparation Method exist that allows the technique introduced in Theorem 3.13 to be applied directly in a Hamiltonian triangulation? This would produce the optimal version of the interlaced algorithm.

Bibliography

- Beauchamps, D Gouyou (1978). “Un algorithme de recherche de circuits Hamiltoniens dans les graphes 4-connexes planaires”. In: *Problemes Combinatoires et Theorie des Graphes* (CNRS, Paris, 1978), pp. 185–187.
- Biedl, Therese C, Prosenjit Bose, Erik D Demaine, and Anna Lubiw (2001). “Efficient algorithms for Petersen’s matching theorem”. In: *Journal of Algorithms* 38.1, pp. 110–134.
- Bose, Prosenjit, Vida Dujmović, and David R Wood (2005). “Induced subgraphs of bounded degree and bounded treewidth”. In: *International Workshop on Graph-Theoretic Concepts in Computer Science*. Springer, pp. 175–186.
- Bose, Prosenjit and Ferran Hurtado (2009). “Flips in planar graphs”. In: *Computational Geometry* 42.1, pp. 60–80.
- Bose, Prosenjit and Sander Verdonschot (2011). “A history of flips in combinatorial triangulations”. In: *Spanish Meeting on Computational Geometry*. Springer, pp. 29–44.
- Bose, Prosenjit, Jurek Czyzowicz, Zhicheng Gao, Pat Morin, and David R Wood (2007). “Simultaneous diagonal flips in plane triangulations”. In: *Journal of Graph Theory* 54.4, pp. 307–330.
- Bose, Prosenjit, Dana Jansens, André Van Renssen, Maria Saumell, and Sander Verdonschot (2014). “Making triangulations 4-connected using flips”. In: *Computational Geometry* 47.2, pp. 187–197.
- Brunet, Richard, Atsuhiko Nakamoto, and Seiya Negami (1996). “Diagonal flips of triangulations on closed surfaces preserving specified properties”. In: *Journal of Combinatorial Theory, Series B* 68.2, pp. 295–309.
- Cardinal, Jean, Michael Hoffmann, Vincent Kusters, Csaba D Tóth, and Manuel Wettstein (2018). “Arc diagrams, flip distances, and Hamiltonian triangulations”. In: *Computational Geometry* 68, pp. 206–225.
- Chartrand, Gary and Frank Harary (1967). “Planar permutation graphs”. In: *Annales de l’IHP Probabilités et statistiques*. Vol. 3. 4, pp. 433–438.
- Chen, Yen-Ju, Jou-Ming Chang, and Yue-Li Wang (2005). “An efficient algorithm for estimating rotation distance between two binary trees”. In: *International Journal of Computer Mathematics* 82.9, pp. 1095–1106.
- Chiba, Norishige and Takao Nishizeki (1989). “The hamiltonian cycle problem is linear-time solvable for 4-connected planar graphs”. In: *Journal of Algorithms* 10.2, pp. 187–211.
- Cortés, Carmen, Clara Grima, Alberto Marquez, and Atsuhiko Nakamoto (2002). “Diagonal flips in outer-triangulations on closed surfaces”. In: *Discrete mathematics* 254.1-3, pp. 63–74.
- Fрати, Fabrizio (2015). “A lower bound on the diameter of the flip graph”. In: *arXiv preprint arXiv:1508.03473*.
- Galtier, Jérôme, Ferran Hurtado, Marc Noy, Stéphane Pérennes, and Jorge Urrutia (2003). “Simultaneous edge flipping in triangulations”. In: *International Journal of Computational Geometry & Applications* 13.02, pp. 113–133.

- Gao, Zhicheng, Jorge Urrutia, and Jianyu Wang (2001). "Diagonal flips in labelled planar triangulations". In: *Graphs and Combinatorics* 17.4, pp. 647–657.
- Hurtado, Ferran, Marc Noy, and Jorge Urrutia (1999). "Flipping edges in triangulations". In: *Discrete & Computational Geometry* 22.3, pp. 333–346.
- Komuro, Hideo (1997). "The diagonal flips of triangulations on the sphere". In: *Yokohama Math* 44 (2), pp. 115–122.
- Kuratowski, Casimir (1930). "Sur le probleme des courbes gauches en topologie". In: *Fundamenta mathematicae* 15.1, pp. 271–283.
- Lubiw, Anna and Vinayak Pathak (2015). "Flip distance between two triangulations of a point set is NP-complete". In: *Computational Geometry* 49, pp. 17–23.
- Mori, Ryuichi, Atsuhiko Nakamoto, and Katsuhiko Ota (2003). "Diagonal flips in Hamiltonian triangulations on the sphere". In: *Graphs and Combinatorics* 19.3, pp. 413–418.
- Nakamoto, Atsuhiko (1996). "Diagonal transformations in quadrangulations of surfaces". In: *Journal of Graph Theory* 21.3, pp. 289–299.
- Negami, Seiya and Atsuhiko Nakamoto (1993). "Diagonal transformations of graphs on closed surfaces". In: *Sci. Rep. Yokohama Nat. Univ. Sect. I Math. Phys.Chem.* 40, pp. 71–97.
- Pallo, Jean (1987). "On the rotation distance in the lattice of binary trees". In: *Information Processing Letters* 25.6, pp. 369–373.
- (2000). "An efficient upper bound of the rotation distance of binary trees". In: *Information Processing Letters* 73.3-4, pp. 87–92.
- Petersen, Julius (1891). "Die Theorie der regulären graphs". In: *Acta Mathematica* 15.1, pp. 193–220.
- Robertson, Neil, Daniel Sanders, Paul Seymour, and Robin Thomas (1997). "The four-colour theorem". In: *Journal of Combinatorial Theory, Series B* 70.1, pp. 2–44.
- Sleator, Daniel D, Robert E Tarjan, and William P Thurston (1988). "Rotation distance, triangulations, and hyperbolic geometry". In: *Journal of the American Mathematical Society* 1.3, pp. 647–681.
- Souvaine, Diane L, Csaba D Tóth, and Andrew Winslow (2011). "Simultaneously flippable edges in triangulations". In: *Spanish Meeting on Computational Geometry*. Springer, pp. 138–145.
- Wagner, K (1936). "Bemerkungen zum Vierfarbenproblem. Jahresber. Math". In: *Deutsch. Math.-Verein.* 46, pp. 26–32.
- Whitney, Hassler (1931). "A theorem on graphs". In: *Annals of Mathematics*, pp. 378–390.

Alphabetical Index

- chord, 8
  - k -chord, 46
  - 2-chord, 46
  - bad chord, 32
  - blocked chord, 32
  - blocking chord, 32
  - earring chord, 48
  - external chord, 8
  - foreign chord, 46
  - internal chord, 8
  - local chord, 46
  - provisioning chord, 48, 64
    - doubly provisioning chord, 48, 64
    - singly provisioning chord, 48, 64
- component, connected component, 21
- corner vertex, 48
- cycle, 8
  - k -cycle, 8
  - empty cycle, 21
    - foreign region, 21
    - local region, 21
    - non-overlapping empty cycles, 46
    - non-overlapping maximal outerplane subgraphs, 46
  - Hamiltonian cycle, 8
  - maximal cycle, 8
- ear, 22
- edge, 6
  - bad pair, 16
  - blocked edge, 17
  - blocking edge, 17
  - bridge, 21
  - consecutive edges, 16
  - loop, 6, 90
  - parallel edges, 6
- face, 8
- flip, diagonal flip, 9
  - canonical triangulation, 6, 10
- ordered set of flippable sets, 46
- seeing, 9
- sequence of flips, 9
- simultaneously flippable set, 16
- standard triangulation, 10
- flip, flippable edge, 9
- graph, 6
  - k -connected, 8
  - k -regular graph, 21
  - combinatorial embedding, 8
    - equivalence class, 8
    - outerface, 8, 11
    - rotation system, 8
  - complete bipartite graph, 7
  - complete graph, 6
  - connected graph, 7
  - dual graph, 21
  - edge induced subgraph, 6
  - edge subdivision, 7
  - flip graph, 12
  - graph subdivision, 7
  - Hamiltonian triangulation, 12
  - induced subgraph, 6
  - isomorphism, 9
  - maximal outerplanar graph, 9
  - maximal outerplane graph, 9
  - order of a graph, 6
  - outerplanar graph, 8
  - outerplane graph, 8
  - partitioned subgraph, 63
  - planar graph, 7
  - plane graph, 7
  - simple graph, 6
  - subgraph, 6
    - external subgraph, 47
    - internal subgraph, 47
  - supergraph, 6
  - triangulation, 8
- matching, 22
  - perfect matching, 22

- neighbourhood, 48
  - closed neighbourhood, 48
- path, 7
  - arc, 32
  - chord-path, 47
- region, 8
- separating triangle, 14
- tree, 8
  - branch, 73
  - dual tree, 21, 22
  - forest, 35
  - leaf, 22
  - unary tree, 35
- used notations
  - $G\langle S \rangle$, 16, 46
  - $G\{C\}$, 25, 47
  - $G\{C\}_{ext}$, 47
  - $G\{C\}_{int}$, 47
  - $N_G(v)$, $N_G[v]$, 48
  - $[\hat{p}\hat{q}]$, $(\hat{p}\hat{q})$, 32
  - $deg_G(v)$, $deg(v)$, 6
  - $ext(C)$, 8
  - $int(C)$, 8
  - union of ordered set of flippable sets $S_1 \cup S_2$, 46
- vertex, 6
  - BRB vertex, 75
  - BRG vertex, 75
  - GRG vertex, 75
  - adjacent vertex, 6
  - avoiding vertex, 46, 48, 64
  - avoiding vertex set, 44, 46
  - avoiding vertices, 63
  - black vertex, 64
  - degree of a vertex, 6
  - dominant vertex, 10
  - dominating vertex, 48
  - earing vertex, 48
  - earing vertex set, 48
  - green vertex, 64
  - independent set, 22
  - non-priority vertex, 76
  - priority vertex, 76
  - red vertex, 64