SIMULATING THE TRANSFORMATION OF PLANE TRIANGULATION USING EDGE FLIP ALGORITHM.

by

Afia Hossain Anushka
18101430
Md. Arman Hossain
18101405
Mukit Kibria
18101373

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
BRAC University
Date of Submission: 20^{th} January, 2022

Declaration

It is hereby declared that

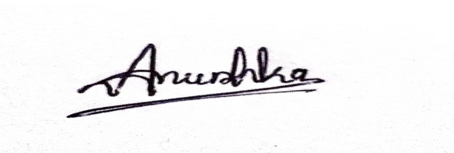
1. The thesis submitted is our own original work while completing degree at BRAC University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

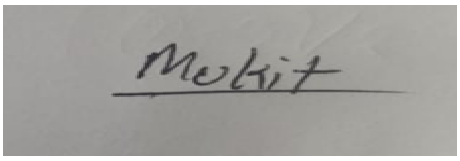
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



Afia Hossain Anushka
18101430

Md. Arman Hossain
18101405



Mukit Kibria
18101373

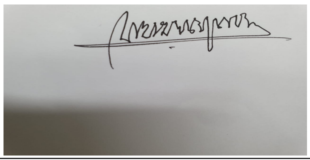
Approval

The thesis titled "Simulating the transformation of plane triangulation using edge flip algorithms." submitted by

1. Afia Hossain Anushka (18101430)
2. Md. Arman Hossain (18101405)
3. Mukit Kibria (18101373)

Of Fall, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on 16^{th} January, 2022.

Examining Committee:
Supervisor:
(Member)



Dr. M. Kaykobad
Distinguished Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Tanvir Kaykobad
PhD Student
School of Computing
Queen's University, Kingston, Canada

Head of Department:
(Chairperson)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
BRAC University

Abstract

In Computational Geometry edge flipping of triangulation is a well-studied topic . Also in computer graphics, triangulations are used to form any kind of shape of an object. Although many algorithms have been introduced for transforming one plane triangulation to any other one, their implementation in the literature could not be found. We have decided to check the behaviour of these algorithms in terms of required flip to transform a triangulation into another triangulation. While worst case behaviour of these algorithms have been established in terms of number of flips, there is a dearth of average case analysis of these algorithms in the literature. We want to gain some insight as to the average behaviour of these algorithms through performing simulation. We would also like to investigate how to visualize the transformation of these plane triangulations in an intuitive way. While the current best-known algorithm for single edge flip is near-optimal, we believe there is much room for improvement when it comes to transforming plane triangulations using a sequence of simultaneous flips.

Keywords: Edge Flipping, algorithms, plane triangulation, visualization, simultaneous flips.

Acknowledgement

All praise to Almighty Allah for whom our thesis has been completed without any major interruption. Firstly, we would like to thank our Supervisor Prof. Mohammad Kaykobad for his support and advises.Secondly, we are really honored to have our co-supervisor Tanvir Kaykobad who helped in each segment when we needed help.Finally to our parents without their throughout support it might not be possible. With their prayer and support we are now on the verge of our graduation.

Table of Contents

Chapter 1

Introduction

Diagonal flipping[2.23] has become an important topic in computational geometry and has been very widely studied. In this thesis, we simulate a number of algorithms related to transforming any triangulation to any other using edge flips. The algorithms are due to Wagner, Komuro and Mori et al[21] among others. Wagner proves that by only flipping edges(diagonals) any triangulation can be transformed into any other triangulation with same number of vertices. According to Wagner's algorithm, $\mathcal{O}(n^2)$ diagonal flips are required for the transformation of triangulation. Then Komuro improved the bound to $\mathcal{O}(n)$. Then Mori et al. improved this bound further to max($6n-30$,0). Recently Cardinal et al. [25], showed that this transformation can be done requiring no more than $5n-23$ edge flips for $n \geq 6$ where n is the number of vertex.

1.1 Problem Statement

Many algorithms have been introduced for transforming one plane triangulation into another with same number of vertices but unfortunately their visualization is missing. While transforming one it is fairly easy to compute this transformation for small number of vertices, when solving this problem can become fairly difficult for large triangulations. If someone wants to visualize the process of transforming a triangulation on twenty vertices in pen and paper, it might take more than an hour. Although, there have been results relating to the upper bound on the number of flips required to transform one triangulation into another, simulated results on the average case appears to be missing in the literature. Hence for our thesis, we have decided to understand some of the best-known algorithm on the topic and also implement the currently best algorithm for transforming triangulations using sequential edge flips. In the course of this research we have also made some observations on better improve the existing algorithm for average case without incurring any cost on worst case.

1.2 Research Objectives

In our research, our motive is to implement the algorithms for transforming a triangulation into a canonical triangulation and show how it works on the average cases. So, the objective of our thesis is to find an estimation of how many edge flips does it requires in an average case scenario to transform one triangulation into another. Many algorithms have tried to decrease the upper bound of required edge flips, but they did not represent the average case scenario. Our goal would be to heuristically compute the performance (number of flips needed) to transform different random triangulations from one to another. After that we can analyze the data and show the difference between the theoretical worst case and the practical average case in required edge flip.

1.3 Organization of the Thesis

In Chapter 1 we have given an introduction to our research problem and then, briefly described about our research objective that what kind of problem is still unknown regarding our topic and how we are going to perform our research about that.

In Chapter 2 we discussed some of the important theories that are available in the literature on sequential flips, provide a brief explanation of the theories.

In Chapter 3 we showed all of our implementations. We showed how Wagner's algorithm works in our code. We showed some flow chart for the better understanding of how we implemented the Mori's theorem to convert a Hamiltonian triangulation to canonical triangulation and Cardinal's theorem to convert a non Hamiltonian triangulation to a Hamiltonian triangulation.

In Chapter 4 we showed the difference between the upper bound given by cardinal and the average case. We compute the average case by simulating 1000 transformation on each vertex (7 to 60).

In Chapter 5 we discovered some points which we observed from the theories. It can bring some positive result and can decrease number of flips. We also showed how we can remove separating triangles from triangulation ($Dummyflip$ and $4-Connector$).

In Chapter 6 we talked about the importance and the future uses of our thesis topic.

Chapter 2

Literature Review

2.1 Triangulation

A graph $G = (V, E)$ consists of a set of vertices V and a set of edges E, each member of which corresponds to a pair of vertices from V. If this pair is unordered we call it a *graph*. Otherwise it is called a *directed graph* or simply a *digraph*. If no two pairs are identical and $(u, v) \in E$ implies that $u \neq v$ then we call it a *simple graph*. Otherwise the graph is called a *multigraph*. A simple graph is said to be *planar* if it can be embedded in such a way that none of its edges intersects other than at the endpoints. If for a graph $G = (V, E)$ there is a planar embedding then G is said to be a plane graph. A region of a plane graph bounded by a sequence of edges and containing no vertices and edges inside is said to be a *face*. If a planar graph $G = (V, E)$ contains all its vertices in a single face then the graph is said to be *outerplanar*. We start with a few terminologies before understanding *triangulation*.

There is a lot of literature on triangulations. For example, simultaneous diagonal flips can be used to transform planar triangulations [26]. There is also further literature on the transformation of a triangulation into another triangulation [1], [16], [18], [22].Many authors presented their conclusions and assumptions regarding diagonal flips while transforming one triangulation into another [11], [12], [17], [24].

Definition 2.1 (Loop) *Given a graph $G = (V, E)$, if an edge connects a vertex to itself then it is called a* loop.

Definition 2.2 (Parallel Edges) *Given a graph $G = (V, E)$ and u, v are two distinct vertices of G, if multiple edges connect the same pair of vertices u, v then they are called* parallel edges.

Definition 2.3 (Simple Graph) *Given a graph $G = (V, E)$, if for every edge $uv \in E$, $u \neq v$ and if no two edges $e_1, e_2 \in E$ connect the same pair of vertices in V, then we can say G is a* simple graph.

Definition 2.4 (Adjacent) *Given a graph $G = (V, E)$, if a vertex A is connected to another vertex B by an edge then B is called* adjacent *to A. It is also called a* neighbor.

In Figure 2.1 the vertices in $\{D, E, B\}$ are adjacent to A.

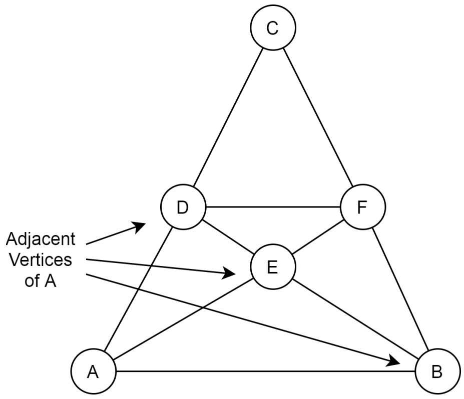


Figure 2.1: Adjacent of vertex of A

Definition 2.5 (Path) *For the ordered set of pair-wise unique vertices* $(u = u_0, u_1, \cdots, u_n = v)$ *if every consecutive pair of vertices is adjacent then this order set of vertices denotes a* path.

Definition 2.6 (Cycle) *Given a graph* $G = (V, E)$*, if for a path* $(u = u_0, u_1, \cdots, u_n = v)$ $u = v$ *then this path is called a cycle.*

Definition 2.7 (Connected) *Given a graph* $G = (V, E)$*, if a path exists between two vertices* u *and* v *where* $u, v \in V$ *then* u *and* v *are said to be* connected.

In Figure 2.2 all the vertices of the graph $\{C, D, E, F, B\}$ are connected to A as there exist a path in all of those vertices.

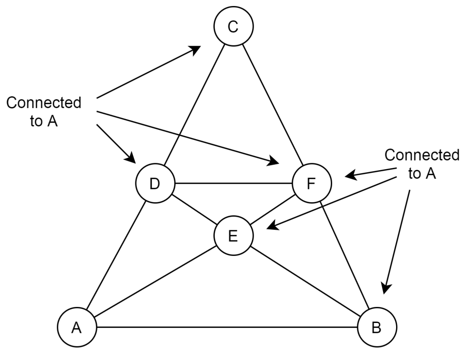


Figure 2.2: Connected vertex of A

Definition 2.8 (Connected graph) *Given a graph* $G = (V, E)$*, if there is no such vertices* u *and* v*, such that* u *and* v *are not connected then the graph is* connected graph. *In another word, every vertices have to be connected with every other vertices to make the graph connected.*

Definition 2.9 (Distance between a pair of vertices) *Given a graph* $G = (V, E)$*, the distance between a pair of vertices* u *and* v *is the minimum number of edges among all paths connecting these two vertices.*

Definition 2.10 (Diameter) *Given a connected graph* $G = (V, E)$*. The longest among the distances between any pair of vertices is called the diameter of graph* G*.*

Definition 2.11 (Plane Graph) *Given a graph G, if it is embedded on a plane such that no two of its edges intersect except at their endpoints, then G is called a* plane graph*. The* outer-face *of G is the (unbounded) face that lies on the outside. However, depending on the problem, sometimes we may nominate a bounded face to be the outerface as well.*

Definition 2.12 (Planar Graph) *Given a graph $G = (V, E)$, if it has a plane embedding with the edges only at the endpoints, then G is called a planar graph.*

Definition 2.13 (Region) *If an area of a plane graph is bounded by a cycle then it is called a* region*. Every cycle of a plane graph divides the graph into two regions; One in its interior and the other in its exterior.*

Definition 2.14 (Face) *Given a region, if there are no vertices or edges strictly inside it, then it is called a* face*.*

Definition 2.15 (Outer-face) *Given an embedding g of a graph, face abc is an outer-face if it has an unbounded area.*

Definition 2.16 (Outerplane Graph) *If a plane graph has all its vertexes on its outerface, then it is called an outerplane graph.*

Definition 2.17 (Maximal Outerplane Graph) *An Outerplane graph is a* Maximal Outerplane Graph *if we cannot add any additional edges to it. Notice that, it will not be an Outerplane Graph anymore if we add any new edge in Maximal Outerplane Graph..*

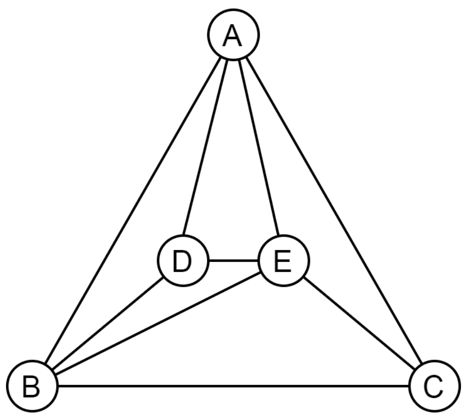


Figure 2.3: Triangulation

Definition 2.18 (Triangulation) *If every face of a planar graph is a 3-cycle, then it is called* triangulation*. Notice that, a triangulation is an (edge) maximal planar graph as inserting any more edges will not keep the graph simple and planar.*

Figure 3.2 illustrates a triangulation where every face is a triangle.

Definition 2.19 (Flip Graph) *Let $G = (V, E)$ be a triangulation. Now consider every non-isomorphic triangulation of G as nodes of g. And two nodes are adjacent if, they can be transformed into one another with a single flip. This graph g is called the flip graph of G.*

2.2 Wagner's Theorem

Wagner's Theorem gives us the idea about how to convert a triangulation into a canonical triangulation. We first start with a few more definitions.

Definition 2.20 (Dominant Vertex) *If a vertex is adjacent to all other vertices then that vertex is called a* dominant vertex*.*

In Figure 2.4 B and C are the dominant vertices.

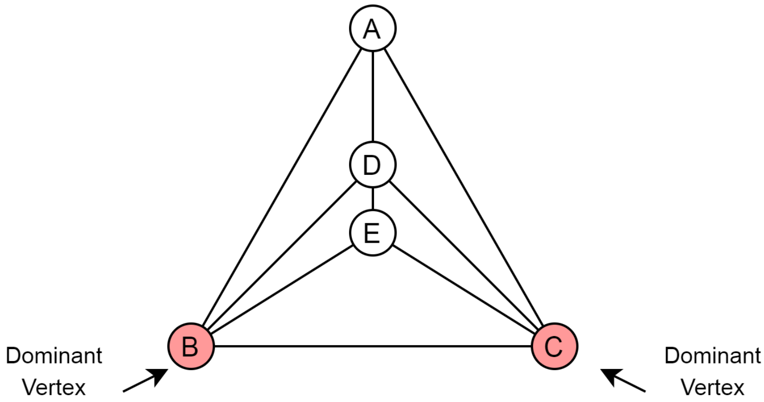


Figure 2.4: Dominant vertex on Canonical Triangulation

Definition 2.21 (Canonical Triangulation) *In a triangulation, if there exists 2 dominant vertices, then the triangulation is called* canonical triangulation*.*

In Figure 2.4 the graph is a canonical triangulation.

Definition 2.22 (Degree) *Given a graph $G = (V, E)$, the number of edges that are incident to a vertex is called the* degree *of that vertex.*

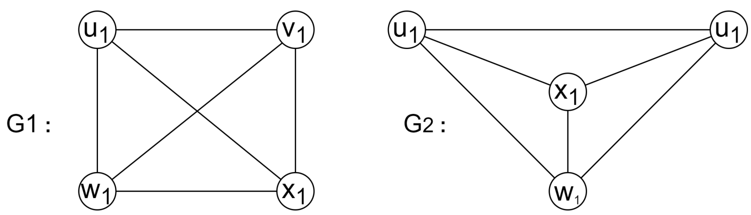


Figure 2.5: Isomorphism

Definition 2.23 (Diagonal Flip) *Let us assume $ABCD$ is a quadrilateral with a diagonal BC where ABC and BCD are faces. Now, if we remove this diagonal and add a new diagonal AD such that AD lies in the same region of $ABCD$ as BC did, then this operation is called* Diagonal flip*.*

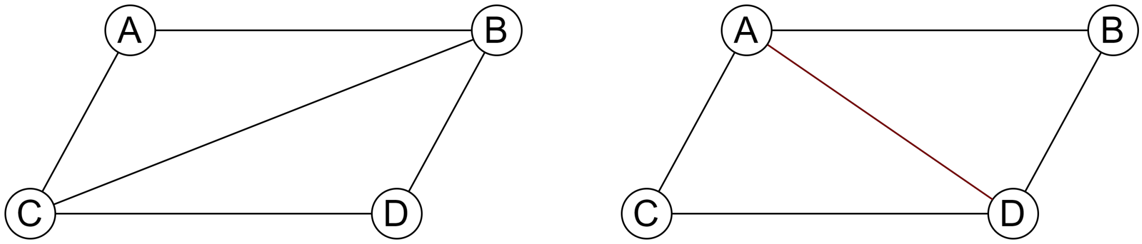


Figure 2.6: Diagonal Flip

In Figure 2.6 here edge CD flipped to edge AD.
To convert any triangulation into any other triangulation of equal order, Wagner first introduced canonical triangulation. Later on it was found that [2] by using Wagner's method we can convert a triangulation with n vertices into a canonical triangulation through at most $2n^2 - 14n + 24$ diagonal flips. Later, Negami and Nakamoto proves that a triangulation can be transform into another triangulation with not more than $2n^2$ diagonal flips. For converting a triangulation G_1 into another triangulation G_2, first, we have to convert G_1 to a canonical triangulation Δ_n. Then if we reversely operate all the flips that are required for converting G_2 into Δ_n, then canonical triangulation G_1' will transform into G_2. Thus G_1 is converted to G_2.

2.2.1 Converting to a Dominant Vertex:

Here AHGE is a quadrilateral shown in Figure 2.7 (part of a triangulation).

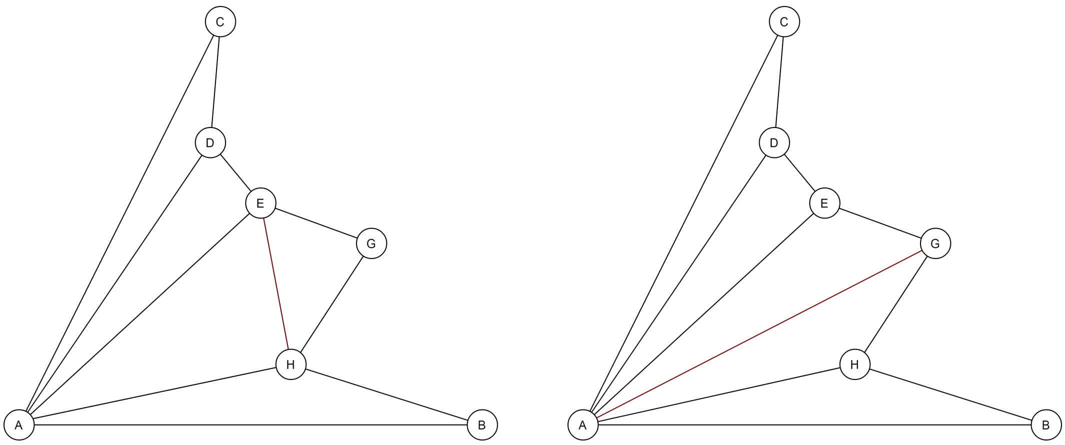


Figure 2.7: Triangulation To Canonical Triangulation Conversion

To convert a triangulation to a canonical triangulation we can follow the given steps:

1. We select a triangular face as outer face and nominate 2 of its vertices to be dominant vertices.

2. We need to check whether the nominated vertices are dominant or not. In Figure 2.8 suppose, we choose face ABC as outer-face and A, C are nominated as the dominant vertices. Next we need to check if A and C are adjacent to all other vertices.

3. If A is not a dominant vertex, then there will exist a triangle, where two of its vertices are adjacent of A but one is not. Here EGH is a triangle where E, H are adjacent of A but G is not.

4. If the triangle AEH is a face, with a single flip we can increase the degree of A. We will flip EH to AG and thus G will be adjacent of A. We will continue this process until A is adjacent to all the vertices and become the dominant vertex.

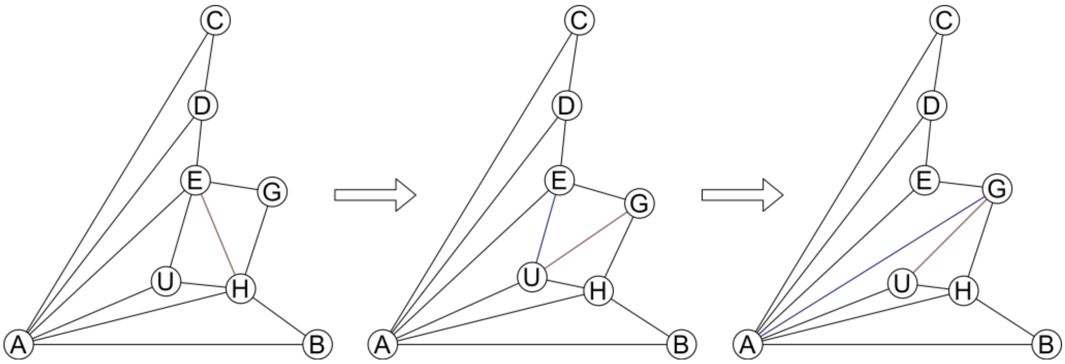


Figure 2.8: Diagonal Flip to make a Canonical Triangulation

5. If AEH is not a face, then we cannot increase the degree of A in a single step. As AEH is a triangle any vertices that exist inside can not be adjacent of G. Now we will increase the degree of G such that, G is adjacent to any of the vertices inside AEH. Then we will start again from step-3.

As this is a finite graph, after certain repetition of these steps eventually A will be adjacent to all the vertices. Next we need to do the same for C. When A and C will become dominant vertex, the graph will convert to a canonical graph.

2.3 Komuro's Bound

In Wagner's result, there was a problem of quadratic in the number of vertices on the diameter of the flip graph. Komuro showed that [14] the diameter of the flip graph was linear by showing upper and lower linear bound to them. By using Wagner's approach, Komuro came up with an idea to transform a triangulation into a canonical triangulation by decreasing the linear upper bound. Let $G = (V, E)$ be a triangulation with n vertices and we want a and b to be dominant vertices. As every vertex in a triangulation has at least 3 edges and each dominant vertices have $n - 1$ edges. So we can say that to make both vertices dominant we need at most $2n-8$ flips, if there is an increment in the flip then the degree of $deg(a)$ or $deg(b)$ will increase at least by 1. But this scenario is not practical. In the Figure 2.9, we can see that, $deg(a)$ or $deg(b)$ does not increase with a single flip. Komuro introduced a function such as: $d_G(a, b) = 3deg(a) + deg(b)$ and stated that, there always exist either one edge flip where $d_G(a, b)$ increases at least by 1 or 2 edge flips where $d_G(a, b)$ increases at least by 2. In some cases, there might be a necessary edge flip which will decrease the degree of b by one, but the next flip will increase the degree of a. As a result the value of $d_G(a, b)$ will increase 2 after 2 flips, which satisfies his claims.

As the dominant vertices has degree of $n-1$, so $d_G(a,b) \leq\ 4n$–4. and we can get an upper bound which is: $4n-4-d_G(a,b)$ to make a and b dominant vertices.

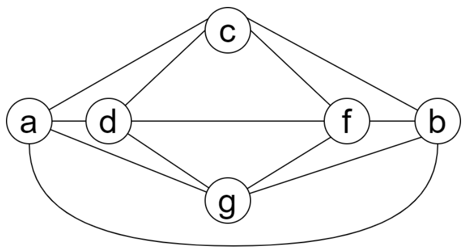


Figure 2.9: No single edge flip can increase the degree of a or b

Definition 2.24 (Separating Triangle) *Given a graph $G = (V, E)$, if removal of a triangle abc (Figure 2.10) disconnects one or more vertices from the graph, then it is called a* separating triangle. *Here abc separates w_1 from the graph.*

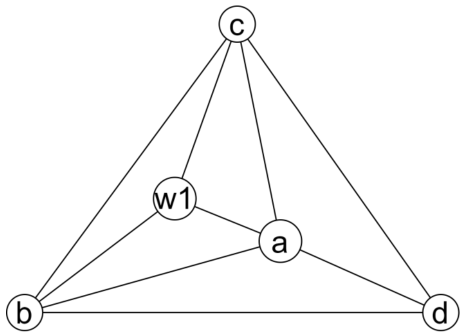


Figure 2.10: Separating Triangulation

Lemma 2.25 *Given a graph $G = (V, E)$ is a triangulation with n vertices, we can make G a canonical triangulation Δ_n where a and b are the dominant vertices with at most $4n-4-d_G(a,b)$ edge flips where $d_G(a,b) = 3deg(a) + deg(b)$.*

Proof: We know every vertex of a triangulation have at least 3 edges. Let uab be a face of G. Here we have two scenario.

- $deg(u) = 3$ or
- $deg(u) > 3$

At first lets consider the 2nd case. In Figure 2.11 a, b, w_1, w_2 be the 5 consecutive neighbours of u in counter-clockwise order. Now if b is not adjacent of w_2 then, flipping uw_1 will increase the degree of b by one and thus $d_G(a,b)$ will also increase by one. And now a, b, w_2 and x_1 are the new four consecutive neighbours of u in counter-clockwise order and we can do the same operation again to increase the $deg(b)$.
Now consider In Figure 2.12. If b adjacent of w_2 then ubw_2 is a separating triangle which separates w_1 form rest of the graph. If we flip ub it will decrease $deg(b)$ by 1 and increase the $deg(a)$ by 1. As a result $d_G(a,b)$ will increase by 2.
Now let us consider case no. 1. Here $deg(u)$ is 3. Let $u1$ be the unique vertex, which is adjacent to a, b and u. Now we can consider 3 different case.

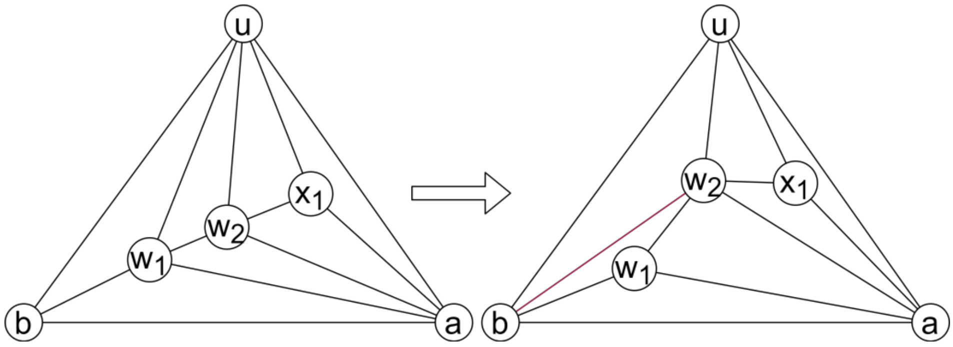


Figure 2.11: One single flip increases the degree of b

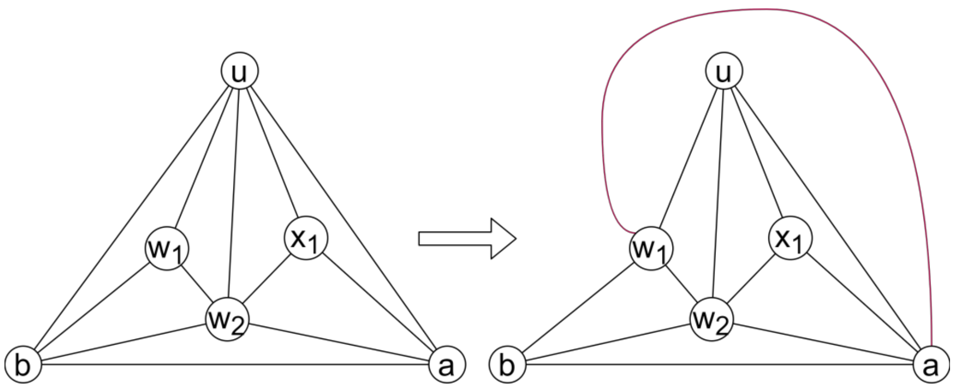


Figure 2.12: One edge flip decreases degree of b but increases degree of a

- $deg(u_1) = 3$
- $deg(u_1) \geq 5$
- $deg(u_1) = 4$

In Figure 2.13 $deg(u_1) = 3$ and the graph is isomorphic to K4 which is Δ_4.

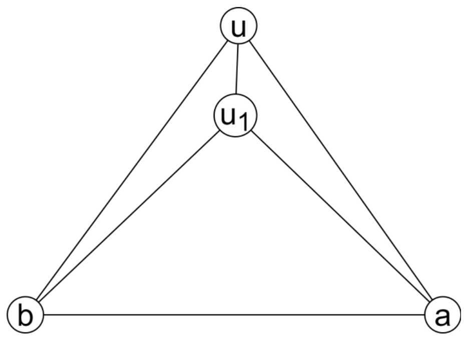


Figure 2.13: isomorphic to Δ_4

Now consider Figure 2.14. Here $deg(u1) \geq 5$, then let a, u, b, w_1 and w_2 be the 5 consecutive neighbours of u_1 in counter-clockwise order. If b is not adjacent to w_2, then flipping edge u_1w_1 will increase the $deg(b)$ by 1 and this will also increase $d_G(a, b)$ by 1.
In Figure 2.15 b is adjacent to $w2$ and u_1bw_2 is a separating triangle which separates w_1 form rest of the graph. Now we will flip u_1b at first and then flip uu_1. this will

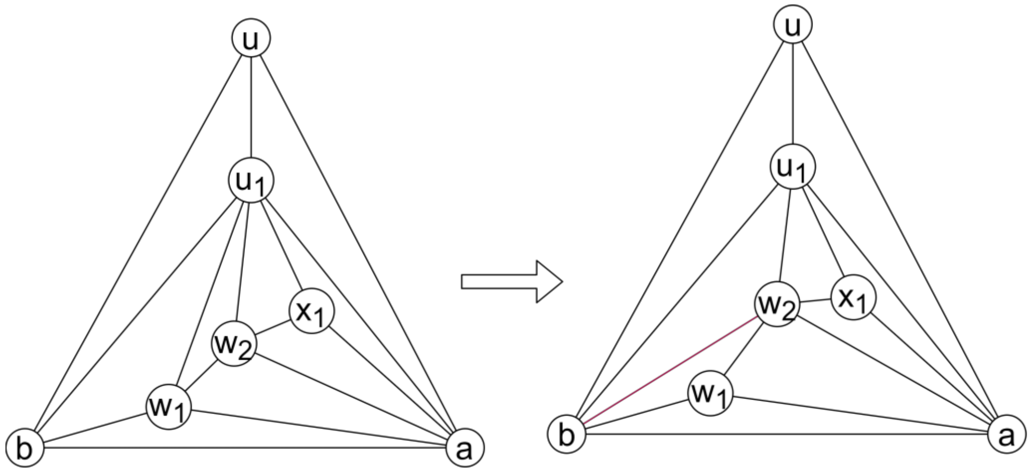


Figure 2.14: One edge flip increases the degree of b

decrease $deg(b)$ by one but increase $deg(a)$ by 1. As a result $d_G(a,b)$ will increase by 2 in two flips. This also satisfy komuro's claim.

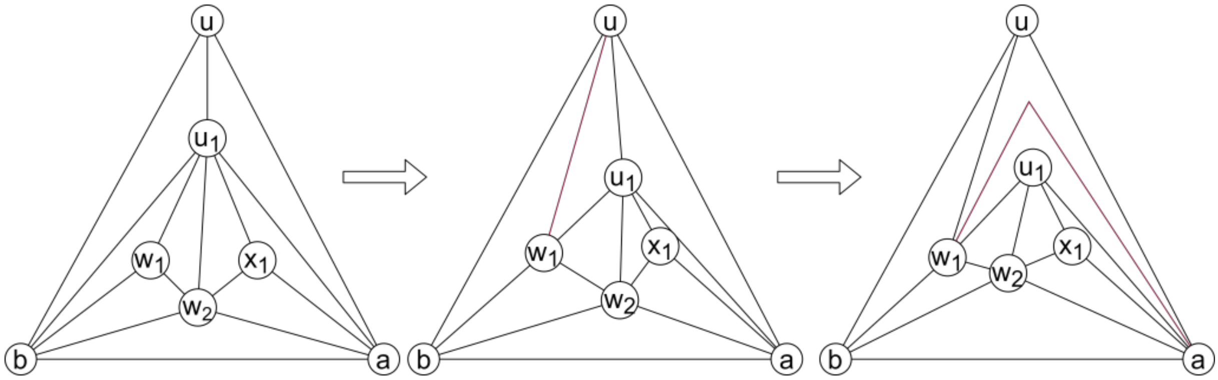


Figure 2.15: 2 edge flips increases $d_G(a,b)$'s value by 2

And finally if $deg(u_1) = 4$ then, there exist another unique vertex u_2 which is adjacent to a, b and u_1. For u_2 we can again consider the same 3 cases of u_1. And we will repeat the same process until we reach u_{n-1} and then a and b will be dominant. Since in every scenario $d_G(a,b)$ is increased by at least 1 in each edge flip, we do not need to do more that $4n - 4 - (3 * deg(a) + deg(b))$.

2.4 Mori et al.'s Bound

Mori, Nakamoto and Ota [19] have improved Komuro's bound for converting a triangulation into a canonical triangulation down to $6n - 30$ where the number of vertices $n \geq 6$. For example, if we have 6 vertices, then we need at most $6*6-30 = 6$ edge flips to convert any triangulation into a canonical triangulation.

Definition 2.26 (Hamiltonian cycle) *A* Hamiltonian cycle *is a cycle where every vertex of the graph occurs exactly once.*

In *Figure* 2.16 notice that, the red marked border is the Hamiltonian cycle.

Definition 2.27 (Hamiltonian Triangulation) *If a triangulation contains a Hamiltonian cycle then it is called* Hamiltonian triangulation*. If the triangulation has n vertices then the length of the cycle is n.*

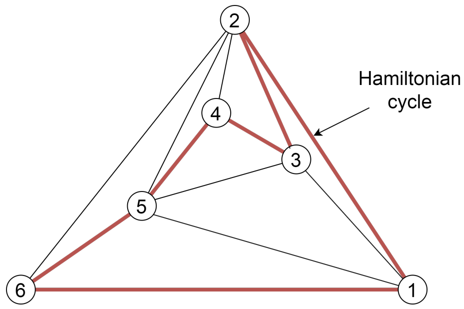


Figure 2.16: Hamiltonian Triangulation

Mori et al. [20] followed only two steps to convert a Hamiltonian triangulation into a Canonical triangulation. In the first step, using the Hamiltonian cycle, we decompose the Hamiltonian triangulation into two outerplanar graphs. Both outer planar graphs contains the Hamiltonian cycle and the left part of the cycle creates one outer planar graph. Similarly the right part of the Hamiltonian cycle creates another outer planar graph. Mori et al. [20] also proved that in a maximal outerplanar

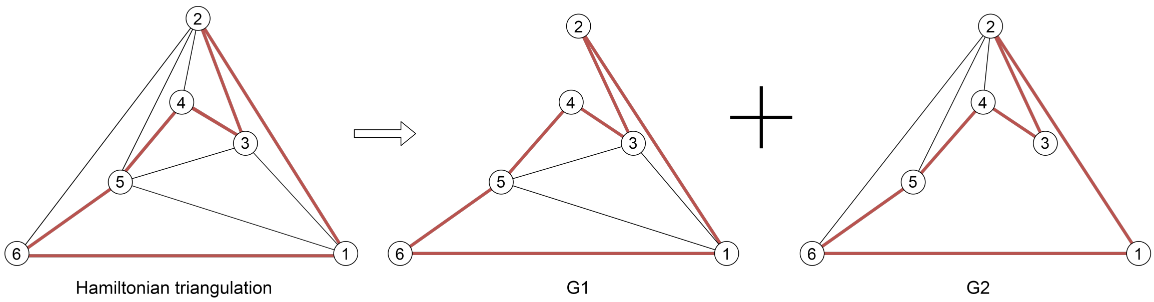


Figure 2.17: Decomposing into two outerplanar graphs.

graph, any vertex v on n vertices can be made dominant by at most $n-1-deg(v)$ edge flips. This property is used in the second step. For example, in $Figure$ 2.17 there are two outerplanar graphs $G1$ and $G2$ which have been decomposed using a Hamiltonian cycle. In $G1$, to make vertex 1 a dominant vertex, we need at most $6-1-4=1$ flip. Similarly in $G2$, to make vertex 2 a dominant vertex, we need at most $6-1-5=0$ flip. So by following the steps, two dominant vertices have created. Here in $Figure$ 2.18, we can see that $G1$ has been converted into G_1' by flipping $(5,3)$ edge to edge $(1,4)$ in quadrilateral $(4,3,1,5)$. Here $(1,4)$ edge increases the degree of vertex 1 which makes vertex 1 a dominant vertex. Similarly, we follow the same method to make vertex 2 dominant in G_2'. Finally by merging the two outerplanar graphs $G1'$ and $G2'$, we get a Canonical Triangulation in $Figure$ 2.19. Again Mori, Nakamoto and Ota [1] proved that any Hamiltonian Triangulation of n vertices which consists a Hamiltonian cycle, can be converted into Canonical triangulation by at most $2n-10$ edge flips. For example, in Figure 15 we have 6 vertices. So to convert this Hamiltonian Triangulation into Canonical Triangulation we need at most $6*2-10=2$ flips. They also proved that any two triangulation on n

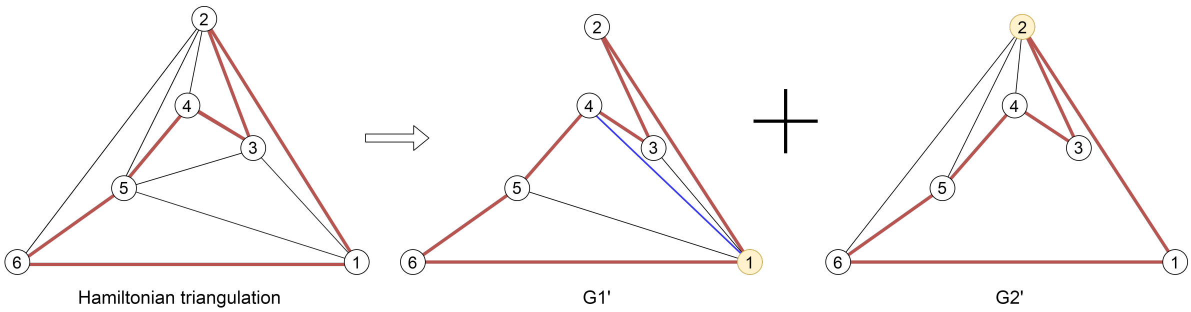


Figure 2.18: Making One vertex dominant from each graph.

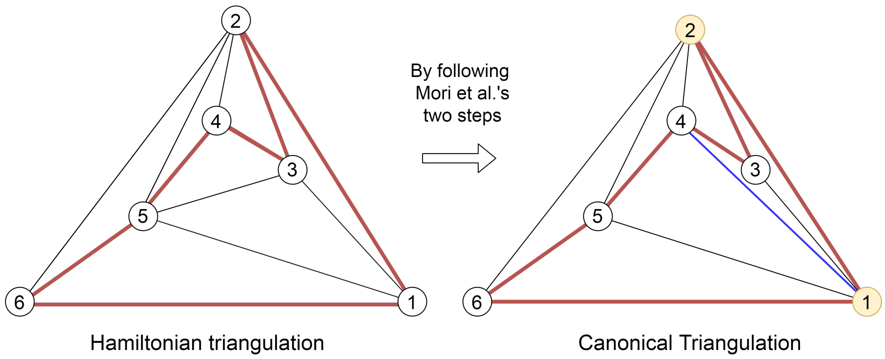


Figure 2.19: Hamiltonian Triangulation to Canonical Triangulation

vertices can be converted into each other by at most $6n - 30$ flips. He also added that a triangulation of n vertices where n≥6, flipping any edge of a separation triangulation will remove the separating triangulation. After flipping the edge no other separating triangle will be created if the selected edge belongs to multiple separating triangle or all the edges of separating triangle not belong to any separating triangle.

2.5 Cardinal et al's Theorem

Cardinal et al. [25] showed that transforming one triangulation into a Canonical Triangulation can be done requiring no more than $5n - 23$ edge flips on a graph of n number of vertices where $n \geq 6$. In Section 2.4, we discussed how Mori et al. transformed a Hamiltonian triangulation into a canonical triangulation. As every 4-connected triangulation is Hamiltonian, Mori et al. first transformed the triangulation into 4-connected triangulation. Bose et al. [23] added that with at most $(3n - 9)/5$ edge flips any triangulation can be transformed into a 4-connected Triangulation.
Cardinal et al. improved the upper bound of Bose et al. by using the fact that we don't necessarily need to transform a triangulation into a 4-connected one to make it Hamiltonian. Cardinal [25] proved that a triangulation can directly be transformed in to a Hamiltonian triangulation with less or at most equal amount of edge flips that is required to transform a triangulation to a 4-connected triangulation.

We start with some few definitions:

Definition 2.28 Interior Edge: *Given a plane graph $G = (V, E)$, if an edge is* *not on the outer face then it is called* Interior Edge*.*

Definition 2.29 Separating Triangle: *Given a triangulation $G = (V, E)$, if removal of a triangle disconnects one or more vertices from the graph, then it is called a* separating triangle*.*
In (Figure 2.10) abc is such a triangle, that deleting abc will separate w_1 from the graph.

Definition 2.30 Hamiltonian Triangulation: *If a triangulation contains a Hamiltonian cycle then it is called* Hamiltonian triangulation*. If the triangulation has n vertices then the length of the cycle is n.*

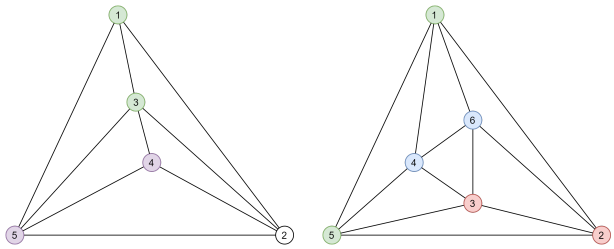


Figure 2.20: Matching

Definition 2.31 Matching: *Given a graph $G = (V, E)$, matching M is a set of edges such that no two edges in M are incident to the same vertex.*
In Figure 2.20 graph G_1 has 2 edges $M_1 = \{\{1, 3\}, \{5, 4\}\}$ which creates a matching for graph G_1. Here M_1 has 2 edges with 4 unique vertices. Note that if a matching of a graph have all the vertices then it is called Perfect Matching*. In Figure 2.20 matching of graph G_2 is edge set $M_2 = \{\{\ 1, 5\}, \{\ 6, 4\}, \{\ 3, 2\}\}$. Here all* 6 *vertices of G_2 is present in M_2. So M_2 is a perfect matching of G_2.*

Definition 2.32 Adjacent Face: *Given a graph $G = (V, E)$, if two of its faces share a common edge then they are called* adjacent faces*. In Figure 2.21 face A and B are adjacent as they have a common edge* $(3, 4)$*.*

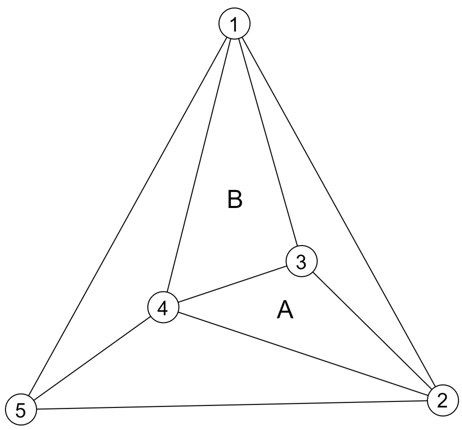


Figure 2.21: Adjacent Face

Definition 2.33 4-connected Graph: *Given a graph $G = (V, E)$ if it cannot be made disconnected by removing up to 3 vertices then the graph is called $4-$connected.*

Note that, as there exists no such case in 4-connected graph that removing 3 vertices will disconnect the graph, so in a 4-connected graph, there also does not exist any separating triangle.

Definition 2.34 Dual Graph: *Given a triangulation $G = (V, E)$, G' is a dual graph of G when every node of G' represent the faces of G and there is a connection between 2 nodes if their represented faces are adjacent to each other.In Figure 2.22 G'(Green colored) is the dual graph of graph G.*

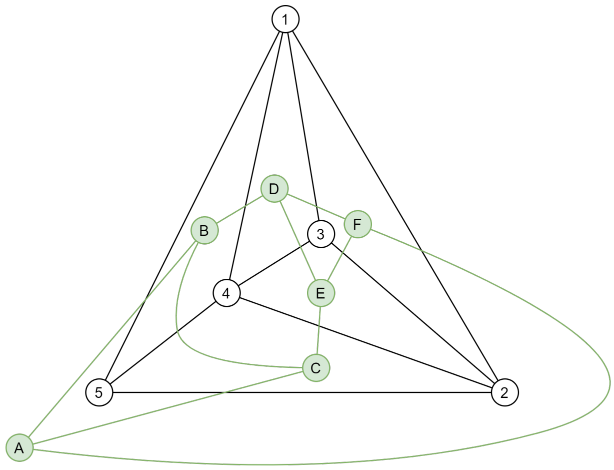


Figure 2.22: Dual Graph

The current best algorithm to transform a triangulation of n vertices into a canonical Δ_n using sequential flips showed that, using 2 steps all triangulation is able to transformed into a canonical triangulation Δ. First of all, no more than $[(3n-9)/5]$ flips are required to get a 4-connected triangulation and after that additionally no more than $2n - 15$ flips are required to transform a 4-connected to a canonical triangulation [25]. Altogether on the diameter of flip graph it has an upper bound of $5.2n - 33.6$ [23]. The upper bound for the second state is tight. So Cardinal focused on the first step. Cardinal at al. [25] proved that, a shorter number of flips are required to assure a Hamiltonian triangulation rather than a 4-connected one. In order to do so, at first he introduced dummy flip. And then he some certain cases, he used this dummy flip to reduce 3 normal flips to 2 normal flips.

Definition 2.35 Dummy Flip: *Given a triangulation G with n vertices where $n \geq 4$, let T be a facial triangle. To make a dummy flip, we will put a dummy vertex v inside of T. Then with v, we will connect all the vertices of T. We will flip every edge of T. As a result the degree of v will increase from 3 to 6. This operation is called Dummy Flip.*

In Figure 2.23 we can see a subgraph of a triangulation and $T = ace$ is a face. We added v in the interior of T and added av, cv, ev. Now we will flip all the edges of T$\{ac, ae, ce\}$ to complete the dummy flip operation.
Later we will see that using this dummy flip we can reduce the number of required flip.

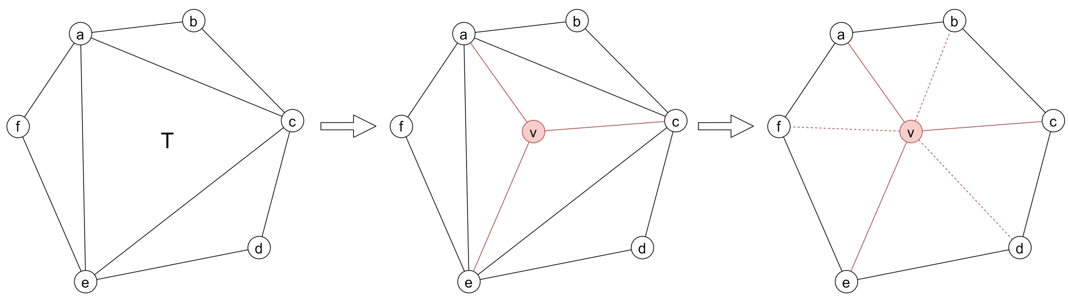


Figure 2.23: For $T = aec$, to make a dummy Flip, we added vertex v and flipped all the edges of T

4-Block Decomposition:

Definition 2.36 4-Block tree: *Given a triangulation $G = (V, E)$, we create a* 4-Block tree *in such way that every node of that tree represents a sub graph of G and every node is also 4-connected. Let g_1 and g_2 be two nodes of 4-block tree. If the outerface of g_2 is an interior face of g_1 then, g_1 is the parent of g_2.*

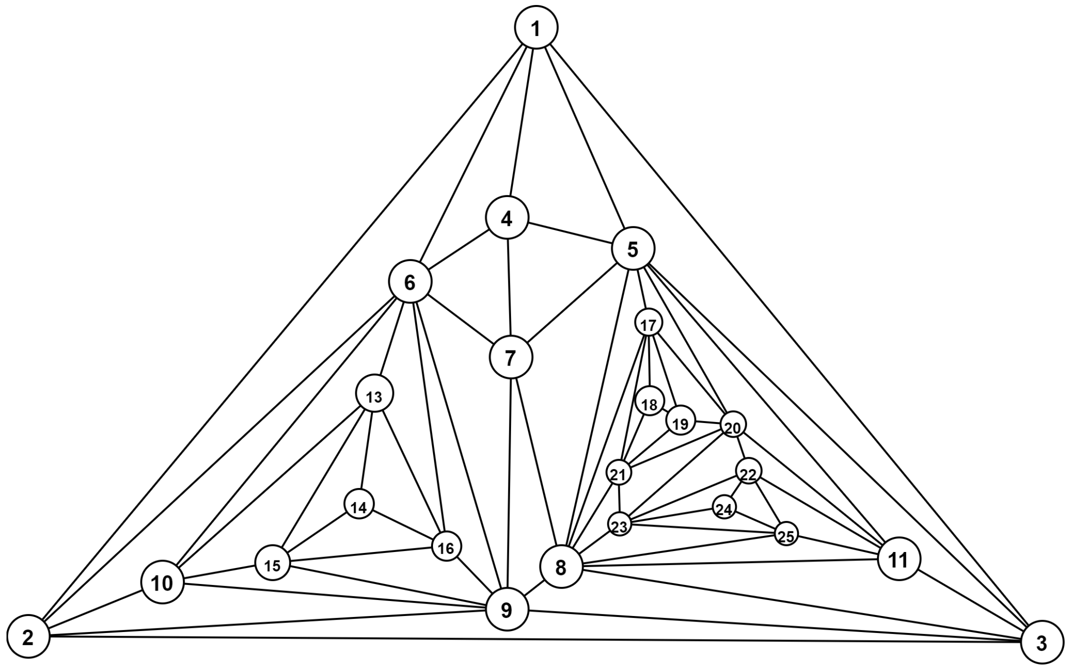


Figure 2.24: A Triangulation with some Separating Triangle.

We apply the 4-block decomposition on the original graph to obtain its corresponding 4-block tree. To achieve this we find the largest separating triangle of the triangulation, remove that triangle and make it a child of the current node. We will continue this process until every node becomes a 4 connected sub graph of the main graph. In Figure 2.24 we have a triangulation with many separating triangles. Then Figure 2.25, it shows how to remove all the separating triangle and add them in their child and get the 4-block tree B in Figure 2.27. Here we denote G_i as a node of 4-block tree, T_i is the outer face of G_i and ST_i is the list of all the separating triangle

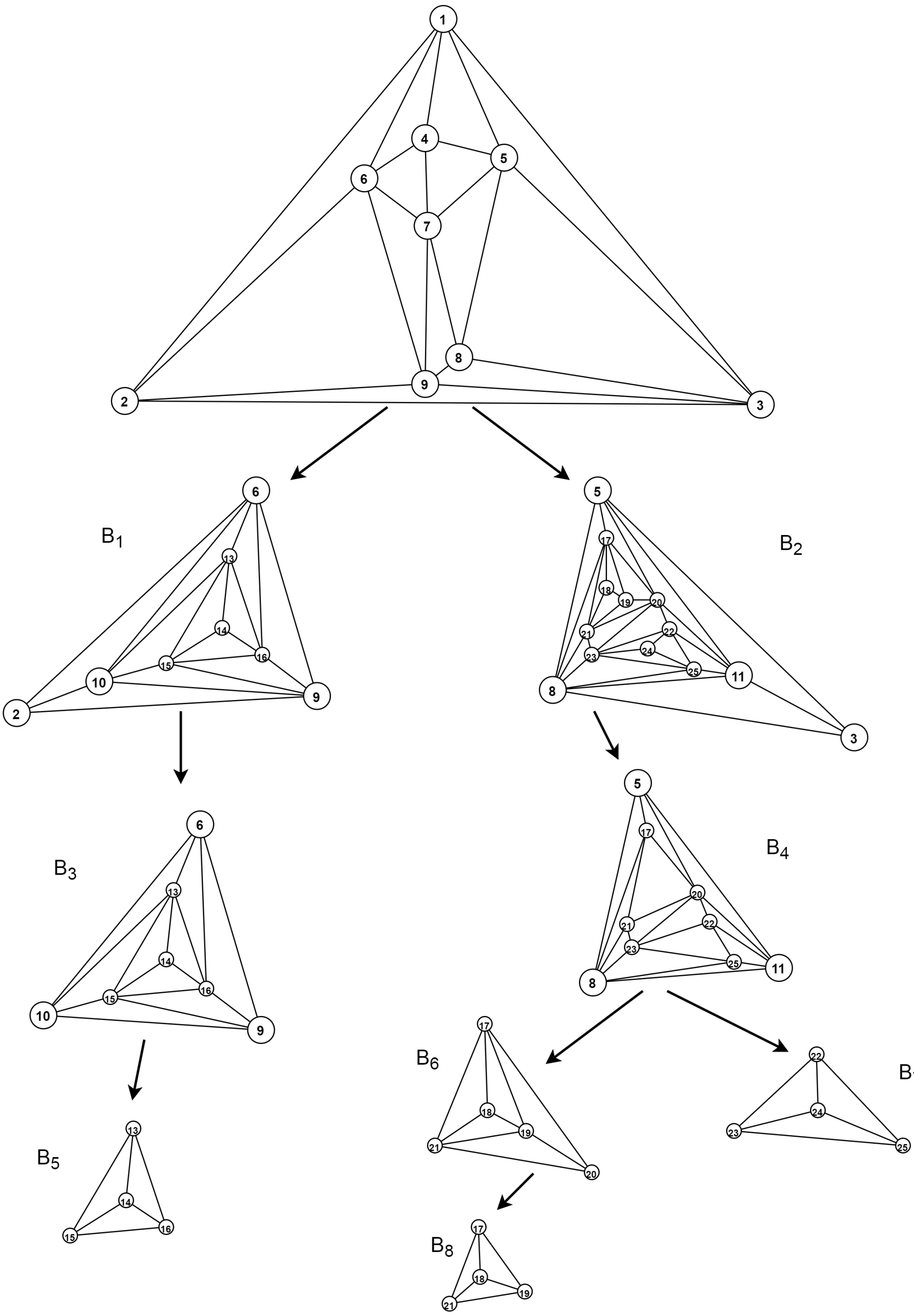


Figure 2.25: 4-Block Decomposition

inside of G_i.

Definition 2.37 Checkerboard: *Given a 4-block G_i from a 4-block tree B, if every interior edge of G_i belongs to exactly one separating triangle from ST_i then it creates a checkerboard.*

Algorithm
Note that in every checkerboard, there exist at least 1 face F such that, every edge of F is part of different separating triangles. There will also exist another face H such that H is adjacent to the outerface, but is not adjacent of F.

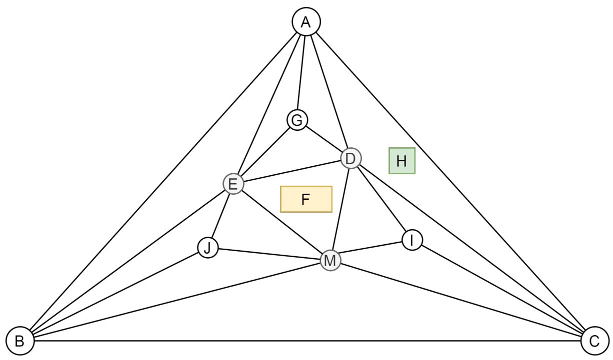


Figure 2.26: Checkerboard

At first we will choose a G_i at penultimate level (such a node which is not a leaf but every child of this node is a leaf). Now we will check if the triangles of ST_i forms a checkerboard or not. If it is not a checkerboard, then we need a 4-connector of G_i. 4-connector of G_i is a set of edges, that if we flip them then it will merge G_i with all of its child and there will be no separating triangle. Meaning it will make G_i a leaf. To find a 4-connector at first we need a dual matching of G_i(matching of dual graph of G_i) such that there exist exactly 1 edge from every triangle in ST_i. Now if we flip these edges it will merge all the child of G_i with G_i. If ST_i forms a checkerboard, then there will exist such a face f that all of its edges are part of different triangles of ST_i. Now we will do a dummy flip in f (this will be replaced by at most 2 normal flips later). We will also need the 4-connector of G_i, but we will remove 3 edges which are incident of 3 separating triangles adjacent of f. Now the child of G_i are merged with G_i and G_i is no longer at a penultimate level, in fact G_i is now a leaf. We will continue this process till all the nodes of 4-block tree B merge together and becomes a single node. And the resulted triangulation G' is a Hamiltonian triangulation.
With this dummy flip operation, we can avoid at least 1 extra flip. Usually for every separating triangles we need 1 edge flip. But this dummy flip will destroy 3 separating triangles at a time.

2.5.1 Remove dummy flip with normal flips

At this point we have a Hamiltonian triangulation G' obtained from G, but this triangulation has dummy vertices and we want to remove them. For that at first we

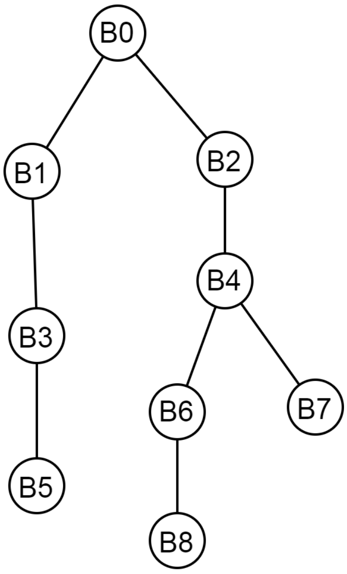


Figure 2.27: 4-Block Tree

will select a Hamiltonian cycle c' in G'. In c' There will always be two edges who are adjacent to vertex v (the new vertex we added for the dummy flip operation). Let uv and wv are such two edges. We can see that the dummy vertex can only connect with 6 vertices. So u and w have to be in those 6 vertices and they will create 3 cases based on their position. Our goal is to remove the dummy vertex and still have the Hamiltonian cycle. Here if we remove v and with some flips we can connect u and w then we can accomplish our goal.

Case 1:

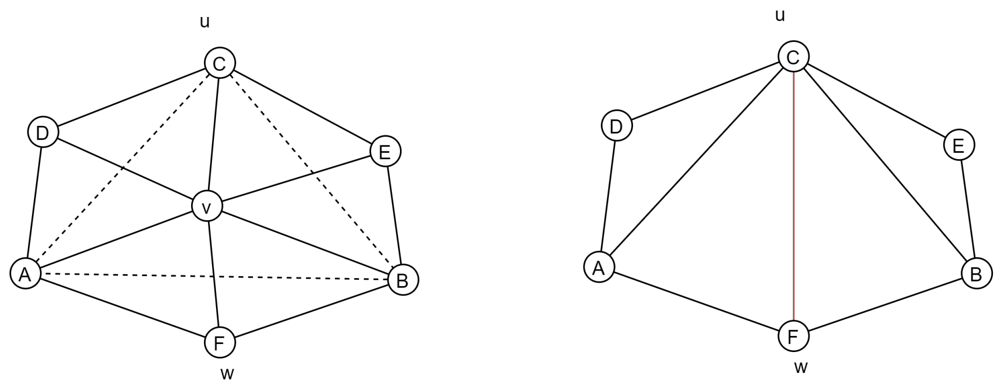


Figure 2.28: Remove dummy flip: Case 1

In Figure 2.28 u and w are in the opposite side (difference of 2 vertices). Here ABC is a facial triangle of G and dotted edges are the part of G. Notice that u and C are actually the same vertex. In this case either uv or wv will always intersect an edge of ABC. In Figure 2.27 edge AB and vw intersect. So in our main triangulation G we will flip the edge AB and it will connect v and w.

Case 2:

In Figure 2.29 u and w are adjacent to each other. In this case u and w are already connected. So there no flip is required in this case.

Case 3:

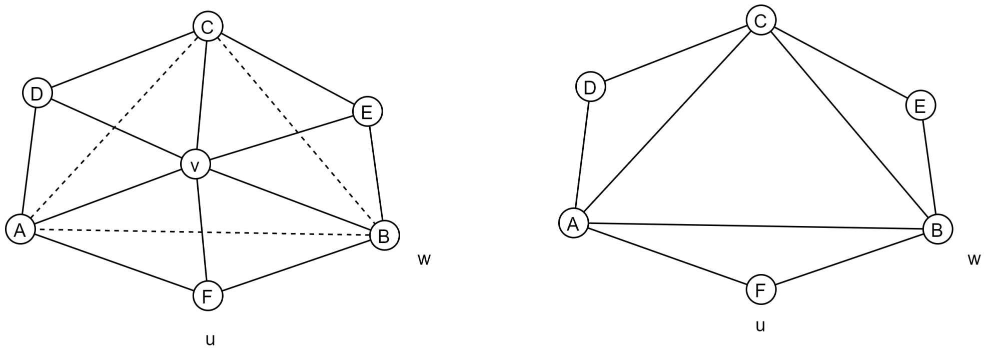


Figure 2.29: Remove dummy flip: Case 2

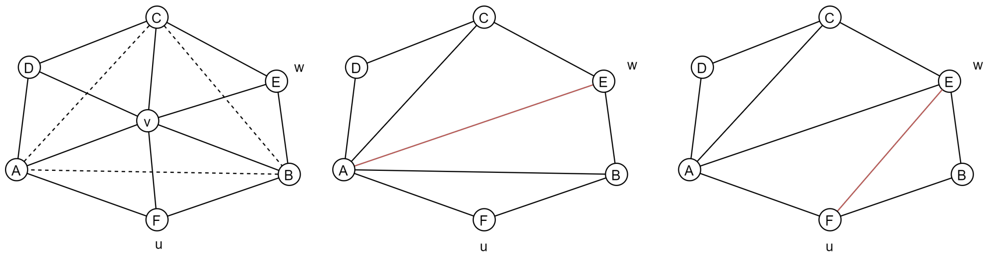


Figure 2.30: Remove dummy flip: Case 3

In Figure 2.30 u and w are 1 edge away from each other. If $u \in ABC$ and $v \in ABC$ then they are already connected. If not, then uv and wv each will intersect 1 edge from ABC. We just need to flip those edges sequentially to connect u and w.

Now by doing all these operations, we can get a Hamiltonian triangulation where separating triangulation does not exists. Then according to Mori, Nakamoto and Ota [19], Hamiltonian Triangulation can be transformed into a caninical triangulation by at most max $2n - 10, 0$ flips. Also if a Graph G is 4-connected then at most max $2n - 11, 0$ flips needed.

Previously we described these steps in the Mori et al.'s bound. Here in [19], they talked about a term called *Apex*. A vertex is called apex if it is connected to all the other vertices of a graph. Using Hamiltonian cycle, Graph G decomposed into two sub graph $G1$ and $G2$. In $G1$ we try to find a vertex which has degree 2 and name it v. Then in Graph $G2$, the degree of v has to be $deg(v) \geq 3$ by the 3-connectedness of graph.

If there is a triangular face vxy in $G2$ with xy, then xy can be switched into vz in the quadrilateral $vxyz$ formed by two triangular faces vxy and xyz in $G2$, without breaking the simpleness of the graph. Now $G2$ can be transformed into maximal outer plane graph with apex v by at most $n-4$ diagonal flip. Then from $G1$, we will remove vertex v. The maximum outer planer graph $G1$ will thus have precisely $n-1$ vertices. Then we can transform this Triangulation into a Canonical Triangulation. So these are the algorithms we have covered so far. And the time complexity of the algorithms is:

Algorithm	Complexity
Wagner's algorithm	$4n^2 - 28n + 48$
Kumoru's algorithm	$4n - 4 - d_G(a,b)$
Mori's algorithm	$6n - 30$
Cardinal's algorithm	$5n - 23$

Figure 2.31: Time Complexity Table

Chapter 3

Implementation

3.1 Wagner's algorithm implementation

The notion of transforming a triangulation into a canonical triangulation comes from Wagner's Theorem. Wagner's algorithm requires $\mathcal{O}(n^2)$ edge flips to transform a triangulation into a Canonical triangulation.

Our full implementation can be found in GitHub at https://github.com/mukit136337/Transform-a-triangulation-with-edge-flips link which has a piece of code that contains 2 classes. We have implemented this by using SageMath and python as programming language. In the code one is triangulation class that works like a data structure. It creates and stores triangulation. We need to pass a cyclic list and external face to its constructor and it will create all the necessary properties. For example in a Triangulation object we store all the edges, faces also edge to face map (stores which edge is connected to which faces) etc. For the cyclic list, we used a map where every vertex is mapped to a list which represent the cyclic order of all the vertices in counter clockwise. For external face, any edge will work, and the given edge will be in the outerface.

```
c1 = {0:[6,4,5,1], 1:[0,5,2,6], 2:[5,4,3,6,1], 3:[4,6,2], 4:[5,0,6,3,2], 5:[0,4,2,1], 6:[1,2,3,4,0]}
external_face1 = (6,1)
t1 = Triangulation(c1,external_face1)
t1.g.show()

c2 = {0:[2,4,3,1], 1:[0,3,5,6,2], 2:[1,6,4,0], 3:[4,5,1,0], 4:[2,6,5,3,0], 5:[4,6,1,3,0], 6:[4,2,1,5]}
external_face2 = (1,2)
t2 = Triangulation(c2,external_face2)
t2.g.show()
```

Figure 3.1: Driver Code

In the figure 3.1 we have given a example of the driver code and in figure 3.2 shows how our triangulation graph looks like.

Our another class contains the implementation of Wagner's theorem. Here 2 triangulations are passed in the constructor (t_1 and t_2). The method "show_transformation()" shows a list of flips that is required to transform t_1 into t_2 using Wagner's algorithm. In figure 3.2 shows the required flips using Wagner's algorithm. Here we can see

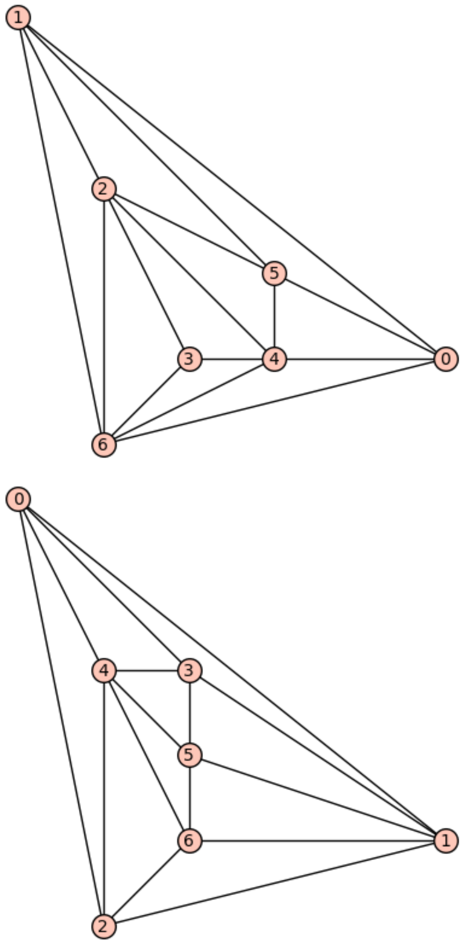


Figure 3.2: Output: Showing 2 triangulations

```
w = Wagner(t1, t2)
w.show_transformation()

```

```
[[0, 4], [2, 5], [2, 4], [0, 2], [3, 4], [4, 5]]
```

Figure 3.3: Flip Sequence using Wagner's theorem

that 6 flips were required for transforming t_1 into t_2. But according to Wagner's theorem the upper bound is $2n^2 - 14n + 24 = 24$, however in practical we need only 6 flips.

3.2 Identifying and removing Separation Triangulation

Identifying and removing separating triangulation is a very important part of our thesis. For identifying and removing separating triangulation, we have reviewed some theorems. Then we came to our own conclusion which is kind of similar to H. De Fraysseix, J. pach and R. Pollack's theorem for identifying the separating triangulation and using Cardinal et al. [25] theorem we have removed the separating triangulation.

In the De fraysseix,J Pach, and R. Pollack's theorem they mainly talked about fáry embedding. They demonstrate that on the $2n - 4$ by $n - 2$ grid, any planar graph having n vertices got a straight-line embedding or Fáry embedding, and they provide an O(n) space, O(n log n) time approach to achieve this embedding. It was unknown in the past whether somebody can always find a polynomial-sized grid to accommodate such an embedding. On the contrary, they prove that every set F with cardinality at least $n + (1 - o(1))\sqrt{n}$ may support a Fáry embedding of any planar graph of size n, which solves a Mohar issue.

Theorem 1. Any plane graph with n vertices has a Fáry embedding on the 2n-4 by n-2 grid.

According to the theorem of I. Fáry [3] in the *Fáry embedding*, the points in the plane are the vertices and straight line segments are the edges. At first in the paper of Tutte, there have been numerous algorithms presented for constructing a Fáry embedding [4], [8], [10]. However, all of these publications have certain flaws, such as it requiring high precision real arithmetic in relation to the graph's size, and vertices prefer to pack together with the idea that the ratio of the smallest to the biggest distance is unnecessarily little.
Also, it is not clear that every planar graph of size n has a Fáry embedding on a grid of side length bounded by n^k for some fixed k. These Questions is related about how compactly graphs can be embedded on grids are related to the problems of VLSI layout design ([7], [9], [6]). Theorem 1 of De fraysseix, J. Pach and R. Pollack's gives an good answer to this question and its proof provides an algorithm constructing such an embedding.

So these are the main ideas of De Fraysseix, J. Pach and R. Pollack's theorem.

In our algorithm implementation, we used same kind of implementation like DE FRAYSSEIX, J. PACH and R. POLLACK's theorem for identifying separating triangulation.

Here in figure 3.4, we can see a Triangulation where ABC is the outer face. For

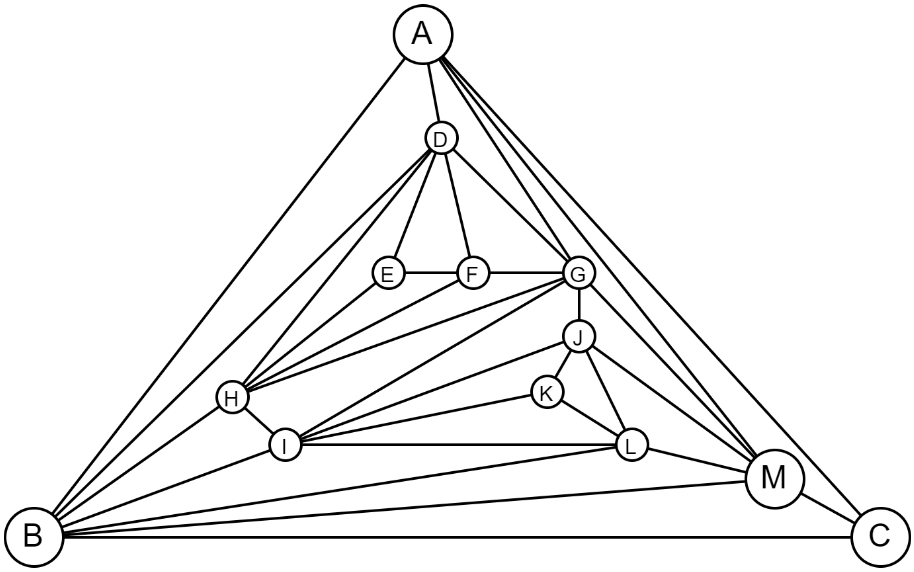


Figure 3.4: Identifying Separating Triangulation

vertex A, the cyclic order is (C, M, G, D, B) where B and C is in the outer face. For identifying the separating triangulation, we will ignore the outer vertices for this case. So now we are considering M, G, D vertices. Cyclic order for vertex M is (C, B, L, J, G, A), for vertex G is (M, J, I, H, F, D, A) and for vertex D is (G, F, E, H, B, A). Now we can see only vertex M has connection with all the outer vertices. And in the cyclic order of A vertex, vertex B and vertex M has a edge connection. So ABM creates a separating triangle in the ABC Triangulation.

After identifying the separating triangles, we can easily remove them by following the steps explained in 2.5.1.

3.3 Transforming Non Hamiltonian into Hamiltonian

In 2.5, we discussed we can make a non Hamiltonian Triangulation a Hamiltonian Triangulation without making the triangulation 4-connected. This improved the minimum required flip from $(3n - 9)/5$ to $5n - 23$. Figure 3.5 shows the flow chart of how the code segment works for this transformation. We used recursively 4-Block Decomposition and created a 4-Block Tree. Here B_0 is the root which has the same outerface of the main triangulation. We will run a main loop, where we will take an arbitrary node G_i from the 4-Block Tree at the penultimate level. Then we will get the 4-connector of G_i, and flipping every edge of the 4-connector will merge G_i with all of its child. It means G_i has no more separating triangles in it. We will update the 4-Block Tree and in the updated 4-Block Tree, G_i will become a leaf node. There is a chance that G_i might form a checkerboard. In that case we put a dummy vertex on face F and perform a dummy flip which destroys 3 separating triangles with at most 2 normal flips. If a checkerboard appears we delete 3 edges in 4-connectors which are responsible for breaking the separating triangles adjacent to face f. This loop will continue until all the child are merged to B_0 and in the

4-Block Tree there exists only one node. Then the root B_0 is a triangulation which is a Hamiltonian Triangulation.

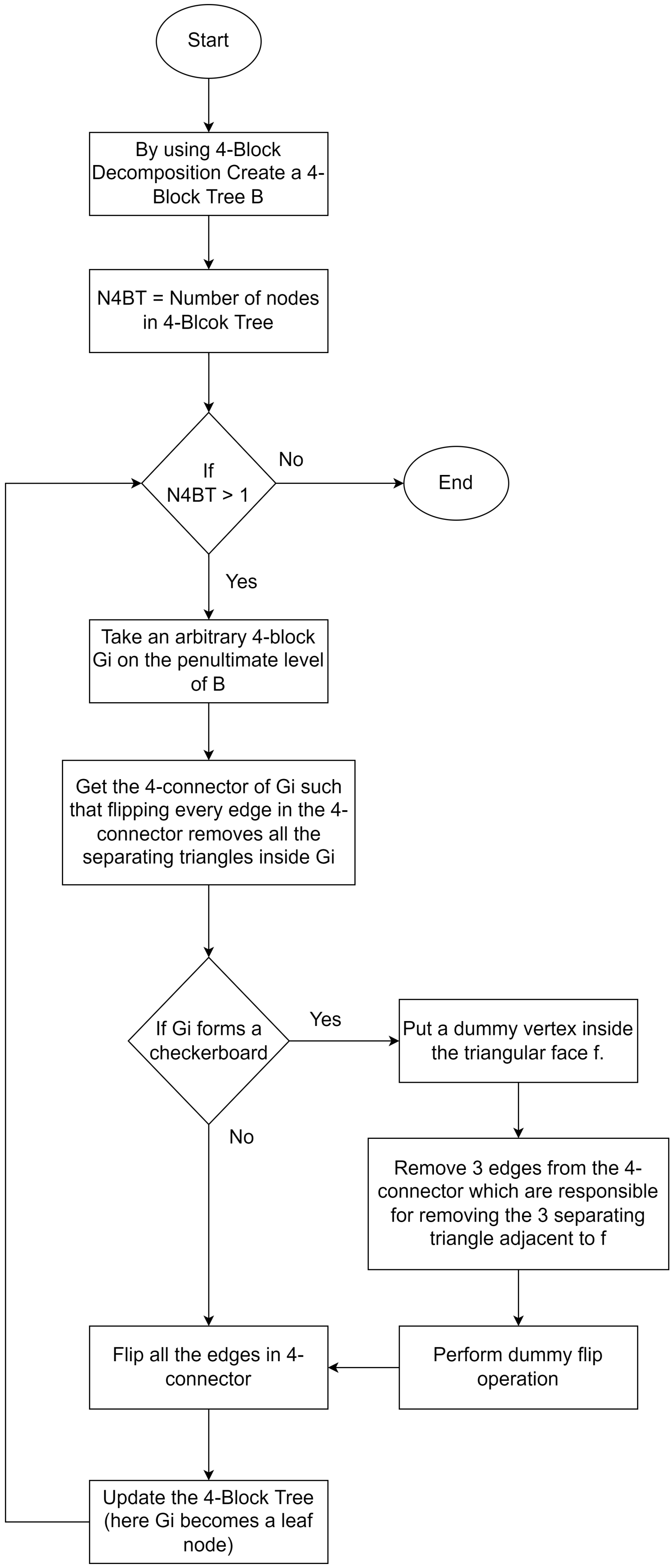


Figure 3.5: Flow Chart for Transforming non Hamiltonian triangulation to Hamiltonian Triangulation

3.4 Transforming Hamiltonian triangulation into Canonical Triangulation

In Section 2.4, we discussed using Hamiltonian cycle we can transform a triangulation to Canonical triangulation. Figure 3.6 shows the flow chart of how the code segment works. We decomposed The graph G into 2 sub-graph $G1$ and $G2$ considering HC (Hamiltonian Cycle) as the outer cycle. On each sub-graph we created an apex vertex v. As the sub-graphs were strictly divided by the HC, a flip on sub-graph $G1$ will not change anything on sub-graph $G2$ and vice versa. By using this algorithm we can a Hamiltonian Triangulation to a Canonical Triangulation by at most $2n - 10$ flips [19].

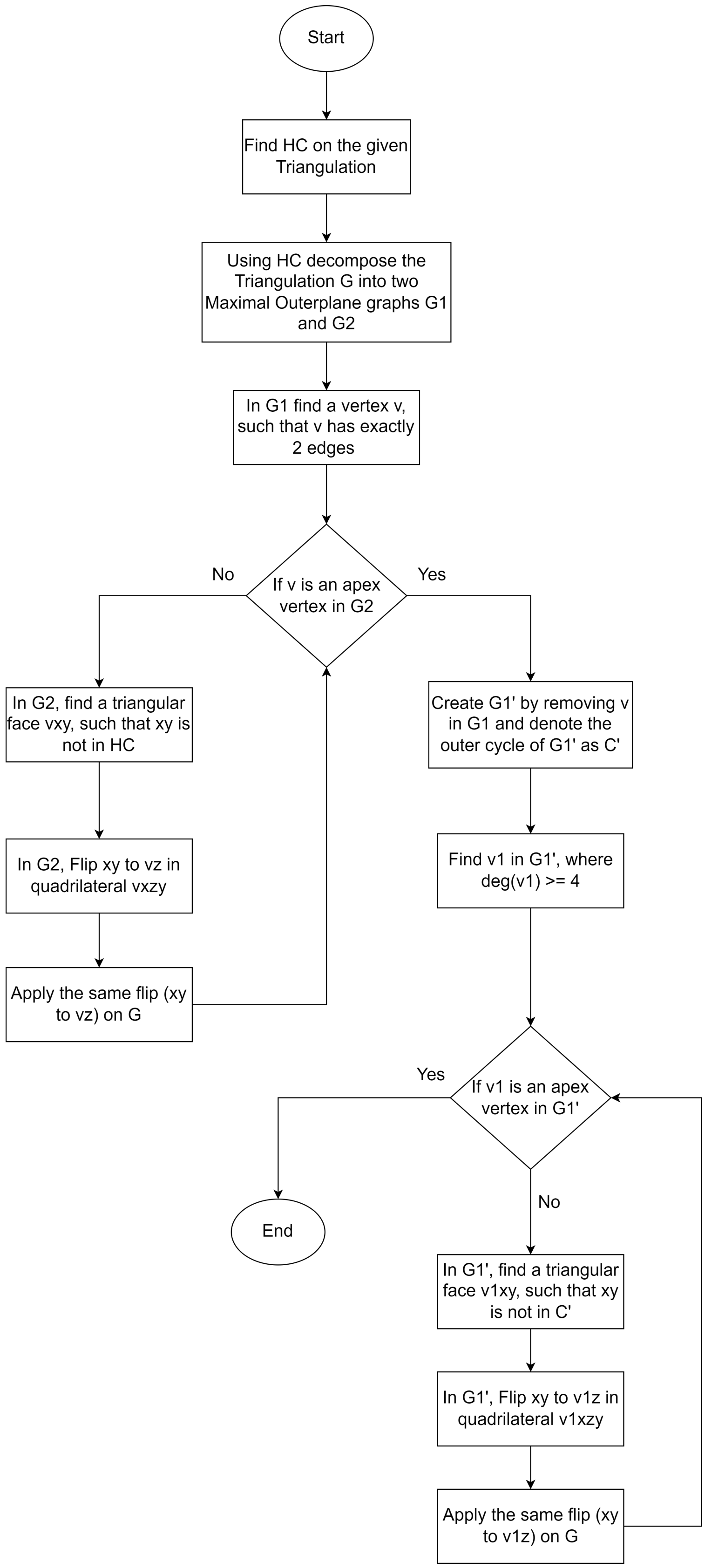


Figure 3.6: Flow Chart for Transforming Hamiltonian triangulation to Canonical Triangulation

Chapter 4

Result

There are many algorithms on transforming one triangulation into another with edge flips in Graph Theory. But there are no readily available implementation for investigating the performance of these algorithms for transforming different triangulation.While the existing algorithms talk about the worst case, not much information is available on the number of flips needed for the average case. So, for this reason we showed the following in this thesis:

- We implemented the Wagner's algorithm and showed the flip sequence that is generated by Wagner's algorithm
- We implemented the current best known algorithm in sequential flips (cardinal) for transforming triangulation.
- We want to observe the number of flips required for this algorithms in average case to transform one triangulation into another.

So far we have learned some of the famous algorithms which use sequential edge flip to transform one triangulation to any other triangulation of the same number of vertices. We learned the following algorithms:

- Wagner's algorithm
- Komuro's algorithm
- Mori et al.'s algorithm
- Cardinal et al's algorithm

We described our own thoughts on these algorithms in Chapter 2. In order to implement the algorithms, we used SageMath application. In SageMath we are using python to write our code. We have implemented a class to represent a triangulation.SageMath has a built-in function for finding the right vertices' positions whilst the graph can be a planar graph. We need to provide a clockwise cyclic order and external-face(outer-face) and the class will plot an embedding of the triangulation and store it in a global variable. But the most important part of this class is the function which can flip an edge. This function checks if the edge is flippable or not and if it is flippable, then this function flips the edge and updates all the related information. This function is very crucial as every algorithm will use this function. So we are trying to make this function as optimal as possible.

We have implemented Wagner's algorithm. We can successfully follow the steps of wagner's algorithm and show the sequence of edge flip that is required to transform one triangulation to another.
We also worked on the algorithm of Cardinal et al. [25] which is known as the best algorithm for using sequential edge flips.
An important task for our thesis is to compute the sequence of flips for the algorithm. We also want to heuristically measure the performance (quantity of flips required) of different random triangulation transformations from one to another. Then we want to analyze the data and show how much better it performs on the average case than the theoretical worst case.

In Figure 4.2 the line graph shows the required flips for worst case and average case. Here X-axis is the number of vertices (n) and in the Y-axis we put the required flips to convert a n-vertices Triangulation to another n-vertices Triangulation.
In Figure 4.1 we showed the data set we got (form 7 to 20). For the worst case we computed the required flip by the equation $5n - 23$. And for the last column we showed the average flips required in the 1000 iteration.

1	No. of Vertices	No. of flips on Worst Case	No. of flips on Average Case(1000)
2	7	12	5.23
3	8	17	8.58
4	9	22	12.45
5	10	27	16.18
6	11	32	20
7	12	37	23.94
8	13	42	27.86
9	14	47	31.74
10	15	52	35.49
11	16	57	39.29
12	17	62	43.19
13	18	67	47.41
14	19	72	51.14
15	20	77	55.04

Figure 4.1: Worst Case VS Average Case Table

In order to get the result of average case, we run 1000 iteration on every single vertex(7 to 60). And for every iteration, we generated 2 random triangulation and transformed one into another. We stored the numbers of required flip for every transformation and then we compute the average required flip for every vertex.

In Figure 4.2 green line is a straight line which represent how much flips we need in the worst case. And the orange line represent the required flips on average case. Notice that although the orange line segments together appears as a straight line,

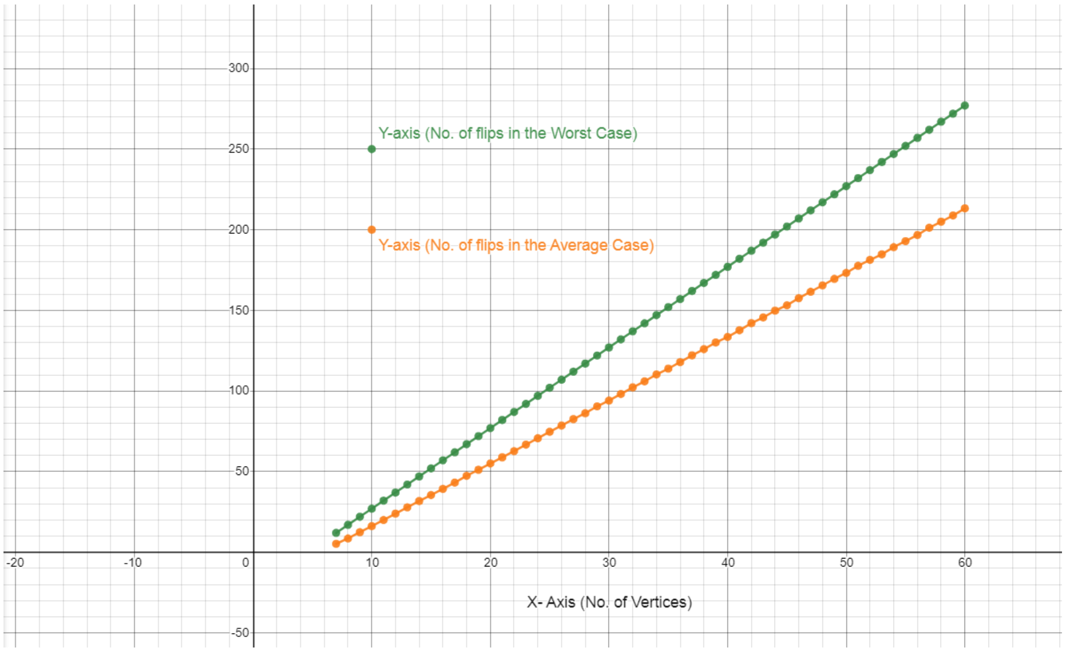


Figure 4.2: Line Graph of Worst Case VS Average Case

it is not. That is because we used 1000 iteration for every vertices, as a result the increase of required flip for n to n+1 is almost equal for every vertex. Orange line is strictly below green line, as it is supposed to because, average is always less than worse case.

Chapter 5

Analysis

In our thesis, we are mainly implementing some algorithms to transform one triangulation into another. In this process, we have learned about different algorithms which we reviewed in Chapter 2. We have implemented Wagner's algorithm so far and now we are in the process of implementing the algorithm of Cardinal et al. [25]. Also, we have gone through some papers [5], [13], [15] to see the run-time of the implementation of the algorithm 4-Connected graph, duel perfect matching, 4 block tree. While studying their algorithm, we noticed the following observation:

Observation 5.1 *According to Cardinal et al., in triangulation we can remove all the separating triangles in two ways. They are Dummy flip operation and another one is 4-Connector. We can only perform Dummy flip operation when there is a checkerboard in the triangulation. Otherwise we will perform 4-connector. But we observed that in some cases, we can perform dummy flip operation even if the triangulation is not a checkerboard. And it performs better than the 4-connector.*

Although we still do not know if this will decrease the upper bound, but the chances of using dummy flip operation will surely increase as we can use dummy flip operation without having a Checkerboard. Here is an example for better understanding.

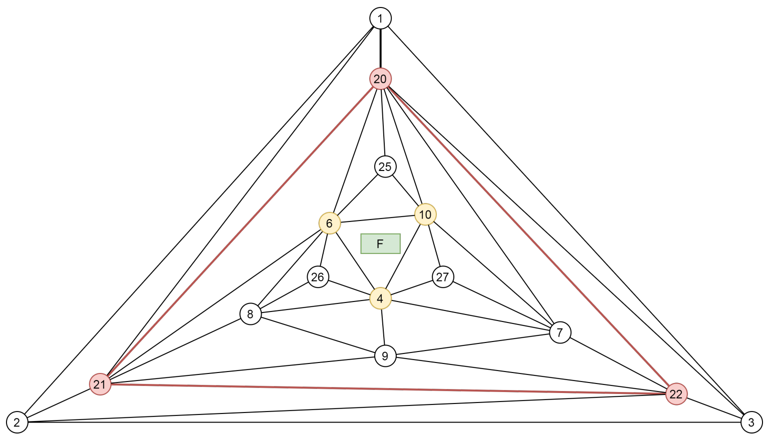


Figure 5.1: A Triangulation without Checkerboard

According to the Cardinal et al.'s theorem if a triangulation has a Checkerboard only then we can perform the Dummy flip operation to remove all the separating triangles. Otherwise we will perform 4-connector. In Figure 5.1, it shows a Triangulation which has no such 4-block that contains a Checkerboard. So we are performing 4-connector in every step according to the Cardinal's theorem.

In Figure 5.2, we selected a 4-block {20,21,22} from the triangulation which con-

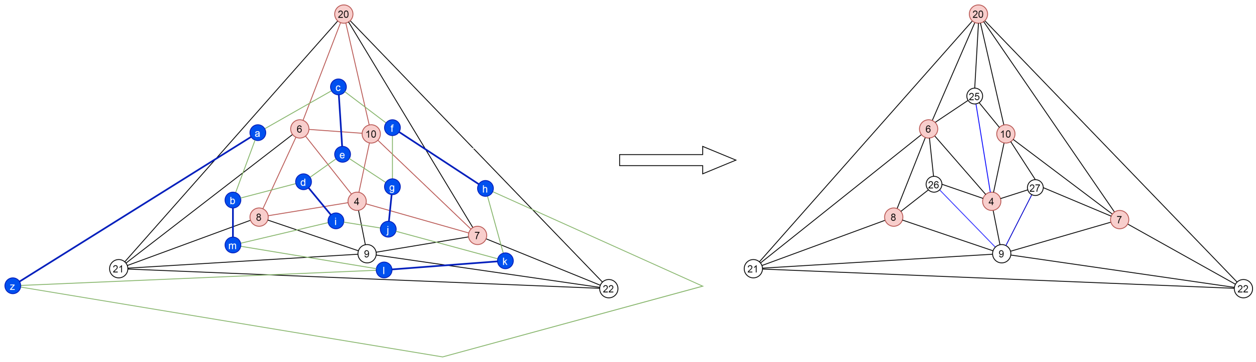


Figure 5.2: 4-Connector

tains all the separating triangles. From this 4-block we find {{a,z}, {b,m}, {d,i}, {c,e}, {g,j}, {f,h}, {l,k}} as dual matching and we will flip the edges of the separating triangles which contains the dual matching. So we flip {{8,4}, {6,10}, {4,7}} edges to remove all the separating triangles. So it clearly shows that 4-connector will require 3 edge flips to remove all the separating triangles. But if we use dummy

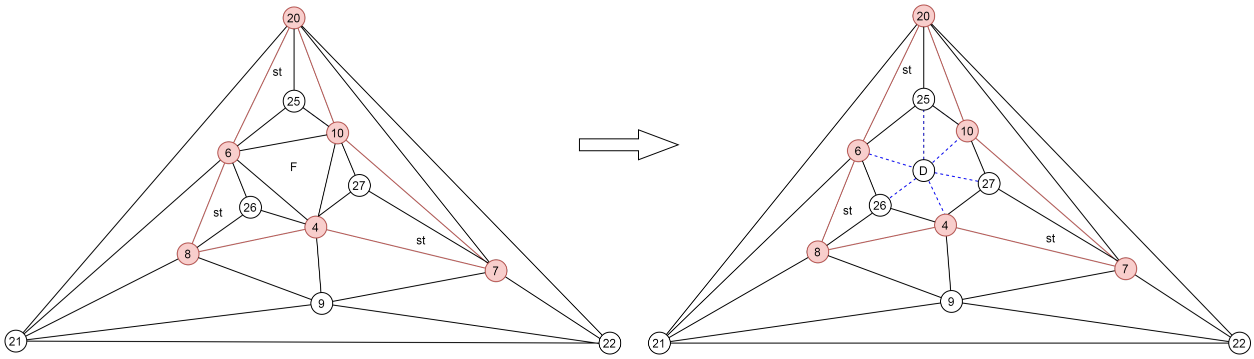


Figure 5.3: Dummy Flip Operation

flip operation in Figure 5.3, according to 2.5.1 these 3 separating triangles can be removed by at most 2 flips.

Chapter 6

Conclusion

Edge flipping is one of the very important topics in graph theory. We have reviewed Wagner's theorem, Komuro's bound, Mori et al.'s theorem and Cardinal et al's thorem in this research so far. Using edge flips, here we have showed the simulation of Wagner's theorem in Sagemath as there is no step by step simulation of the process. Also there is no such instances where it shows what is the actual cost(required edge flip) for the transformation of one triangulation into another. So, we computed how many flips it actually requires for the transformation and leave a clear difference between practical result and academical upper bound of edge flips. We have also implemented Cardinal et al.'s algorithm. Thus, it will help the researchers, students and others (who are interested in these field) to learn about these algorithm perfectly and can have better visualization. it also shows how the algorithm works.
In the future, if more vertices can be simulated then we might find a better result. The fact that, the occurrence of a Triangulation without Hamiltonian cycle is very low in lower number of vertices the cardinal's algorithm almost performs at $4n - 20$ for a triangulation of n vertices. So if further simulation can be done on more than 100 vertices which may increase the chance of appearance of non Hamiltonian triangulation. Another possible scope for further work can be comparing multiple algorithms average case. We only simulated the current best known algorithm and showed the difference. But other algorithms simulation might have a different result where the average case is much closer the the worst case or even much further from the worst case. This leaves an open field for further research.

References

[1] H. Whitney, "A theorem on graphs," *Annals of Mathematics*, pp. 378–390, 1931.

[2] K. Wagner, "Bemerkungen zum vierfarbenproblem," *Jahresbericht der Deutschen Mathematiker-Vereinigung*, vol. 46, pp. 26–32, 1936.

[3] F. István, "On straight-line representation of planar graphs," *Acta scientiarum mathematicarum*, vol. 11, no. 229-233, p. 2, 1948.

[4] W. T. Tutte, "How to draw a graph," *Proceedings of the London Mathematical Society*, vol. 3, no. 1, pp. 743–767, 1963.

[5] J. A. Bondy, U. S. R. Murty, *et al.*, *Graph theory with applications*. Macmillan London, 1976, vol. 290.

[6] L. G. Valiant, "Universality considerations in vlsi circuits," *IEEE Transactions on Computers*, vol. 100, no. 2, pp. 135–140, 1981.

[7] F. T. Leighton, *Complexity issues in VLSI: optimal layouts for the shuffle-exchange graph and other networks.* MIT press, 1983.

[8] N. Chiba, "Linear algorithms for convex drawings of planar graphs," *Progress in graph theory*, 1984.

[9] J. D. Ullman, "Computational aspects of vlsi computer science press," *Rockville, Md*, 1984.

[10] R. C. Read, *A new method for drawing a planar graph given the cyclic order of the edges at each vertex.* Faculty of Mathematics, University of Waterloo, 1986.

[11] D. D. Sleator, R. E. Tarjan, and W. P. Thurston, "Rotation distance, triangulations, and hyperbolic geometry," *Journal of the American Mathematical Society*, vol. 1, no. 3, pp. 647–681, 1988.

[12] R. Brunet, A. Nakamoto, and S. Negami, "Diagonal flips of triangulations on closed surfaces preserving specified properties," *journal of combinatorial theory, Series B*, vol. 68, no. 2, pp. 295–309, 1996.

[13] G. Kant, "A more compact visibility representation," *International Journal of Computational Geometry & Applications*, vol. 7, no. 03, pp. 197–210, 1997.

[14] H. Komuro *et al.*, "The diagonal flips of triangulations on the sphere," 1997.

[15] N. Robertson, D. Sanders, P. Seymour, and R. Thomas, "The four-colour theorem," *journal of combinatorial theory, Series B*, vol. 70, no. 1, pp. 2–44, 1997.

[16] F. Hurtado, M. Noy, and J. Urrutia, "Flipping edges in triangulations," *Discrete & Computational Geometry*, vol. 22, no. 3, pp. 333–346, 1999.

[17] T. Nakamigawa, "A generalization of diagonal flips in a convex polygon," *Theoretical computer science*, vol. 235, no. 2, pp. 271–282, 2000.

[18] Z. Gao, J. Urrutia, and J. Wang, "Diagonal flips in labelled planar triangulations," *Graphs and Combinatorics*, vol. 17, no. 4, pp. 647–657, 2001.

[19] R. Mori, A. Nakamoto, and K. Ota, "Diagonal flips in hamiltonian triangulations on the sphere," *Graphs and Combinatorics*, vol. 19, no. 3, pp. 413–418, 2003.

[20] P. Bose, J. Czyzowicz, Z. Gao, P. Morin, and D. R. Wood, "Simultaneous diagonal flips in plane triangulations," *Journal of Graph Theory*, vol. 54, no. 4, pp. 307–330, 2007.

[21] P. Bose and S. Verdonschot, "A history of flips in combinatorial triangulations," in *Spanish Meeting on Computational Geometry*, Springer, 2011, pp. 29–44.

[22] D. L. Souvaine, C. D. Tóth, and A. Winslow, "Simultaneously flippable edges in triangulations," in *Spanish Meeting on Computational Geometry*, Springer, 2011, pp. 138–145.

[23] P. Bose, D. Jansens, A. Van Renssen, M. Saumell, and S. Verdonschot, "Making triangulations 4-connected using flips," *Computational Geometry*, vol. 47, no. 2, pp. 187–197, 2014.

[24] A. Lubiw and V. Pathak, "Flip distance between two triangulations of a point set is np-complete," *Computational Geometry*, vol. 49, pp. 17–23, 2015.

[25] J. Cardinal, M. Hoffmann, V. Kusters, C. D. Tóth, and M. Wettstein, "Arc diagrams, flip distances, and hamiltonian triangulations," *Computational Geometry*, vol. 68, pp. 206–225, 2018.

[26] M. T. Kaykobad, "Transforming plane triangulations by simultaneous diagonal flips," Ph.D. dissertation, Université d'Ottawa/University of Ottawa, 2020.