12-2016

Embedding of hypercube graphs on orientable surfaces

Prateek Kunwar
University of Maine, prateek.kunwar@maine.edu

EMBEDDING OF HYPERCUBE GRAPHS ON ORIENTABLE SURFACES

By

Prateek Kunwar

B.E, Delhi College of Engineering, 2013

A THESIS

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Arts

(in Mathematics)

The Graduate School

The University of Maine

December 2016

Advisory Committee:

Dr. Robert Franzosa, Professor, Advisor

Dr. Andrew Knightly, Associate Professor

Dr. Benjamin Weiss, Assistant Professor

THESIS ACCEPTANCE STATEMENT

On behalf of the Graduate Committee for Prateek Kunwar, I affirm that this manuscript is the final and accepted thesis. Signatures of all committee members are on file with the Graduate School at the University of Maine, 42 Stodder Hall, Orono, Maine.

Dr. Robert Franzosa, Professor (Date)

LIBRARY RIGHTS STATEMENT

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at The University of Maine, I agree that the Library shall make it freely available for inspection. I further agree that permission for “fair use” copying of this thesis for scholarly purposes may be granted by the Librarian. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Prateek Kunwar (Date)

EMBEDDING OF HYPERCUBE GRAPHS ON ORIENTABLE SURFACES

By Prateek Kunwar

Thesis Advisor: Dr. Robert Franzosa

An Abstract of the Thesis Presented
in Partial Fulfillment of the Requirements for the
Degree of Master of Arts
(in Mathematics)
December 2016

The purpose of this thesis is to study hypercube graphs and their embeddings on orientable surfaces. We use rotation systems to represent these embeddings. We prove some results about the effect of adjacent switches in rotation system and create a rotation system called the ABC rotation system and prove general results about it. Using this rotation system, we give a general theorem about the minimal embedding of Q_n. We also look at some interesting types of maximal embedding of Q_n, such as the Eulerian walk embedding and the "big-face embedding". We prove a theorem that gives a recursively constructive way of obtaining the latter embedding in Q_n

ACKNOWLEDGEMENTS

I would firstly like to thank the University of Maine and the department of Mathematics and Statistics for giving me an opportunity to pursue my passion in mathematics by offering me admission into the master's program. I am very grateful to my advisor Dr. Robert Franzosa for not only guiding me through the mathematics but also for having great patience in editing and going through my written work and giving me continuous feedback and encouragement. I would also like to thank the members of my committee for their valuable input and support. Lastly, I thank my parents, roommates, friends and the universe for being how they are.

TABLE OF CONTENTS

Chapter

LIST OF FIGURES

Chapter 1
INTRODUCTION

1.1 History and Motivation

The city of Konigsberg in Prussia (present day Russia) was built such that there were two parts of the city on either side of the river Pregel. The parts of the city, which included two large islands, were connected to the mainland through seven bridges. A very persistent problem was to prove or disprove whether it was possible to traverse the city, crossing each bridge exactly once, under the condition that one could reach the islands only through a bridge (and not a boat for example) and that a bridge once accessed must be crossed(Figure1.1). Such a path is called an Eulerian walk, as we will see later . It was the genius of Leonhard Euler who settled this problem in 1736 by proving that it was impossible to do such a traverse, and that work is considered to have given birth to the branches of mathematics known as graph theory and topology. We will see later how this problem is concerned with walks on graphs and Eulerian circuits, important topics in modern-day graph theory

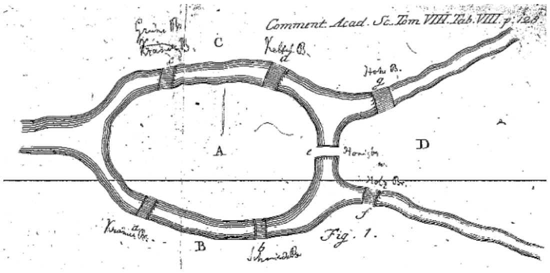


Figure 1.1. The Konigsberg Bridges

The Utility Problem asks whether it is possible in a plane to connect three houses to three supply stations (like gas, water and electricity) without using a third dimension and without the lines overlapping, as suggested in Figure 1.2. Unlike the Konigsberg bridges problem, the origin of the Utility problem is not known, although it has been around for a long time. Yet, quite similar to the Konigsberg Bridges Problem, it plays an important role in graph theory. This problem can be represented by what is known as the bi-partite graph $K_{3,3}$ (Figure 1.3) and the question is whether $K_{3,3}$ is planar (i.e , whether it can be located in the plane in such a way that the edges do not cross, unlike the case in Figure 1.3). With basic tools from graph theory and topology it can be proven that this graph is not planar. Hence, the utility layout cannot be done in the desired way. As it often happens in mathematics, this problem gives rise to more general questions such as "When is a graph planar?" and "If a graph is not planar, on what surface can such

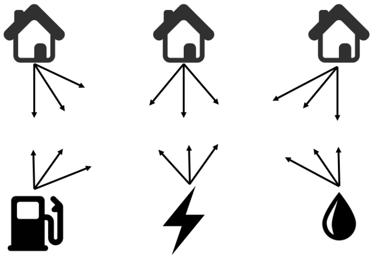


Figure 1.2. The utility problem

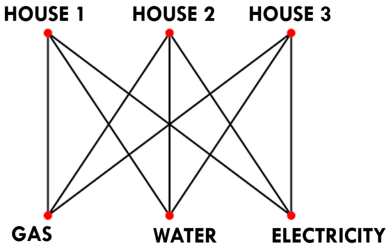


Figure 1.3. The bi - partite graph $K_{3,3}$

a layout be done?". It turns out that the surface on which the utility problem can be done is the torus (the surface of the familiar doughnut) as shown in Figure 1.4.

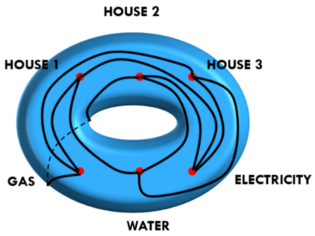


Figure 1.4. The utilities layout on a torus

With the above ideas in mind, this thesis is an exploration of the embeddings of a class of graphs, called hypercube graphs, on orientable surfaces. We will make all of these notions precise in what follows. Furthermore we will address ways of representing embeddings and how a graph can be maximally or minimally embedded on a surface. Before delving into these topics, however, we need to define some of the recurrent terms that will be used throughout, and we also need to make clear any conventions that will be followed.

1.2 Graph Theory

In this section we introduce the basic concepts of graph theory. The text Graph Theory [6] is a reference for the topics and results that we present in this section.

A **graph** is a set of **vertices** that are connected to each other by **edges**. Figure 1.5 illustrates the graph K_5, a graph known as the complete graph on five vertices (complete because each vertex is connected to every other vertex by a single edge). The **degree** of a vertex in a graph is defined as the number of edges that are incident on it. If there is a loop, it is counted twice; i.e if an edge connects a vertex

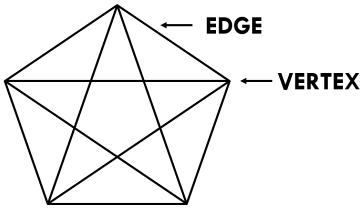


Figure 1.5. The complete graph K_5 with labeled edges and vertex

to itself, it is counted twice in the degree of the vertex. Figure 1.6 shows a graph with the degree of each vertex labeled.

If an edge in a graph connects distinct vertices, then we can assign a direction to the edge by indicating the vertices in order (i.e, a start vertex and an end vertex). A **directed edge** is an edge along with a direction assigned to it. In a graph, each

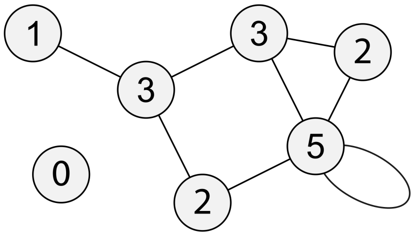


Figure 1.6. A Graph showing the degree of it's vertices

edge that is not a loop, has two directed edges associated to it.

A **regular graph** is a graph where every vertex has the same degree. If a graph has vertices of different degree, it is called a **non-regular** graph. In Figure 1.7, we have a regular graph with each vertex of degree 3. Adding a vertex to this graph at the center, connected to five other vertices as shown, makes it a non-regular graph because the new vertex has degree 5, different from the degree of all the other vertices. A **simple graph** is an graph containing no loops and no cases

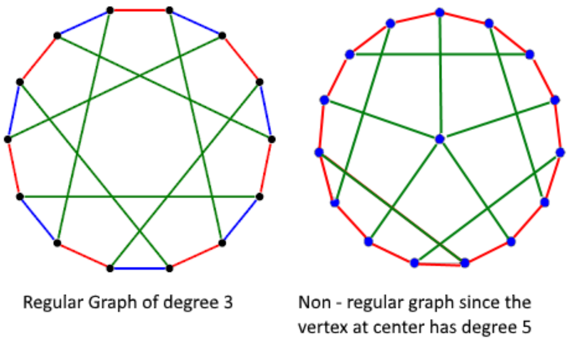

Regular Graph of degree 3
Non - regular graph since the vertex at center has degree 5

Figure 1.7. Examples of a regular graph and a non-regular graph

where two or more edges connect the same pair of vertices. In Figure 1.8 and Figure 1.9 , we can see an example of a graph that is simple and an example of a graph that is not, respectively.

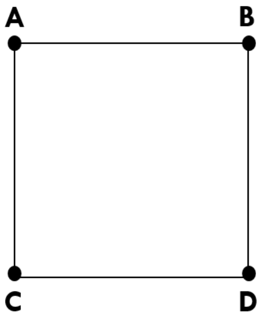


Figure 1.8. Example of a simple graph

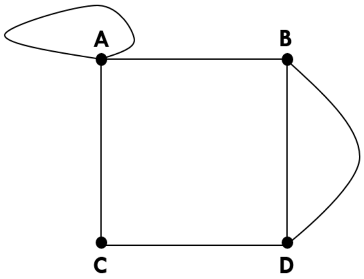


Figure 1.9. A graph that is not simple

A **path** in a graph is sequence of vertices and edges $(v_0, e_0, v_1, e_1, ...v_{n-1}, e_{n-1}, v_n)$ such that for each $j = 0, 1, 2, ...n - 1$, edge e_j connects vertices v_j and v_{j+1}. A **cycle** is a closed path, a path that ends where it begins. In the graph on the left in Figure 1.10, $1 \rightarrow 5 \rightarrow 8 \rightarrow 6$ is a path, and in the graph on the right $1 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 1$ is a path that is a cycle. A path that visits every edge in a graph exactly once is called an **Eulerian path**, and if the path is closed, then it is called a **Eulerian circuit**. In Figure 1.11 we have an Eulerian path that goes from vertex to vertex as follows :

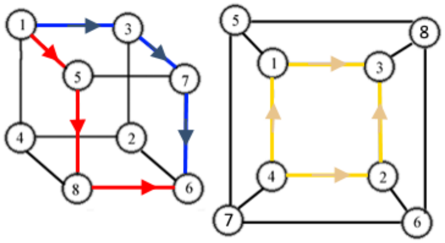


Figure 1.10. Examples of Paths and a Cycle

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1 \rightarrow 5 \rightarrow 2 \rightarrow 4 \rightarrow 5.$$

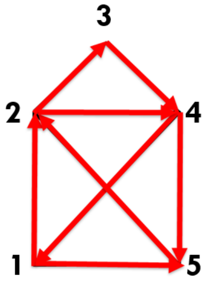


Figure 1.11. Example of an Eulerian Path

There are well known necessary and sufficient conditions for a graph to have an Eulerian path or an Eulerian circuit :

Theorem 1. *A graph has an Eulerian Circuit if and only if every vertex has even degree. A graph has an Eulerian path, but not an Eulerian circuit if exactly two vertices have an odd degree. In this case, the path begins and ends at these vertices.*

It is interesting to note that in the Konigsburg Bridges Problem, if we treat each region as a vertex and each bridge as an edge, we obtain the graph illustrated in Figure 1.12. The problem now reduces to finding an Eulerian path on this graph. By Theorem 1, it follows that an Eulerian path does not exist because more than two vertices have odd degree. Thus, there is no walk through Konigsberg crossing each bridge exactly once.

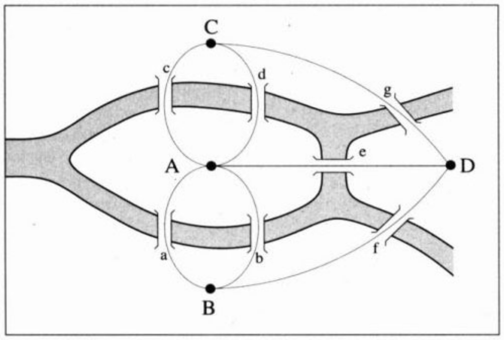


Figure 1.12. The Konigsburg Bridges Problem modeled by a graph

A **hypercube graph** Q_n is defined as the vertices and edges in an n - dimensional cube. The hypercube graph Q_n is a regular graph that has 2^n vertices, each with degree n, and has $n2^{n-1}$ edges. The vertices in Q_n can be expressed by binary strings of length n where connected vertices differ in their binary strings by exactly one bit. In Figure 1.13, we show Q_1, Q_2 and Q_3, the hypercube graphs in one, two and three dimensions, respectively, along with the binary strings that label the vertices.

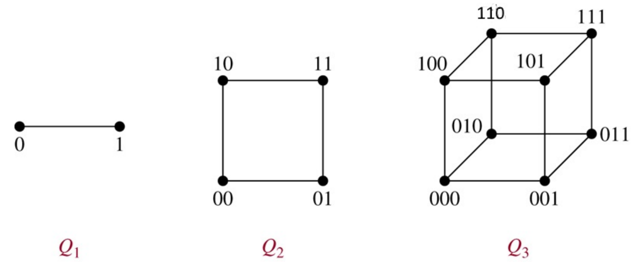


Figure 1.13. Hypercube graphs for $n = 1, 2, 3$

Since Hypercube graphs are simple graphs, for the remainder of the thesis we restrict our attention to simple graphs and assume that all graphs we consider are simple.

1.3 Topology Background

The theory of topological spaces and continuous functions is the main setting for the work in this thesis.The text Introduction to Topology: Pure and Applied [5] is a reference for the topology topics and results presented throughout this thesis.

The main notion of equivalence in topology is **homeomorphism**, a continuous bijective function between topological spaces $f : X \to Y$, having a continuous inverse. Two spaces are said to be topologically **equivalent** or **homeomorphic** if there is a homeomorphism between them.

An **embedding** from a topological space A to a topological space X is a continuous injective function $f : A \to X$ that is a homeomorphism onto it's image $f(A) \subset X$. We can think of an embedding as placing a copy of space A within space X. An illustration of an embedding of the cube on a sphere is shown in Figure 1.14.

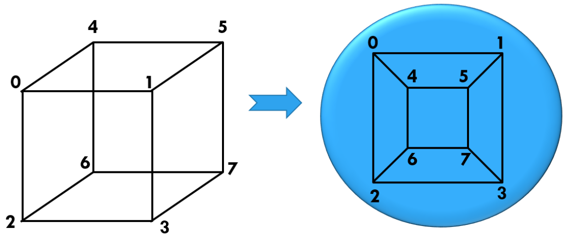


Figure 1.14. Example of an embedding of the cube on a sphere

The **open disk** in the plane plays an important role in what is to come. It is the set $\{(x,y)|x^2+y^2<1\}$.

The primary purpose of this paper is to investigate embeddings of the hypercube graphs on what are known as compact orientable surfaces. So, we now introduce some background on compact orientable surfaces. We will be skipping the detailed formal definitions, but will present important characterizing properties below.

Two important aspects of surfaces are that they are connected and locally homeomorphic to $\mathbf{R}^2$. That is, if you are on a surface, locally it looks like you are on a plane.

The compact orientable surfaces are the sphere and the n-hole tori, the latter of which can be thought of as surfaces of doughnuts with n holes, as illustrated in Figure 1.15. The **genus** of a compact orientable surface is essentially the number

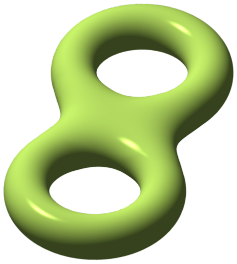


Figure 1.15. A two hole torus

of holes in it. Every compact orientable surface of genus $n \geq 1$ can be represented as a 4n-gon with edges glued in pairs, as illustrated in Figure 1.16 and Figure 1.17.

An important question that arises is whether given a collection of polygons whose edges altogether are glued in pairs, does a compact orientable surface always result?

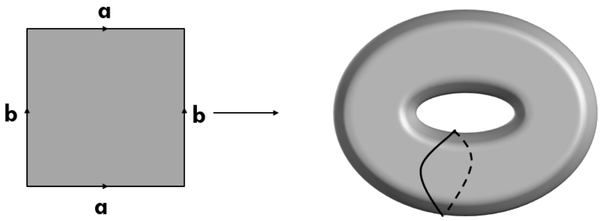


Figure 1.16. Folding a square into a torus

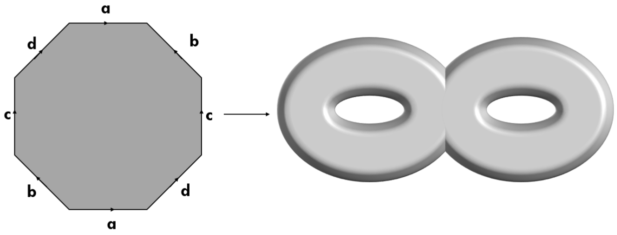


Figure 1.17. Folding an octagon into a two hole torus

It turns out that it does not necessarily happen. But, if the gluing is orientation preserving (a property we describe below), then compact orientable surfaces result.

Suppose we have a collection of polygons whose edges we glue in pairs (as shown in Figure 1.18). We can orient each polygon either clockwise or counterclockwise. If there is a choice of orientation on each polygon such that, considering each glued pair of edges, one edge's gluing direction is in the direction of orientation on its

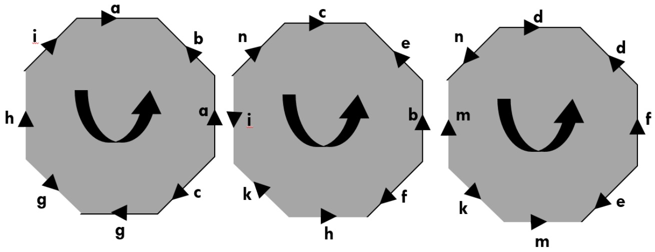


Figure 1.18. Example for orientation preserving gluing

polygon and the other edge's gluing direction is opposite the direction of orientation, then the polygon gluing is called **orientation preserving**.

Theorem 2. *A collection of polygons whose edges are glued in pairs by an orientation preserving gluing results in a collection of compact orientable surface.*

In Figure 1.18 we can see an example of when a gluing is orientation preserving. An important question for this thesis is, When we have an orientation preserving gluing of polygons that result in a single compact orientable surface, how do we determine the genus of the resulting surface?

It turns out, we can find that out using what is known as Euler's formula. Note that with a polygon gluing that results in a single compact orientable surface, each polygon interior results in a face on the compact orientable surface, each glued pair of edges results in a single edge, and the polygon vertices result in some number of vertices on the surface. We denote the number of faces,edges, and vertices on the resulting compact orientable surface by F, E, and V, respectively.

Theorem 3. *If an orientation preserving gluing results in a single compact orientable surface, then $V - E + F = 2 - 2g$, where g is the genus of the compact orientable surface.*

When we consider regular polyhedra in three dimensions, since they are homeomorphic to the sphere, they have a genus of 0, and we get the well known relation $V - E + F = 2$. An example is shown in Figure 1.19, where the convex icosahedron has $V = 12$, $E = 30$ and $F = 20$.

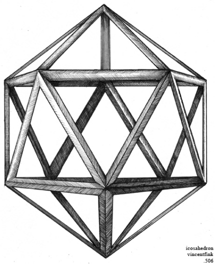


Figure 1.19. A convex icosahedron

Chapter 2
GRAPH EMBEDDING INTO SURFACES

In this chapter we introduce the main concepts and results related to graph embeddings and their representations ia rotation systems. The following are references for the material in sections 2.1-2.3 : [4],[2],[3]. In section 2.4 we introduce our first main result. It details how rotation system changes impact the associated embedding. It will be important to us in Chapter 3 when we investigate particular embeddings of hypercube graphs.

2.1 Graph embedding and Rotation systems

Given a graph G and a surface S, a **graph embedding** is an embedding $f : G \to S$. Informally a graph embedding is a representation of the graph G on the surface such that two edges intersect only in vertices they have in common. A graph embedding is called a **2-cell embedding** if every component of the complement of the embedded graph in the surface is homeomorphic to the open 2-disk.

For example we have the embedding of K_5 on a Torus, as shown in Figure 2.1

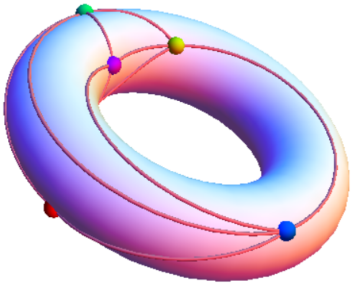


Figure 2.1. Embedding of K_5 on a Torus

Once we look at a few embeddings, we realize that it is cumbersome to represent them pictorially each time we want to talk about them. Thankfully, there is an easier way of representing embeddings using rotation systems. A **rotation system** for a graph G is an ordering of the edges around every vertex. For example, consider the embedding of Q_3 on a sphere shown in Figure 2.2. The information about this graph embedding can be stored at each vertex as an ordering of the edges that appear around it when traversing it in a counterclockwise fashion. For example, around 0, the ordering of the edges here in counterclockwise order is $01, 04, 02$. For simplicity, instead of representing the rotation system as a list of edges, we represent it using the vertex at the other end of the edge. We can do this because we are working with simple graphs and there are no repeated edges. For example, the rotation system around vertex 0 in Figure 2.2 is represented as $(0 : 1, 4, 2)$. If this is done around each vertex, we obtain the rotation system as follows :

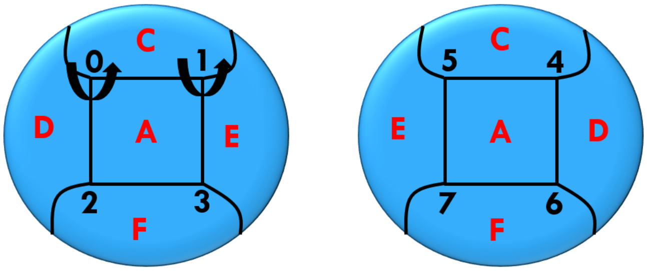


Figure 2.2. Q_3 on a sphere : Front and Rear view

$(0 :\ 1,\ 4,\ 2)$

$(1 :\ 0,\ 3,\ 5)$

$(2 :\ 3,\ 0,\ 6)$

$(3 :\ 2,\ 1,\ 7)$

(4 : 5, 6, 0)

(5 : 4, 7, 1)

(6 : 7, 4, 2)

(7 : 6, 5, 3)

Consider the rotation system about 3 - (3 : 2, 1, 7). The entry that follows 1 is 7 - in terms of embedding this translates to the fact that around vertex 3, the edge that appears next, counterclockwise, after the edge 31 is 37.

Thus, with this scheme, given a graph embedding, we can write an associated rotation system. The natural question to ask here is if the reverse process can be done as well. Given a rotation system, can we construct an embedding from it?

It turns out that with straightforward assumptions, the answer to this question is yes. We address this in the next section.

2.2 Boundary Walks

As indicated previously, given a graph embedding, there is an associated rotation system. Now we show that when we have a rotation system, we also have an associated embedding. From the rotation system, we can create cyclic sequences of edges traversed around the boundary of a polygon that by itself, or with other polygons, will glue to provide a surface and an embedding of the graph. The boundary walk algorithm described below is the process by which we create these sequences of edges.

Definition 1. *Given a rotation system and a directed edge vw, the* **succeeding edge** *to vw is the directed edge wv' such that wv' immediately follows wv in the ordering in the rotation system at vertex w. The* ***preceding edge*** *to vw is the* *directed edge $w'v$ such that vw' precedes vw in the rotation system at vertex v. Note that if there is only one edge incident to vertex w then wv is the succeeding edge to vw, and if there is only one edge incident to vertex v, then wv is the preceding edge to vw.*

Boundary Walk Algorithm : Pick a directed edge e_0 and form a walk $e_0e_1e_2...e_n$ by letting e_{i+1} be the succeeding edge to e_i for each i. End the walk at e_n where the succeeding edge to e_n is e_0 (we explain below why this is always possible). The resulting walk is called a **boundary walk** associated to the rotation system. If all the directed edges have appeared in the boundary walks constructed, the process ends. Otherwise pick a directed edge that has not appeared and use it as the starting edge in another boundary walk. The set of boundary walks constructed by the boundary walk algorithm is called the **boundary walk collection** associated with the rotation system.

The idea behind the boundary walk algorithm is that we look at it as if we are walking along the edges in the boundary of a polygon whose interior is a component of the complement of the graph in a surface embedding, and we are walking in such a way that the component lies to our right. As we approach a vertex along an edge, we need to know which edge to take next to continue the walk. As shown in Figure 2.3, the departure edge should be the next edge after the approach edge in the ordering of the edges around the vertex in the rotation system, that is the approach edge's succeeding edge.

There are a number of important questions and results to be addressed regarding the boundary walk algorithm, but first we consider an example. Here we consider Q_3 with the rotation system :

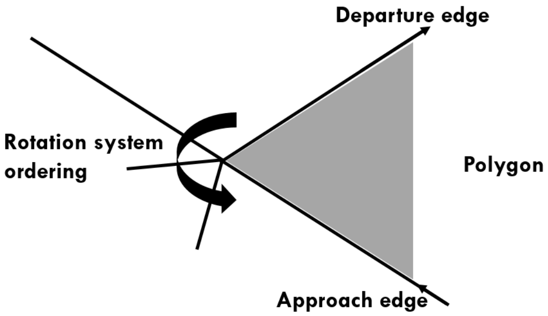


Figure 2.3. Boundary Walk

$$(0:\ 1,\, 2,\, 4)$$
$$(1:\ 0,\, 3,\, 5)$$
$$(2:\ 3,\, 0,\, 6)$$
$$(3:\ 2,\, 1,\, 7)$$
$$(4:\ 5,\, 6,\, 0)$$
$$(5:\ 4,\, 7,\, 1)$$
$$(6:\ 7,\, 4,\, 2)$$
$$(7:\ 6,\, 5,\, 3)$$

If we apply the boundary walk algorithm to the rotation system above, starting with edge 01, we obtain the following walk :

$$0-1-3-2-6-4-0$$

Here we are listing the vertices visited in order; the corresponding directed edges are $01, 13, 32$, etc. Since 01 is the succeeding edge to 40, the walk stopped at 40. We continue to construct boundary walks choosing the starting edge to be a directed edge that has not been traversed yet. We get the following additional walks :

$$0-2-3-7-5-1-0-2$$

$$0 - 4 - 5 - 7 - 6 - 2 - 0 - 4$$

$$1 - 0 - 2 - 3 - 7 - 5 - 1 - 0$$

Each of these boundary walks can be represented as a hexagon with directed edges, as illustrated in Figure 2.4. Note that, as expected, every edge appears exactly twice, once in each direction. Hence the sum of the walk lengths is 24 where there are four components, each of size six. The four hexagons in Figure 2.4 can be glued according to the labeled edges to obtain an embedding of Q_3 on a torus, as illustrated in Figure 2.5. In Figure 2.6 we show the correspondence of the interior of the polygons with the regions in the torus with the embedding. To verify that this is indeed the surface it maps into, we can look at Figure 2.6, where

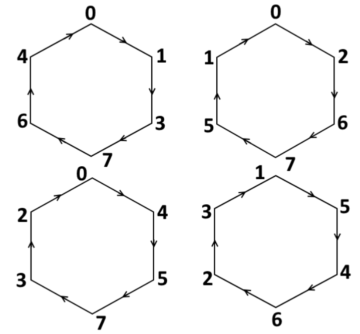


Figure 2.4. The boundary walks

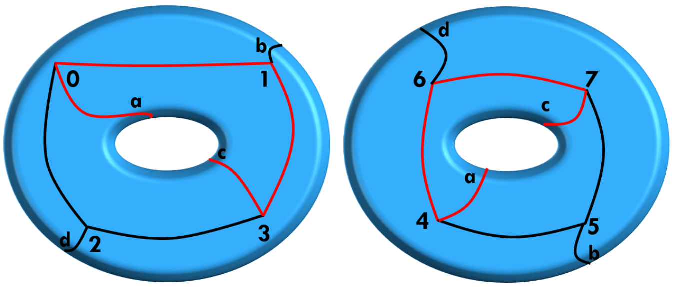


Figure 2.5. Top and bottom view of embedding of Q_3 on a torus

we have marked each of the boundary walk on the hexagons and their corresponding representation on the torus. We get a mapping of each region to the torus. Note that the embedding that results from the boundary walk algorithm is a 2-cell embedding since the components of the complement of the graph in the embedding surface is made up of the interiors of the polygons and each is homeomorphic to an open disk.

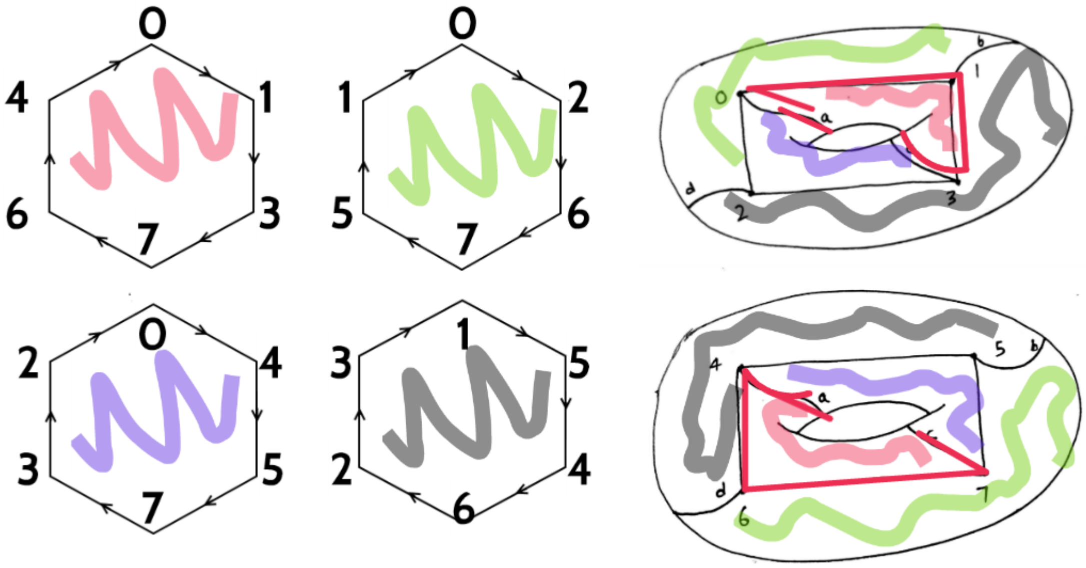


Figure 2.6. Mapping of regions on Q_3

Next, via Theorem 4-6 we address some natural questions that arise regarding the Boundary Walk Algorithm.

Theorem 4. *In constructing a boundary walk as in the boundary walk algorithm, at some point an edge is chosen whose succeeding edge is the initial directed edge (and therefore each boundary walk ends as indicated).*

Proof. The boundary walk will end because there are a finite number of edges and after some iterations of the boundary walk algorithm, the edges will start repeating. Since, the rotation system is fixed, if a particular directed edge repeats then so would its preceding edge, unless the directed edge is the initial directed edge in the boundary walk. Thus, the initial directed edge is the first to repeat. □

Theorem 5. *Each directed edge appears exactly once in a boundary walk collection.*

Proof. Let a directed edge xy appear in two different boundary walks and let the succeeding edges in each of these boundary walks be yz and yw. If $z \neq w$, we have an ambiguity in the rotation system around y. Hence, $z = w$. Similarly, in both boundary walks, the preceding edge to xy must be the same and it follows that the two boundary walks are the same. □

Given a rotation system and corresponding boundary walk collection, we can associate to each boundary walk a polygon whose edges are labeled in clockwise order with the directed edges in the boundary walk. The result can be seen as a collection of polygons whose edges are glued together in pairs (since each directed edge appears exactly once in the boundary walk collection). We refer to the result of the polygon gluing as the **rotation system polygon gluing**.

Theorem 6. *Assume we have a connected simple graph G and a given rotation system for G. The rotation system polygon gluing results in a compact orientable surface of genus $g = 1 - \frac{V-E+B}{2}$, where E is the number of edges in the graph, V is the number of vertices in the graph and B is the number of boundary walks in the boundary walk collection. Furthermore, the polygon edges and vertices glue together so that the result is 2-cell embedding of G in the surface.*

Proof. We provide a sketch of the proof here. For further details, see the references mentioned previously. Because the graph is connected, it follows that the polygon gluing results in a single compact surface.

It easily follows that a rotation system polygon gluing is an orientation preserving gluing. By Theorem 2, the resulting surface is a compact orientable surface.

Since the sides and the vertices of the polygon correspond to the edges and the vertices of the graph, and since the polygons glue together with the appropriate incidence relation between vertices and edges, the graph embedding obtained is an embedding of G.

The interior of each polygon is homeomorphic to an open 2-cell, implying that the embedding obtained by gluing together the polygons is a 2-cell embedding.

Theorem 3 implies that $V - E + B = 2 - 2g$ and therefore the genus of the resulting surface is $g = 1 - \frac{V-E+B}{2}$.

□

Definition 2. *The* ***rotation system defined embedding*** *of G is the embedding that is obtained from a rotation system using the boundary walk algorithm.*

2.3 Maximal and Minimal genus of graph embeddings

We have seen that representing embeddings as rotation systems is much easier than making drawings of them.This raises a number of questions about the relationship between the rotation systems and the embedding -

- How does a change in the rotation system affect the embedding?
- Can we control how the embedding of the graph changes by making specific changes in the rotation system?

For example, let us take the rotation system on Q_3 that we defined previously and reverse the ordering of the edges around vertex 0. We change the rotation system ordering from $(1, 2, 4)$ to $(1, 4, 2)$ and the new rotation system is now

$$(0 : 1, 4, 2)$$

$$(1 : 0, 3, 5)$$

$$(2 : 3, 0, 6)$$

$$(3 : 2, 1, 7)$$

$$(4 : 5, 6, 0)$$

$$(5 : 4, 7, 1)$$

$$(6 : 7, 4, 2)$$

$$(7 : 6, 5, 3)$$

Let us use the boundary walk algorithm to find the boundary walks of this embedding. Since the entry around 0 has changed, when we start the walk with $0 - 1$, and reach vertex 0 from 4, instead of going towards 1 and closing this boundary walk, we now go towards 2 and continue for a while before this boundary walk closes up. As a result, we now get only two boundary walks, instead of the four that we previously had. These boundary walks are :

$$0 - 1 - 3 - 7 - 5 - 4 - 6 - 2 - 3 - 1 - 5 - 7 - 6 - 4 - 0 - 2 - 6 - 7 - 3 - 2, \text{ and}$$

$$1 - 0 - 4 - 5$$

This means we have two polygons, to be glued for the embedding, and one is a 20-gon as represented in Figure 2.7. The effect of this seemingly minor change is

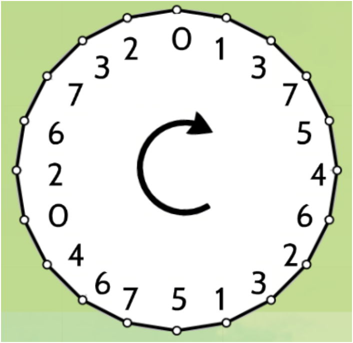


Figure 2.7. 20-gon corresponding to the walk from the modified rotation system

that now the embedding of Q_3 has only two boundary walks. Hence the Euler characteristic of the embedding surface is -2 and the embedding now is in a 2-hole torus instead of a 1-hole torus, as shown in Figure 2.8

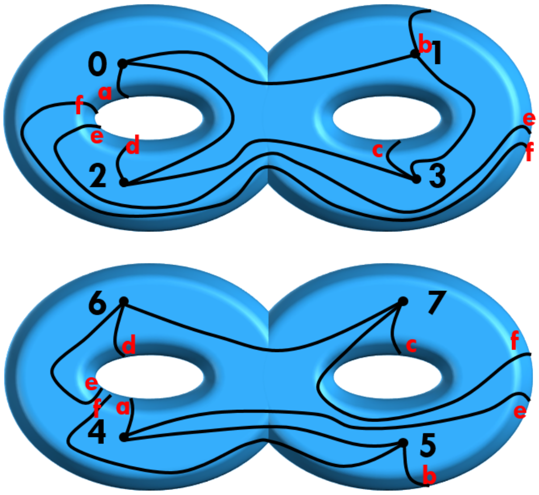


Figure 2.8. Top and bottom view of Q_3 in a 2-hole torus after 1 switch

Notice that, so far, we have seen the same graph Q_3 embedded in a sphere, a 1-hole torus, and a 2-hole torus; that is, in surfaces of genus $0, 1$, and 2. Is there a limit to how large the genus of the embedding surface can be? This brings us to some of the specific kinds of embeddings that we are interested in looking at. The **maximal genus** of a graph is the maximum integer n such that the graph can be 2-cell embedded in an orientable surface of genus n. The **minimal genus** of a graph is the minimum integer n such that the graph can be 2-cell embedded in an orientable surface of genus n. The minimal genus is also called the **genus** of the graph. The genus and the maximal genus of a graph G are denoted by $\gamma(G)$ and $\gamma_M(G)$, respectively.

Note that the embedding of Q_3 that we have seen in surfaces of genus $0, 1, 2$ are 2-cell embeddings. It follows that $\gamma(Q_3) = 0$ since zero is the least value of genus possible. Furthermore $\gamma_M(Q_3) = 2$ since we have a 2-cell embedding of Q_3 in the genus 2 surface, and the following theorem indicates that that is the largest genus for a 2-cell embedding of Q_3.

Theorem 7. *The minimal genus of the hypercube graph Q_n satisfies :*

$$\gamma(Q_n) \geq (n-4)2^{n-3} + 1,$$

and the maximal genus satisfies :

$$\gamma_M(Q_n) \leq (n-2)2^{n-2}$$

Proof. For a given graph embedding, the number of vertices and the number of edges are fixed. In Q_n, we have $V = 2^n$, and $E = n2^{n-1}$. The number of boundary walks though, depends on the rotation system and the embedding. From Theorem 6, we have $g = 1 - \frac{V-E+B}{2}$. Note that since E and V are even and g must be an integer, it follows that B is even. For maximal genus, we need B to be the least possible, which is 2. Plugging in the values, we get $\gamma_M(Q_n) \leq (n-2)2^{n-2}$. Now, for minimal genus, we need the number of boundary walks to be as high as possible. This means that we need the size of each boundary walk to be as small as possible. Since, the total boundary walk length is $n2^n$ and the smallest cycle in each hypercube graph is of size 4, the total number of boundary walks is at most $n2^{n-2}$. Plugging this into the equation for genus, we get $\gamma(Q_n) \geq (n-4)2^{n-3} + 1$. □

In chapter 3 we prove that equality is attainable in each of the inequalities in Theorem 8.

2.4 Adjacent changes in a Rotation System

Given a rotation system, the simplest change we could make to it is to switch the order of a single pair of adjacent edges incident on a particular vertex. Note that, with a sequence of such changes we can obtain all possible rotation systems for a graph, given a rotation system to start with. In this section, we examine the impact of such a switch on the collection of boundary walks and the embedding.

Let us switch the ordering of edges about the vertex 0 in Figure 2.9 and look at the effect that this action has on the embedding. We will state a result about how the boundary walks change when these switches are made. In Figure 2.9, we start with an initial configuration of edges incident to vertex 0 in a rotation system and then switch the edges labeled 01 and 02 to get to a new configuration. We assume that there is at least one other edge incident to 0, the edge $0x$ in Figure 2.9, otherwise switching the edges 01 and 02 has no impact on the ordering of the edges around vertex 0 since they are the only edges incident to the vertex. It is possible that edges $0x$ and $0y$ in the figure are the same edge.

In Figure 2.9, P_1, P_2, P_3 represent polygons that are glued in the embedding as a result of the parts of the boundary walk shown. They are not necessarily distinct. For example, P_1 will coincide with P_2 if directed edges 01 and 10 appear in the same boundary walk. We will be referring to these polygons as the associated polygons. P_1 has a boundary walk $x01B'$ where B' represents the sequence of edges in the boundary walk as we go from 01 back to $x0$. Similarly, P_2 has its boundary walk given by the sequence $102B''$, and P_3 has boundary walk $20yB'''$.

Definition 3. *Given a rotation system, an* ***adjacent change*** *at a vertex is a switch of the ordering of two adjacent edges around the vertex.*

Theorem 8. *With vertices, edges and polygons as above, making an adjacent change to edges* 01 *and* 02 *at vertex* 0 *in a rotation system increases the number of boundary walks by 2, decreases it by 2 or leaves it unchanged. In particular, the*

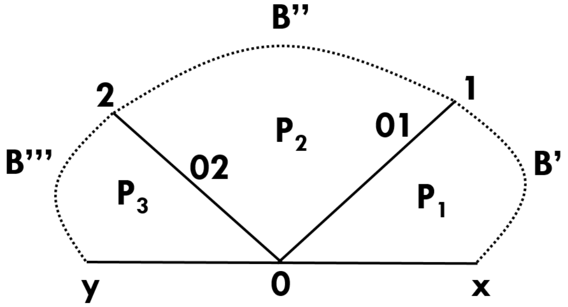


Figure 2.9. The configuration prior to the switch of edges and polygons

change in the number of boundary walks depends on the initial configuration and on where the directed edges 01, 10, 02, 20 *are located in the boundary walks:*

1. *If the associated polygons* P_1, P_2 *and* P_3 *are distinct, then the number of boundary walks decreases by 2 (three faces combine to form one).*

2. *If the associated polygons* P_1, P_2 *and* P_3 *coincide and the directed edges* 01, 10, 02, 20 *appear in the order* $01 - 20 - 10 - 02$ *in the corresponding boundary walk, then the number of boundary walks increases by 2 (the faces breaks apart to form 3 faces), whereas if they appear in the order* $01 - 10 - 02 - 20$*, then there is no change in the number of boundary walks*

3. *If two of the associated polygons* P_1, P_2 *and* P_3 *coincide and the third is distinct, then there is no change in the number of boundary walks.*

Proof. 1. Note that since we are switching the edges 01 and 02, only the boundary walks containing 01, 10, 02, 20 are impacted by the switch. This is because by making the switch in the edges around 0, we are making a change in the rotation system around 0 only and hence the only boundary walks that are affected are the ones that contain the directed edges 01, 10, 20, 02. Referring to Figure 2.9 as the initial configuration, we can write the corresponding boundary walks. Since the edges $0x, 02, 01, 0y$ are adjacent edges around 0, a boundary walk that starts with 01 will close off only when we come in from the edge $x0$ (01 is the succeeding edge to $x0$). Hence, the boundary walk is going to be in the form $01B_{1x}x0$, where B_{1x} is the part of the boundary walk going from 1 to x. It is important to note that this is just a string (of possibly zero length) of vertices traversed, and the exact nature of the path taken does not matter. Similarly, the boundary walk starting 02 is in the form $02B_{21}10$ and the boundary walk starting $0y$ is in the form $0yB_{y2}20$. Note that these boundary walks cover each of the four directed edges we are concerned with (namely $01, 10, 02, 20$), and the directed edges in B_{1x}, B_{21} and B_{y2} are all distinct.

The polygons P_1, P_2, P_3 corresponding to these boundary walks are represented in Figure 2.10. Now we switch the adjacent edges 01 and 02.

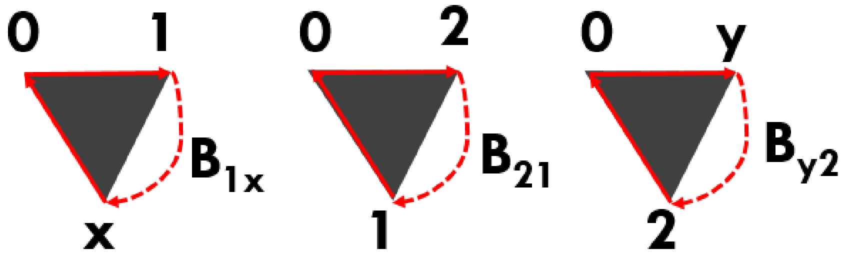


Figure 2.10. The polygons P_1, P_2 and , P_3 and their boundary walks

This amounts to switching 1 and 2 in the boundary walks illustrated in Figure 2.10. Once we have done this, the resulting polygon or polygons will have parts as shown in Figure 2.11. We have to determine where B_{1x}, B_{21} and B_{y2} lie within these boundary walks.

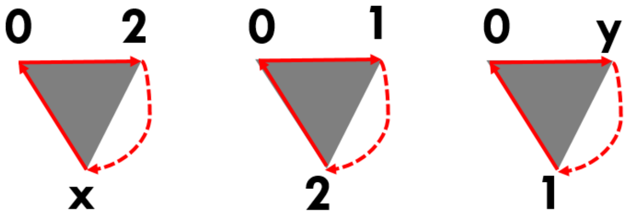


Figure 2.11. Parts of the polygon(s) after switching 01 and 02

To determine the resulting boundary walks, note that if we begin a walk $x02$, we must then traverse B_{21} following edge 02 as before. That brings us to edge 10 which, with the switch, is followed by $0y$. Continuing in this manner we find that we now have a single boundary walk $x02B_{21}10yB_{y2}201B_{1x}$, as illustrated in Figure 2.12. Hence, in this case the number of boundary walks

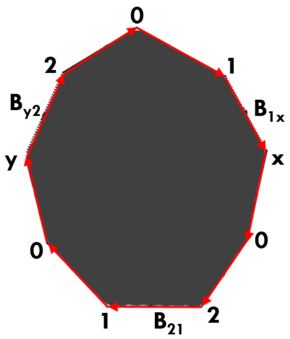


Figure 2.12. 3 faces have combined into 1

has reduced by 2 as claimed.

2. Now we consider a different initial configuration. In this case, the associated polygons P_1, P_2, P_3 coincide and result from a single boundary walk. When this happens, we could have two kinds of boundary walks. In the first case, the boundary walk has the edges in order $01 - 20 - 10 - 02$, as follows :

$$01B_{12}20yB_{y1}102B_{2x}x.$$

This corresponds to one face, as shown in Figure 2.13]. Now we switch the adjacent edges $01, 02$. In doing so, as in the previous case, the resulting polygon(s) contain parts as shown in Figure 2.11.

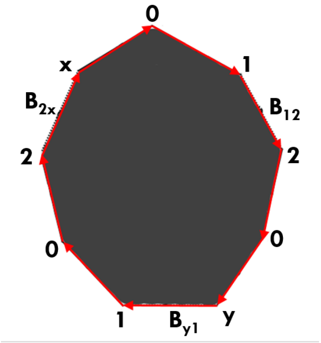


Figure 2.13. Case 2: Initial configuration

To determine the resulting boundary walk(s), consider the boundary walk that includes $x02$. Following edge 02, according to the original rotation system, we must have B_{2x}. That then brings us back to $x0$ and closes the boundary walk. The resulting boundary walk is $x02B_{2x}$. Similarly, we obtain boundary walks $10yB_{1y}$ and $201B_{12}$, as illustrated in Figure 2.14. Thus, the

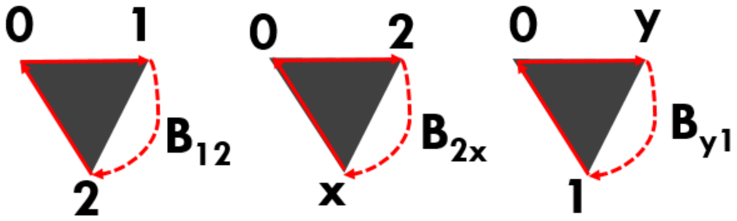


Figure 2.14. Case 2: The boundary walk after switching edges 01 and 02

number of boundary walks has increased by two from the initial configuration.

Next, in the case the edges appear in the order $01 - 10 - 02 - 20$, we have a boundary walk that appears as follows:

$$x01B_{11}102B_{22}20yB_{yx}x0.$$

This corresponds to the polygon shown in Figure 2.15, with the vertices labeled as required. We switch the adjacent edges $01, 02$ and again obtain a boundary walk collection with boundary walks with parts as shown in Figure 2.14

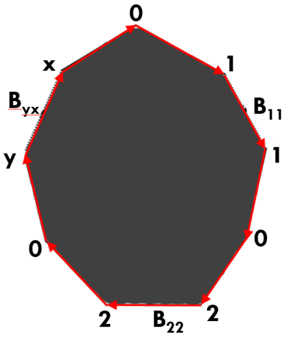


Figure 2.15. Case 2: The boundary walk before switching edges 01 and 02

Arguing as in the previous cases, we find that in this case the three parts lie in a single boundary walk given by $x02B_{22}201B_{11}10yB_{yx}$. The only effect of the switch is to change the ordering of the edges in the boundary walk to the new ordering, as seen in Figure 2.16. Thus, in this case, the number of boundary walks is unchanged as a result of the edge switch.

3. In this case, when the associated polygon P_1 and P_2 coincide, the boundary walk collection is given by $x01B_{11}102B_{2x}$ and $20yB_{y2}$. Arguing as before we

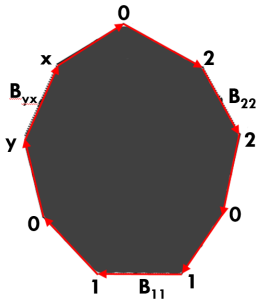


Figure 2.16. Case 2: The boundary walk after switching edges 01 and 02

find that on switching, the new boundary walk collection is $x02B_{2x}$ and $201B_{11}10yB_{y2}$. Similarly, if the associated polygons P_1 and P_3 coincide, then the boundary walk collection is given by $x01B_{12}20yB_{yx}$ and $102B_{21}$. On switching, the new boundary walk collection is $x02B_{21}10yB_{yx}$ and $201B_{12}$. Lastly, if the associated polygons P_2 and P_3 coincide, the boundary walk collection is given by $x01B_{1x}$ and $102B_{22}yB_{y1}$. On switching, the new boundary walk collection is $x02B_{22}201B_{1x}$ and $10yB_{y1}$. In all of these subcases of case 3, the number of boundary walks remained unchanged, as claimed.

□

Corollary 1. *Given a graph and a rotation system for it, an adjacent change at a vertex increases the genus of the associated embedding surface by 1, decreases it by 1 or leaves it the same.*

Proof. From Theorem 9 we know that an adjacent switch amounts to a decrease in number of boundary walks by two, an increase in number of boundary walks by two or no change in the number of boundary walks. Since, $g = 1 - \frac{V-E+B}{2}$, where B is the number of boundary walks in the boundary walk collection, an adjacent change would correspond to an increase in the genus of the embedding surface by one, a decrease in the genus of the embedding surface by one, or no change in the genus of the embedding surface. □

Corollary 2. *For a graph G, if $\gamma(G) \leq g \leq \gamma_M(G)$, then there is a 2-cell embedding of G in a surface of genus g.*

Proof. Given g such that $\gamma(G) \leq g \leq \gamma_M(G)$, using Corollary 1 we know that if we start with an embedding in a surface of genus $\gamma(G)$ and an associated rotation system, then if we perform a sequence of edge switches, then we increment the genus by at most one with each switch. Since, the maximal genus and the minimal genus are attainable, there are embeddings and associated rotation systems for them. We can take a rotation system for the minimal genus and via a sequence of edge switches attain the rotation system for the maximal genus. It follows that via these edge switches we go through embeddings with each genus g, such that $\gamma(G) \leq g \leq \gamma_M(G)$. □

Note that it is not necessary that each switch increase the genus by exactly one, but it is necessary that by making switches each and every genus in between the minimal and the maximal genus is achieved.

Corollary 3. *For a graph G, either every 2-cell embedding has an even number of faces or every 2-cell embedding has an odd number of faces.*

Proof. This follows from the fact that according to edge switching in rotation systems, the number of boundary walks can change only by 2, by 0 or by -2, and that all rotation systems can be attained via edge switches. □

Chapter 3
HYPERCUBE GRAPH EMBEDDINGS ON ORIENTABLE SURFACES

3.1 Introduction

In this chapter we explore examples of embeddings of hypercube graphs in orientable surfaces and establish results about some general families of such embeddings. These results include the existence of minimal and maximal embeddings for Q_n for all n. Before doing that, we will define some of the terms that will be used throughout the chapter.

3.2 Common Terminology

A **bit** is defined as a binary digit that has a single binary value, 1 or 0. A **binary string** is a sequence of bits. The **complement of a bit** a_i is denoted by $\tilde{a_i}$ and equals 1-a_i. **The complement of a binary string** $\tilde{v}$ is the string made up of the complement of each of its bits; that is if $v = a_1a_2...a_n$, then $\tilde{v} = \tilde{a_1}\tilde{a_2}...\tilde{a_n}$.

Given two binary strings of equal length, the **Hamming distance** between them is defined as the number of bits in which they differ. For example, 01101 and 10010 differ in five bits and hence the Hamming distance between them is 5. In the hypercube graph Q_n, the Hamming distance between every pair of vertices connected by an edge is one. For example, in Figure 3.1, consider the vertex 7 in Q_3. The binary representation of 7 is 111. Since there are exactly three bits that can be complemented, the vertex is connected by single edges to three vertices, which is consistent with the fact that in Q_3 all vertices have degree 3. The vertices to which it is connected can be obtained by complementing the bits one at a time to obtain $011, 101, 110$, which are $3, 5, 6$, respectively, as is shown in Figure 3.1

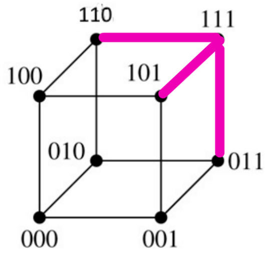


Figure 3.1. Vertices connected to 7 in Q_3

A **Gray code** is a sequence of binary strings where every term is Hamming distance 1 away from the term preceding it. The usual numeral system $0, 1, 2, 3$ is not a Gray code because in binary this is $00, 01, 10, 11$, and in going from 01 to 10 we have made changes in two bits, instead of one. In order to make it a Gray code, we need the order to be $00, 01, 11, 10$ in binary, which is $0, 1, 3, 2$, in the decimal system. The following lemma, whose proof is straightforward, will be helpful to us later in the chapter:

Lemma 1. *The vertices in Q_n can be listed in a Gray-code order beginning with* $00...00, 00...01$ *and ending with* $10...01, 10...00$

In a boundary walk for a rotation system on the hypercube graph Q_n, the sequence of vertices around the boundary walk component is a Gray code since each numerically consecutive pair of vertices is connected by an edge in Q_n.

In what follows, we use either the binary or the decimal representation for the vertices in Q_n, depending on which is most convenient and relevant for the situation.

3.2.1 The "ABC" rotation system

Recall that a rotation system is an ordering of the edges incident to each vertex in a graph. When we represent the rotation system, instead of listing the edges, we list the vertices at the other end of the edges. We can do this with hypercube graphs since we have at most one edge between each pair of vertices. For example, looking at Figure 3.1 the rotation system around 7 is represented as $(7 : 3, 6, 5)$, which means that when we traverse counter clockwise around vertex 7, the edges appear in the order $(73, 76, 75)$.

Definition 4. *The* ***Alternate Bit Change (ABC) rotation system*** *for every vertex* $v = a_1a_2...a_{n-1}a_n$ *is defined as :*

$$(a_1a_2...a_{n-1}a_n : a_1a_2...a_{n-1}\tilde{a_n}, a_1a_2...\tilde{a_{n-1}}a_n, ..., a_1\tilde{a_2}...a_{n-1}a_n, \tilde{a_1}a_2...a_{n-1}a_n)$$

For example, in Q_4 to get the ABC rotation system entry corresponding to vertex 9, we start with the binary representation of 9, $9 = 1001$. The rotation system entry at vertex 9 is :

$$(1001 : 1000, 1011, 1101, 0001)$$

Note that for each vertex we have exactly four entries in the rotation system because in Q_4, each vertex has degree 4 and there are exactly four bits to be complemented.

We also define a rotation system, called the **Reverse ABC rotation system** as :

$$(a_1a_2...a_{n-1}a_n : \tilde{a_1}a_2...a_{n-1}a_n, a_1\tilde{a_2}...a_{n-1}a_n, ..., a_1a_2...\tilde{a_{n-1}}a_n, a_1a_2...a_{n-1}\tilde{a_n})$$

Here the ordering around each vertex is the opposite of the ordering in the ABC rotation system.

3.2.2 The embedding corresponding to the ABC rotation system

In Chapter 2, we established that given a rotation system, we can use the boundary walk algorithm to find the corresponding embedding. Let us see what embedding the ABC rotation system gives us. In Q_3, the ABC rotation system is given by :

$$(0 : 1, 2, 4)$$
$$(1 : 0, 3, 5)$$
$$(2 : 3, 0, 6)$$
$$(3 : 2, 1, 7)$$
$$(4 : 5, 6, 0)$$
$$(5 : 4, 7, 1)$$
$$(6 : 7, 4, 2)$$
$$(7 : 6, 5, 3)$$

The boundary walk algorithm gives us four polygons which are shown in Figure 3.2. When these polygons are glued together, we get the surface on which this graph is embedded, and that is a torus. The embedding is shown in Figure 3.3. One of the polygons is represented on the embedding using a red line. In Q_4, the ABC rotation system is given by :

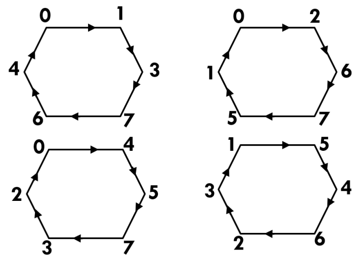


Figure 3.2. Polygons corresponding to the ABC rotation system in Q_3

$$(0 : 1, 2, 4, 8)$$
$$(1 : 0, 3, 5, 9)$$
$$(8 : 9, 10, 12, 0)$$
$$(9 : 8, 11, 13, 1)$$

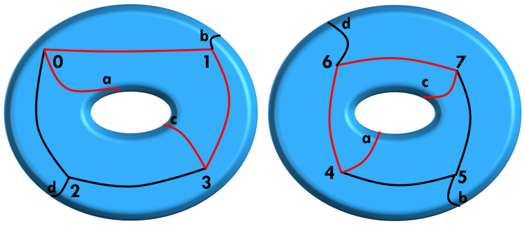


Figure 3.3. Top and bottom view of the embedding of Q_3 with the ABC rotation system

$$(2:\ 3,\ 0,\ 6,\ 10)$$
$$(3:\ 2,\ 1,\ 7,\ 11)$$
$$(4:\ 5,\ 6,\ 0,\ 12)$$
$$(5:\ 4,\ 7,\ 1,\ 13)$$
$$(6:\ 7,\ 4,\ 2,\ 14)$$
$$(7:\ 6,\ 5,\ 3,\ 15)$$
$$(10:\ 11,\ 8,\ 14,\ 2)$$
$$(11:\ 10,\ 9,\ 15,\ 3)$$
$$(12:\ 13,\ 14,\ 8,\ 4)$$
$$(13:\ 12,\ 15,\ 9,\ 5)$$
$$(14:\ 15,\ 12,\ 10,\ 6)$$
$$(15:\ 14,\ 13,\ 11,\ 7)$$

In this case, using the boundary walk algorithm, we get eight boundary walks of size eight each. When we glue these polygons together, we get the surface on which the graph is embedded. Since $V = 16, E = 32$, and $B = 8$, we have g$=$ 5, which means that the embedding is on a 5-hole torus. Figure 3.4 shows one of the polygons for this embedding on a 5-hole torus. These observations can be generalized by the following theorem, which tells us exactly how the embedding of Q_n corresponding to the ABC rotation system appears.

Theorem 9. *For $n \geq 3$ the ABC rotation system on the hypercube graph Q_n has 2^{n-1} boundary walks of size $2n$, resulting in an embedding of Q_n in a surface of genus : $g = 1 - n - 2^{n-1} + n2^{n-2}$*

Proof. For a given hypercube graph Q_n, we can write the vertex in binary form as $v = (a_1, a_2, ..., a_n)$. The ABC rotation system ordering around the vertex v is :

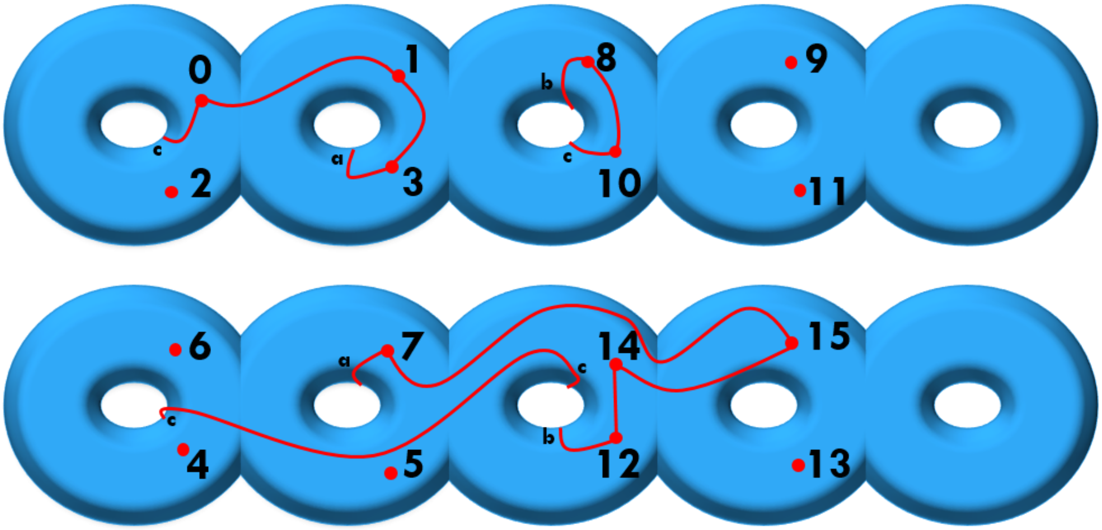


Figure 3.4. Top and bottom views of one of the faces in the embedding of Q_4 with the ABC rotation system

$$((a_1, a_2, ..., a_n) : (a_1, ..., \tilde{a_n}), (a_1, ...\tilde{a_{n-1}}, a_n), ...(\tilde{a_1}, a_2, ..., a_n))$$

By definition, every pair of vertices connected by an edge is separated by Hamming distance of 1. To create a boundary walk , we start at any edge, say v-w. Since these vertices are separated by a Hamming distance of 1, by appropriately choosing the bits for their representation, we can have our boundary walk start as $(a_1, ...a_q, ...a_n) \to (a_1, ...\tilde{a_q}, ...a_n)$ for some a_q. But according to the rotation system, the next entry in the boundary walk is the vertex appearing to the right of $(a_1, ...\tilde{a_q}, ...a_n)$ in the rotation system ordering at $(a_1, ...a_n)$. That vertex is $(a_1, ...\tilde{a_{q-1}}, \tilde{a_q}, ...a_n)$. Continuing the boundary walk in this way will give us n entries until we reach $\tilde{v_i}$ (see Figure 3.5). Repeating this process another n times gets us back to v. This completes the boundary walk and the length of the walk is $2n$, as required.

In Q_n the number of vertices is $V = 2^n$ and the number of edges is $E = n2^{n-1}$. Since each edge in the graph is traversed twice, once in each direction, the total length of the boundary walk collection is twice the number of edges, which in the hypercube graph is $n2^n$. In the ABC rotation system, every walk is of size $2n$. Hence, the number of boundary walks is $B = n2^n/2n = 2^{n-1}$. Substituting this into the equation $g = 1 - \frac{V+E-B}{2}$, we get $g = 1 - n - 2^{n-1} + n2^{n-2}$, as claimed. □

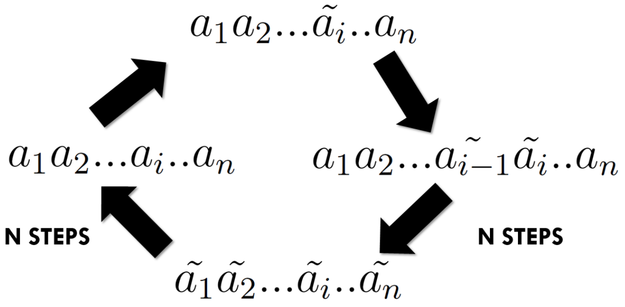

$a_1a_2...\tilde{a}_i..a_n$
$a_1a_2...a_i..a_n$
$a_1a_2...\tilde{a_{i-1}}\tilde{a}_i..a_n$
N STEPS
N STEPS
$\tilde{a}_1\tilde{a}_2...\tilde{a}_i..\tilde{a}_n$

Figure 3.5. ABC Proof sketch

3.3 The Minimal Embedding of Q_n

3.3.1 Definition

Recall that a 2-cell embedding of a graph on a surface of minimal genus is called a minimal embedding. From Theorem 8, we have that $\gamma(Q_n) \geq (n-4)2^{n-3} + 1$. So the question we want to ask here is if $\gamma(Q_n) = (n-4)2^{n-3} + 1$ is always attainable. In the next section, we will provide some examples and answer the question generally for Q_n

3.3.2 Examples of minimal embedding

We have seen before that the minimal embedding of Q_3 is on a sphere. The rotation system that corresponds to this embedding is given by:

$$(0 : 1, 4, 2)$$

$$(1 : 3, 5, 0)$$

$$(2 : 3, 0, 6)$$

$$(3 : 7, 1, 2)$$

$$(4 : 5, 1, 6)$$

$$(5 : 1, 4, 7)$$

$$(6 : 4, 2, 7)$$

$$(7 : 5, 6, 3)$$

Given this rotation system, we can use the boundary walk algorithm to obtain the embedding polygons as shown in Figure 3.6. When these polygons are glued together, they give the surface on which Q_3 is minimally embedded, as shown in Figure 3.7. Note that there are exactly six faces of size four labeled A, B, C, D, E, F.

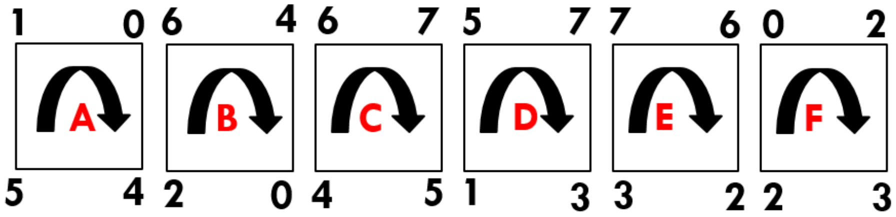


Figure 3.6. Faces in the minimal embedding of Q_3

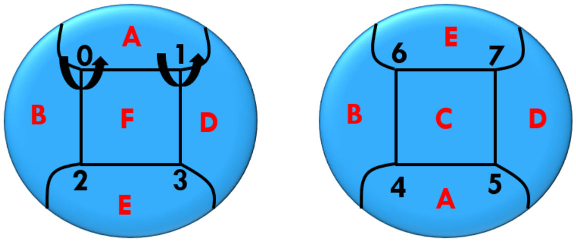


Figure 3.7. Front and rear view of the minimal embedding of Q_3 on a sphere

Now let us look for the minimal embedding of Q_4. By Theorem 8 we have that $\gamma(Q_4) \geq 1$ so the minimal embedding is possible in a surface of genus 1. In fact, we can find such an embedding. The rotation system corresponding to it is given by :

$(0 : 1, 2, 4, 8)$

$(1 : 9, 5, 3, 0)$

$(2 : 10, 6, 0, 3)$

$(3 : 2, 1, 7, 11)$

$(4 : 12, 0, 6, 5)$

$(5 : 4, 7, 1, 13)$

$(6 : 7, 4, 2, 14)$

$(7 : 15, 3, 5, 6)$

$(8 : 0, 12, 10, 9)$

$(9 : 8, 11, 13, 1)$

$(10 : 11, 8, 14, 2)$

$(11 : 3, 15, 9, 10)$

$(12 : 13, 14, 8, 4)$

$(13 : 5, 9, 15, 12)$

$(14 : 6, 10, 12, 15)$

$(15 : 14, 13, 11, 7)$

Using the boundary walk algorithm, we obtain 16 embedding polygons, which can be represented in the grid as shown in Figure 3.8. Note that the left and right side coincide in the figure, as do the top and bottom sides. When these sides are glued together, we get the embedding of Q_4 on a torus, as shown in Figure 3.9.

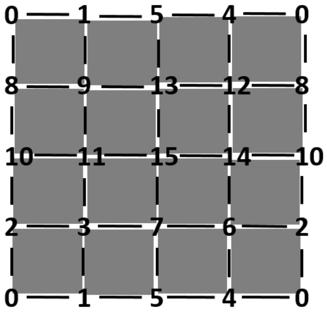


Figure 3.8. Lattice for the minimal embedding of Q_4

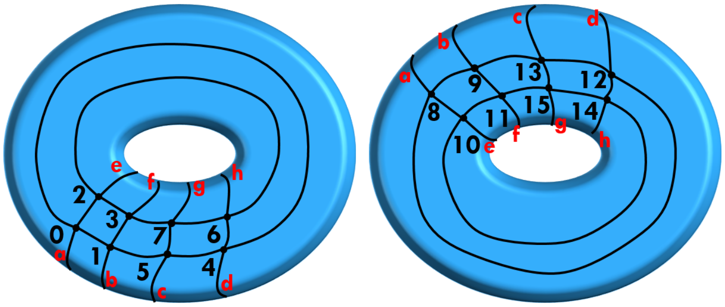


Figure 3.9. Top and bottom view of the minimal embedding of Q_4 on a torus

3.3.3 Minimal Embedding Theorem

Here, we address the minimal embedding for Q_n. By Theorem 8 we have that $\gamma(Q_n) \geq 1 + (n-4)2^{n-3}$. In the examples we presented, we obtained equality for $n = 3$ and $n = 4$ where we saw $\gamma(Q_3) = 0$ and $\gamma(Q_4) = 1$.

In "The genus of the n-cube" [7] Lowell Bieneke and Frank Harary proved that equality holds for all n via a proof that recursively builds the minimal embedding for Q_n from the minimal embedding for Q_{n-1}. We provide an alternative proof where the minimal embedding of Q_n for each n is found via a specific rotation system.

Definition : The **4-cycle rotation system on** Q_n with vertices represented in their decimal notation, is the rotation system defined as having the ABC rotation system at even vertices and the reverse ABC rotation system at odd vertices.

Theorem 10. *The genus of the surface for the embedding associated with the 4-cycle rotation system is* $1 + (n-4)2^{n-3}$*, and therefore* $\gamma(Q_n) = 1 + (n-4)2^{n-3}$.

Proof. We know that $g \geq 1 + (n-4)2^{n-3}$. We prove equality by showing every boundary walk has 4 edges.

To prove the theorem we claim that every boundary walk is of size 4. Pick any vertex v to start with. There are two cases - the vertex in its decimal representation is either odd or even. Consider the vertex v of graph Q_n in its binary notation as $a_1a_2...a_n$. Let the decimal representation of v be even. Since the connected vertices in the hypercube differ in exactly one bit, the next entry in the walk is $a_1, ...\tilde{a_i}...a_n$ for some i. Since this vertex is at an odd Hamming distance from v, it will have the reverse ABC rotation system. In the reverse ABC rotation system, we begin by complementing the leftmost bit and move progressively towards the right, until each bit has been complemented exactly once. Thus, the next entry is going to have the next right bit complemented, which is $a_1, ...\tilde{a_i}, \tilde{a_{i+1}}, ...a_n$. This vertex is an even Hamming distance away from vertex v and hence has the ABC rotation system. The next entry is thus going to have the next left bit complemented, which gives us $a_1, ...a_i, \tilde{a_{i+1}}, ...a_n$ (see Figure 3.10). Once again, this vertex is at an odd Hamming distance from v and hence has the reverse ABC rotation system. This gives us the next entry by complementing the next bit to the right to get $a_1, ..., a_n$, thus completing the walk. The face size is thus exactly 4, as required. Hence, this rotation system gives us exactly the minimal embedding on a surface of genus $g = 1 + (n-4)2^{n-3}$, and therefore this g is the genus of the graph. A similar argument can be made when the vertex v is odd in the decimal representation as shown in Figure 3.10. □

The process used in the proof is illustrated in Figure 3.10.

3.4 Maximal Embedding

As indicated previously, a maximal embedding of a graph is a 2-cell embedding in an orientable surface with g as large as possible. We also indicated previously that for an embedding of Q_n the fewest number of faces possible is 2 (since every 2-cell embedding of Q_n must have an even number of faces). Is an embedding of Q_n with 2 faces possible? Recall that Theorem 8 indicated $\gamma_M(Q_n) \leq (n-2)2^{n-2}$. The

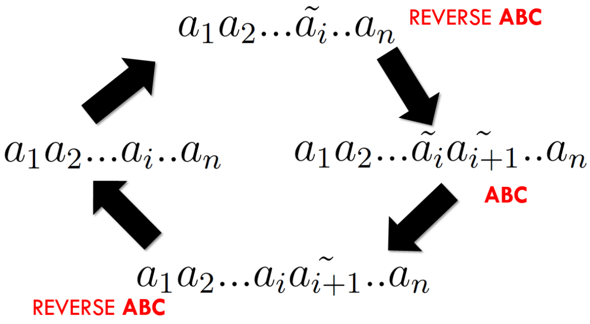

$a_1a_2...\tilde{a_i}..a_n$
REVERSE ABC
$a_1a_2...a_i..a_n$
$a_1a_2...\tilde{a_i}\tilde{a_{i+1}}..a_n$
ABC
$a_1a_2...a_i\tilde{a_{i+1}}..a_n$
REVERSE ABC

Figure 3.10. Sketch of a boundary walk for the Minimal Embedding

upper bound arises from the fact that the fewest faces possible in a two cell embedding of Q_n is two. So, here we are asking if we can achieve $\gamma_M(Q_n) = (n-2)2^{n-2}$.

The answer is known to be affirmative and is based on a theorem by Mark Jungerman [1] that indicates that every 4-edge connected graph is upper embeddable. (A graph is 4-edge connected if it cannot be disconnected by removal of any 3 of its edges, and upper embeddable means has a 2-cell embedding with 1 or 2 faces). It is not difficult to show that for $n \geq 4$, Q_n is 4-edge connected and therefore is upper embeddable. Q_2 has just one embedding on the sphere, so the maximal genus is realized trivially. In Q_3, we have seen an example in Chapter 1 where we switched the rotation system around vertex 0 to get two boundary walks, which corresponds to achieving the maximal genus.

Jungerman's proof of the upper embeddability of Q_n is based on a particular property of Q_n and does not demonstrate an actual maximal embedding. In this section, we examine some examples of maximal embeddings of Q_3 and Q_4, and we prove, via a recursive construction, a general theorem about the existence of a particular type of maximal embedding of Q_n that we call a "big face" embedding.

Unlike minimal embeddings of Q_n, which can be achieved only by having all faces be of size 4, maximal embeddings can be obtained in a number of ways. This is because there are a number of choices in how the two faces can appear. For example, in Q_3 we give examples of maximal embeddings with faces of sizes 18 and 6, 14 and 10, 12 and 12, and 20 and 4. It is interesting to note that by considering properties of boundary-walk lengths under rotation-system switches, it can be proven that there is no maximal embedding where the boundary walks are of sizes 8 and 16. We do not provide a proof here though.

3.4.1 Various maximal embeddings of Q_3

In Q_3 we have $V = 8$ and $E = 12$. Since we are talking about the maximal embedding, we know that $B = 2$. This gives us that $g = 2$. Hence, the maximal embedding of Q_3 is on a 2-hole torus. As remarked before, the complete boundary walk length in Q_3 is 24 and there are a number of ways in which the maximal embedding can be achieved. Let us look at some of the examples. In all of these examples, whenever we are talking of making adjacent changes, it is with respect to the ABC rotation system as the starting point.

Case 1: We make one adjacent switch in the rotation system about any one vertex, say we do it for vertex 0. This gives us an embedding of Q_3 where the two boundary walks are of sizes 18 and 6 respectively. The boundary walks in this case are :

0-1-3-7-6-4-0-2-6-7-5-1-0-4-5-7-3-2-0

1-5-4-6-2-3-1

When the corresponding polygons are glued together we get the embedding on the 2-hole torus shown in Figure 3.11.

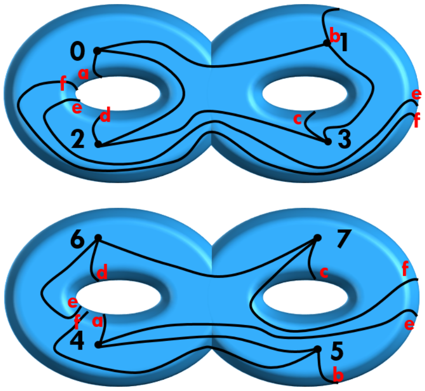


Figure 3.11. Maximal embedding of Q_3 with boundary walks of size 18 and 6.

Case 2: In this case, we make adjacent switches in the rotation system on vertices 0, 1, 3 and 7. This gives us an embedding of Q_3 where the two boundary walks are of size 12 each. The boundary walks in this case are :

0-1-5-4-6-2-3-7-5-1-3-2

0-4-5-7-6-4-0-2-6-7-3-1

When the corresponding polygons are glued together, we get the embedding on the 2-hole torus shown in Figure 3.12.

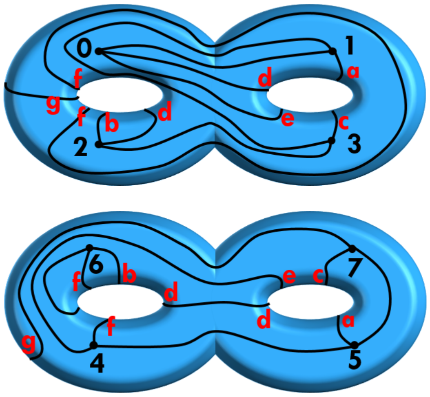


Figure 3.12. Maximal embedding of Q_3 with boundary walks of size 12 and 12

Case 3: In this case, we make adjacent switches in the rotation system on vertices 0 and 3, that is vertices that are Hamming distance 2 apart. This gives us an embedding of Q_3 where the two boundary walks are of sizes 20 and 4. The boundary walks in this case are :

0-1-3-2

0-4-5-7-3-1-5-4-6-2-3-7-6-4-0-2-6-7-5-1.

When the corresponding polygons are glued together we get the embedding on the 2-hole torus shown in Figure 3.13.

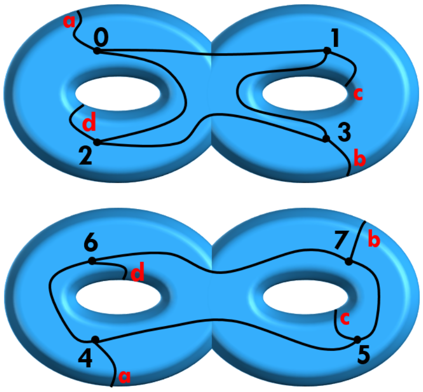


Figure 3.13. Maximal embedding of Q_3 with boundary walks of size 20 and 4

Case 4: For this we make adjacent switches in the rotation system on two connected vertices, for example, on vertex 0 and vertex 1. This gives us an embedding of Q_3 where the two boundary walks are of sizes 14 and 10 respectively. The boundary walks in this case are :

0-1-5-4-6-2-3-1-0-4-5-7-3-2

0-2-6-7-5-1-3-7-6-4.

When the corresponding polygons are glued together we get the embedding on the 2-hole torus given by Figure 3.14.

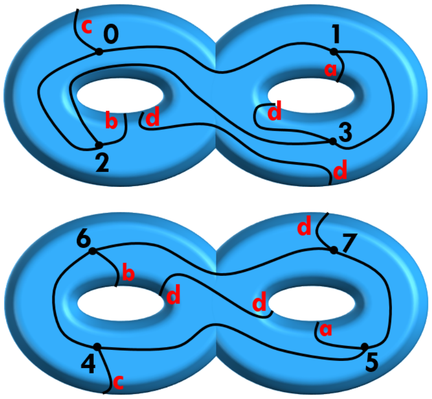


Figure 3.14. Maximal embedding of Q_3 with boundary walks of size 14 and 10

3.4.2 Maximal Eulerian Circuit Embedding

Definition 5. *A **maximal Eulerian circuit embedding** is defined as a maximal embedding where each of the boundary walks is an Eulerian circuit on the graph.*

Recall that an Eulerian circuit is a closed path in a graph that visits every edge of the graph exactly once. Also, an Eulerian circuit is possible only when the degree of each vertex in the graph is even. Hence, here we will be looking only at hypercube graphs of even dimension.

Let us look at some examples. First, we have the simple case of Q_2. Here we have $V = 4$, $E = 4$ and $B = 2$ (the latter because it is a maximal embedding). Using this, we have $g = 1 - \frac{V-E+B}{2} = 0$. Hence, this embeds on a surface of genus 0; that is, on the sphere. The two Eulerian circuits in this case are as follows

$$0 - 1 - 3 - 2 - 0$$

$$1 - 0 - 2 - 3 - 1$$

When these faces are glued together, we get the surface on which it is embedded, the sphere, as shown in the Figure 3.15. In Q_4, we have $V = 16$, $E = 32$. Since, we are considering the maximal embedding, we have exactly two boundary walks, or

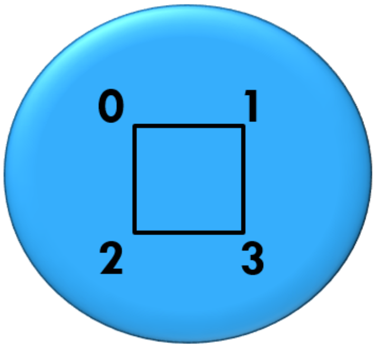


Figure 3.15. Eulerian Circuit embedding for Q_2

$B = 2$. Hence, $g = 8$, which means that the embedding is on a 8-hole torus. We obtain this embedding using the following rotation system :

(0 : 8, 4, 2, 1)
(1 : 0, 3, 5, 9)
(2 : 10, 6, 0, 3)
(3 : 2, 1, 7, 11)
(4 : 5, 6, 0, 12)
(5 : 4, 7, 1, 13)
(6 : 7, 4, 2, 14)
(7 : 6, 5, 3, 15)

(8 : 0, 12, 10, 9)
(9 : 8, 11, 13, 1)
(10 : 11, 8, 14, 2)
(11 : 10, 9, 15, 3)
(12 : 13, 14, 8, 4)
(13 : 12, 15, 9, 5)
(14 : 15, 12, 10, 6)
(15 : 14, 13, 11, 7)

This rotation system gives two boundary walks, each of which is an Eulerian circuit on Q_4.

0-1-3-7-15-14-12-8-10-14-6-7-5-1-9-8-0-4-12-13-15-11-3-2-10-11-9-13-5-4-6-2-0

1-0-8-12-4-5-7-3-11-10-8-9-11-15-7-6-4-0-2-3-1-5-13-12-14-10-2-6-14-15-13-9-1

Using four copies of the lattice from Figure 3.9, the first boundary walk can be represented as in Figure 3.16.

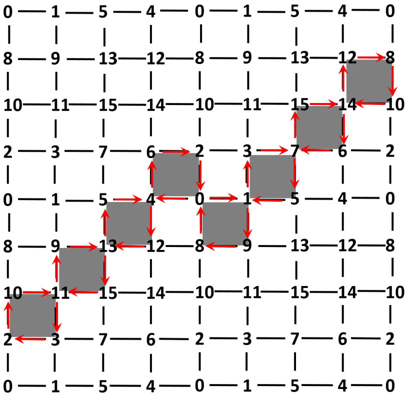


Figure 3.16. Eulerian Circuit lattice

It is interesting to note that, starting with the ABC rotation system, if we make changes at the vertices $8, 2$ and 10, we obtain another maximal Eulerian circuit embedding, with the corresponding boundary walks having the added property that each Eulerian circuit is made up of two Hamiltonian circuits (cycles visiting each vertex exactly once). The boundary walks are as follows :

0-1-3-7-15-14-12-8-10-11-9-13-5-4-6-2-0-4-12-13-15-11-3-2-10-14-6-7-5-1-9-8-0

0-2-3-1-5-13-12-14-10-8-9-11-15-7-6-4-0-8-12-4-5-7-3-11-10-2-6-14-15-13-9-1-0

To try to find an Eulerian circuit embedding for Q_6, we wrote a computer code (that we refer to as the boundary walk program) that iteratively searched for a pair of Eulerian circuits that when glued together would give a maximal embedding. However, we were not able to find one. We found an embedding where one of the boundary walks is an Eulerian circuit, but we had more than one other face, which means that that it was not a maximal embedding.

Conjecture 1. *Every hypercube of even dimension has a maximal Eulerian circuit embedding.*

3.4.3 Big-Face Maximal Embedding

We define a **big-face embedding** of Q_n as a maximal embedding where one face is the smallest possible, that is of size 4, and the other is the largest possible, that is of size of $n2^n - 4$. Before we provide a general construction of these kinds of embeddings, let us look at some examples.

In Q_3, we have already seen the big-face maximal embedding. In Figure 3.17, the face of size 4, with boundary walk $0-1-3-2$, is highlighted. The big-face here is:

$$0-4-5-7-3-1-5-4-6-2-3-7-6-4-0-2-6-7-5-1.$$

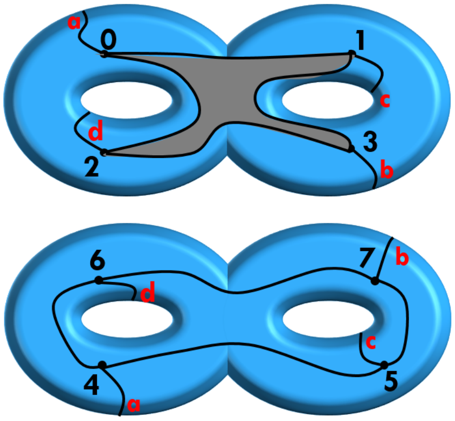


Figure 3.17. The big-face maximal embedding of Q_3

Using the lattice representation, the big-face embedding in Q_4 can be represented as shown in Figure 3.18. Here we start with the minimal embedding and make adjacent changes at the vertices circled in red. Making seven adjacent switches takes us from the minimal embedding to the big-face maximal embedding. We discuss this transition from minimal embedding to maximal embedding further in the next chapter. Experimenting with the boundary walk program, we were also able to find the big- face maximal embedding of Q_5, which has boundary walks as

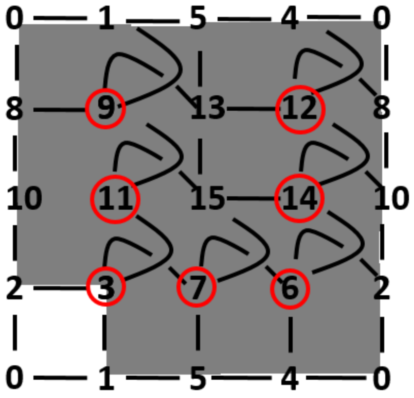
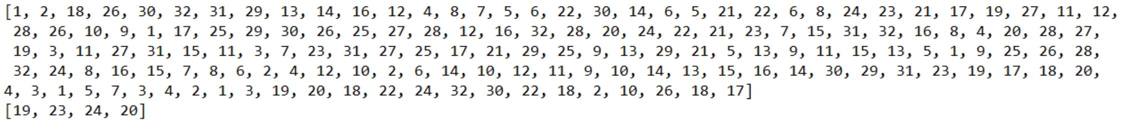


Figure 3.18. The big-face maximal embedding of Q_4

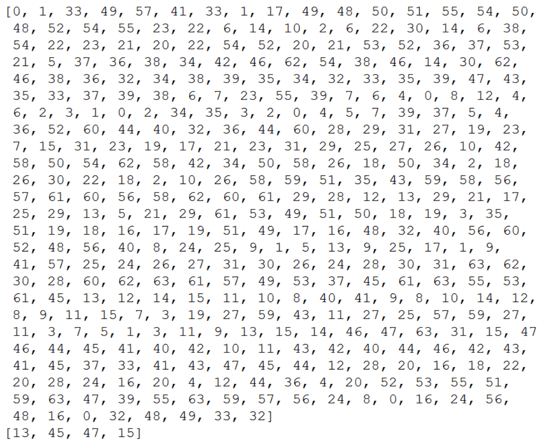
shown in Figure 3.19. The boundary walks are of sizes 156 and 4 respectively. The



```
[1, 2, 18, 26, 30, 32, 31, 29, 13, 14, 16, 12, 4, 8, 7, 5, 6, 22, 30, 14, 6, 5, 21, 22, 6, 8, 24, 23, 21, 17, 19, 27, 11, 12,
 28, 26, 10, 9, 1, 17, 25, 29, 30, 26, 25, 27, 28, 12, 16, 32, 28, 20, 24, 22, 21, 23, 7, 15, 31, 32, 16, 8, 4, 20, 28, 27,
 19, 3, 11, 27, 31, 15, 11, 3, 7, 23, 31, 27, 25, 17, 21, 29, 25, 9, 13, 29, 21, 5, 13, 9, 11, 15, 13, 5, 1, 9, 25, 26, 28,
 32, 24, 8, 16, 15, 7, 8, 6, 2, 4, 12, 10, 2, 6, 14, 10, 12, 11, 9, 10, 14, 13, 15, 16, 14, 30, 29, 31, 23, 19, 17, 18, 20,
4, 3, 1, 5, 7, 3, 4, 2, 1, 3, 19, 20, 18, 22, 24, 32, 30, 22, 18, 2, 10, 26, 18, 17]
[19, 23, 24, 20]
```

Figure 3.19. Boundary walks for the big-face maximal embedding of Q_5

boundary walk corresponding to the big-face embedding in Q_6 is given in Figure 3.20. Here the boundary walks are of sizes 380 and 4 respectively. In the next



```
[0, 1, 33, 49, 57, 41, 33, 1, 17, 49, 48, 50, 51, 55, 54, 50,
 48, 52, 54, 55, 23, 22, 6, 14, 10, 2, 6, 22, 30, 14, 6, 38,
 54, 22, 23, 21, 20, 22, 54, 52, 20, 21, 53, 52, 36, 37, 53,
 21, 5, 37, 36, 38, 34, 42, 46, 62, 54, 38, 46, 14, 30, 62,
 46, 38, 36, 32, 34, 38, 39, 35, 34, 32, 33, 35, 39, 47, 43,
 35, 33, 37, 39, 38, 6, 7, 23, 55, 39, 7, 6, 4, 0, 8, 12, 4,
 6, 2, 3, 1, 0, 2, 34, 35, 3, 2, 0, 4, 5, 7, 39, 37, 5, 4,
 36, 52, 60, 44, 40, 32, 36, 44, 60, 28, 29, 31, 27, 19, 23,
 7, 15, 31, 23, 19, 17, 21, 23, 31, 29, 25, 27, 26, 10, 42,
 58, 50, 54, 62, 58, 42, 34, 50, 58, 26, 18, 50, 34, 2, 18,
 26, 30, 22, 18, 2, 10, 26, 58, 59, 51, 35, 43, 59, 58, 56,
 57, 61, 60, 56, 58, 62, 60, 61, 29, 28, 12, 13, 29, 21, 17,
 25, 29, 13, 5, 21, 29, 61, 53, 49, 51, 50, 18, 19, 3, 35,
 51, 19, 18, 16, 17, 19, 51, 49, 17, 16, 48, 32, 40, 56, 60,
 52, 48, 56, 40, 8, 24, 25, 9, 1, 5, 13, 9, 25, 17, 1, 9,
 41, 57, 25, 24, 26, 27, 31, 30, 26, 24, 28, 30, 31, 63, 62,
 30, 28, 60, 62, 63, 61, 57, 49, 53, 37, 45, 61, 63, 55, 53,
 61, 45, 13, 12, 14, 15, 11, 10, 8, 40, 41, 9, 8, 10, 14, 12,
 8, 9, 11, 15, 7, 3, 19, 27, 59, 43, 11, 27, 25, 57, 59, 27,
 11, 3, 7, 5, 1, 3, 11, 9, 13, 15, 14, 46, 47, 63, 31, 15, 47
 46, 44, 45, 41, 40, 42, 10, 11, 43, 42, 40, 44, 46, 42, 43,
 41, 45, 37, 33, 41, 43, 47, 45, 44, 12, 28, 20, 16, 18, 22,
 20, 28, 24, 16, 20, 4, 12, 44, 36, 4, 20, 52, 53, 55, 51,
 59, 63, 47, 39, 55, 63, 59, 57, 56, 24, 8, 0, 16, 24, 56,
 48, 16, 0, 32, 48, 49, 33, 32]
[13, 45, 47, 15]
```

Figure 3.20. Boundary walks for the big-face maximal embedding of Q_6

section, we prove that big-face embeddings exist for Q_n for all n.

3.4.4 The Big-Face Maximal Embedding Theorem

We begin with a construction lemma that enables us to add edges to a graph and a rotation system without impacting the number of components in the corresponding embedding (although the genus of the embedding surface increases by one).

Lemma 2. *Given a graph G, and a rotation system such that in one boundary walk, directed edges e,e', connecting vertices v_0,v_1 and v_0', v_1', respectively, appear as illustrated in Figure 3.21. If a new graph G^* is formed by adding edges e_0, connecting v_0 and v_0', and e_1, connecting v_1 and v_1', and the rotation system for G is modified so that :*

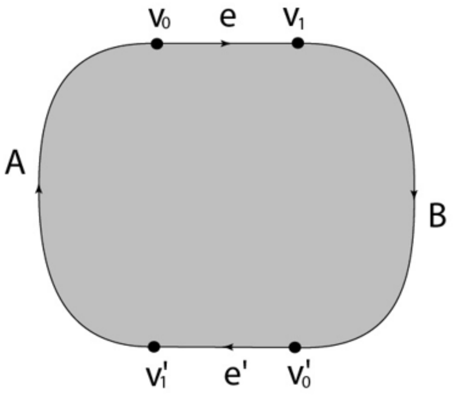


Figure 3.21. Boundary walk

1. *e_0 immediately precedes e in the ordering around v_0,*

2. *e_1 immediately follows e in the ordering around v_1,*

3. *e_0 immediately precedes e' in the ordering around v_0',*

4. *e_1 immediately follows e' in the ordering around v_1'.*

Then the original boundary walk $eBe'A$ becomes $ee_1Ae_0e'e_1^{-1}Be_0^{-1}$, with all the other boundary walks unchanged.

Proof. The proof of the lemma is going to be done by what is known as "cutting and pasting". Beginning with Figure 3.21, we connect vertices v_0 and v_0' by adding an edge e_0 consistent with the prescribed rotation system. So, now we have Figure 3.22. Now, in order to be able to add the edge e_1, connecting vertices v_1 and v_1', we need a tube that goes over the edge e_0. Note that adding a tube amounts to increasing the genus of the embedding surface by 1. The tube can be added by removing an open 2-disk from either side of the edge e_0 and gluing the circle boundaries together. In Figure 3.23, this is represented by the circle X.

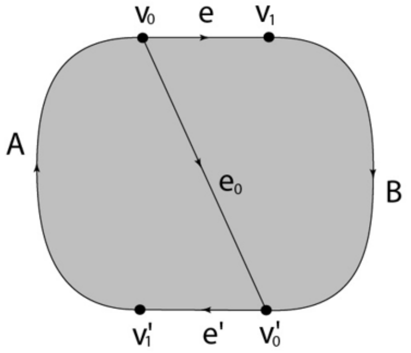


Figure 3.22. Inserting edge e_0

Now, we can insert the edge e_1 connecting vertices v_1 and v_1'. To do this, we connect v_1 to the circle X via edge e_{1a} which then come out of the other copy of circle X, from the corresponding position along edge e_{1b} before terminating at v_1'. This is shown in Figure 3.24.

To see that we now have the desired boundary walk, we do some cutting and pasting on Figure 3.24. To begin, we cut along the edges e_0 and e_1 to obtain

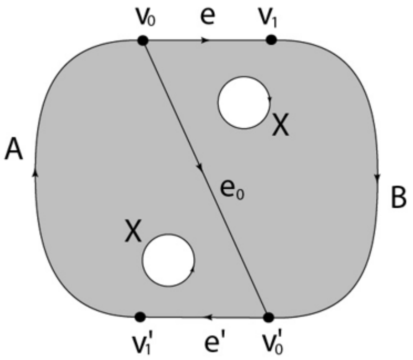


Figure 3.23. Inserting the tube

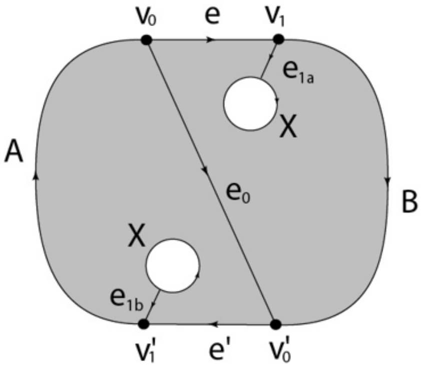


Figure 3.24. Inserting edge e_1

Figure 3.25, which gives us two parts, both of which are hexagons, with sides as indicated in Figure 3.26. What is implicit here is that these cut parts are glued back together again in the construction of the embedding surface from the polygon components. Finally, we paste the hexagons along X, and e_{1a} and e_{1b} combine to form e_1. The result is a face with the desired boundary walk, as shown in Figure 3.27. No other boundary walk component is impacted in doing this because the rotation system ordering at all other vertices did not change and because the four new directed edges appear in the modified boundary walk.

□

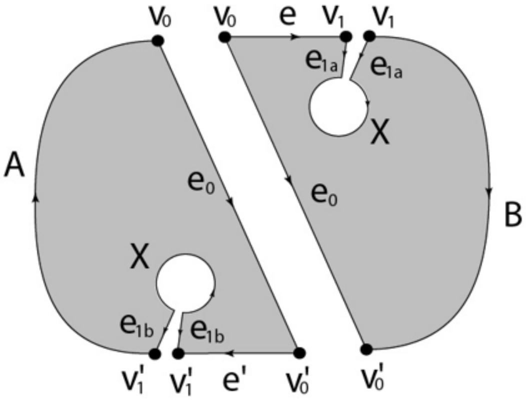


Figure 3.25. Cut along edges e_0 and e_1

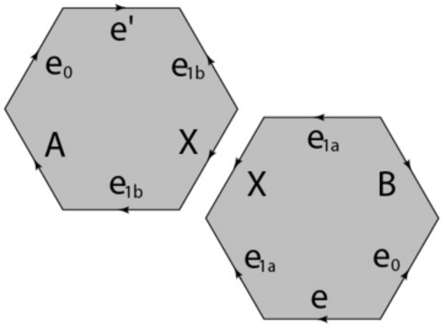


Figure 3.26. Hexagons resulting from the cutting

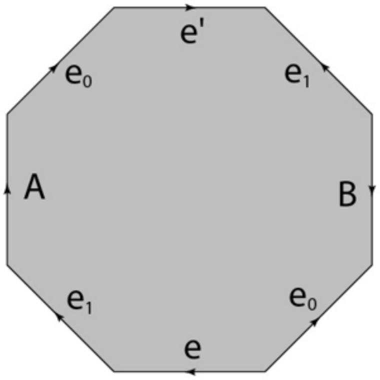


Figure 3.27. Result of pasting along X

With this lemma, we can now prove our big-face maximal embedding theorem.

Theorem 11. *For all $n \geq 2$, there is a rotation system for Q_n with two boundary walks where one is* $00...00 - 00...01 - 10...01 - 10...00 - 00...00$ *and the other is made up of the remaining directed edges in Q_n, and therefore the corresponding embedding is a big-face embedding of Q_n.*

Proof. The result holds for $n = 2$, the boundary walks being $00 - 01 - 11 - 10$ and $01 - 00 - 10 - 11$ and the rotation system being :

$$0 : 1,\ 2$$
$$1 : 3,\ 0$$
$$2 : 3,\ 0$$
$$3 : 1,\ 2$$

We will now induct on the dimension of the graph (n). Assume that the statement is true and we have a rotation system on Q_{n-1} as specified by the statement of the theorem.

Let S be the embedding surface for Q_{n-1}. From S, we remove the "small face" $00...00 - 00...01 - 10...01 - 10...00$. We call the resulting surface S_0 (Figure 3.28). Now, to each of the vertices in S_0 we append a 0 on the left in its binary representation to obtain a vertex in Q_n. (Note that although the decimal

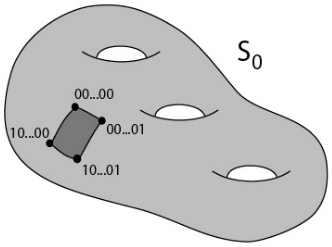


Figure 3.28. Surface S_0 after removing 4 sided face

equivalent has not changed, we have increased the length of the binary string from $n-1$ to n.)

Next, we take another copy of Q_{n-1} with the rotation system corresponding to the big-face embedding. For this copy, we reverse the rotation system at each vertex. This gives us a big-face embedding with the small face

$$00...00 - 10...00 - 10...01 - 00...01 - 00...00$$

Let S_1 be the embedding surface with the "small face" removed. We now append 1 on the left in the binary representation of the vertices in S_1 to obtain the rest of the vertices in Q_n. Note that we have the rotation system orderings for each of the vertices in the original Q_{n-1} that we started with.

To obtain Q_n from these two copies of Q_{n-1}, we need to add edges connecting $0a_1a_2...a_{n-1}$ and $1a_1a_2...a_{n-1}$, which are the edges connecting each pair of vertices, one on each of the Q_{n-1} hypercubes. Furthermore, we need to place these edges properly in each vertex's rotation system ordering to obtain the desired rotation system and embedding.

Align S_0 and S_1 and add the four edges going across as illustrated in Figure 3.29. We add these edges to the rotation system ordering at each vertex in such a way

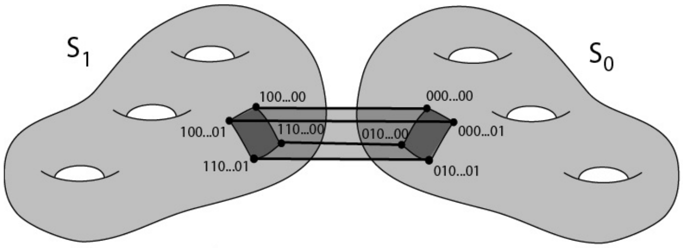


Figure 3.29. Surface S_0 and S_1 with four edges connected across

that four 4-sided faces result, creating a tube that connects S_0 and S_1. Denote the faces T(top), B(bottom), F(front), K(back), as they appear in Figure 3.29.

For example, originally 00...01 immediately followed 01...00 in the rotation system ordering at 00...00. We add 10...00 between them to get the new rotation system around 00...00. Let G_0 denote the graph that results. It is made up of two copies of Q_{n-1} and 4 new edges. We have an embedding of G_0 in a surface such that there are six faces : S_0, S_1, T, B, F, K.

In the rotation system at 11...00 switch the edges corresponding to the vertices 01...00 and 11...01. By Theorem 5, of the 6 original faces, S_1, K, B combine to form one face, say H. Hence, we now have 4 faces.

Next, in the rotation system at 01...01 switch the edges corresponding to the vertices 11...01 and 00...01. By Theorem 5, of the four faces from the last step, H, F, S_0 combine to form one face. This results in a new rotation system and an embedding of G_0 with 2 faces: the original face T,

$$00...00 - 00...01 - 10...01 - 10...00 - 00...00,$$

and T', the face obtained by merging the rest.

Now, we are in the position to add the missing edges to form Q_n. We can do this two edges at a time so that at each stage Lemma 2 implies we have graph embedding with two faces, T and a face determined by the other directed edges in the graph. When this addition process is completed, we have the desired rotation system and a big-face maximal embedding.

It remains to prove that the edge-addition process can be carried out. To that end, by Lemma 1, we can order the vertices in $Q_{n-1} - \{00...00, 00...01, 10...01, 10...00\}$

with a Gray code ordering $v_1, v_2, ..., v_p$. In the boundary walk for T' in G_0, directed edges $0v_1 - 0v_2$ and $1v_1 - 1v_2$ appear (as do $0v_2 - 0v_1$, and $1v_2 - 1v_1$). By Lemma 2, we can add edges $0v_1 - 1v_1$ and $0v_2 - 1v_2$ to G_0 and the rotation system to obtain a graph G_1 such that the resulting surface embedding has two faces, T and a face made up of the rest of the directed edges in G_1. Similarly, we can build G_2, adding edges $0v_3 - 1v_3$ and $0v_4 - 1v_4$ to G_1, and we can continue the process, adding the needed edges to obtain Q_n and a rotation system with the desired property.

Thus, if result holds for $n - 1$, it holds for n as well, and by induction we can conclude that for all n, there exists a big-face maximal embedding of Q_n.

□

Chapter 4
FURTHER THOUGHTS

4.1 The spectrum of embeddings from minimal to maximal

In this section we demonstrate a process where we can sequentially go from the minimal embedding of Q_4 to a maximal embedding of Q_4. In doing so, we cover a spectrum of embeddings of Q_4 in surfaces of genus $g, \gamma(G) \leq g \leq \gamma_M(G)$, where $\gamma(G)$ is the genus of the graph and $\gamma_M(G)$ is the maximal genus. We achieve this by using Theorem 5 and making a sequence of adjacent changes.

In Q_4 we start with the rotation system corresponding to the minimal embedding. As we have seen before, the polygons resulting from the boundary walks corresponding to this rotation system can be represented as the lattice in Figure 4.1. Now, using Theorem 5, if we switch the rotation system around vertex 3, specifically switching the entries 7 and 11, the 3 associated faces will combine to form one bigger face corresponding to the shaded region in Figure 4.2. Note that

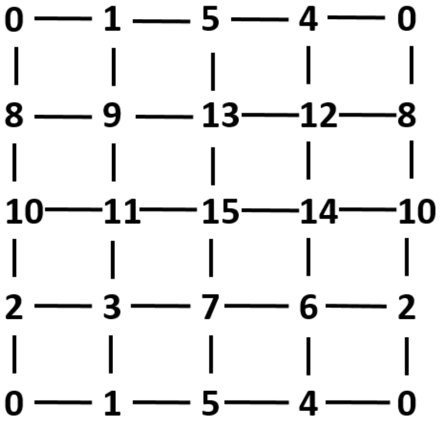


Figure 4.1. Lattice corresponding to the minimal embedding of Q_4

this amounts to adding a handle in the $3-7-15-11$ square and switching the edges so that $3-11$ goes over the handle and $3-7$ goes through the handle to accomplish the edge switch. We can continue this process, changing the rotation

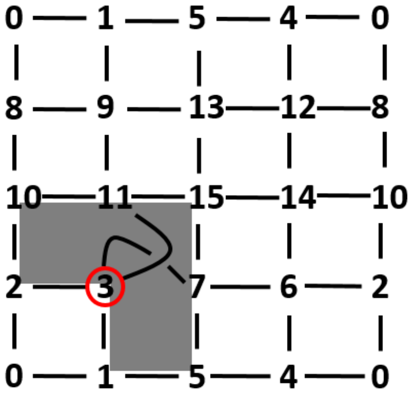


Figure 4.2. Lattice after switching the rotation system around 3

system around 7, to combine 2 more faces and obtain the lattice as shown in Figure 4.3. We can continue this process further until we have combined all the faces

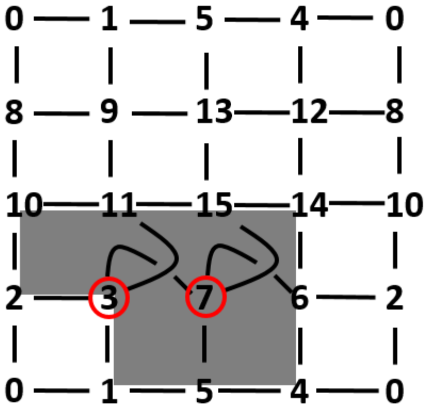


Figure 4.3. Lattice after switching the rotation system around the circled vertices

except the one labeled 0132. At this point, we have attained the big-face maximal embedding in Q_4. The vertices around which the rotation system is changed are circled in the Figure 4.4, and the regions that combine to form the big-face has been shaded. Notice that each switch adds a handle to the embedding surface, so

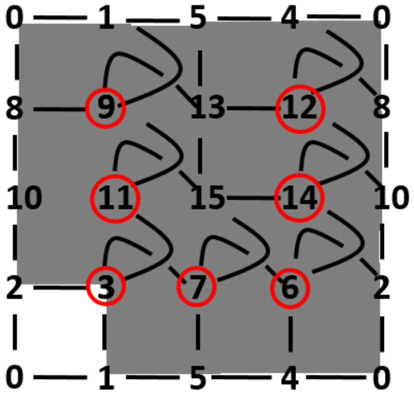


Figure 4.4. Lattice for the big-face maximal embedding in Q_4

we start with a torus for the minimal embedding, and with each switch increase the genus by one, ending with a maximal embedding in the 8-hole torus.

4.2 Some open questions

Here is a list of some open questions that we encountered during our investigation :

- For Q_n can we find a set of adjacent changes that takes us monotonically from the minimal embedding to the maximal embedding? We demonstrated an example in Q_4 of how this can be done. But, can such a series of switches be found for higher dimensions as well?

- Does there exist an Eulerian circuit embedding for every Q_n of even dimension? We found an example of a rotation system on Q_4 where each of the boundary walks is an Eulerian circuit on Q_4. Can this be generalized for higher dimensions?

- Can we find an explicit rotation system for the big-face maximal embedding of Q_n? We showed a recursive construction for the big-face maximal embedding, but is there a closed form for the rotation system corresponding to the big-face maximal embedding?

REFERENCES

[1] Jungerman, Mark "A characterization of upper embeddable graphs." American Mathematical Society, Volume 241, July 1978

[2] Mohar, Bojan, and Thomassen, Carsten "Graphs on Surfaces" Johns Hopkins University Press

[3] Gross, L. Jonathan, and Tucker, W. Thomas "Topological Graph Theory" Courier Corporation, 1987

[4] Potoczak, Sophia N., and Robert D. Franzosa "A survey of graph embeddings into compact surfaces" 2014. Print

[5] Robert D. Franzosa, and Adams Colin "Introduction to Topology: Pure and Applied" Pearson, 2009

[6] Harary, Frank "Graph Theory" Addison-Wesley Publishing Company, 1969

[7] Bieneke, W. Lowell, and Harary, Frank "The genus of the n-cube" Canadian Journal of Mathematics, 1965

BIOGRAPHY OF THE AUTHOR

Prateek Kunwar was born in Siliguri, India and completed his schooling from Don Bosco School in Delhi, India. He completed his bachelor's degree in electrical engineering from Delhi College of Engineering, Delhi in May, 2013. After that he worked as an associate in technology at Sapient Corporation, Gurgaon for a year. Prateek Kunwar is a candidate for the Master of Arts degree in Mathematics from The University of Maine in December 2016.