TWO NETWORKS DERIVED FROM THE HYPERCUBE

by

Yiu Ming Sammy Ma

B.Sc. (Hons.), Simon Fraser University, 1993

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the Department of Mathematics and Statistics

SIMON FRASER UNIVERSITY
August 1995

APPROVAL

Name: Ma, Yiu Ming Sammy

Degree: Master of Science

Title of Thesis: Two Networks Derived from the Hypercube

Examining Committee: Dr. G.A.C. Graham
Chair

Dr. B. Alspach, Senior Supervisor

Dr. L. Goddyn

Dr. P. Hell

Dr. T.C. Brown, External Examiner
Department of Mathematics and Statistics, SFU

Date Approved: August 1, 1995

Title of Thesis/Project/Extended Essay

TWO NETWORKS DERIVED FROM THE HYPERCUBE

Author:_
(signature)

MA, YIU MING
(name)

JUNE 30, 1995
(date)

Abstract

The hypercube is one of the most popular interconnection networks. Not only does it have good topological structure but also nice symmetric properties. However, it has a major drawback that the degree is not bounded as the dimension increases. Because of this, some networks with bounded degree have been derived from the hypercube. Two of the most popular are butterfly graphs and cube-connected-cycles. They both inherit some properties from the hypercube. This thesis investigates these two networks.

Acknowledgment

I would like to thank Dr. Brian Alspach for suggesting the topic and his guidance of completing the thesis.

Contents

List of Figures

Chapter 1

Introduction

Networks can be divided into two classes: static networks and dynamic networks. The physical structure of dynamic networks is not fixed. It can be changed by modifying the configuration of the switches in the connection cables. On the other hand, the physical structure of static networks is fixed. It can be modelled by the tools from Graph Theory. In this thesis, only static networks are considered.

Before constructing a network, one has to consider many factors such as the hardware cost, the performance, the reliability and the expandibility. The first three are obviously important factors. For the expandibility, the network must be designed to minimize the total amount of modification when more processors are being added to it. The simplest method of building a network is by putting a link between a new processor and one of the processors randomly chosen in the network. The cost for this kind of network is very low because not many links are required. The expandibility is clearly very high because almost no modification is required when a new processor is added. However, this kind of connection is unpredicable. The network may turn out to be a simple path. That means if any one of the links or one of the processors breaks down, the network will not be connected. Furthermore, the time for communication is very long for the processors at the end of the path. Thus, this kind of network in general is neither reliable nor efficient. Another extreme method is to connect every pair of processors. This kind of network has the maximum performance and reliability. Each processor in the network can directly communicate with every other. Also, the network is always connected no matter how many processors are out-of-function. However, the cost will be very high. The expandibility is also not very good because the number of links going out from each processors is not the same for the networks of different size. Thus, every processor must be modified if the network need to be expanded. These two examples show that building a network is a trade-off problem.

For the hardware cost, one should consider the number of links being used. In order to increase the performance, the network should be designed so that any pair of processors can communicate easily. One measure of this is that the diameter of the network should be small.

The design of a network also affects the software cost. If processors can be addressed using binary numbers, the system software will be simpler because most of the other components in the whole system are binary-based. Futhermore, if the network is exactly "the same" with respect to each processor, the processors can share a single routing table. The network is said to have good symmetry properties. Symmetry properties also affect the reliability of the network. Good symmetric network can re-order the processors or communication lines so that some particular processors can still communicate,

This thesis will present two methods for constructing a network that has good symmetry properties. One of them is the Cayley graph construction and the other one is the set graph construction. This thesis will mainly discuss the Cayley graph construction.

The n-dimensional (binary) hypercube is one of the most popular networks that is formed by the Cayley graph construction. It does not have too many links but it has very good performance. The routing in the hypercube is extremely easy. The reliability is also very high. Moreover, it possesses all the symmetry properties that one usually studies. Unforturnately, it has one major drawback in that the number of links going out from the processors increases as the dimension increases. It reduces the expandibility dramatically.

There are some extensions coming from the hypercube. The two popular ones are the butterfly graph and the cube-connected-cycle. Both of them use an n-cycle to replace each processor in the the hypercube so that the degree of each processor can be fixed. They inherit many of the topological properties from the hypercube, but they also destroy many of th6e symmetry properties. This means the performance of the butterfly graph and the cube-connected-cycle will be almost the same as the hypercube, but they have lower reliability. This thesis will investigate these two networks, and will discuss the topological structure as well as the symmetry properties of the butterfly graph and the cube-connected-cycles. It also does some comparisons between these two kinds of networks with the hypercube.

Chapter 2

Groups of Permutations

2.1 Permutation Groups

Permutation groups play a very important role in group theory. In fact, Cayley's Theorem [5] says that every finite group is isomorphic to a group of permutations. In this section, certain notions regarding permutation groups will be presented.

Given the set $\{1, \ldots, n\}$, one can think of a permutation as a rearrangement of the numbers. The following is a formal definition of a permutation [5].

Definition 2.1 A function $f : A \rightarrow B$ is *one-to-one* if every element of B has at most one element of A mapped to it.

Definition 2.2 A function $f : A \rightarrow B$ is *onto* if every element of B has at least one element of A mapped to it.

Definition 2.3 A *permutation* of a set A is a one-to-one and onto function from A to A.

In this chapter, all permutations are on the set $\{1, \ldots, n\}$. The collection of all permutations of $\{1, \ldots, n\}$ is usually denoted by S_n.

A permutation, $\sigma \in S_n$ can written in several ways. One of them is to list all $(x, \sigma(x))$ pairs in an array as:

$$\sigma = \begin{pmatrix} 1 & 2 & \dots & i & \dots & n \\ \sigma(1) & \sigma(2) & \dots & \sigma(i) & \dots & \sigma(n) \end{pmatrix}.$$

In fact, all permutations in S_n have the same first row. This row is actually redundant. Hence, σ can be written as:

$$\sigma = \sigma(1)\sigma(2)\dots\sigma(i)\sigma(n)$$

Definition 2.4 Let $\rho, \sigma \in S_n$. A binary operation $\cdot$ is defined as the composition of functions, that is, $(\rho \cdot \sigma)(x) = \rho(\sigma(x))$ for all $x \in \{1, \dots, n\}$.

Proposition 2.5 *S_n is closed under $\cdot$.*

Proof: Let $\rho, \sigma \in S_n$. For any $x, y \in \{1, \dots, n\}, (\rho \cdot \sigma)(x) = (\rho \cdot \sigma)(y) \Rightarrow \rho(\sigma(x)) = \rho(\sigma(y))$. Since ρ is one-to-one, $\sigma(x) = \sigma(y)$. Again, σ is one-to-one implying that $x = y$. Hence, $\rho \cdot \sigma$ is one-to-one.

Now for any $y \in \{1, \dots, n\}$, there is $x \in \{1, \dots, n\}$ such that $y = \rho(x)$. Again, there is $z \in \{1, \dots, n\}$ such that $x = \sigma(z)$. Hence, $y = \rho(\sigma(z)) = (\rho \cdot \sigma)(z)$ and $\rho \cdot \sigma$ is onto. Thus $\rho \cdot \sigma$ is a permutation in S_n. $\square$

Proposition 2.6 *$(S_n, \cdot)$ is a group.*

Proof: Let e be the permutation such that $e(x) = x$ for all $x \in \{1, \dots, n\}$. Then for any $\rho \in S_n$, $(\rho \cdot e)(x) = \rho(e(x)) = \rho(x)$ for all $x \in \{1, \dots, n\}$. Hence, e is an identity.

For any $\rho, \sigma, \tau \in S_n$, $((\rho \cdot \sigma) \cdot \tau)(x) = (\rho \cdot \sigma)(\tau(x)) = \rho(\sigma(\tau(x))) = \rho((\sigma \cdot \tau)(x)) = (\rho \cdot (\sigma \cdot \tau))(x)$ for all $x \in \{1, \dots, n\}$. Hence, $\cdot$ is associative.

Let ρ be a permutation. As ρ is one-to-one and onto, the inverse function ρ^{-1} of ρ exists, and ρ^{-1} is also one-to-one and onto. We have $(\rho \cdot \rho^{-1})(x) = x = e(x)$ for all $x \in \{1, \dots, n\}$. Hence, $(S_n, \cdot)$ is a group. $\square$

$(S_n, \cdot)$ is usually called the symmetric group of degree n.

2.2 Orbits and Cycles

Given a permutation σ, one can partition $\{1,\ldots,n\}$ using an appropriate relation $\sim$ defined as follows: For any $a, b \in \{1,\ldots,n\}, a \sim b$ if and only if $b = \sigma^n(a)$ for some integer n.

Proposition 2.7 *The relation $\sim$ is an equivalence relation.*

Proof:

Reflexive $a \sim a$ because $a = e(a) = \sigma^0(a)$.

Symmetric $a \sim b \Rightarrow b = \sigma^n(a)$ for some integer n. So $a = \sigma^{-n}(b)$ and $b \sim a$.

Transitive $a \sim b, b \sim c \Rightarrow b = \sigma^n(a), c = \sigma^m(b)$ for some integers m and n. So $c = \sigma^m(\sigma^n(a)) = \sigma^{m+n}(a)$ and $a \sim c$. $\square$

Definition 2.8 Let $\sigma \in S_n$, the equivalence classes determined by $\sim$ are called the *orbits* of σ.

Hence a permutation partitions the set $\{1,\ldots,n\}$ into orbits. This idea provides a method to decompose a permutation into a set of simple permutations.

Another way to describe a permutation is to use a digraph. Let $\sigma \in S_n$ and $V = \{1,\ldots n\}$ be the vertex-set. There is an arc from i to j if and only if $\sigma(i) = j$. We denote this digraph by D_σ. Clearly, a directed cycle in D_σ corresponds to an orbit in σ.

Theorem 2.9 *The associated digraph D_σ of σ consists of a set of vertex-disjoint directed cycles.*

Proof: As σ is a function, the outdegree of each vertex in D_σ is 1. Since σ is one-to-one and onto, the indegree of each vertex in D_σ is also 1. Hence, D_σ consists of a set of vertex-disjoint directed cycles. $\square$

Theorem 2.9 says that we can decompose D_σ into a set of vertex-disjoint directed cycles, $C_1, C_2, \ldots, C_k$. Let $B_1, B_2, \ldots, B_k$ be the directed spanning subgraphs induced by $C_1, C_2, \ldots, C_k$, respectively, and add a directed loop to each isolated vertex. Each B_i will give us a permutation. Those permutations have at most one orbit containing more than one element. Since the permutations come from the cycles of D_σ, these permutations are also named *cycles*.

Definition 2.10 A permutation $\sigma \in S_n$ is a *cycle* if σ has at most one orbit containing more than one element. The *length* of a cycle is the number of elements in the largest orbit.

Definition 2.11 Two cycles are said to be *disjoint* if their orbits that contain more than one element do not have any element in common.

Corollary 2.12 *Every permutation $\sigma \in S_n$ can be written as a product of disjoint cycles.*

2.3 Cyclic Notation

By the definition, a cycle has at most one orbit containing more than one element. So given the list of the elements in the largest orbit of a cycle is sufficient to determine the whole structure of the permutation. For example, if the largest orbit of a cycle $c \in S_8$ is $(1\ 8\ 3\ 5)$, then

$$c = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 8 & 2 & 5 & 4 & 1 & 6 & 7 & 3 \end{pmatrix}.$$

The notation, $c = (1\ 8\ 3\ 5)$ is called *cyclic notation.*

Corollary 2.12 says that every permutation can be written as a product of disjoint cycles. For example $p = 6734152$ can be written as $(1\ 6\ 5)(2\ 7)$.

2.4 Transpositions and Inversions

Other than the identity, every cycle has length at least 2. This means that a cycle of length 2 has the simplest structure. However, it has a special property. In cyclic notation, a cycle of length 2 can be written as $(i\ j)$. In explicit notation, it will look like $1 \ldots i-1\ j\ i+1 \ldots j-1\ i\ j+1 \ldots n$. Given a cycle p of length 2 and a permutation $\sigma = a_1 a_2 \ldots a_n$ in explicit notation. If p is multiplied on the right of σ, it will exchange the ith and jth elements in σ. Similarly if p is multiplied on the left of σ, it will exchange i and j in σ. Hence, a cycle of length 2 is called a *transposition*. Using this property, one can obtain any permutation by exchanging suitable pairs of elements.

Theorem 2.13 *Every permutation in $S_n, n \geq 2$, can be written as a product of transpositions.*

Proof: It is sufficient to prove that every cycle can be written as a product of transpositions. For any cycle $c = (a_1\ a_2\ \cdots\ a_k)$ in cyclic notation, $k \geq 2$, it can be written as

$$c = (a_1\ a_k)(a_1\ a_{k-1}) \cdots (a_1\ a_2).$$

If the length of c is 1, then c is the identity and can be written as $(1\ 2)(1\ 2)$.

□

Another property of a transposition is that every transposition is the inverse of itself. Suppose σ can be written as a product $p_1 p_2 \cdots p_k$ of k transpositions. Then $e = \sigma p_k p_{k-1} \cdots p_1$, where e is the identity. This is the basic idea of a sorting algorithm.

Definition 2.14 Let σ be a permutation. If $\sigma(j) < \sigma(i)$ for $i < j$, then the pair $(\sigma(j), \sigma(i))$ is called an *inversion* of σ.

Transpositions and inversions in fact are the crucial parts of the sorting algorithm based on *comparison*. The rest of this section will discuss the relationship between transpositions and inversions.

Definition 2.15 A permutation σ is said to be an *odd* or *even* permutation if the number of inversions in σ is odd or even, respectively.

Lemma 2.16 *Let $w = a_1a_2 \cdots a_n$ be a permutation and $p = 1 \ldots i-1\ j\ i+1 \ldots j-1\ i\ j+1 \ldots n$ be a transposition. The number of inversions in w and in wp has different parity.*

Proof: We have $wp = a_1a_2 \cdots a_{i-1}a_ja_{i+1} \cdots a_{j-1}a_ia_{j+1} \cdots a_n$. It is sufficient to consider the subsequence $a_ja_{i+1} \cdots a_{j-1}a_i$.

If $a_i < a_j$, then (a_i, a_j) is an inversion in wp but not in w. For any $a_i < a_k < a_j$, $i+1 \leq k \leq j-1$, both (a_k, a_j) and (a_i, a_k) are inversions in wp but not in w. If $a_i < a_j < a_k$ or $a_k < a_i < a_j$, $i+1 \leq k \leq j-1$, then the number of inversions involving a_k in wp is the same as in w. Thus, the number of inversions in wp is increased by an odd number.

If $a_i > a_j$, then (a_j, a_i) is an inversion in w but not in wp. For any $a_i > a_k > a_j$, $i+1 \leq k \leq j-1$, both (a_k, a_i) and (a_j, a_k) are inversions in w but not in wp. If $a_i > a_j > a_k$ or $a_k > a_i > a_j$, $i+1 \leq k \leq j-1$, then the number of inversions involving a_k in w is the same as in wp. Thus, the number of inversions in wp is decreased by an odd number. The result follows. $\square$

Theorem 2.17 *Let $w = p_1p_2 \cdots p_k$ be a product of k transpositions. Then w is even (or odd) if and only if k is even (or odd).*

Proof: Let e be the identity. The number of inversions in e is 0. When $k = 1$, k is odd and $w = p_1 = ep_1$. By Lemma 2.16, w has an odd number of inversions, that is, w is odd. Suppose it is true for $k = r-1$. Consider $k = r$. By Lemma 2.16, $w = p_1 \cdots p_{r-1}p_r$ and $p_1 \cdots p_{r-1}$ have different parity. The result follows. $\square$

Corollary 2.18 *If σ is even (or odd), then σ can only be written as a product of an even (or odd) number of transpositions.*

Proof: By Theorem 2.13, σ is a product of transpositions. Let $\sigma = p_1p_2 \cdots p_k$, where the p_i's are transpositions. By Theorem 2.17, the result follows. $\square$

2.5 Conjugacy

We again consider the associated directed graph D_σ of the permutation σ again. D_σ consists of a set of vertex-disjoint cycles. The length of the cycles can be any number from 1 to n, so let $[\lambda_1(\sigma), \lambda_2(\sigma), \ldots \lambda_n(\sigma)]$ be an n-tuple, where $\lambda_i(\sigma)$ is the number of cycles of length i. We define a relation $\sim_c$ as follows: For any σ_1, $\sigma_2 \in S_n$, $\sigma_1 \sim_c \sigma_2$ if and only if

$$[\lambda_1(\sigma_1), \lambda_2(\sigma_1), \ldots \lambda_n(\sigma_1)] = [\lambda_1(\sigma_2), \lambda_2(\sigma_2), \ldots \lambda_n(\sigma_2)].$$

Definition 2.19 The relation $\sim_c$ is called *conjugacy.*

Theorem 2.20 *Conjugacy is an equivalence relation.*

Proof: The proof is trivial. □

2.6 Stabilizer

There are special subgroups in a permutation group which we now define.

Definition 2.21 Let $(B, \cdot)$ be a permutation group. Let $u \in \{1, \ldots, n\}$ and $B_u = \{\alpha : \alpha \in B \text{ and } \alpha(u) = u\}$. B_u is called the *stabilizer* of u.

Proposition 2.22 *We have that* $(B_u, \cdot)$ *is a subgroup of* $(B, \cdot)$.

Proof: Let $\alpha, \beta \in B_u$. Then $(\alpha \cdot \beta)(u) = \alpha(\beta(u)) = \alpha(u) = u$. So $\alpha \cdot \beta \in B_u$. Since $e(u) = u$, $e \in B_u$ If $\alpha \in B_u$, $\alpha(u) = u$. So $\alpha^{-1}(u) = u$, that is, $\alpha^{-1} \in B_u$. Hence, $(B_u, \cdot)$ is a subgroup of $(B, \cdot)$. □

2.7 Transitive groups and regular groups

There are certain permutation groups that play important roles in group theory. Some of them will be used in the coming chapters.

Definition 2.23 Let Γ be a permutation group on $\{1,\ldots,n\}$. Γ is *transitive* if for each pair $i,j \in \{1,\ldots,n\}$, there exists a $\sigma \in \Gamma$ such that $\sigma(i) = j$.

Definition 2.24 A permutation group Γ on $\{1,\ldots,n\}$ is said to be *regular* if Γ is transitive and for each $i \in \{1,\ldots,n\}$, the stabilizer Γ_i of i is $\{e\}$.

Theorem 2.25 *A permutation group Γ on $\{1,\ldots,n\}$ is regular if and only if for any pair $i,j \in \{1,\ldots,n\}$, there is a unique permutation $\sigma \in \Gamma$ such that $\sigma(i) = j$.*

Proof: Since Γ is transitive, it is sufficient to show that σ is unique. Suppose there are two permutations σ_1 and σ_2 such that $\sigma_1(i) = \sigma_2(i) = j$. Then $e(i) = i = \sigma_1^{-1}\sigma_1(i) = \sigma_1^{-1}\sigma_2(i)$. Therefore, $\sigma_1^{-1}\sigma_1 = \sigma_1^{-1}\sigma_2$ implies that $\sigma_1 = \sigma_2$. $\square$

Chapter 3

Cayley Graphs and Transposition Graphs

3.1 Cayley Graphs

Cayley graphs are an important class of graphs constructed from groups. They reflect not only the group structure but also possess some nice graph properties.

Definition 3.1 Let $(\Gamma, \cdot)$ be a finite group with identity e. Let S be a subset of Γ such that

1. if $g \in S$, then $g^{-1} \in S$, and
2. $e \notin S$.

The *Cayley graph* $G(\Gamma, S)$ is defined as follows.

1. The vertex set of $G(\Gamma, S) = \Gamma$.
2. The edge set of $G(\Gamma, S) = \{xy : x, y \in \Gamma$ and there exists $g \in S$ such that $y = x \cdot g\}$.

The set S is called the *symbol* of $G(\Gamma, S)$.

Proposition 3.2 *S generates Γ if and only if $G(\Gamma, S)$ is connected.*

Proof: Let $S = \{a_1, \ldots, a_r\}$. Suppose S generates Γ. Let $x, y \in G(\Gamma, S)$ and $g = x^{-1}y$. Since $\Gamma = \langle S \rangle$, $g = x^{-1}y = a_{i_1}a_{i_2}\cdots a_{i_l}$. Hence, $y = xx^{-1}y = xa_{i_1}a_{i_2}\cdots a_{i_l}$. This implies x, y are connected by a path. Conversely, suppose $G(\Gamma, S)$ is connected. Let $g \in \Gamma$. There is a path from e to g in $G(\Gamma, S)$. So $g = a_{i_1}a_{i_2}\cdots a_{i_k}$. This implies S generates Γ. □

Since there is no reason to consider a disconnected interconnection network, all symbol sets in this thesis will be assumed to be generator sets.

3.2 Transposition Graphs

Another kind of graph that is determined by a permutation group is a transposition graph. Studying transposition graphs is not useful because every simple graph is a transposition graph. However, the transposition graph corresponding to the Cayley graph generated by a permutation group has some special characteristics.

Definition 3.3 Let $(\Gamma, \cdot)$ be a permutation group on A. Let S be a set of transpositions in Γ. The *transposition graph* $TG(A, S)$ is defined as follows:

1. The vertex set of $TG(A, S) = A$, and
2. The edge set of $TG(A, S) = \{xy : (x, y) \in S\}$.

Definition 3.4 A transposition graph which is a tree is called a *transposition tree.*

Theorem 3.5 (Pòlya) *A set $\Omega \subseteq S_n$ of $(n-1)$ transpositions generates the symmetric group S_n if and only if the transposition graph $TG(S_n, \Omega)$ is a transposition tree.*

Proof: Suppose $TG(S_n, \Omega)$ is a tree. Then any two vertices are connected by a unique path. Let $a, b \in \{1, \ldots, n\}$ and

$$a, (a\ x_1), x_1, (x_1\ x_2), \ldots, (x_{k-1}\ x_k), x_k, (x_k\ b), b$$

be the path joining a and b. Then

$$(a\ b) = (a\ x_1)(x_1\ x_2)\cdots(x_{k-1}\ x_k)(x_k\ b)(x_{k-1}\ x_k)\cdots(x_1\ x_2)(a\ x_1)$$

which is a product of transpositions in Ω.

Conversely, suppose Ω generates S_n. Let $(x\ y) = p_1 p_2 \cdots p_k$, where $p_1, p_2, \ldots, p_k \in \Omega$. Then $p_{i_1} = (x\ x_1)$ for some x_1 and $1 \leq i_1 \leq k$. Similarly, $p_{i_2} = (x_1\ x_2)$ for some x_2 and $1 \leq i_2 \leq k$, $p_{i_3} = (x_2\ x_3)$ for some x_3 and $1 \leq i_3 \leq k$, and so forth. Finally, $p_{i_r} = (x_{r-1}\ y)$ for some $1 \leq i_r \leq k$. Clearly x and y are joined by a walk $x, x_1, x_2, \ldots, x_{r-1}, y$ in $TG(S_n, \Omega)$. Hence, $TG(S_n, \Omega)$ is connected with $n-1$ edges, that is, it is a tree. □

Corollary 3.6 *A set $\Omega \subseteq S_n$ of transpositions generates the symmetric group S_n if and only if the transposition graph $TG(S_n, \Omega)$ is connected.*

Proof: Every connected graph has a spanning tree. By Theorem 3.5, the result follows. □

Chapter 4

Symmetry in Graphs

Symmetry is an important issue in interconnection networks. It affects not only the performance but also the cost of the network. For instance, if a network has symmetry on the nodes, the same routing algorithm can be used on each node. This simplifies both the hardware of the control center and the system software of the operating system. This chapter will discuss certain symmetry that a network can have.

4.1 Automorphisms on Graphs

Given a square, one can rotate it and flip it. The square is still a square. However, if one tries to "twist it", the square will no longer be a square. On the other hand, no transformation can make a complete graph structually different. This kind of transformation that perserves the structure of the graph is called an ***automorphism***.

Definition 4.1 A *vertex automorphism* α of G is a permutation of the vertex-set that preserves the adjacency. That is, if the edge $xy \in E$, then the edge $\alpha(x)\alpha(y) \in E$.

4.2 Transitivity

In a network, it will be useful if the network looks the same when viewed through any node. In other words, each node lead to the same network by relabelling the other nodes. This property is called vertex-transitivity.

Definition 4.2 G is said to be *vertex-transitive* if given any pair of vertices x and y, there exists $\alpha \in Aut(G)$ such that $y = \alpha(x)$.

Definition 4.3 G is said to be *edge-transitive* if given any pair of edges xy and uv, there exists $\alpha \in Aut(G)$ such that $x = \alpha(u)$ and $y = \alpha(v)$, or $x = \alpha(v)$ and $y = \alpha(u)$.

From this definition, it is easy to see that a vertex-transitive graph has to be regular because no automorphism can map a vertex to one of different degree.

Vertex-transitivity can be generalized. Let D be the diameter of the graph G. For $0 \leq k \leq D$, G is said to be k-distance-transitive if given four vertices, x, y, u and v such that $d(x,y) = d(u,v) = k$, then there exists an $\alpha \in Aut(G)$ such that $u = \alpha(x)$ and $v = \alpha(y)$. If G is k-distance-transitive for all $0 \leq k \leq D$, then it is called distance-transitive.

Clearly, vertex-transitivity is 0-distance-transitivity.

Proposition 4.4 *If a graph G is 1-distance-transitive, then G is edge-transitive.*

Proof: Suppose G is 1-distance-transitive. Let $e_1 = xy$ and $e_2 = uv$ be two edges in G. Then there exists $\alpha \in Aut(G)$ such that $u = \alpha(x)$ and $v = \alpha(y)$. The result follows. □

The rest of this section will discuss transitivities of the graphs. For $u \in V(G)$, define $N_i \subseteq V(G)$ as $N_i = \{v : v \in V(G) \text{ and } d(u,v) = i\}$ and $d_i = |N_i(u)|$. Then the following is the characterization of distance-transitive graph [8, 2].

Lemma 4.5 *Let D be the diameter of the graph G. G is distance-transitive if and only if it is vertex-transitive and the vertex stabilizer A_u is transitive on the set $N_i(u)$ for all $i \in \{0, 1, \ldots, D\}$ and for each $u \in V(G)$.*

Proof: G is distance-transitive implying that G is 0-distance-transitive. Thus G is vertex-transitive. Let u be any vertex and $p, q \in N_i(u), 0 \leq i \leq D$. Since G is distance-transitive, there exists $\alpha \in Aut(G)$ such that $u = \alpha(u), p = \alpha(q)$. Since $u = \alpha(u), \alpha \in A_u(G)$. So $A_u(G)$ is transitive on $N_i(u)$.

Conversely, G is vertex-transitive and $A_u(G)$ is transitive on $N_i(u)$, for all $u \in V(G)$ and $i \in \{0, \ldots D\}$. Let $x, y, u, p \in V(G)$ so that $d(x, y) = d(u, p) = d$. Let $w \in V(G)$. There exists $\alpha \in Aut(G)$ such that $w = \alpha(x)$. Let $y' = \alpha(y)$. Also, there exists $\beta \in Aut(G)$ such that $w = \beta(u)$. Let $p' = \beta(p)$. Since α and β are automorphisms, $d(w, y') = d(x, y) = d = d(u, p) = d(w, p')$. So $y', p' \in N_d(w)$. Since $A_w(G)$ is transitive on $N_d(w)$, there exists an automorphism $\tau \in A_w(G)$ such that $w = \tau(w)$ and $p' = \tau(y')$. Then $u = \beta^{-1}\tau\alpha(x)$ and $p = \beta^{-1}\tau\alpha(y)$. Thus, G is distance-transitive. □

4.3 Intersection Number

Lemma 4.5 provides a method to check whether a graph is distance-transitive. The procedure in fact is quite tedious. There is a necessary condition for a graph being distance-transitive [8, 2]. First we define $n_{hi}(u, v) = |\{w : w \in V,\ d(u, w) = h \text{ and } d(v, w) = i\}| = |N_h(u) \cap N_i(v)|$. If a graph G is distance-transitive, $n_{hi}(u, v)$ is independent of u and v but depends only on j which is $d(u, v)$. This means $n_{hi}(u, v)$ can be denoted as n_{hij}.

Definition 4.6 Let D be the diameter of the distance-transitive graph G. The $(D+1)^3$ integers n_{hij} for $0 \leq h, i, j \leq D$ are called *intersection numbers*.

Proposition 4.7 *We have $n_{1ij} = 0$ for $i \notin \{j-1, j, j+1\}$.*

Proof: As the graph is distance-transitive, it is sufficient to consider one pair u and v of vertices with distance j between them. Let w be a vertex that is adjacent to u. Then $d(u, w) = 1$. Since $d(u, v) = j$, we have $j - 1 \leq d(v, w) \leq j + 1$. In other words, $n_{1ij} = 0$ for $i \notin \{j-1, j, j+1\}$. □

For this reason, if D is the diameter of a distance-transitive graph, we can let

$$\begin{array}{rclcl} a_j &=& n_{1jj} &=& |N_1(u) \cap N_j(v)| \\ b_j &=& n_{1,j+1,j} &=& |N_1(u) \cap N_{j+1}(v)| \\ c_j &=& n_{1,j-1,j} &=& |N_1(u) \cap N_{j-1}(v)|, \end{array}$$

where u and v are any pair of vertices with distance j between them, $0 \leq j \leq D$. Furthermore, b_D and c_0 are undefined. These $3D+1$ integers can be arranged as an array.

Definition 4.8 The array

$$IA(G) = \begin{bmatrix} a_0 & a_1 & a_2 & \dots & a_{D-1} & a_D \\ b_0 & b_1 & b_2 & \dots & b_{D-1} & * \\ * & c_1 & c_2 & \dots & c_{D-1} & c_D \end{bmatrix}$$

is called the *intersection array* of the distance-transitive graph G.

The intersection array has the following properties [2].

Lemma 4.9 *If G is distance-transitive, then the entries of $IA(G)$ satisfy:*

1. $a_0 = 0,\ b_0 = d_1,\ c_1 = 1,$
2. $c_i + a_i + b_i = d_1$ *for all* $1 \leq i \leq D-1,$
3. $1 \leq c_2 \leq c_3 \leq \ldots \leq c_D,$
4. $d_1 \geq b_1 \geq b_2 \geq \ldots \geq b_{D-1},$
5. $d_{i-1}b_{i-1} = d_i c_i$ *for* $1 \leq i \leq D,$

where $d_i = |N_i(u)|$ *and* $D =$ *diameter.*

Proof:

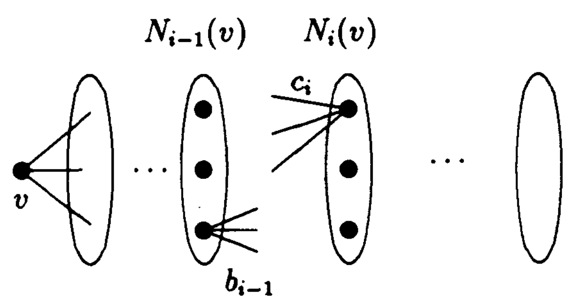
1. We have $a_0 = n_{100} = |N_1(u) \cap N_0(v)| = 0$ as $d(u,v) = 0$, that is, $u = v$. Also, $b_0 = n_{110} = |N_1(u) \cap N_1(v)| = |N_1(u)| = d_1$ as $u = v$, and $c_1 = n_{101} = |N_1(u) \cap N_0(v)| = |\{v\}| = 1$ as v is adjacent to u.

2. If $d(u,v) = i$ and w is adjacent to u, then $i-1 \leq d(v,w) \leq i+1$. So $a_i + b_i + c_i = n_{1ii} + n_{1,i+1,i} + n_{1,i-1,i} = |N_1(u) \cap N_i(v)| + |N_1(u) \cap N_{i+1}(v)| + |N_1(u) \cap N_{i-1}(v)| = |N_1(u)| = d_1$.

3. Suppose $d(u,v) = i+1$, $1 \leq i \leq D-1$. Pick a path $v, x, \ldots, u$ of length $i+1$. Then $d(x,u) = i$. If $w \in N_{i-1}(x) \cap N_1(u)$, then $w \in N_i(v) \cap N_1(u)$. So $N_{i-1}(x) \cap N_i(u) \subseteq N_i(v) \cap N_1(u)$, that is, $|N_{i-1}(x) \cap N_1(u)| \leq |N_i(v) \cap N_1(u)|$. In other words, $c_i = n_{1,i-1,i} \leq n_{1,i,i+1} = c_{i+1}$ for $1 \leq i \leq D-1$.

4. Suppose $d(u,v) = i$, $1 \leq i \leq D-1$. Pick a path $v, x, \ldots, u$ of length i. Then $d(x,u) = i-1$. If $w \in N_1(u) \cap N_{i+1}(v)$, then $w \in N_1(u) \cap N_i(x)$. So $N_1(u) \cap N_{i+1}(x) \subseteq N_1(u) \cap N_i(x)$, i.e. $b_{i-1} = n_{1,i,i-1} = |N_1(u) \cap N_i(x)| \geq |N_1(u) \cap N_{i+1}(v)| = b_i$.

5. Pick any vertex v. The number of edges from $N_{i-1}(v)$ to $N_i(v)$ is equal to the number of edges from $N_i(v)$ to $N_{i-1}(v)$, $1 \leq i \leq D$.



The number of edges from $N_i(v)$ to $N_{i-1}(v) = c_i|N_i(v)| = c_i d_i$.

The number of edges from $N_{i-1}(v)$ to $N_i(v) = b_{i-1}|N_{i-1}(v)| = b_{i-1}d_{i-1}$.

So $b_{i-1}d_{i-1} = c_i d_i$. ◫

In the rest of this section, we will consider properties of Cayley graphs and transposition graphs.

Theorem 4.10 *Every Cayley graph is vertex-transitive.*

Proof: Let $G(\Gamma, S)$ be a Cayley graph. Pick any two vertices u and v and define $\alpha : V \rightarrow V$ by $\alpha(x) = vu^{-1}x,\ x \in V$.

1. If $\alpha(x_1) = \alpha(x_2)$, then $vu^{-1}x_1 = vu^{-1}x_2$. So $x_1 = x_2$ and α is one-to-one.

2. For any $y \in V$, $\alpha(uv^{-1}y) = vu^{-1}uv^{-1}y = y$. Thus, α is onto.

3. If x_1 is adjacent to x_2, then $x_2 = x_1 g$ for some $g \in S$. So $vu^{-1}x_2 = vu^{-1}x_1 g$, or $\alpha(x_2) = \alpha(x_1)g$. That is, $\alpha(x_1)$ is adjacent to $\alpha(x_2)$ implying that α is an automorphism of $G(\Gamma, S)$.

Furthermore, $\alpha(u) = vu^{-1}u = v$. Hence, α is an automorphism that maps u to v. □

Using the Cayley graph construction, we can obtain a vertex-transitive graph. If we use a group with certain properties, those properties may be reflected in the graph. The Proposition [2] is one such example.

Proposition 4.11 *Let G be a connected graph. The subgroup H of the automorphism group $Aut(G)$ acts regularly on G if and only if G is a Cayley graph $G(H, S)$ for some symbol set S that generates H.*

Proof: Suppose $G = G(H, S)$. For each $h \in H$, let $\alpha_h : H \rightarrow H$ be defined by $\alpha_h(x) = hx$. The mapping α_h is definitely a permutation. If x is adjacent to y, then $y = xs$ for some $s \in S$. So $\alpha_h(y) = hy = hxs = \alpha_h(x)s$. That is, $\alpha_h(x)$ is adjacent to $\alpha_h(y)$. Therefore, α_h is an automorphism. The set of all α_h is a subgroup of $Aut(G)$ isomorphic to H. Let $\overline{H}$ be this subgroup. For any pair of vertices x and y, $x, y \in H$. There is a unique $h \in H$ such that $hx = y$. Hence, there is an automorphism in $\overline{H}$, α_h such that $\alpha_h(x) = y$. Notice that if e is the identity in H, then it is the unique element such that $\alpha_e(x) = x$. Hence, $\overline{H}$ acts on G regularly.

Conversely, suppose H is regular and $H \leq Aut(G)$. Let $V(G) = \{v_1, v_2, \ldots, v_n\}$ denote the vertex-set of G. Since H is transitive, for each i, there exists $h_i \in H$ such that $h_i(v_1) = v_i$. Suppose $h_i(v_1) = h_i'(v_1) = v_i$. Then $h_i^{-1}h_i(v_1) = h_i^{-1}h_i'(v_1) = v_1$. Since H is regular, $h_i^{-1}h_i' = e$ and $h_i' = h_i$. This implies h_i is the unique element in H that maps v_1 to v_i. Now let $S = \{h_i \in H : v_i \text{ is adjacent to } v_1 \text{ in } G\}$. Clearly $e \notin S$. If $h_i \in H$, then v_i is adjacent to v_1 and $h_i(v_i)$ is adjacent to $h_i(v_1) = v_i$. So $h_i^{-1}(h_i(v_i)) = h_i^{-1}(v_1)$ is adjacent to $h_i^{-1}(v_i) = v_1$. By the definition of S, $h_i^{-1} \in S$. Therefore, S satisfies the conditions of being a symbol set. Let $\phi : G \to G(H, S)$ be defined by $\phi(v_i) = h_i$. Since there is a unique h_i corresponding to v_i, ϕ is one-to-one. Since H is transitive, ϕ is onto.

Suppose v_i is adjacent to v_j. Then $h_i^{-1}(v_i) = v_1$ is adjacent to $h_i^{-1}(v_j) = h_i^{-1}h_j(v_1)$. So $h_i^{-1}h_j \in S$. Since $h_j = h_i h_i^{-1} h_j$, h_i is adjacent to h_j in $G(H, S)$. Conversely suppose h_i is adjacent to h_j. Then $h_j = h_i h_l$, for some $h_l \in S$. Since v_1 is adjacent to v_l, $h_i(v_1) = v_i$ is adjacent to $h_i(v_l) = h_i h_l(v_1) = h_j(v_1) = v_j$. Thus, $G \cong G(H, S)$. □

Now we consider some relationships between Cayley graphs and transposition graphs [8].

Lemma 4.12 *Let $G(\Gamma_1, S_1)$ and $G(\Gamma_2, S_2)$ be two Cayley graphs on the permutation groups Γ_1 and Γ_2 acting on the sets A_1 and A_2, respectively. Let Γ_1 and Γ_2 be generated by the sets of transpositions S_1 and S_2 respectively, where $|S_1| = |S_2|$. If the transposition graphs $TG(A_1, S_1)$ and $TG(A_2, S_2)$ are isomorphic, then $G(\Gamma_1, S_1)$ and $G(\Gamma_2, S_2)$ are isomorphic too.*

Proof: Let $\omega : TG(A_1, S_1) \to TG(A_2, S_2)$ be an isomorphism. Define $\beta : \Gamma_1 \to \Gamma_2$ by $\beta(u) = \omega \cdot u \cdot \omega^{-1}$. Then $\beta(u)$ is a composition of one-to-one and onto functions, so $\beta(u)$ is a permutation on A_2.

1. β is one-to-one.

   If $\beta(u_1) = \beta(u_2)$, then $\omega \cdot u_1 \cdot \omega^{-1}(y) = \omega \cdot u_2 \cdot \omega^{-1}(y)$ for all $y \in A_2$, or $\omega(u_1(\omega^{-1}(y))) = \omega(u_2(\omega^{-1}(y)))$ for all $y \in A_2$. Since ω is one-to-one, $u_1(\omega^{-1}(y)) = u_2(\omega^{-1}(y))$ for all $y \in A_2$. Thus, $u_1(x) = u_2(x)$ for all $x \in A_1$, or $u_1 = u_2$.

2. β is onto.

   Pick any $p \in \Gamma_2$. Let $u = \omega^{-1} \cdot p \cdot \omega$. Then $\beta(u) = \beta(\omega^{-1} \cdot p \cdot \omega) = \omega \cdot \omega^{-1} \cdot p \cdot \omega \cdot \omega^{-1} = p$.

3. β preserves the adjacency.

   If uv is an edge in $G(\Gamma_1, S_1)$, then $u = vs$ for some transposition $s = (i\ j) \in S_1$. Since ω is an isomorphism from $TG(A_1, S_1)$ to

$TG(A_1, S_1)$, $(\omega(i)\ \omega(j)) \in S_2$. Thus, $\beta(u) = \omega \cdot u \cdot \omega^{-1} = \omega \cdot v \cdot s \cdot \omega^{-1}$ $= \omega \cdot v \cdot \omega^{-1}\omega \cdot s \cdot \omega^{-1} = \beta(v)(\omega(i)\ \omega(j))$. That is, $\beta(u)$ is adjacent to $\beta(v)$.

Hence, β is an isomorphism from $G(\Gamma_1, S_1)$ to $G(\Gamma_2, S_2)$. □

Theorem 4.13 *Let $G(\Gamma, S)$ be a Cayley graph on a permutation group Γ acting on A with the set of transpositions S. If the transposition graph $TG(A, S)$ is edge-transitive, then $G(\Gamma, S)$ is 1-distance-transitive.*

Proof: Let e be the identity in Γ and let $G = G(\Gamma, S)$. Let $u, v, x, y \in \Gamma$ be such that $uv \in E(G)$ and $xy \in E(G)$. Since G is vertex-transitive, there exist automorphisms α and τ such that $e = \alpha(u)$ and $e = \tau(x)$. Let $v' = \alpha(v)$ and $y' = \tau(y)$. Since $d(u, v) = d(e, v') = d(e, y') = d(x, y) = 1$, v' and y' are transpositions.

Let $v' = (i\ j)$ and $y' = (l\ k)$. Since the transposition graph $TG(A, S)$ is edge-transitive, there exists an automorphism σ such that $l = \sigma(i)$ and $k = \sigma(j)$, or $k = \sigma(i)$ and $l = \sigma(j)$. Notice that σ is a permutation on A.

Define $\beta : \Gamma \to \Gamma$ by $\beta(p) = \sigma p \sigma^{-1}$. By the proof of Lemma 4.12, β is an automorphism of G so that $\beta(v') = \sigma v' \sigma^{-1} = y'$ and $\beta(e) = \sigma e \sigma^{-1} = e$.

Hence, $\tau^{-1}\beta\alpha$ is an automorphism such that $\tau^{-1}\beta\alpha(u) = x$ and $\tau^{-1}\beta\alpha(v)$ $= y$. That is, $G(\Gamma, S)$ is 1-distance-transitive. □

4.4 Set Graphs

Although the construction of Cayley graphs gives us a way to build vertex-transitive graphs, it does not produce all vertex-transitive graphs. For example, the Petersen graph is not a Cayley graph. Before showing that the Petersen graph is a vertex-transitive graph but not a Cayley graph, let's consider another construction of vertex-transitive graphs.

Definition 4.14 Let $S = \{1, \ldots, n\}$. The *set graph* $G(S, k)$ is a graph whose vertex-set is the set of all k-subsets of S. Two vertices are adjacent if and only if the intersection of the corresponding subsets is empty.

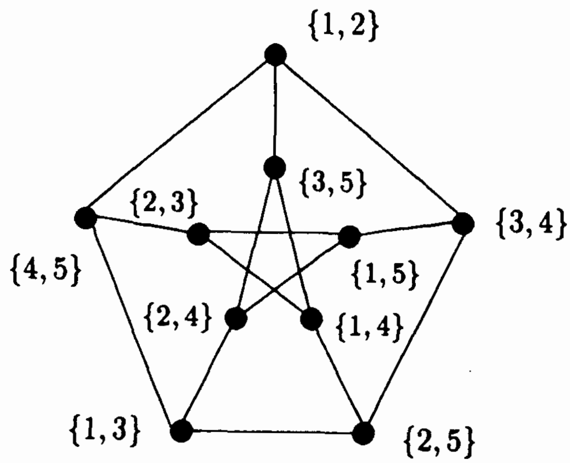


Figure 4.1: The Petersen Graph Modelled as a Set Graph

Example : Let $S = \{1,2,3,4,5\}$ and $k = 2$. We get the Petersen graph (Figure 4.1.)

Theorem 4.15 *Every set graph is vertex-transitive.*

Proof: Let $G(S,k)$ be a set graph. For any pair of vertices u and v, we need to find an automorphism ϕ_{uv} such that $\phi_{uv}(u) = v$. Let $u = \{a_1, a_2, \ldots, a_k\}$ and $v = \{b_1, b_2, \ldots, b_k\}$. There exists a permutation σ such that $\sigma(a_i) = b_i$ for all $1 \leq i \leq k$.

Let $\phi_{uv} : G \to G$ be defined by $\phi_{uv}(x) = \phi_{uv}(\{x_1, x_2, \ldots, x_k\}) = \{\sigma(x_1), \sigma(x_2), \ldots, \sigma(x_k)\}$. Suppose we have $\{\sigma(x_1), \sigma(x_2), \ldots, \sigma(x_k)\} = \{\sigma(y_1), \sigma(y_2), \ldots, \sigma(y_k)\}$, then we can relabel the elements so that $\sigma(x_i) = \sigma(y_i)$, $1 \leq i \leq k$. Since σ is a permutation, $x_i = y_i$ for all $1 \leq i \leq k$, that is, $\{x_1, x_2, \ldots, x_k\} = \{y_1, y_2, \ldots, y_k\}$. Hence, ϕ_{uv} is one-to-one.

For any $\{z_1, z_2, \ldots, z_k\}$, Let $w = \{\sigma^{-1}(z_1), \sigma^{-1}(z_2), \ldots, \sigma^{-1}(z_k)\}$. Then $\phi_{uv}(w) = \{z_1, z_2, \ldots, z_k\}$. So ϕ_{uv} is onto.

If $\{x_1, x_2, \ldots, x_k\}$ is adjacent to $\{y_1, y_2, \ldots, y_k\}$, then $\{x_1, x_2, \ldots, x_k\} \cap \{y_1, y_2, \ldots, y_k\} = \emptyset$. Since σ is a permutation, $\{\sigma(x_1), \sigma(x_2), \ldots, \sigma(x_k)\} \cap \{\sigma(y_1), \sigma(y_2), \ldots, \sigma(y_k)\} = \emptyset$. Therefore, $\phi_{uv}(\{x_1, x_2, \ldots, x_k\})$ is adjacent to $\phi_{uv}(\{y_1, y_2, \ldots, y_k\})$.

Also, $\phi_{uv}(\{a_1, a_2, \ldots, a_k\}) = \{\sigma(a_1), \sigma(a_2), \ldots, \sigma(a_k)\} = \{b_1, b_2, \ldots, b_k\}$. Hence, ϕ_{uv} is an automorphism mapping u to v. □

Corollary 4.16 *The Petersen graph is vertex-transitive.*

Although Cayley graphs and set graphs are both vertex-transitive, in general, the Cayley graph construction cannot produce set graphs. Again we can show that the Petersen graph is not a Cayley graph.

Theorem 4.17 *The Petersen graph is not a Cayley graph.*

Proof: Suppose the Petersen graph is a Cayley graph $G(\Gamma, S)$ for some group Γ and symbol set S. From the fact of the group theory, there are only two possible groups of order 10[2, 13]. Suppose $\Gamma = \langle g \rangle$ for some element g. Then $\Gamma = \{e, g^1, g^2, \ldots g^9\}$. The symbol set can only be $S_i = \{g^i, g^5, g^{-i}\}$, where $i = 1, 2, 3, 4$. Then $e, eg^i, eg^ig^5, eg^ig^5g^{-i}, eg^ig^5g^{-i}g^5 = e$ is a 4-cycle. But the Petersen graph does not have any 4-cycle.

Then $\Gamma = \{e, g, g^2, g^3, g^4, x, xg, xg^2, xg^3, xg^4\}$, where $xg^ix = g^{-i}$ and $x^2 = e$. Then the possible symbol sets are $S_1 = \{xg^i, g, g^4\}$, $S_2 = \{xg^i, g^2, g^3\}$ and $S_3 = \{xg^j, xg^k, xg^l\}$, where $0 \leq i, j, k, l \leq 4$.

If the symbol set is S_1, then $e, exg^i, exg^ig, exg^igxg^i, exg^igxg^ig = e$ is a 4-cycle. If the symbol set is S_2, then $e, exg^i, exg^ig^2, exg^ig^2xg^i, exg^ig^2xg^ig^2 = e$ is a 4-cycle. For S_3, we let $S_3 = \{g_1, g_2, g_3\}$. We know that $g_1^2 = g_2^2 = g_3^2 = e$. We can label the edges by the symbols in S_3 such that adjacent edges have different symbols assigned.

Consider the outermost 5-cycle. Two of the symbols in S_3 must be used twice. Without loss of generality, we label the outermost 5-cycle as Figure 4.2.

Then we can continue to label the edges, and finally we will get two adjacent edges having the same symbol. So we get the contradiction. Hence the Petersen graph is not a Cayley. □

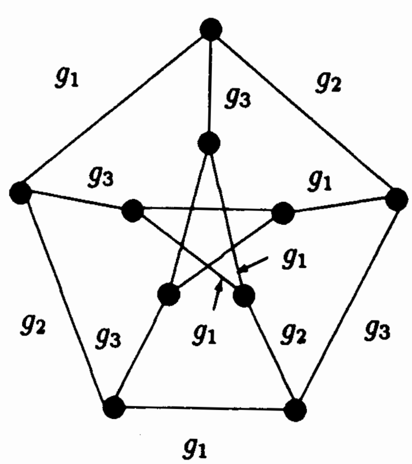


Figure 4.2: Diagram for Theorem 4.17

Chapter 5

The Hypercube

The hypercube is usually considered to be an efficient networks for parallel computation. The construction of the hypercube is based on the binary numbers. As a consequence, routing algorithms for the hypercube are very easy to implement. Also, the hypercube is highly symmetric. In fact, it is distance-transitive. Furthermore, one can simulate most of the popular networks on the hypercube such as the grid and the binary tree. Hence, the hypercube is a good architecture for general purpose parallel systems [9].

5.1 Modelling

The hypercube is usually defined as follows [9].

Definition 5.1 Let $G(V, E)$ be a graph with $|V| = 2^r$ and $|E| = r2^{r-1}$ for some positive integer r. The vertices in G are labelled with a binary sequence of length r. Two vertices are adjacent if and only if their binary sequences differ in precisely one bit. G is called the *r-dimensional hypercube* and denoted as Q_r.

Figure 5.1 is a 3-dimensional hypercube. Besides the above definition, The hypercube can be defined as a Cayley graph too [8].

Proposition 5.2 *Let $\Gamma = \langle(1\ 2), (3\ 4), \ldots, (2r\text{-}1\ 2r)\rangle$ be a subgroup of S_{2r}. Let $S = \{(1\ 2), (3\ 4), \ldots, (2r\text{-}1\ 2r)\}$ be the set of symbols. Then the Cayley graph $G(\Gamma, S) \cong Q_r$.*

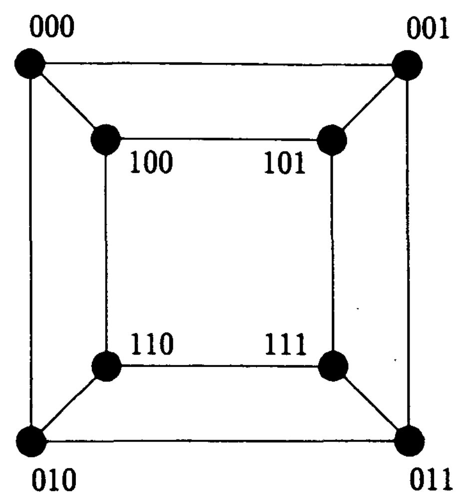


Figure 5.1: The 3-dimensional Hypercube

Proof: Let v be a vertex in Q_r. Let $a_1a_2\cdots a_r$ be the binary sequence corresponding to v. Define $\phi : Q_r \to G(\Gamma, S)$ by $\phi(v) = \phi(a_1a_2\cdots a_r) = p_1p_2\cdots p_r$ where

$$p_i = \begin{cases} (2i\text{-}1\ 2i) & \text{if } a_i = 1 \\ e & \text{otherwise.} \end{cases}$$

Let v_1, $v_2 \in Q_r$ and $v_1 = a_1\cdots a_r$ and $v_2 = b_1\cdots b_r$. Since $(1\ 2), (3\ 4), \ldots,$ $(2r\text{-}1\ 2r)$ are disjoint cycles, none of them can be generated by the others. Thus, if $v_1 \neq v_2$, then $a_1\cdots a_r \neq b_1\cdots b_r$. There are some $1 \leq i \leq r$ such that $a_i \neq b_i$. So $(2i\text{-}1\ 2i)$ is contained either in $\phi(v_1)$ or in $\phi(v_2)$ but not in both, that is, $\phi(v_1) \neq \phi(v_2)$.

Let $p \in \Gamma$. Let $a_1a_2\cdots a_r$ be a binary sequence such that

$$a_i = \begin{cases} 1 & \text{if } (2i\text{-}1\ 2i) \text{ is in } p \\ 0 & \text{otherwise.} \end{cases}$$

Then $\phi(a_1a_2\cdots a_r) = p$. Hence ϕ is a bijection.

Now if $a_1\cdots a_r$ is adjacent to $b_1\cdots b_r$, then there is exactly one i, $1 \leq i \leq r$ such that $a_j = b_j$ for $i \neq j$ and $a_i \neq b_i$. Then $\phi(a_1\cdots a_r) = \phi(b_1\cdots b_r)(2i - 1\ 2i)$. So $\phi(a_1\cdots a_r)$ is adjacent to $\phi(b_1\cdots b_r)$.

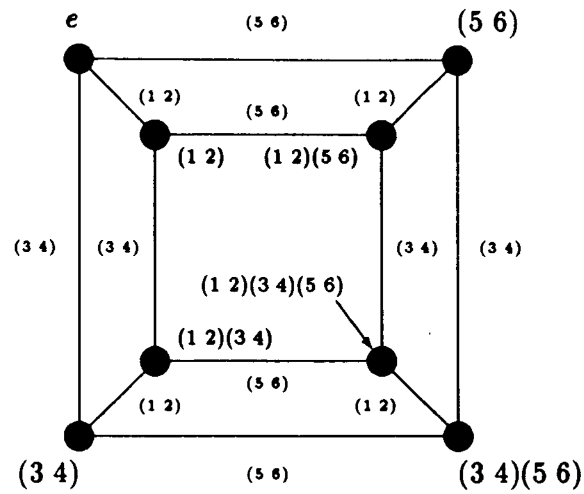


Figure 5.2: The 3-dimensional Hypercube Modelled as a Cayley Graph

Conversely, if $\phi(a_1 \cdots a_r)$ is adjacent to $\phi(b_1 \cdots b_r)$, then $\phi(a_1 \cdots a_r) = \phi(b_1 \cdots b_r)(2i-1\ 2i)$ for some i. This implies that $a_j = b_j$ for $i \neq j$ and $a_i \neq b_i$. Hence, $a_1 \cdots a_r$ is adjacent to $b_1 \cdots b_r$. Therefore, ϕ is an isomorphism. Figure 5.2 is the Cayley graph version of the 3-dimensional hypercube. $\square$

One drawback of the hypercube is that the degree of each vertex is equal to $\log_2 |V|$. That means, when the network is getting bigger, the communication lines going out from the vertex will increase too. If a processor is designed for the 4-dimensional hypercube, it cannot be used for the 8-dimensional hypercube because four communication ports are missing from each processor. This drawback reduces the expandibility of the network.

5.2 Symmetry

However, the hypercube has very good symmetry properties. Since the hypercube is a Cayley graph, it is vertex-transitive. The transposition graph of the hypercube is a perfect matching, so it is 1-distance-transitive by Lemma 4.13. As mentioned before, the hypercube is in fact distance-transitive [2].

Lemma 5.3 *The r-dimensional hypercube has diameter r.*

Proof: For any pair of vertices u and v in Q_r, we can flip the necessary bits of u one by one to get v. This also gives a route from u to v. Thus, the diameter must be at most r. Since from $00\cdots 0$ to $11\cdots 1$ we have to flip at least r bits, the diameter must be at least r. The result follows. $\square$

Theorem 5.4 *The hypercube is distance-transitive.*

Proof: From the above Lemma, we know that the diameter of Q_r is r. Since Q_r is a Cayley graph, it is vertex-transitive. Let $u = p_1 p_2 \cdots p_r$ be any vertex, where $p_j = (2j\text{-}1\ 2j)$ or $p_j = e$. Let x and $y \in N_i(u)$, $0 \leq i \leq r$. There are precisely i transpositions either in x or in u but not in both. Simlarly, there are precisely i transpositions either in y or in u but not in both.

Let $p_{k1}, p_{k2}, \ldots, p_{ki}$ be the transpositions either in x or in u but not in both, and $p_{l1}, p_{l2}, \ldots, p_{li}$ be the transpositions either in y or in u but not in both. Let $p_{kj} = (r_{kj}\ r_{kj}{+}1)$ and $p_{lj} = (s_{lj}\ s_{lj}{+}1)$ for $1 \leq j \leq i$. Consider the mapping $\beta : \langle(1\ 2), \ldots (2r\text{-}1\ 2r)\rangle \rightarrow \langle(1\ 2), \ldots (2r\text{-}1\ 2r)\rangle$ defined by

$$\begin{aligned}\beta(v) \;=\;& u(r_{k1}\ s_{l1})(r_{k1}{+}1\ s_{l1}{+}1)(r_{k2}\ s_{l2})(r_{k2}{+}1\ s_{l2}{+}1)\cdots\\ & (r_{ki}\ s_{li})(r_{ki}{+}1\ s_{li}{+}1)uv(r_{k1}\ s_{l1})(r_{k1}{+}1\ s_{l1}{+}1)\\ & (r_{k2}\ s_{l2})(r_{k2}{+}1\ s_{l2}{+}1)\cdots(r_{ki}\ s_{li})(r_{ki}{+}1\ s_{li}{+}1).\end{aligned}$$

Notice that every components in β is the inverse of itself. We have

$$\beta(u(t\ t{+}1)) = \begin{cases} u(r_{kj}\ r_{kj}{+}1) & \text{if } t = s_{kj}\\ u(s_{kj}\ s_{kj}{+}1) & \text{if } t = r_{kj}\\ u(t\ t{+}1) & \text{otherwise}\end{cases}$$

and

$$\begin{aligned}&\beta(u(t_1\ t_1{+}1)(t_2\ t_2{+}1)\cdots(t_m\ t_m{+}1))\\ &\quad = \beta(u(t_1\ t_1{+}1))u\beta(u(t_1\ t_1{+}1))\cdots u\beta(u(t_m\ t_m{+}1)).\end{aligned}$$

Also, $\beta^{-1} = \beta$. Clearly β is a permutation on the vertices in Q_r. If v_1 is adjacent to v_2 in Q_r, then there is precisely one transposition in one of the v_j's but not in both. There is also precisely one transposition in one of the $\beta(v_j)$'s but not in both. Hence, β is an automorphism. Since $\beta(u) = u$ and $\beta(x) = y$, $\beta \in A_u$. By Lemma 4.5, Q_r is distance-transitive. $\square$

5.3 Connectivity

To be a good network, a graph should have very high connectivity. The connectivity tells us how many nodes can be malfunctioned and the network is still connected. It also tells us how many node-disjoint paths between a pair of nodes. The more node-disjoint paths the network has, the more subproblems can be handled simultaneously.

Definition 5.5 Let b be a binary digit. we define

$$\bar{b} = \begin{cases} 0 & \text{if } b = 1 \\ 1 & \text{if } b = 0. \end{cases}$$

Lemma 5.6 *An n-regular graph has connectivity n if for any pair of vertices u and v, there are n vertex-disjoint paths joining them.*

Proof: If any two vertices are joined by n vertex-disjoint paths, then we should remove at least n vertices to disconnect the graph. However, G is n-regular, so G has connectivity n.

Note: the converse of the lemma is also true (See the Menger's Theorem [3]), but we will not prove it here. □

Theorem 5.7 *The r-dimensional hypercube Q_r has connectivity r.*

Proof: In the 2-dimensional hypercube, any pair of vertices are joined by two vertex-disjoint paths. We assume that any pair of vertices of Q_r are joined by r vertex-disjoint paths. Since Q_{r+1} is vertex-transitive, it is sufficient to show that there are $r+1$ vertex-disjoint paths joining the vertex $0\cdots0$ and $b_1\cdots b_{r+1}$, where $b_i \in \{0,1\}$, $1 \le i \le r+1$.
Case 1: The bit $b_{r+1} = 0$. Let S be the subgraph induced by the vertices whose $(r+1)$th bit is 0, and let T be the subgraph induced by the vertices whose $(r+1)$th bit is 1. S and T are r-dimensional hypercubes. There are r vertex-disjoint paths from $0\cdots0$ to $b_1\cdots b_r0$ in S. There is a path P from $0\cdots01$ to $b_1\cdots b_r1$ in T. Therefore, we have a path starting from $0\cdots0$, passing through P and ending at $b_1\cdots b_r0$ in Q_{r+1}. Together with the r

vertex-disjoint paths in S, we have $r+1$ vertex-disjoint paths from $0\cdots 0$ to $b_1\cdots b_r0$.
Case 2: The bit $b_{r+1}=1$. We use the same definitions of S and T. There are r vertex-disjoint paths from $0\cdots 0$ to $b_1\cdots b_r0$ in S. We remove $b_1\cdots b_r0$ from each of these paths. Let P_i be the path from $0\cdots 0$ to $b_1\cdots b_{i-1}\bar{b}_ib_{i+1}\cdots b_r0$, $1\le i\le r$. We extend P_i by adding the 2-path $b_1\cdots b_{i-1}\bar{b}_ib_{i+1}\cdots b_r0$, $b_1\cdots b_{i-1}\bar{b}_ib_{i+1}\cdots b_r1$, $b_1\cdots b_{i-1}b_ib_{i+1}\cdots b_r1$ for $1\le i\le r-1$. We extend P_r by adding the 2-path $b_1\cdots b_{r-1}\bar{b}_r0$, $b_1\cdots b_{r-1}b_r0$, $b_1\cdots b_{r-1}b_r1$. We translate P_r to T by changing the $(r+1)$th bit of each vertex to 1 and call it P_r'. Then we have the path from $0\cdots 0$ passing through P_r' to $b_1\cdots b_{r-1}b_r1$. Hence, we have $r+1$ vertex-disjoint paths from $0\cdots 0$ to $b_1\cdots b_r1$. By Lemma 5.6, the result follows. □

Corollary 5.8 *The edge-connectivity of the r-dimensional hypercube is r.*

Proof: Suppose the edge-connectivity is k, where $k<r$. Let T be the set of k edges whose removal will disconnect the graph. Then for each edge in T we can remove one of the incident vertices to disconnect the graph. But it is a contradiction. Since the r-dimensional hypercube is r-regular. The edge-connectivity must be r. □

5.4 Other Known Results

It is not difficult to see that the hypercube is bipartite. We can get the bipartition by letting one of the partition sets be the set of vertices with an even number of 1's.

Definition 5.9 Let $G=(X,Y)$ be a bipartite graph. If for any pair of vertices, $x\in X$ and $y\in Y$, there is a Hamilton path from x to y, then G is said to be *Hamilton-laceable.*

Definition 5.10 Let G be a graph. If for any pair of vertices x and y in G, there is a Hamilton path from x to y, then G is said to be *Hamilton-connected.*

Theorem 5.11 *The r-dimensional hypercube is Hamilton-laceable.*

Proof: Q_2 is Hamilton-laceable. Assume Q_r is Hamilton-laceable. Let $u = u_1u_2 \ldots u_{r+1}$ and $v_1v_2 \ldots v_{r+1}$ be any vertices in Q_{r+1}, where $u_i, v_i \in \{0, 1\}$ for $1 \leq i \leq r$. Suppose $u_i \neq v_i$. Let P_1 be a Hamilton path from $u_1 \ldots u_{i-1}u_{i+1} \ldots u_{r+1}$ to $u_1 \ldots u_{i-1}\overline{u}_{i+1} \ldots u_{r+1}$ in Q_r and P_2 a be a Hamilton path from $u_1 \ldots u_{i-1}\overline{u}_{i+1} \ldots u_{r+1}$ to $v_1 \ldots v_{i-1}v_{i+1} \ldots v_{r+1}$ in another copy of Q_r (These exist because $u_1 \ldots u_{i-1}\overline{u}_{i+1} \ldots u_{r+1}$ and $v_1 \ldots v_{i-1}v_{i+1} \ldots v_{r+1}$ differ in an odd number of bits.)

Now we insert u_i in the ith position for every vertex in P_1 and denote the new path as $P_1(u_i)$. We also insert v_i in the ith position for every vertex in P_2 and denote the new path as $P_2(v_i)$. Clearly $P_1(u_i)P_2(v_i)$ is a Hamilton path from $u_1 \ldots u_{r+1}$ to $v_1 \ldots v_{r+1}$ in Q_{r+1}. By induction, the result follows. $\square$

Corollary 5.12 *The r-dimensional hypercube is hamiltonian.*

Chapter 6

The Butterfly Network

The butterfly network is one of the modification of the hypercube. It inherits some of the properties of the hypercube, but its degree is bounded. In fact, the butterfly graph is a 4-regular graph. Like the hypercube, the butterfly network can simulate most of the networks with bounded degree with acceptable slowdown [9].

6.1 Modelling

The following definition of the butterfly graph is taken from [9].

Definition 6.1 Let $G(V, E)$ be a graph with $|V| = r2^r$ and $|E| = r2^{r+1}$ for some positive integer r. The vertices in G are labelled as $\langle w, i\rangle$, where w is a binary sequence of length r that is called the row of the vertex, and i is the level of the vertex ($1 \leq i \leq r$). Two vertices $\langle w, i\rangle$ and $\langle w', i'\rangle$ are adjacent if and only if either

1. $w = w'$ and $i' \equiv i \pm 1 \pmod r$ or

2. w and w' differ in precisely the i'th bit when $i' \equiv i + 1 \pmod r$ or w and w' differ in precisely the ith bit when $i' \equiv i - 1 \pmod r$

G is called the *r-dimensional butterfly graph* and denoted as B_r.

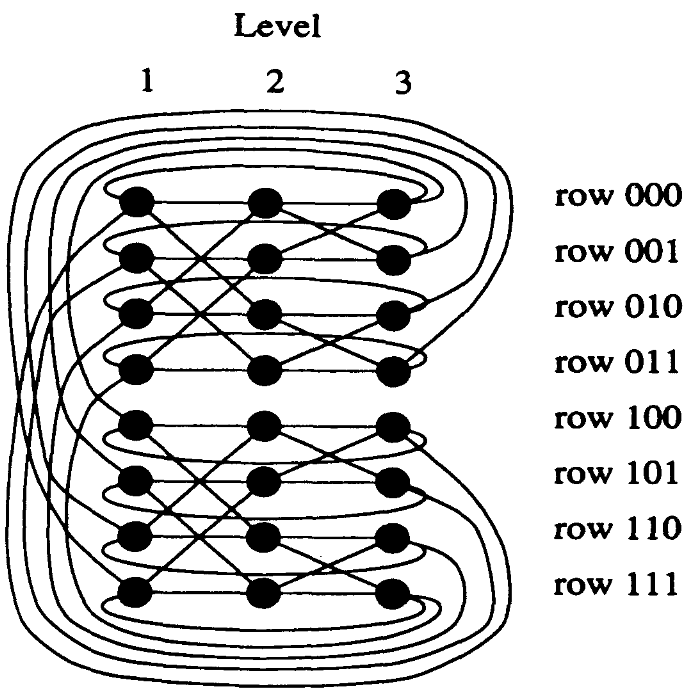


Figure 6.1: The 3-dimensional Butterfly Graph

Figure 6.1 exhibits the 3-dimensional butterfly graph. Notice that if we identify the vertices in the same row, and remove all the loops and multiple edges, we will get an r-dimensional hypercube.

Proposition 6.2 *Let B_r be an r-dimensional butterfly graph. If G is the graph obtained from B_r by identifying the vertices in the same row and removing all the loops and multiple edges, then $G \cong Q_r$.*

Proof: This follows directly from the definitions of the butterfly graph and the hypercube. □

Again, the butterfly graph can be modelled as a Cayley graph. But the group structure will not be as simple as the one for the hypercube. For the butterfly graph, each vertex has two coordinates. The first one is related to the one in the hypercube. Thus, we will extend the group for the hypercube to the one for the butterfly graph [11, 1].

Let $\Gamma_r = \{(p, i) : p = p_1 \cdots p_r,\ p_i \in \{(2j\text{-}1\ 2j) : j = 1, \ldots, r\} \cup \{e\},\ 0 \leq i \leq r-1\}$ and $S = \{(1\ 2), (3\ 4), \ldots, (2r\text{-}1\ 2r)\}$. Define $\pi_i : S \to S$ by $\pi_i((2j\text{-}1\ 2j)) = (2(i+j)\text{-}1\ 2(i+j))$ reduced modulo $2r$ and $\pi_i(e) = e$. Clearly, $\pi_{i+k}(p) = \pi_i(\pi_k(p))$.

Define a binary operator $\cdot$ as follows:

$$(p, i) \cdot (p', i') = (p_1 \cdots p_r, i) \cdot (p'_1 \cdots p'_r, i') = (p_1 \cdots p_r \pi_i(p'_1) \cdots \pi_i(p'_r), i + i'),$$

where $i + i'$ is reduced modulo r.

Proposition 6.3 *We have that $(\Gamma_r, \cdot)$ is a group.*

Proof: Note that

$$\begin{aligned}
& [(p, i) \cdot (p', i')] \cdot (p'', i'') \\
= & [(p_1 \cdots p_r, i) \cdot (p'_1 \cdots p'_r, i')] \cdot (p'', i'') \\
= & (p_1 \cdots p_r \pi_i(p'_1) \cdots \pi_i(p'_r), i + i') \cdot (p'', i'') \\
= & (p_1 \cdots p_r \pi_i(p'_1) \cdots \pi_i(p'_r), i + i') \cdot (p''_1 \cdots p''_r, i'') \\
= & (p_1 \cdots p_r \pi_i(p'_1) \cdots \pi_i(p'_r) \pi_{i+i'}(p''_1) \cdots \pi_{i+i'}(p''_r), i + i' + i'') \\
= & (p_1 \cdots p_r \pi_i(p'_1) \cdots \pi_i(p'_r) \pi_i(\pi_{i'}(p''_1)) \cdots \pi_i(\pi_{i'}(p''_r)), i + i + i'') \\
= & (p_1 \cdots p_r, i) \cdot (p'_1 \cdots p'_r \pi_{i'}(p''_1) \cdots \pi_{i'}(p''_r), i' + i'') \\
= & (p_1 \cdots p_r, i) \cdot [(p'_1 \cdots p'_r, i') \cdot (p''_1 \cdots p''_r, i'')] \\
= & (p, i) \cdot [(p', i') \cdot (p'', i'')].
\end{aligned}$$

So $\cdot$ is associative.

Since $(e,0)\cdot(p,i)=(p,i)\cdot(e,0)=(p,i)$, $(e,0)$ is the identity.
For any $(p,i)=(p_1\cdots p_r,i)$,

$$\begin{aligned}(p_1\cdots p_r,i)\cdot(\pi_{-i}(p_1)\cdots\pi_{-i}(p_r),-i) &= (p_1p_1\cdots p_rp_r,0)\\ &= (e,0),\end{aligned}$$

and every element has an inverse. □

Now we can model the butterfly graph as a Cayley graph.

Proposition 6.4 *Let Γ_r be the group in Proposition 6.3 and $S=\{(e,1),(e,r-1),((1\ 2),1),((2r\text{-}1\ 2r),r-1)\}$. The Cayley Graph $G(\Gamma_r,S)\cong B_r$.*

Proof: Let $\langle w,i\rangle$ be a vertex in B_r, where $w=a_1a_2\cdots a_r$ is a binary sequence of length r. Define $\phi : B_r \rightarrow G(\Gamma,S)$ by $\phi(\langle w,i\rangle)=\phi(\langle a_1\cdots a_r,i\rangle)=(p_1\cdots p_r,i)$, where

$$p_i=\begin{cases}(2i-1\ 2i) & \text{if } a_i=1\\ e & \text{otherwise.}\end{cases}$$

Let $\langle w_1,i_1\rangle,\langle w_2,i_2\rangle\in B_r$ where $w_1=a_1\cdots a_r$ and $w_2=b_1\cdots b_r$. If $i_1\neq i_2$, then clearly $\phi(\langle w_1,i_1\rangle)\neq\phi(\langle w_2,i_2\rangle)$. If $w_1\neq w_2$, then using the same argument in Proposition 5.2, We have $\phi(\langle w_1,i_1\rangle)\neq\phi(\langle w_2,i_2\rangle)$.

Let $(p,i)\in\Gamma$. Let $a_1\cdots a_r$ be a binary sequence such that

$$a_i=\begin{cases}1 & \text{if } (2i-1\ 2i) \text{ is in } p\\ 0 & \text{otherwise.}\end{cases}$$

Then $\phi(\langle a_1\cdots a_r,i\rangle)=(p,i)$. Hence ϕ is a bijection.

If $\langle w_1,i_1\rangle$ is adjacent to $\langle w_2,i_2\rangle$, then there are two cases.

Case 1: $w_1=w_2$ and $i_1\equiv i_2+1 \pmod r$. Then $\phi(\langle w_1,i_1\rangle)(e,1)=\phi(\langle w_2,i_2\rangle)$. Thus, $\phi(\langle w_1,i_1\rangle)$ is adjacent to $\phi(\langle w_2,i_2\rangle)$.

Case 2: w_1 and w_2 differ in the i_1th bit and $i_1\equiv i_2+1 \pmod r$. Then $\phi(\langle w_1,i_1\rangle)((1\ 2),1)=\phi(\langle w_2,i_2\rangle)$. Thus, $\phi(\langle w_1,i_1\rangle)$ is adjacent to $\phi(\langle w_2,i_2\rangle)$.

Conversely, if $\phi(\langle w_1,i_1\rangle)$ is adjacent to $\phi(\langle w_2,i_2\rangle)$, there are four cases.

Case 1: if $\phi(\langle w_1, i_1\rangle)(e, 1) = \phi(\langle w_2, i_2\rangle)$, then $w_1 = w_2$ and $i_2 \equiv i_1 + 1 \pmod r$. Hence, $\langle w_1, i_1\rangle$ is adjacent to $\langle w_2, i_2\rangle$.

Case 2: if $\phi(\langle w_1, i_1\rangle)(e, r-1) = \phi(\langle w_2, i_2\rangle)$, then $w_1 = w_2$ and $i_2 \equiv i_1 + r - 1 \pmod r$, or $i_1 \equiv i_2 + 1 \pmod r$. Hence, $\langle w_1, i_1\rangle$ is adjacent to $\langle w_2, i_2\rangle$.

Case 3: if $\phi(\langle w_1, i_1\rangle)((1\ 2), 1) = \phi(\langle w_2, i_2\rangle)$, then w_1 and w_2 differ in the i_1th bit and $i_2 \equiv i_1 + 1 \pmod r$. Hence, $\langle w_1, i_1\rangle$ is adjacent to $\langle w_2, i_2\rangle$.

Case 4: if $\phi(\langle w_1, i_1\rangle)((2r-1\ 2r), r-1) = \phi(\langle w_2, i_2\rangle)$, then w_1 and w_2 differ in the $(i_1 - 1)$th bit and $i_2 \equiv i_1 + r - 1 \pmod r$. That is, w_1 and w_2 differ in the i_2th bit and $i_1 \equiv i_2 + 1 \pmod r$. Hence, $\langle w_1, i_1\rangle$ is adjacent to $\langle w_2, i_2\rangle$.

□

Corollary 6.5 *All butterfly graphs are vertex-transitive.*

Proof: Since all Cayley graph are vertex-transitive, the result follows. □

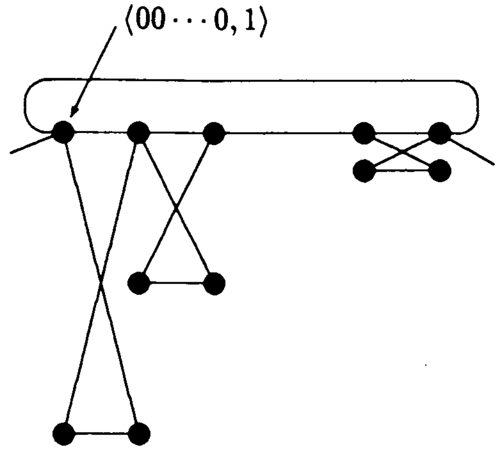
6.2 Symmetry

Although the butterfly graph is derived from the hypercube, unfortunately it does not inherit all the symmetry properties from the hypercube. In fact, the butterfly graph is not even edge-transitive. This means that it is not distance-transitive or k-distance-transitive because those transitivities imply edge-transitivity.

Theorem 6.6 *Butterfly graphs are not edge-transitive for $r \geq 3$.*

Proof: For $r \geq 3$, consider the r-cycle,

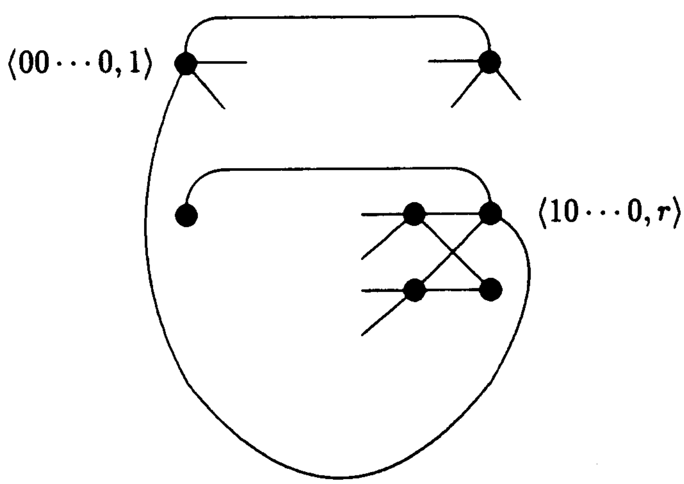
$$\langle 00\cdots 0, 1\rangle\langle 00\cdots 0, 2\rangle \cdots \langle 00\cdots 0, r-1\rangle\langle 00\cdots 0, r\rangle\langle 00\cdots 0, 1\rangle.$$



Each edge $\langle 00\cdots 0, i\rangle\langle 00\cdots 0, i+1\rangle$ in this cycle lies in the unique 4-cycle,

$$\langle 00\cdots 0, i\rangle\langle\underbrace{00\cdots 0}_{i}10\cdots 0, i+1\rangle\langle\underbrace{00\cdots 0}_{i}10\cdots 0, i\rangle\langle 00\cdots 0, i+1\rangle\langle 00\cdots 0, i\rangle.$$

These 4-cycles are edge-disjoint. If the butterfly graph is edge-transitive, the edge $\langle 00\cdots 0, 1\rangle\langle 10\cdots 0, r\rangle$ must lie in an r-cycle with the same property described above. Suppose such a cycle exists. Then $\langle 00\cdots 0, 1\rangle$ and $\langle 10\cdots 0, r\rangle$ are the first and the second vertex.



The third vertex cannot be $\langle 10\cdots 0,1\rangle$ because $\langle 00\cdots 0,1\rangle\langle 10\cdots 0,r\rangle$ and $\langle 10\cdots 0,r\rangle\langle 10\cdots 0,1\rangle$ are in the same 4-cycle

$$\langle 00\cdots 0,1\rangle\langle 10\cdots 0,r\rangle\langle 10\cdots 0,1\rangle\langle 00\cdots 0,r\rangle\langle 00\cdots 0,1\rangle.$$

Hence, the third vertex must be either $\langle 10\cdots 0,r-1\rangle$ or $\langle 10\cdots 01,r-1\rangle$. The fourth vertex cannot be in the rth level. Otherwise, the second and the third edge will be in the same 4-cycle (see the figure above). Similarly, the fifth vertex cannot be in the $(r-1)$th level. Otherwise, the third and the fourth edge will be in the same 4-cycle, and so on. It forces the last vertex v_r to be in the second level. Since the path from $\langle 00\cdots 0,1\rangle$ to the last vertex v_r passes through the rth level exactly once at the second vertex $\langle 10\cdots 0,r\rangle$, the first bit of the last vertex is 1. Hence, v_r is not adjacent to $\langle 00\cdots 0,1\rangle$. That is, the cycle in fact does not exist. Therefore, the butterfly graph is not edge-transitive. □

6.3 Topological Structure

Besides the properties of symmetry, topological properties are also very important in studying interconnection networks. For example, people are unlikely to use a network with the large diameter because in general it takes longer time to communicate. Furthermore, it may be good news for anyone who wants to pipeline the job if the network is hamiltonian. We will now consider the topological properties of the butterfly graph.

Proposition 6.7 *The r-dimensional butterfly graph has girth 4, for $r \geq 4$.*

Proof: Since $\langle 00\cdots 0,1\rangle\langle 010\cdots 0,2\rangle\langle 010\cdots 0,1\rangle\langle 00\cdots 0,2\rangle\langle 00\cdots 0,1\rangle$ is a 4-cycle, the butterfly graph has girth at most 4. Now suppose there is a triangle in the butterfly graph. Let $\langle w_1,i_1\rangle$, $\langle w_2,i_2\rangle$ and $\langle w_3,i_3\rangle$ be the vertices of this triangle. Then $|i_1-i_2|=1$, $|i_2-i_3|=1$ and $|i_1-i_3|=1$ which is impossible unless $r=3$. □

Before determining the diameter of the butterfly graph, we present a simple routing algorithm that is completely based on the definition of the butterfly graph.

Let $s = \langle w, i \rangle$ be the source node and $t = \langle w', i' \rangle$ be the destination. Let $w = a_1 a_2 \cdots a_r$ and $w' = b_1 b_2 \cdots b_r$. If $q = c_1 c_2 \cdots c_r$, then denote $flip(q, k) = c_1 c_2 \cdots c_{k-1} \bar{c}_k c_{k+1} \cdots c_r$, where

$$\bar{c}_k = \begin{cases} 0 & \text{if } c_k = 1 \\ 1 & \text{if } c_k = 0. \end{cases}$$

Algorithm 6.1: Simple Routing Algorithm for Butterfly Graph

The level indices are reduced modulo r and in the range from 1 to r.

1. Let $p \leftarrow i$, $l \leftarrow 0$ and $q_0 \leftarrow w$.
2. $l \leftarrow l + 1$, $p \leftarrow p + 1$.
3. If $a_p \neq b_p$, then

   $q_l \leftarrow flip(q_{l-1}, p)$,

   else

   $q_l \leftarrow q_{l-1}$.
4. If $p \neq i$ then go to step 2.
5. If $(i' - i) < (i - i')$ then

   the route is:

   $$\langle w, i \rangle \langle q_1, i+1 \rangle \langle q_2, i+2 \rangle \cdots \langle q_r, i \rangle \langle q_r, i+1 \rangle \cdots \langle q_r, i' \rangle,$$

   else

   the route is:

   $$\langle w, i \rangle \langle q_1, i+1 \rangle \langle q_2, i+2 \rangle \cdots \langle q_r, i \rangle \langle q_r, i-1 \rangle \cdots \langle q_r, i' \rangle.$$

Example: In the 4-dimensional butterfly graph, the route from $\langle 1011, 2 \rangle$ to $\langle 0110, 4 \rangle$ will be

$$\langle 1011, 2 \rangle \langle 1011, 3 \rangle \langle 1010, 4 \rangle \langle 0010, 1 \rangle \langle 0110, 2 \rangle \langle 0110, 1 \rangle \langle 0110, 4 \rangle.$$

The above algorithm changes the bits in w one by one to get w'. Then it continues the path in the row w' to get the correct level i'. The algorithm actually works. Notice that the loop from step 2 to step 4 runs r times. In step 5, it augments the path with at most $\left\lfloor \frac{r}{2} \right\rfloor$ vertices. Thus the length of the route that the algorithm produces is at most $r + \left\lfloor \frac{r}{2} \right\rfloor = \left\lfloor \frac{3r}{2} \right\rfloor$. With this result, we can show that the diameter of the butterfly graph is $\left\lfloor \frac{3r}{2} \right\rfloor$.

Theorem 6.8 *The r-dimensional butterfly graph has diameter $\left\lfloor \frac{3r}{2} \right\rfloor$.*

Proof: By Algorithm 6.1, we know that the length of the path between any two vertices is at most $\left\lfloor \frac{3r}{2} \right\rfloor$, so the diameter of the butterfly graph is at most $\left\lfloor \frac{3r}{2} \right\rfloor$ too. Now consider the shortest path from $\langle 00\cdots 0, r\rangle$ to $\langle 11\cdots 1, \left\lfloor \frac{r}{2} \right\rfloor\rangle$. We must change all 0's of the source to 1's and move from level r to level $\left\lfloor \frac{r}{2} \right\rfloor$. No matter how we move, we have to take at least $r + \left\lfloor \frac{r}{2} \right\rfloor = \left\lfloor \frac{3r}{2} \right\rfloor$ steps. Thus, the diameter of the butterfly is at least $\left\lfloor \frac{3r}{2} \right\rfloor$. The result follows. □

Algorithm 6.1 is not so good because it always produces a walk of length at least r. Of course, we can identify the repeated vertices and remove the vertices in between to make the walk be a path to get some improvement. However, the result is still not a shortest path in general. For example, consider a path between $\langle 000, 3\rangle$ and $\langle 011, 1\rangle$ in the 3-dimensional butterfly graph. Figure 6.2 shows that Algorithm 6.1 does not give the shortest path.

Before presenting a shortest path algorithm, we consider a simple optimization problem. Given an n-cycle

$$v_0 e_1 v_1 e_2 v_3 \cdots v_{n-2} e_{n-2} v_{n-1} e_n v_0,$$

where $v_0, v_1, \ldots, v_{n-1}$ are vertices and $e_1, e_2, \ldots e_n$ are edges, let $A \subseteq E$ be a subset of the edge-set, and $s, t \in V$ be any two vertices. The problem is to find a shortest walk from s to t so that it covers all the edges in A (See Figure 6.3).

Lemma 6.9 *The shortest walk covers each vertex at most twice.*

Proof: Suppose the walk covers the vertex v three times. Then the walk will look like one of the diagrams in Figure 6.4.

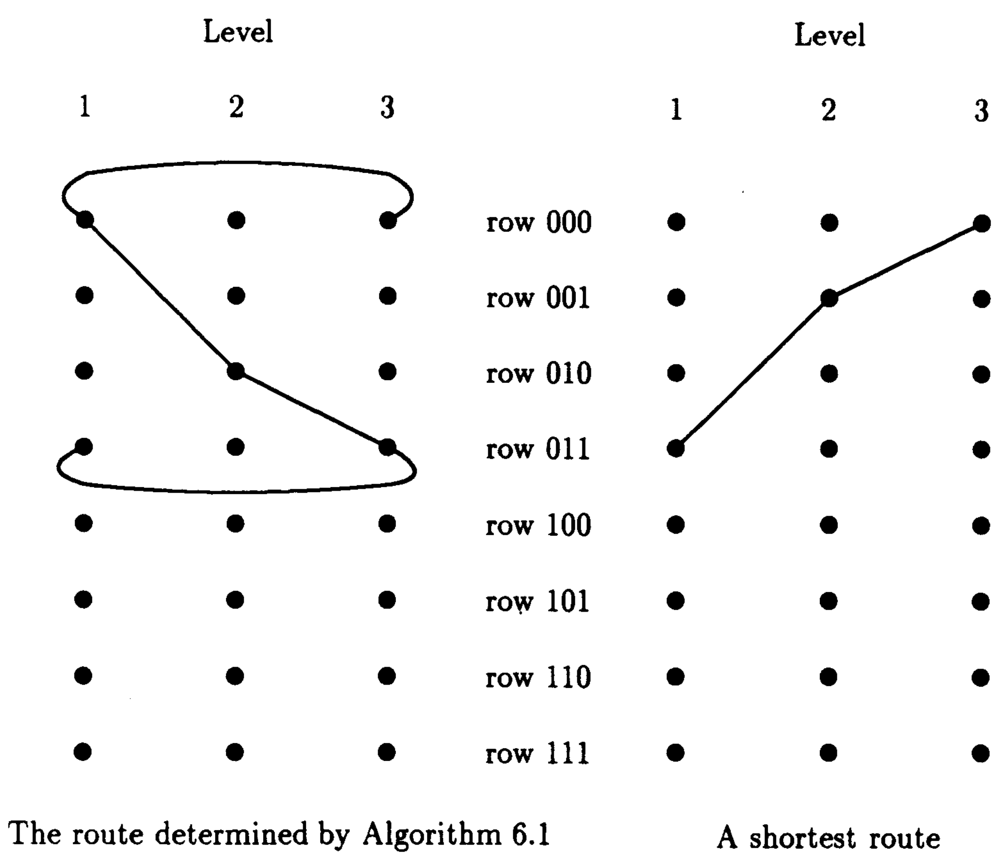


Figure 6.2: The Routes from $\langle 000, 3\rangle$ to $\langle 011, 1\rangle$

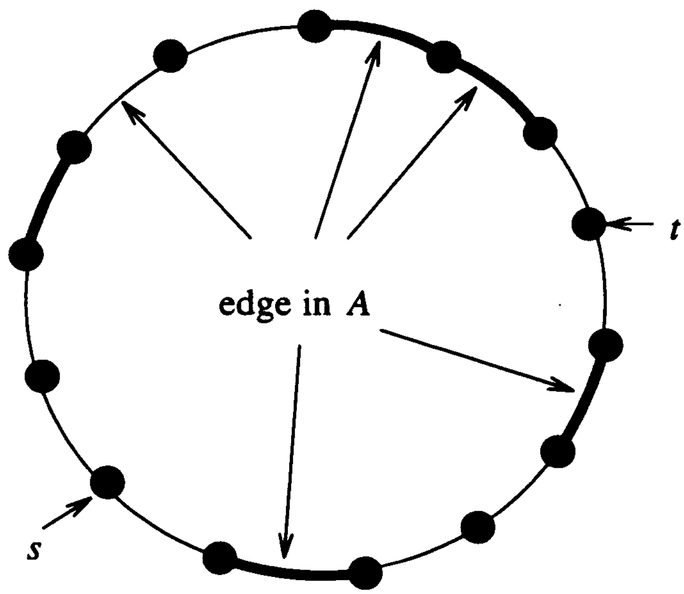


Figure 6.3: Diagram for the Shortest Walk Problem

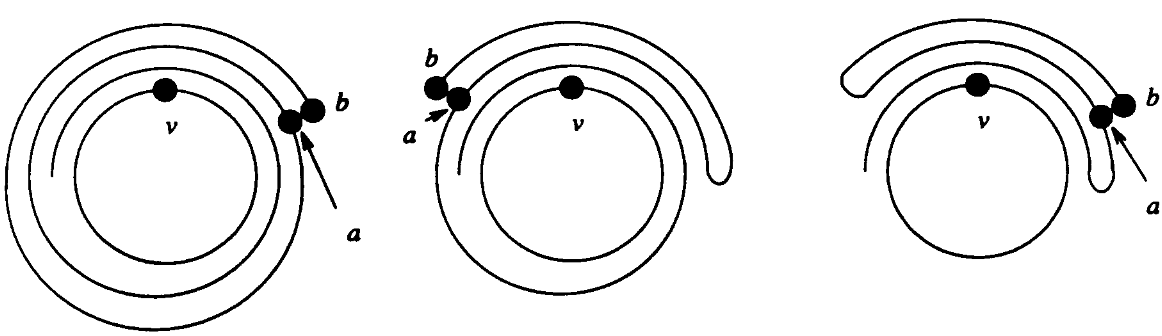


Figure 6.4: Diagram for Lemma 6.9

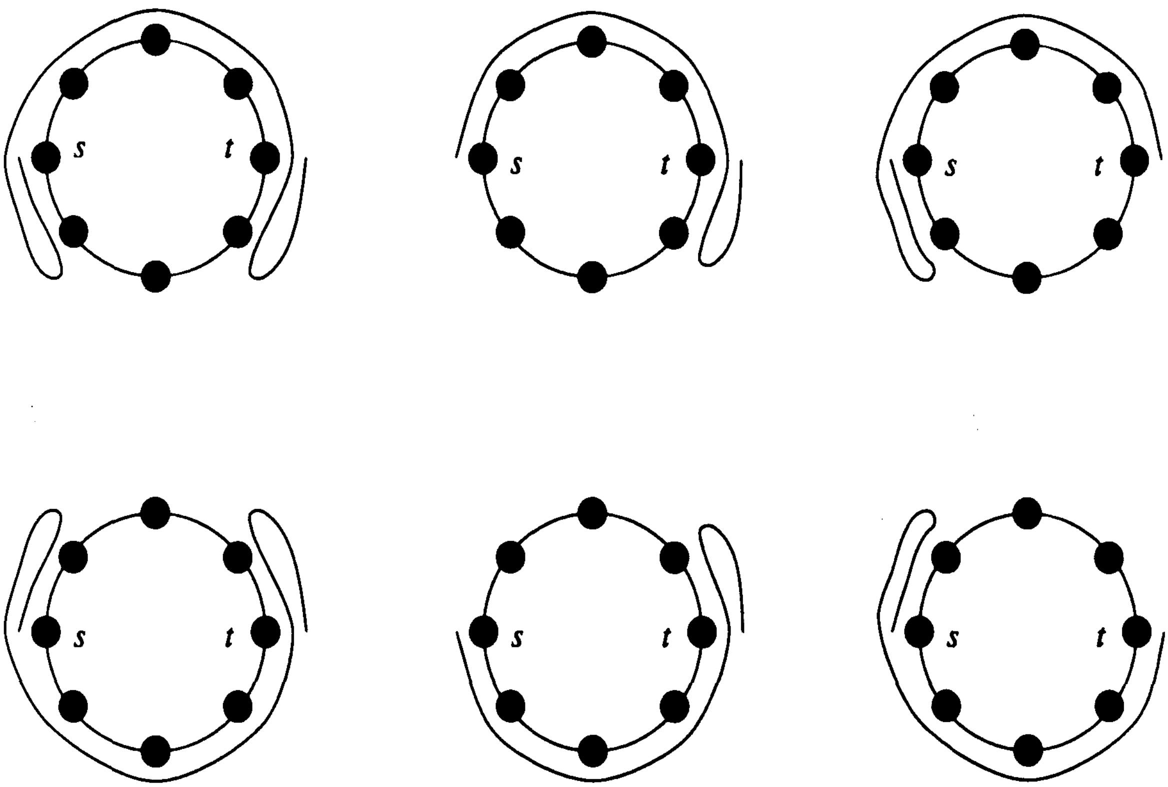


Figure 6.5: Possible Shortest Walks

Clearly, the walks from a to b are redundant in all cases. If we remove all the redundant parts, we can get a shorter walk that covers v at most once. If the walk covers the vertex more than three times, we can use this operation repeatedly to reduce the number of times the walk covers the vertex. □

The walk has to start from s and stop at t, so it must contain a path from s to t. Based on Lemma 6.9, the shortest walk will look like one of the diagrams in Figure 6.5.

Suppose $A = \{e_{i_1}, e_{i_2}, \ldots, e_{i_k}\}$. Then the walk we need to consider will be $sP_1e_{i_p}\overline{P}_1P_2tP_3e_{i_q}\overline{P}_3$ as illustrated in Figure 6.6.

Note that P_1 and P_2 may be empty and $\overline{P}_1$ and $\overline{P}_2$ are the reverse paths of P_1 and P_2, respectively. There are in fact k walks to check, and the shortest one will be the shortest walk from s to t that covers all edges in A.

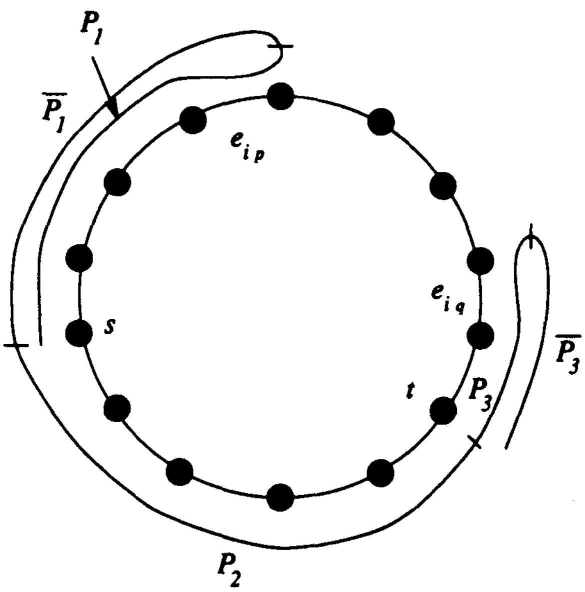


Figure 6.6: A Walk Covering All Edges in A

Now we can model the shortest path problem in the butterfly graph as the above optimization problem. Given any two vertices $\langle w, i \rangle$ and $\langle w', i' \rangle$ where $w = a_1 a_2 \ldots a_r$ and $w' = b_1 b_2 \ldots b_r$, let

$$C = v_r e_1 v_1 e_2 v_2 e_3 \cdots e_{r-2} v_{r-2} e_{r-1} v_{r-1} e_r v_r$$

be an r-cycle, $A = \{e_i : a_i \neq b_i\}$, $s = v_i$ and $t = v_{i'}$

Once we get the shortest walk in C, we can transform the solution to the shortest path of the butterfly graph. Let

$$W = v_{j_1} e_{i_1} v_{j_1} e_{i_2} v_{j_2} e_{i_3} \cdots e_{i_p} v_{j_p},$$

where $j_0 = i$ and $j_p = i'$, be a shortest walk that covers all edges in A

Algorithm 6.2: The shortest path algorithm for the butterfly graph.

The level indices are reduced modulo r and in the range from 1 to r.

1. Let $W = v_{j_1} e_{i_1} v_{j_1} e_{i_2} v_{j_2} e_{i_3} \cdots e_{i_p} v_{j_p}$, where $j_0 = i$ and $j_p = i'$, be a shortest walk that covers all edges in A

2. Let $q_0 \leftarrow w$.

3. For $l = 1$ to p

   if $e_{i_l} \in A$ then

   $q_l \leftarrow flip(q_{l-1}, i_l)$, $A \leftarrow A - \{e_{i_l}\}$,

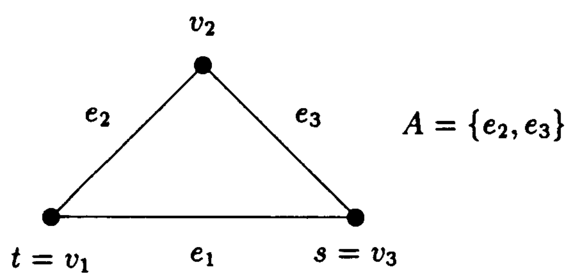
   else

   $q_l \leftarrow q_{l-1}$.

4. The shortest path from $\langle w, i\rangle$ to $\langle w', i'\rangle$ is

$$\langle q_0, j_0\rangle\langle q_1, j_1\rangle \cdots \langle q_p, j_p\rangle.$$

Example : Consider the same example mentioned before. We want to find the shortest path from $\langle 000, 3\rangle$ to $\langle 011, 1\rangle$ in a 3-dimensional butterfly graph. The associated 3-cycle is:



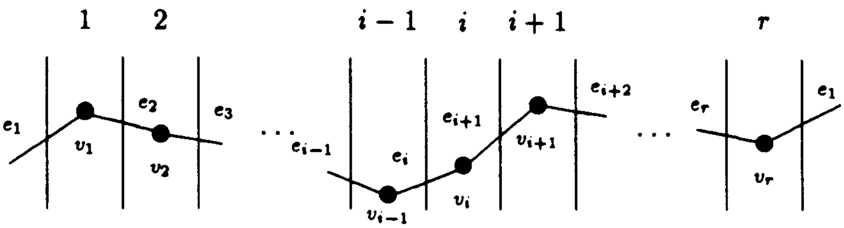
Clearly, the shortest walk is $v_3e_3v_2e_2v_1$. Using algorithm 6.2, we get

$$\langle 000, 3\rangle\langle 001, 2\rangle\langle 011, 1\rangle$$

which is the desired result.

The above algorithm works because from $\langle w, i\rangle$ to $\langle w', i'\rangle$ we have to change w bit by bit to get w'. Each time we change one bit, i will be changed to be either $i+1$ or $i-1$. Finally we have to make i become i' too. Consider the following diagram:

Level



We want to start at row i and finish at row i'. If we move to the right at row j, we can change the $(j+1)$th bit (e_{j+1} is marked) or keep it the same (e_{j+1} is not marked). If we move to the left at row j, we can change the jth bit (e_j is marked) or keep it the same (e_j is not marked). It is exactly the shortest walk problem that we have discussed.

Of course, Algorithm 6.1 is much easier to implement. It is not necessary to pre-determine the route before the node sends the message. This also means that no extra memory is required to store the route if Algorithm 6.1 is used. Thus, there is some trade-off between Algorithm 6.1 and Algorithm 6.2.

6.4 Hamilton Cycles and Hamilton Paths

Next we show that the butterfly graph is hamiltonian and discuss an algorithm to determine a Hamilton cycle. The following is an algorithm to determine a Hamilton cycle in butterfly graphs.

Algorithm 6.3: Hamilton cycle algorithm for butterfly graphs

The level indices additions are reduced modulo r and in the range from 1 to r.

1. Set $\langle w_0, i_0 \rangle \leftarrow \langle 00\cdots 0, r \rangle$.

2. For $l = 1$ to $r2^r$

   let $k = \begin{cases} 0 & \text{if } w_{l-1} = 00\cdots 0 \\ max\{j : j\text{th bit of } w_{l-1} \text{ is } 1\} & \text{otherwise.} \end{cases}$

   If $i_{l-1} < k$ then

$w_l \leftarrow w_{l-1},\ i_l \leftarrow i_{l-1} - 1,$

else

$w_l \leftarrow flip(w_{l-1}, i_{l-1}),\ i_l \leftarrow i_{l-1} - 1.$

3. The Hamilton cycle is

$$\langle w_0, i_0\rangle\langle w_1, i_1\rangle \cdots \langle w_{r2^r}, i_{r2^r}\rangle.$$

Figure 6.7 is a Hamilton cycle of the 5-dimensional butterfly graph generated by Algorithm 6.3.

Theorem 6.10 *Algorithm 6.3 generates a Hamilton cycle of the butterfly graph and hence, all butterfly graphs are hamiltonian.*

Proof: We need to show that Algorithm 6.3 generates all vertices in the butterfly graph. Suppose we start at the vertex $\langle a_1 \cdots a_{r-1}0, r\rangle$. Algorithm 6.3 will generate the following:

$$\langle a_1 \cdots a_{r-1}1, 1\rangle\langle a_1 \cdots a_{r-1}1, r-1\rangle\langle a_1 \cdots a_{r-1}1, r-2\rangle \cdots$$
$$\cdots \langle a_1 \cdots a_{r-1}1, 1, r\rangle\langle a_1 \cdots a_{r-1}0, r-1\rangle.$$

So all vertices in row $a_1 \cdots a_r$ are generated. The path also contains the vertex, $\langle a_1 \cdots a_{r-1}0, r\rangle$ and returns to $\langle a_1 \cdots a_{r-1}0, r-1\rangle$. Assume that if we start at the vertex $\langle a_1 \cdots a_j 0 \cdots 0, r\rangle$, then Algorithm 6.3 will generate the path that contains all vertices in row $a_1 \cdots a_j b_{j+1} \cdots b_r$, $b_l \in \{0,1\}$ and $\langle a_1 \cdots a_j 0 \cdots 0, l\rangle$, where $j+1 \leq l \leq r$. Also the path will return to $\langle a_1 \cdots a_j 0 \cdots 0, j\rangle$.

Now suppose we start at $\langle a_1 \cdots a_{j-1}0 \cdots 0, r\rangle$. Then by the assumption, Algorithm 6.3 will generate the path that contains all vertices in row $a_1 \cdots a_{j-1}0b_{j+1} \cdots b_r$, $b_l \in \{0,1\}$ and $\langle a_1 \cdots a_{j-1}0 \cdots 0, l\rangle$, where $j+1 \leq l \leq r$. The path will return to $\langle a_1 \cdots a_{j-1}0 \cdots 0, j\rangle$. The path will continue to $\langle a_1 \cdots a_{j-1}10 \cdots 0, j-1\rangle\langle a_1 \cdots a_{j-1}10 \cdots 0, j-2\rangle \ldots \langle a_1 \cdots a_{j-1}10 \cdots 0, r\rangle$. Then by the assumption again, Algorithm 6.3 will generate the path that contains all vertices in row $a_1 \cdots a_{j-1}1b_{j+1} \cdots b_r$, $b_l \in \{0,1\}$ and $\langle a_1 \cdots a_{j-1}10 \cdots 0, l\rangle$, where $j+1 \leq l \leq r$. The path will then return to $\langle a_1 \cdots a_{j-1}10 \cdots 0, j\rangle$ and then it continues to $\langle a_1 \cdots a_{j-1}0 \cdots 0, j-1\rangle$. By induction, if we start at $\langle 00 \cdots 0, r\rangle$, Algorithm 6.3 will generate all vertices in the butterfly graph. ❑

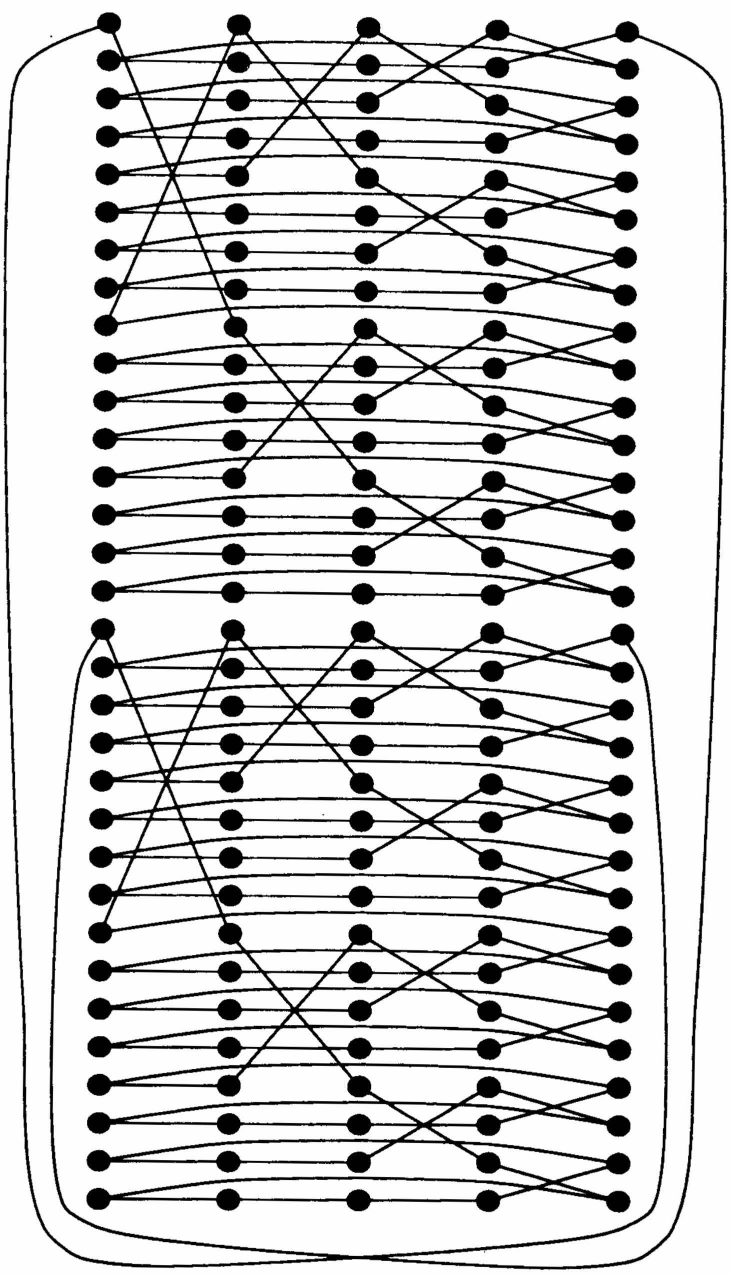


Figure 6.7: A Hamilton Cycle in the 5-dimensional Butterfly Graph

The butterfly graph is not only hamiltonian, but also hamilton-connected for odd dimension and hamilton-laceable for even dimension. We are going to show this stronger result [12].

Theorem 6.11 *The butterfly graph B_r is Hamilton-laceable when n is even.*

Proof: First we show that if B_{r-2} is Hamilton-laceable, then B_r is also Hamilton-laceable. Let $R_{r-2}(x,y)$ be the rows $xyb_3\cdots b_r$, where $b_i \in \{0,1\}$ for all $3 \leq i \leq r$. Since B_r is vertex-transitive, it is sufficient to show that there is Hamilton path from $\langle 0\cdots 0, r\rangle$ to any vertex $v = \langle a_1\cdots a_r, l\rangle$, where l is odd.

Case 1: The bits $a_1a_2 = 00$. If $l \geq 3$, then let P_{r-2} be a Hamilton path from $\langle 0\cdots 0, r-2\rangle$ to $\langle a_3\cdots a_r, l-2\rangle$ in B_{r-2}. If $l < 3$, then let P_{r-2} be a Hamilton path from $\langle 0\cdots 0, r-2\rangle$ to $\langle a_3\cdots a_r, 1\rangle$ such that the edges $\langle a_3\cdots a_r, r-2\rangle\langle \overline{a}_3\cdots a_r, 1\rangle$ and $\langle a_3\cdots a_r, r-2\rangle\langle a_3\cdots a_r, 1\rangle$ are not in P_{r-2}.

First we construct a path in $R_{r-2}(x,y)$ from P_{r-2}. We relabel each vertex $\langle b_1\cdots b_{r-2}, i\rangle$ in P_{r-2} to $\langle xyb_1\cdots b_{r-2}, i+2\rangle$ and augment level 1 and level 2 to P_{r-2}. We replace the edge $\langle xyb_1\cdots b_{r-2}, i+2\rangle\langle xy\overline{b}_1\cdots b_{r-2}, i+2\rangle$ by the path

$$\langle xy\overline{b}_1\cdots b_{r-2}, 3\rangle\langle xyb_1\cdots b_{r-2}, 2\rangle\langle xyb_1\cdots b_{r-2}, 1\rangle\langle xyb_1\cdots b_{r-2}, r\rangle,$$

and replace the edge $\langle xyb_1\cdots b_{r-2}, r\rangle\langle xyb_1\cdots b_{r-2}, 3\rangle$ by the path

$$\langle xyb_1\cdots b_{r-2}, 3\rangle\langle xyb_1\cdots b_{r-2}, 2\rangle\langle xyb_1\cdots b_{r-2}, 1\rangle\langle xyb_1\cdots b_{r-2}, r\rangle.$$

If $l = 1$, we also put a path

$$\langle xyb_1\cdots b_{r-2}, 3\rangle\langle xyb_1\cdots b_{r-2}, 2\rangle\langle xyb_1\cdots b_{r-2}, 1\rangle$$

(it is possible because of the choice of P_{r-2}). Let the resultant path be $S_{r-2}(0,0)$.

Using Algorithm 6.3, we can get a path from $\langle xy0\cdots 0, r\rangle$ to $\langle xy0\cdots 0, 2\rangle$ which contains all but $\langle xy0\cdots 0, 1\rangle$ in $R_{r-2}(x,y)$. Therefore, by putting the edge $\langle xy0\cdots 0, 1\rangle\langle xy0\cdots 0, 2\rangle$, we get a Hamilton path from $\langle xy0\cdots 0, r\rangle$ to $\langle xy0\cdots 0, 1\rangle$ in $R_{r-2}(x,y)$. We denote this path by $T_{r-2}(0,0)$.

Since P_{r-2} is a Hamilton path, it must use an edge whose endpoints are in the first and the rth level. That means S_{r-2} contains an edge whose

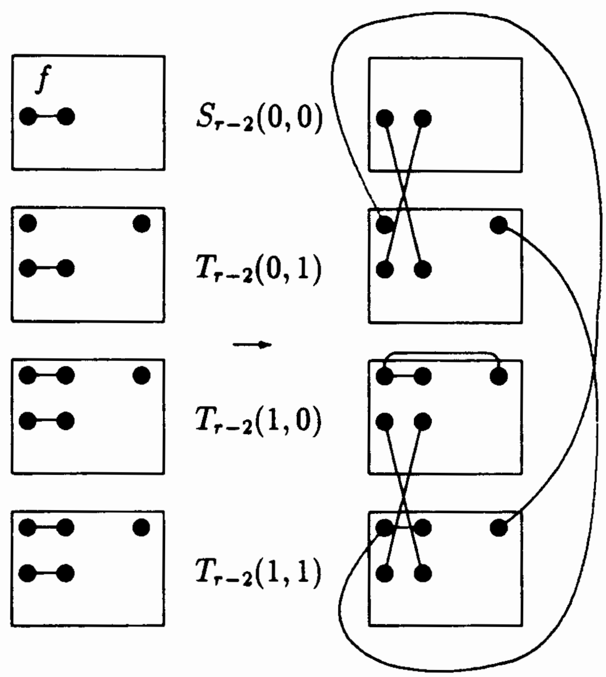


Figure 6.8: Diagram 1 for Theorem 6.11

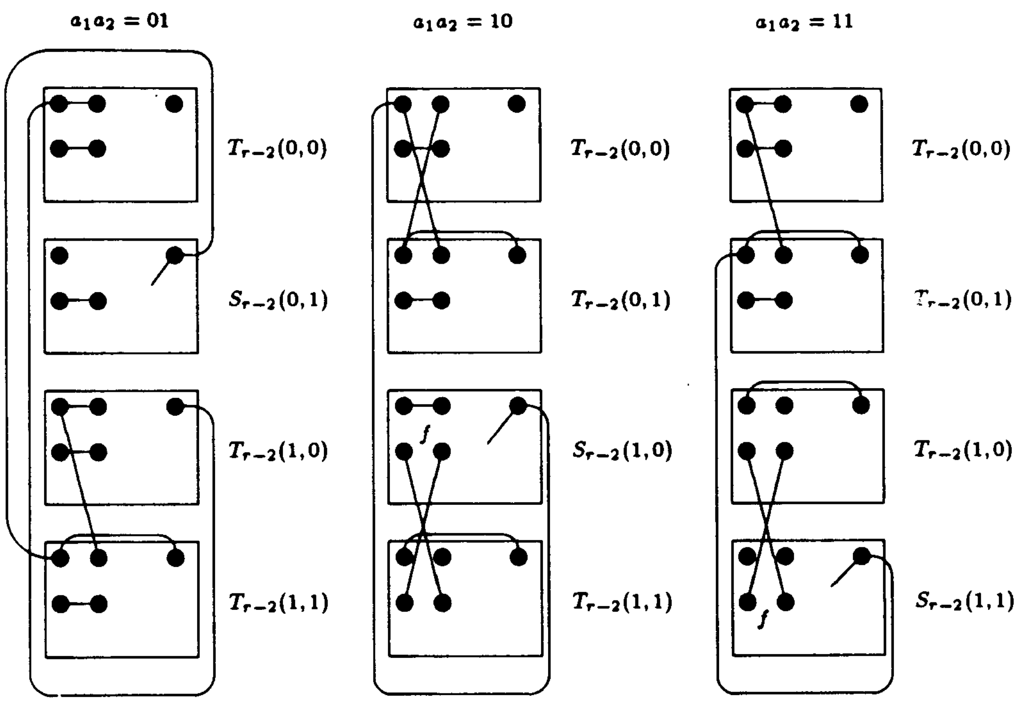


Figure 6.9: Diagram 2 for Theorem 6.11

endpoints are in the first and the second level. Let f be that edge. Now we connect S_{r-2} and three T_{r-2}'s together to get P_r as Figure 6.8 shown.

For the isolated vertices $\langle 00d_1 \cdots d_{r-2}, 1\rangle$ and $\langle 00d_1 \cdots d_{r-2}, 2\rangle$. We remove the edge $\langle 01d_1 \cdots d_{r-2}, 1\rangle\langle 01d_1 \cdots d_{r-2}, 2\rangle$ and add a path

$$\langle 01d_1 \cdots d_{r-2}, 1\rangle\langle 00d_1 \cdots d_{r-2}, 2\rangle\langle 00d_1 \cdots d_{r-2}, 1\rangle\langle 01d_1 \cdots d_{r-2}, 2\rangle.$$

Notice that P_r also satisfies the restriction of P_{r-2}.

Case 2: The bits $a_1a_2 \neq 00$ and $a_3a_4 \cdots a_r \neq 0 \cdots 0$. We first connect the paths as Figure 6.9 shown. Then we use the same method as in case 1 to join the isolated vertices into the path. Again the resultant path satisfies the restriction of P_{r-2}.

Case 3: The bits $a_1a_2 \neq 00$ and $a_3 \cdots a_r = 0 \cdots 0$. Let $\overline{S}_{r-2}(x,y)$ be the path from $\langle xy1\cdots 1, r\rangle$ to the vertices in row $xy0\cdots 0$ in $R_{r-2}(x,y)$ obtained by reversing the rows of $S_{r-2}(x,y)$ which starts from $\langle xy0\cdots 0, r\rangle$ to the vertices in row $xy1\cdots 1$. We define $\overline{T}_{r-2}(x,y)$ in a similar manner. We also let $S_{0,r-2}(x,y)$ be the path from $\langle xy0\cdots 0, r\rangle$ to $\langle xy1\cdots 1, 1\rangle$. Now we connect the paths as Figure 6.10 and join the isolated vertices to get P_r.

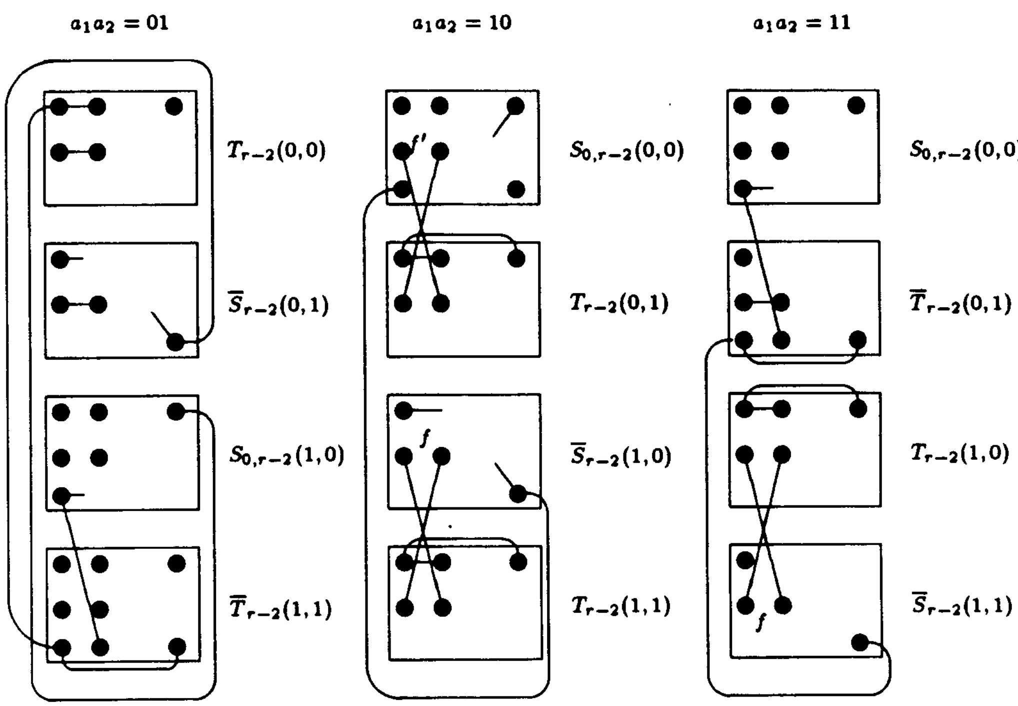
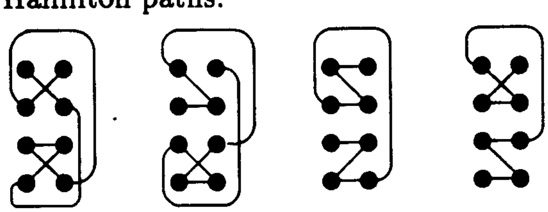


Figure 6.10: Diagram 3 for Theorem 6.11

Notice that all of P_r satisfies the restriction of P_{r-2}. When $r = 2$, we have the following Hamilton paths.



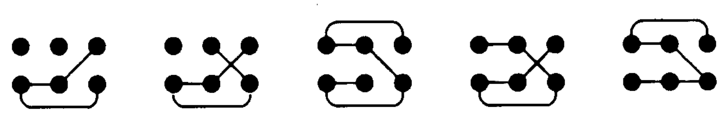
By induction, the result follows. □

Theorem 6.12 *The Butterfly graph B_r is Hamilton-connected when r is odd.*

Proof: we will use the notation in Theorem 6.11. The proof is basically the same as Theorem 6.11. Suppose B_{r-2} is Hamilton-connected. We want to find a Hamilton path from $\langle 0 \ldots 0, r\rangle$ to any vertex $\langle a_1 \cdots a_r, l\rangle$. The same procedure in Theorem 6.11 will be use to get the Hamilton path in B_r. There is one exception: $l = 2$. When $l = 2$ and $a_3 \cdots a_r \neq 0 \cdots 0$. We let P_{r-2} be the path from $\langle 0 \cdots 0, r-2\rangle$ to $\langle a_3 \cdots a_r, r-2\rangle$ such that it does not have the edges $\langle a_3 \cdots a_r, 1\rangle\langle a_3 \cdots a_r, r-2\rangle$ and $\langle a_3 \cdots a_r, r-2\rangle\langle \overline{a}_3 \cdots a_r, 1\rangle$. Then we construct $S_{r-2}(x,y)$ as before and add a path $\langle xya_3 \cdots a_r, r\rangle\langle xya_3 \cdots a_r, 1\rangle\langle xya_3 \cdots a_r, 2\rangle$. The rest will be the same as Theorem 6.11.

If $l = 2$ and $a_3 \cdots a_r = 0 \cdots 0$, the we let $S_{r-2}(x,y)$ be the path from $\langle xy0 \cdots 0, r\rangle$ to $\langle xy1 \cdots 1, r\rangle$. We also let $S_{0,r-2}(x,y)$ be the path from $\langle xy0 \cdots 0, r\rangle$ to $\langle xy1 \cdots 1, 1\rangle$. Then we connect the paths as Figure 6.11 and Figure 6.12 shown.

Notice that all P_r satisfy the restriction of P_{r-2}. For the Hamilton paths starting from $\langle 000, 3\rangle$ in B_3. We can use the following $S_1(x,y)$ to generate them.



By induction, the result follows. □

6.5 Connectivity

Unlike the hypercube, the butterfly graph has the bounded connectivity because of its fixed degree.

Theorem 6.13 *The connectivity of r-dimensional butterfly graph is 4.*

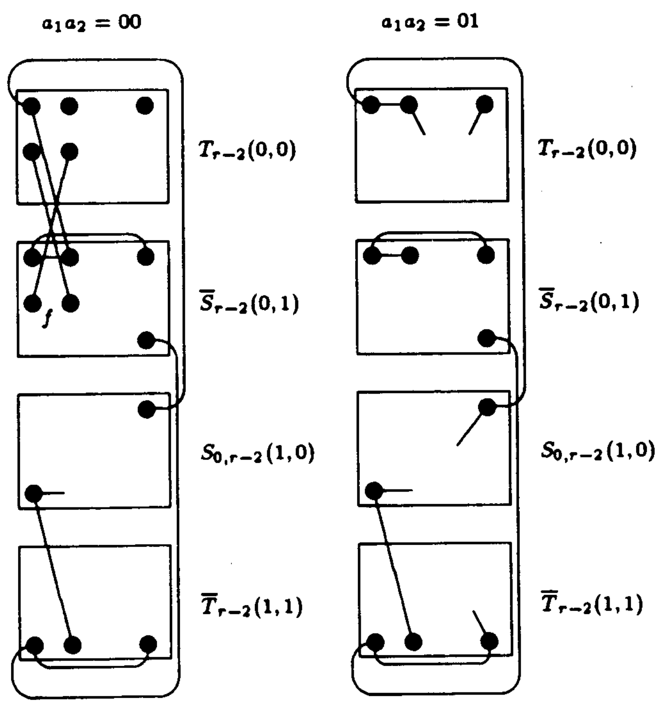


Figure 6.11: Diagram 1 for Theorem 6.12

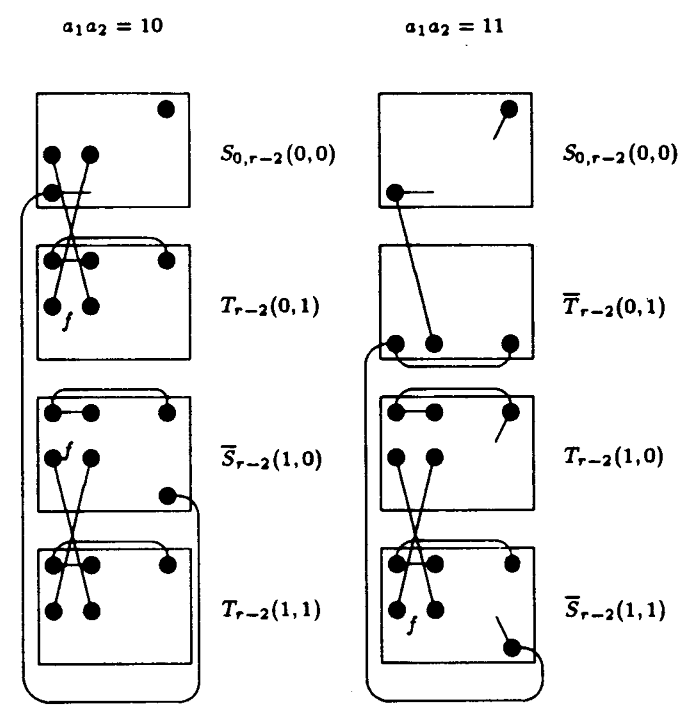


Figure 6.12: Diagram 2 for Theorem 6.12

Proof: Let u, v be any two vertices in B_r. Since B_r is vertex-transitive, we can assume that $u = \langle 0\cdots 0, r\rangle$. Using the Algorithm 6.3, we can get a Hamilton cycle $\langle 0\cdots 0, r\rangle A v B \langle 0\cdots 0, r\rangle$. If A or B is empty, then $B_r - \{\langle 0\cdots 0, r\rangle, v\}$ is connected. We assume that both A and B are non-empty. Then $\langle 0\cdots 01, r-1\rangle \in A$ and $\langle 10\cdots 0, 1\rangle \in B$. Consider the paths

$P_1 : \langle 0\cdots 01, r-1\rangle\langle 0\cdots 01, r\rangle\langle 10\cdots 01, 1\rangle\langle 10\cdots 01, 2\rangle\cdots\langle 10\cdots 01, r-1\rangle$
$\langle 10\cdots 0, r\rangle\langle 10\cdots 0, 1\rangle$.

$P_2 : \langle 0\cdots 01, r-1\rangle\langle 0\cdots 011, r-2\rangle\cdots\langle 01\cdots 11, 1\rangle\langle 11\cdots 1, r\rangle$
$\langle 11\cdots 10, r-1\rangle\cdots\langle 10\cdots 0, 1\rangle$.

$P_3 : \langle 0\cdots 01, r-1\rangle\langle 0\cdots 01, r-2\rangle\cdots\langle 0\cdots 01, 1\rangle\langle 10\cdots 01, r\rangle$
$\langle 10\cdots 0, r-1\rangle\langle 10\cdots 0, r-2\rangle\cdots\langle 10\cdots 0, 1\rangle$.

Those are the vertex-disjoint paths from $\langle 0\cdots 01, r-1\rangle$ to $\langle 10\cdots 0, 1\rangle$. There must exist an edge from A to B. That is, $B_r - \{\langle 0\cdots 0, r\rangle, v\}$ is connected. Therefore, the connectivity of B_r is at least 3.

Let u, v and w be any three vertices in B_r. Again, we can assume that $u = \langle 0\cdots 0, r\rangle$. Using Algorithm 6.3 and relabelling v and w if necessary, we can get a Hamilton cycle $\langle 0\cdots 0, r\rangle A v B w C \langle 0\cdots 0, r\rangle$. If any two of A, B and C are empty, then $B_r - \{\langle 0\cdots 0, r\rangle, v, w\}$ is still connected. If only B is empty, then we can use the same argument above to show that the graph is still connected. Suppose only A is empty. Then B contains $\langle 0\cdots 01, r-2\rangle$ and C contains $\langle 10\cdots 0, 1\rangle$. Consider the paths

$P_1 : \langle 0\cdots 01, r-2\rangle\langle 0\cdots 011, r-1\rangle\langle 10\cdots 011, 1\rangle\langle 10\cdots 011, 2\rangle\cdots$
$\langle 10\cdots 011, r-2\rangle\langle 10\cdots 01, r-1\rangle\langle 10\cdots 0, r\rangle\langle 10\cdots 0, 1\rangle$.

$P_2 : \langle 0\cdots 01, r-2\rangle\langle 0\cdots 0101, r-3\rangle\cdots\langle 01\cdots 101, 1\rangle\langle 11\cdots 101, r\rangle$
$\langle 11\cdots 100, r-1\rangle\langle 11\cdots 100, r-2\rangle\langle 11\cdots 1000, r-3\rangle\cdots\langle 10\cdots 0, 1\rangle$.

$P_3 : \langle 0\cdots 01, r-2\rangle\langle 0\cdots 01, r-3\rangle\cdots\langle 0\cdots 01, 1\rangle\langle 10\cdots 01, r\rangle$
$\langle 10\cdots 0, r-1\rangle\langle 10\cdots 0, r-2\rangle\cdots\langle 10\cdots 0, 1\rangle$.

Those are the vertex-disjoint paths from $\langle 0\cdots 01, r-1\rangle$ to $\langle 10\cdots 0, 1\rangle$. There must exist an edge from B to C. That is, $B_r - \{\langle 0\cdots 0, r\rangle, v, w\}$ is connected.

Suppose only C is empty. Then A contains $\langle 0\cdots 01, r-1\rangle$ and B contains $\langle 110\cdots 0, 2\rangle$. Consider the paths

$P_1 : \langle 0\cdots 01, r-1\rangle\langle 0\cdots 01, r\rangle\langle 10\cdots 01, 1\rangle\langle 110\cdots 01, 2\rangle\cdots$
$\langle 110\cdots 01, r-1\rangle\langle 110\cdots 0, r\rangle\langle 110\cdots 0, r-1\rangle\cdots\langle 110\cdots 0, 2\rangle$.

$P_2 : \langle 0\cdots 01, r-1\rangle\langle 0\cdots 011, r-2\rangle\cdots\langle 01\cdots 11, 1\rangle\langle 11\cdots 1, r\rangle$
$\langle 11\cdots 10, r-1\rangle\cdots\langle 110\cdots 0, 2\rangle$.

$P_3 : \langle 0\cdots 01, r-1\rangle\langle 0\cdots 01, r-2\rangle\cdots\langle 0\cdots 01, 1\rangle\langle 10\cdots 01, r\rangle$
$\langle 10\cdots 0, r-1\rangle\langle 10\cdots 0, r-2\rangle\cdots\langle 10\cdots 0, 2\rangle\langle 110\cdots 0, 1\rangle\langle 110\cdots 0, 2\rangle$.

Those are the vertex-disjoint paths from $\langle 0\cdots 01, r-1\rangle$ to $\langle 110\cdots 0, 2\rangle$. There must exist an edge from A to B. That is, $B_r - \{\langle 0\cdots 0, r\rangle, v, w\}$ is connected.

Finally, suppose all A, B and C are non-empty. Then $\langle 0\cdots 01, r-1\rangle \in A$ and C contains $\langle 10\cdots 0, 1\rangle$. We can use the three vertex-disjoint path from $\langle 0\cdots 01, r-1\rangle$ to$\langle 10\cdots 0, 1\rangle$ discribed above to show that there is a path from A to C. If there is an edge from A or C to B, then $B_r - \{\langle 0\cdots 0, r\rangle, v, w\}$ is connected. Otherwise, since the connectivity of B_r is at least 3, either $\langle 0\cdots 0, r-1\rangle$ or $\langle 0\cdots 0, 1\rangle$ is in B. For the first case, we have the following three vertex-disjoint paths from $\langle 0\cdots 01, r-1\rangle$ to $\langle 0\cdots 0, r-1\rangle$:

$P_1 : \langle 0\cdots 01, r-1\rangle\langle 0\cdots 01, r\rangle\langle 0\cdots 0, r-1\rangle$.

$P_2 : \langle 0\cdots 01, r-1\rangle\langle 0\cdots 011, r-2\rangle\langle 0\cdots 011, r-1\rangle\langle 0\cdots 010, r\rangle$
$\langle 0\cdots 010, 1\rangle\cdots\langle 0\cdots 010, r-2\rangle\langle 0\cdots 0, r-1\rangle$.

$P_3 : \langle 0\cdots 01, r-1\rangle\langle 0\cdots 01, r-2\rangle\cdots\langle 0\cdots 01, 1\rangle\langle 10\cdots 01, r\rangle$
$\langle 10\cdots 01, r-1\rangle\langle 10\cdots 0, r\rangle\langle 0\cdots 0, 1\rangle\langle 0\cdots 0, 2\rangle\cdots\langle 0\cdots 0, r-1\rangle$.

For the second case, we have the following three vertex-disjoint paths from $\langle 0\cdots 01, r-1\rangle$ to $\langle 0\cdots 0, 1\rangle$:

$P_1 : \langle 0\cdots 01, r-1\rangle\langle 0\cdots 01, r\rangle\langle 0\cdots 0, r-1\rangle\langle 0\cdots 0, 1\rangle$.

$P_2 : \langle 0\cdots 01, r-1\rangle\langle 0\cdots 01, r-2\rangle\cdots\langle 0\cdots 01, 1\rangle\langle 10\cdots 01, r\rangle$
$\langle 10\cdots 01, r-1\rangle\cdots\langle 10\cdots 0, r\rangle\langle 0\cdots 0, 1\rangle$.

$P_3 : \langle 0\cdots 01, r-1\rangle\langle 0\cdots 011, r-2\rangle\cdots\langle 01\cdots 1, 1\rangle\langle 01\cdots 1, 2\rangle$
$\langle 0101\cdots 1, 3\rangle\cdots\langle 010\cdots 0, r\rangle\langle 010\cdots 0, 1\rangle\langle 010\cdots 0, 2\rangle\ \langle 0\cdots 0, 1\rangle$.

Hence, there must be a path from A to B. $B_r - \{\langle 0\cdots 0, r\rangle, v, w\}$ is connected. Since B_r is 4-regular, it has connectivity 4. □

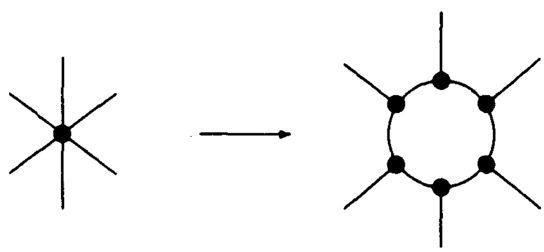
Corollary 6.14 *The edge-connectivity of r-dimensional butterfly graph is 4.*

Proof: Since B_r is 4-regular and has connectivity 4, the result follows. □

Chapter 7

Cube-connected-cycles

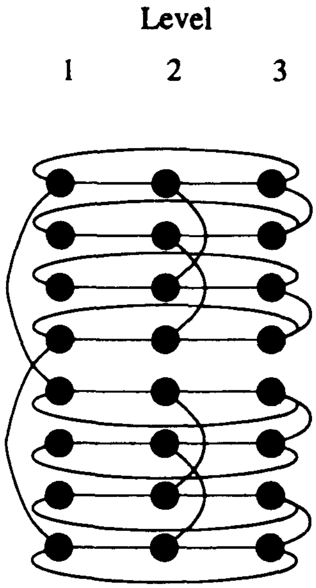
In the previous chapter, we discuss an extension of the hypercube, the butterfly graph. It not only inherits some properties from the hypercube, but also has bounded degree. In this chapter, we will discuss another extension of the hypercube which also has bounded degree. Consider the following operation:



Let v be a vertex of a hypercube Q_r, $r \geq 3$. We replace each vertex by an r-cycle as illustrated above. Then the resultant graph will become a 3-regular graph and is called the r-dimensional cube-connected-cycles. Figure 7.1 is a 3-dimensional cube-connected-cycles.

7.1 Modelling

The following is the formal definition of r-dimensional cube-connected-cycles [9].



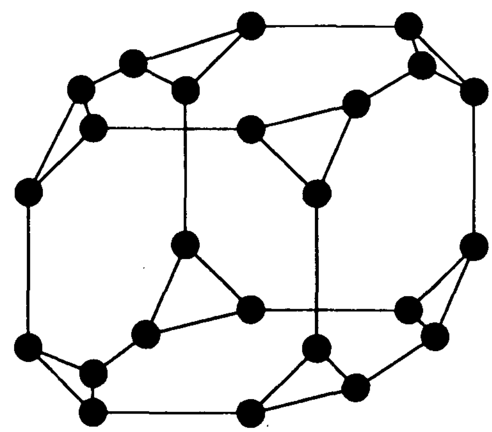


Figure 7.1: The 3-dimensional cube-connected-cycles

Definition 7.1 Let $G(V, E)$ be a graph with $|V| = r2^r$ and $|E| = 3r2^{r-1}$ for some positive integer r. The vertices in G are labelled by $\langle w, i\rangle$, where w is a binary sequence of length r that denotes the row of the vertex and i is the level of the vertex ($1 \le i \le r$). Two vertices $\langle w, i\rangle$ and $\langle w', i'\rangle$ are adjacent if and only if either:

1. $w = w'$ and $i' \equiv i \pm 1 (\mathrm{mod}\ r)$ or
2. w and w' differ in precisely the ith bit and $i' = i$.

The graph is called *r-dimensional cube-connected-cycles and* is denoted as CCC_r.

It is not suprising that cube-connected-cycles are also Cayley graphs. By comparing the cube-connected-cycles and the butterfly graphs, we can discover some similarities between these two classes of graphs. In fact, the group used to generate the cube-connected cycles is exactly the one for the butterfly graphs [11, 1].

Proposition 7.2 *Let Γ_r be the group in Proposition 6.3 and $S = \{(e,1), (e,r\text{-}1), ((1\ 2),0)\}$. The Cayley graph $G(\Gamma_r, S) \cong CCC_r$.*

Proof: Let $\langle w,i\rangle$ be a vertex in CCC_r, where $w = a_1a_2\cdots a_r$ is a binary sequence of length r. Define $\phi : CCC_r \to G(\Gamma, S)$ by $\phi(\langle w,i\rangle) = \phi(\langle a_1\cdots a_r, i\rangle) = (p_1\cdots p_r, i)$, where

$$p_i = \begin{cases} (2i\text{-}1\ 2i) & \text{if } a_i = 1 \\ e & \text{otherwise.} \end{cases}$$

We have already shown that this mapping is a bijection in Proposition 6.4. Let $\langle w_1,i_1\rangle, \langle w_2,i_2\rangle \in CCC_r$, where $w_1 = a_1\cdots a_r$ and $w_2 = b_1\cdots b_r$. If $\langle w_1,i_1\rangle$ is adjacent to $\langle w_2,i_2\rangle$, then there are two cases.
Case 1: $w_1 = w_2$ and $i_1 \equiv i_2 + 1 \pmod r$. Then $\phi(\langle w_1,i_1\rangle)(e,1) = \phi(\langle w_2,i_2\rangle)$. Therefore, $\phi(\langle w_1,i_1\rangle)$ is adjacent to $\phi(\langle w_2,i_2\rangle)$.
Case 2: w_1 and w_2 differ in the i_1th bit and $i_1 = i_2$. Then $\phi(\langle w_1,i_1\rangle)((1\ 2),0) = \phi(w_2,i_2)$. Thus, $\phi(\langle w_1,i_1\rangle)$ is adjacent to $\phi(\langle w_2,i_2\rangle)$.

Conversely, if $\phi(\langle w_1,i_1\rangle)$ is adjacent to $\phi(\langle w_2,i_2\rangle)$, there are three cases.
Case 1: If $\phi(\langle w_1,i_1\rangle)(e,1) = \phi(\langle w_2,i_2\rangle)$, then $w_1 = w_2$ and $i_2 \equiv i_1 + 1 \pmod r$. Hence, $\langle w_1,i_1\rangle$ is adjacent to $\langle w_2,i_2\rangle$.
Case 2: If $\phi(\langle w_1,i_1\rangle)(e,r-1) = \phi(\langle w_2,i_2\rangle)$, then $w_1 = w_2$ and $i_2 \equiv i_1+r-1 \pmod r$, or $i_1 \equiv i_2 + 1 \pmod r$. Hence, $\langle w_1,i_1\rangle$ is adjacent to $\langle w_2,i_2\rangle$.
Case 3: If $\phi(\langle w_1,i_1\rangle)((1\ 2),0) = \phi(\langle w_2,i_2\rangle)$, then w_1 and w_2 differ in the i_1th bit and $i_2 = i_1$. Hence, $\langle w_1,i_1\rangle$ is adjacent to $\langle w_2,i_2\rangle$.

This shows that ϕ is an isomorphism. □

Corollary 7.3 *All cube-connected-cycles are vertex-transitive.* □

7.2 Symmetry

As with the butterfly graph, the cube-connected-cycles is not edge-transitive. This also means that the cube-connected-cycles cannot be distance-transitive or even k-distance-transitive.

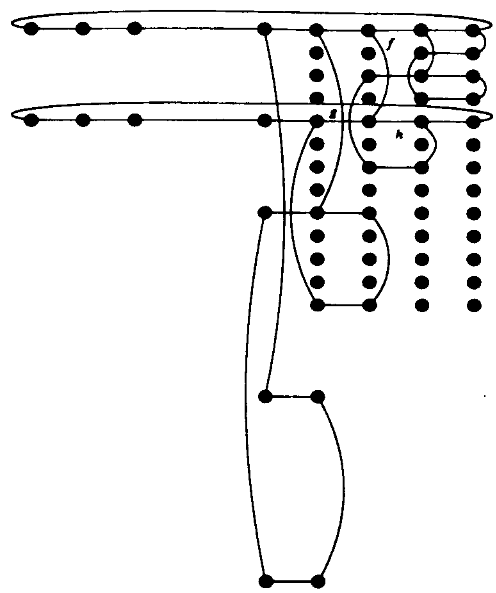


Figure 7.2: Diagram 1 for Theorem 7.4

Theorem 7.4 *The CCC_r is not edge-transitive for* $r \geq 3$.

Proof: Consider the r-cycle of row $\langle 00\cdots 0\rangle$ when $r \neq 8$ (figure 7.2.)

$$\langle 00\cdots 0,1\rangle\langle 00\cdots 0,2\rangle\cdots\langle 00\cdots 0,r-1\rangle\langle 00\cdots 0,r\rangle\langle 00\cdots 0,1\rangle.$$

Each edge $\langle 00\cdots 0,i\rangle\langle 00\cdots 0,i+1\rangle$ in this cycles lies in the unique 8-cycle

$$\langle 0\cdots 0,i\rangle\langle\underbrace{0\cdots 0}_{i-1}10\cdots 0,i\rangle\langle\underbrace{0\cdots 0}_{i-1}10\cdots 0,i+1\rangle\langle\underbrace{0\cdots 0}_{i-1}110\cdots 0,i+1\rangle$$
$$\langle\underbrace{0\cdots 0}_{i-1}10\cdots 0,i\rangle\langle\underbrace{0\cdots 0}_{i}10\cdots 0,i\rangle\langle\underbrace{0\cdots 0}_{i}10\cdots 0,i+1\rangle\langle 0\cdots 0,i+1\rangle$$
$$\langle 0\cdots 0,i\rangle.$$

If the cube-connected-cycles is edge-transitive, then we can find an automorphism that maps the edge $\langle 0\cdots 0,1\rangle\langle 0\cdots 0,2\rangle$ to the edge $\langle 0\cdots 0,r-2\rangle\langle 0\cdots 0100,r-2\rangle$. Thus, the edge $\langle 0\cdots 0,r-2\rangle\langle 0\cdots 0100,r-2\rangle$ must lie in an r-cycle with the same property. Let W be this r-cycle. The edge $\langle 0\cdots 0,r-2\rangle\langle 0\cdots 100,r-2\rangle$ lies in two 8-cycles (see figure 7.2). If the edge g is in W, then

$$\langle 0\cdots 0100,1\rangle\langle 0\cdots 0100,2\rangle\cdots\langle 0\cdots 0100,r\rangle$$

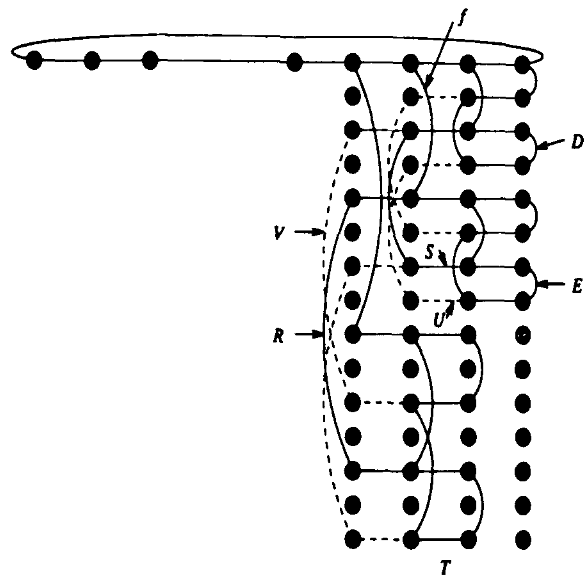


Figure 7.3: Diagram 2 for Theorem 7.4

will be the 8-cycle containing g. Similarly, if the edge h is in W, then the same cycle

$$\langle 0\cdots 0100,1\rangle\langle 0\cdots 0100,2\rangle\cdots\langle 0\cdots 0100,r\rangle$$

will be the 8-cycle containing h. Hence, r must be 8, but it is a contradiction.

When $r = 8$, there are two types of 8-cycles. We call the 8-cycle of each row be the 8-cycle of type I, and the 8-cycle lying in more than one row be the 8-cycle of type II. Two 8-cycles are said to be adjacent if they have an edge in common. Consider those 8-cycles of type II in Figure 7.2 again. Any two disjoint 8-cycles of type II have only one common adjacent 8-cycle of type I which is the cycle of row $00\cdots 0$. If the cube-connected-cycles is edge-transitive, again we want to find an automorphism that maps the edge $\langle 0\cdots 0,1\rangle\langle 0\cdots 0,2\rangle$ to the edge $\langle 0\cdots 0,r-2\rangle\langle 0\cdots 0100,r-2\rangle$. In this case, we can map the cycle $\langle 0\cdots 0,1\rangle\langle 0\cdots 0,2\rangle\cdots\langle 0\cdots 0,r\rangle\langle 0\cdots 0,1\rangle$ to one of the cycles R or S (see figure 7.3). If R is the image, then the disjoint cycles S and T have two common adjacent cycles R and V. This means that this case is impossible. If S is the image, then the disjoint cycles D and E have two common adjacent cycles U and S. Again it is impossible. Hence, it is impossible to map the edge $\langle 00\cdots 0,1\rangle\langle 00\cdots 0,2\rangle$ to the edge

$\langle 00\cdots 0, r-2\rangle\langle 00\cdots 0100, r-2\rangle$. That is, none of the cube-connected-cycles are edge-transitive.

7.3 Topological Structure

We now consider topological properties of the cube-connected-cycles.

Proposition 7.5 *The r-dimensional cube-connected cycle has girth r for $r \leq 8$ and girth 8 for $r > 8$.*

Proof: Consider a cycle $\langle w_1 i_1\rangle\langle w_2 i_2\rangle\cdots\langle w_k i_k\rangle\langle w_1 i_1\rangle$. If all w_i's are the same, we will get a r-cycle. Since any two rows are joined by at most one edge, it is impossible that a cycle lies in precisely two rows. Similarly, any three rows are joined by at most two edges, so it is impossible that a cycle is lies in precisely three rows. Suppose the cycle lies in precisely four rows. The cycle must use at least one edge from each row. This implies the cycle must have length at least 8.

The edge $\langle a_1\cdots a_r, i\rangle\langle a_1\cdots a_r, i+1\rangle$ lies in the r-cycle

$$\langle a_1\cdots a_r, 0\rangle\cdots\langle a_1\cdots a_r, r\rangle\langle a-1\cdots a_r, 0\rangle$$

and an 8-cycle

$$\langle a_1\cdots a_i a_{i+1}\cdots a_r, i\rangle\langle a_1\cdots a_i a_{i+1}\cdots a_r, i+1\rangle\langle a_1\cdots a_i \overline{a}_{i+1}\cdots a_r, i+1\rangle$$
$$\langle a_1\cdots a_i \overline{a}_{i+1}\cdots a_r, i\rangle\langle a_1\cdots \overline{a}_i \overline{a}_{i+1}\cdots a_r, i\rangle\langle a_1\cdots \overline{a}_i \overline{a}_{i+1}\cdots a_r, i+1\rangle$$
$$\langle a_1\cdots \overline{a}_i a_{i+1}\cdots a_r, i+1\rangle\langle a_1\cdots \overline{a}_i a_{i+1}\cdots a_r, i\rangle\langle a_1\cdots a_i a_{i+1}\cdots a_r, i\rangle.$$

Hence, the result follows. □

As the butterfly graphs and the cube-connected-cycles are generated by the same group, there are some similarities between these two classes of graphs. For instance, if we modify the simple routing algorithm for the butterfly graph (Algorithm 6.1), we will get the following version for the cube-connected-cycles.

Let $s = \langle w, i\rangle$ be the source node and $t = \langle w', i'\rangle$ be the destination, where $w = a_1 a_2\cdots a_r$ and $w' = b_1 b_2\cdots b_r$.

Algorithm 7.1: Simple Routing Algorithm for Cube-Connected-Cycles

The level indices are reduced modulo r and in the range from 1 to r.

1. Let $p \leftarrow i$, $l \leftarrow 0$, $q_0 \leftarrow w$ and $i_0 \leftarrow i$.

2. If $|i+1-i'| < |i-1-i'|$ then

   $s = 1,$

   else

   $s = -1.$

3. $l \leftarrow l+1.$

4. If $a_p \neq b_p$, then

   $q_l \leftarrow flip(q_{l-1}, p),$

   $i_l \leftarrow i_{l-1},$

   else

   $q_l \leftarrow q_{l-1},$

   $i_l \leftarrow i_{l-1} + s,$

   $p \leftarrow p + s.$

5. If $p \neq i - s$, then go to step 2.

6. If $a_{i-s} \neq b_{i-s}$ then

   $l \leftarrow l+1,$

   $q_l \leftarrow flip(q_{l-1}, i-s),$

   $i_l \leftarrow i_{l-1}.$

7. The route is

$$\langle w, i\rangle\langle q_1, i_1\rangle\langle q_2, i_2\rangle \cdots \langle q_l, i_l\rangle\langle q_l, i_l - s\rangle\langle q_l, i_l - 2s\rangle \cdots \langle q_l, i'\rangle.$$

The idea of this algorithm is the same as that of Algorithm 6.1. It changes the bits in w one-by-one to get w'. Then it continues the route in row w' to get the correct level i'. Notice that the loop from step 3 to step 5 in Algorithm 7.1 runs at most $2r-1$ times. Because of step 2, the algorithm augments the route with at most $\left\lfloor \frac{r-2}{2} \right\rfloor$ vertices if $r \geq 4$, or 1 if $r = 3$. Hence, the algorithm always gives a route of length at most $2r-1+\left\lfloor\frac{r-2}{2}\right\rfloor=\left\lfloor\frac{5r-4}{2}\right\rfloor$, when $r \geq 4$, and at most $2(3)-1+1=6$, when $r=3$. This leads to the following result [7].

Theorem 7.6 *The CCC_r has diameter $\left\lfloor\frac{5r-4}{2}\right\rfloor$, when $r \geq 4$, and 6 when $r=3$.*

Proof: When $r=3$, the length of the shortest route from $\langle 000,3\rangle$ to $\langle 111,2\rangle$ is 6. The result follows. When $r \geq 4$, the algorithm gives a route of length at most $\left\lfloor\frac{5r-4}{2}\right\rfloor$. Consider the route from $\langle 00\cdots 0,r\rangle$ to $\langle 11\cdots 1,\left\lfloor\frac{r}{2}\right\rfloor\rangle$. The route must hit at least r rows. The route also comes across each level at least once. The ith level is only connected to the $(i-1)$th and $(i+1)$th level. This forces the route to traverse at least $r-1+\left\lfloor\frac{r}{2}\right\rfloor-1=\left\lfloor\frac{3r-4}{2}\right\rfloor$ vertices if the route starts from level r, ends at level $\left\lfloor\frac{r}{2}\right\rfloor$ and crosses every level at least once. Hence, the length of the route from $\langle 00\cdots 0,r\rangle$ to $\langle 11\cdots 1,\left\lfloor\frac{r}{2}\right\rfloor\rangle$ is at least $r+\left\lfloor\frac{3r-4}{2}\right\rfloor=\left\lfloor\frac{5r-4}{2}\right\rfloor$. The result follows. $\square$

Algorithm 7.1 does not give the shortest route in general. For instance, consider the route from $\langle 0000,4\rangle$ to $\langle 1100,2\rangle$. Algorithm 7.1 will generate the following route:

$$\langle 0000,4\rangle\langle 0000,3\rangle\langle 0000,2\rangle\langle 0100,2\rangle\langle 0100,1\rangle\langle 1100,1\rangle\langle 1100,2\rangle.$$

However, the shortest route is

$$\langle 0000,4\rangle\langle 0000,1\rangle\langle 1000,1\rangle\langle 1000,2\rangle\langle 1100,2\rangle.$$

We are now going to investigate the shortest path algorithm for cube-connected-cycles. In fact, the idea is exactly the same as that for the butterfly graphs. We know that if the source and the destination differ in k bits, the route must traverse at least k rows. There is no way to reduce this number. However, we can minimize the number of levels that the route hits. First consider a simple optimization problem. Given an n-cycle

$$C=v_1v_2\cdots v_r,$$

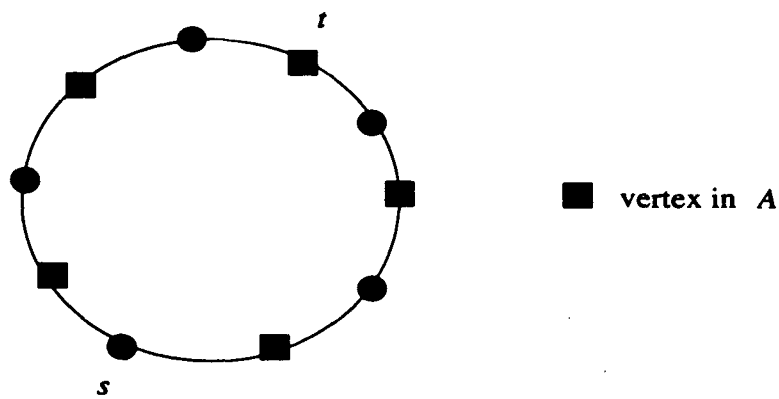


Figure 7.4: Diagram for the Shortest Walk Problem.

where $v_1v_2\cdots v_r$ are vertices, let $A \subseteq V$ be a subset of the vertex-set, and $s, t \in V$ be any two vertices. The problem is to find a shortest walk from s to t so that it covers all the vertices in A (see figure 7.4).

This optimization problem is almost the same as that for the butterfly graphs. In fact, we can use the same method to solve this problem. This solution corresponds to the minimum number of levels that the route must hit.

Now we can model the shortest path problem in the cube-connected-cycles as the above optimization problem. Given any two vertices $\langle w, i\rangle$ and $\langle w', i'\rangle$, where $w = a_1a_2\cdots a_r$ and $w' = b_1b_2\cdots b_r$, let

$$C = v_1v_2v_3\cdots v_r$$

be an r-cycle, $A = \{v_k : a_k \neq b_k, 1 \leq k \leq r\}$, $s = v_i$ and $t = v_{i'}$.

Suppose we have a shortest walk in C that covers all vertices in A. Let

$$W = v_{j_0}v_{j_1}v_{j_2}v_{j_3}\cdots v_{j_p},$$

where $j_0 = i$ and $j_p = i'$, be such a shortest walk. We can now transform the solution to the shortest path in the cube-connected-cycles.

Algorithm 7.2: The shortest path algorithm for the cube-connected-cycles.

The level indices are reduced modulo r and in the range from 1 to r.

1. Let $W = v_{j_0}v_{j_1}v_{j_2}v_{j_3}\cdots v_{j_p}$ be the shortest walk of the associated optimization problem.

2. Let $q_0 \leftarrow w$, $l \leftarrow 0$, $k \leftarrow 0$ and $i_0 \leftarrow i$.

3. $l \leftarrow l + 1$.

4. If $v_{j_k} \in A$ then

   $q_l \leftarrow flip(q_{l-1}, j_k)$, $i_l \leftarrow j_k$, $A \leftarrow A - \{v_{j_k}\}$,

   else

   $k \leftarrow k + 1$, $q_l \leftarrow q_{l-1}$, $i_l \leftarrow j_k$.

5. If $k < p$ then go to step 2.

6. If $v_{j_p} \in A$ then

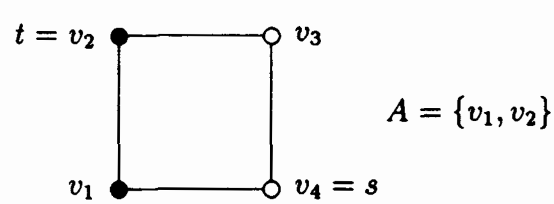
   $l \leftarrow l + 1$,

   $q_l \leftarrow flip(q_{l-1}, j_p)$, $i_l \leftarrow j_p$.

7. The shortest path is

$$\langle q_0, i_0\rangle\langle q_1, i_1\rangle\langle q_2, i_2\rangle\cdots\langle q_l, i_l\rangle.$$

Example: Consider the same example mentioned before. We are looking for a shortest path from $\langle 0000, 4\rangle$ to $\langle 1100, 2\rangle$. The associated 4-cycle is shown below.



Clearly the shortest walk is $W = v_4v_1v_2$. Algorithm 7.2 will generate the path

$$\langle 0000, 4\rangle\langle 0000, 1\rangle\langle 1000, 1\rangle\langle 1000, 2\rangle\langle 1100, 2\rangle.$$

which is the desired result.

7.4 Hamilton Cycles

The cube-connected-cycles is again hamiltonian. The result has been proven by R. Stong [11].

Theorem 7.7 *All cube-connected-cycles are hamiltonian.*

Proof: Let R_j be the subgraph of an r-dimensional cube-connected-cycles induced by the rows

$$0\cdots 0a_{j+1}a_{j+2}\cdots a_r, \qquad a_i \in \{0,1\}, \quad j+1 \leq i \leq r.$$

Suppose R_j is hamiltonian. Consider R_{j-2}. It consists of four copies of R_j which are $R_j(0,0)$, $R_j(0,1)$, $R_j(1,0)$ and $R_j(1,1)$.

$$\boxed{R_j(0,0)}$$

$$\boxed{R_j(0,1)}$$

$$\boxed{R_j(1,0)}$$

$$\boxed{R_j(1,1)}$$

The vertices $\langle\underbrace{0\cdots0}_{j-2}xy0\cdots0, j-1\rangle$ and $\langle\underbrace{0\cdots0}_{j-2}xy0\cdots0, j\rangle$ have degree 2 in $R_j(x,y)$. Therefore, we can let $P_j(x,y)$ be a Hamilton path starting at $\langle\underbrace{0\cdots0}_{j-2}xy0\cdots0, j-1\rangle$ and ending at $\langle\underbrace{0\cdots0}_{j-2}xy0\cdots0, j\rangle$. Let $\overline{P}_j(x,y)$ be the reverse of this Hamilton path. Then we can get a Hamilton cycle of R_{j-2}, namely,

$$P_j(0,0)\overline{P}_j(0,1)P_j(1,1)\overline{P}_j(1,0).$$

When r is even, R_r is an r-cycle, so the result follows. When r is odd, we have the Hamilton cycle of R_{r-3} (Figure 7.5). By induction, the result follows. $\square$

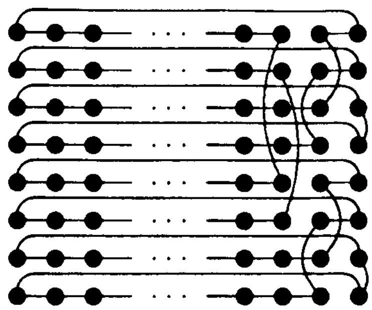


Figure 7.5: Hamilton Cycle on R_{r-3}

Figure 7.6 illustrates Hamilton cycles of the 4-dimensional and the 5-dimensional cube-connected-cycles.

Theorem 7.7 gives a recursive construction of a Hamilton cycle of the cube-connected-cycles. In practice, we may want to have an algorithm that can determine the next vertex of the Hamilton cycle directly. With the algorithm, we do not need to store a Hamilton cycle in memory.

In the recursive construction, the even cases and the odd cases are separated. In the following algorithm, we still separate it into two different cases. For the even cases, let $v = \langle a_1a_2\cdots a_{2p}, i\rangle$. We choose

$$\langle 0\cdots 0, r\rangle\langle 0\cdots 0, r-1\rangle\cdots\langle 0\cdots 0, 1\rangle\langle 0\cdots 0, r\rangle.$$

be the base cycle. The Hamilton cycle of R_{j-2} is given by

$$P_j(0,0)\overline{P}_j(0,1)P_j(1,1)\overline{P}_j(1,0).$$

This implies that we either flip the ith bit or add 1 to i (opposite direction of R_r) to get the next one when the number of 1's in $a_1a_2\cdots a_{2p}$ is odd. Similarly, if the number of 1's in $a_1a_2\cdots a_{2p}$ is even, we either flip the ith bit or substract 1 for i (same direction of R_r). If v is in the form $\langle a_1\cdots a_{i-2}xy\underbrace{0\cdots 0}_{s}, i\rangle$, s even, and $xy = 00$ or $xy = 11$, then we flip the ith bit according to the recursive construction. Let $l = max\{j : a_j = 1\}$ and k = number of 1's in $a_1\cdots a_{2p}$. Hence, we flip the ith bit when $i \geq l$, and i

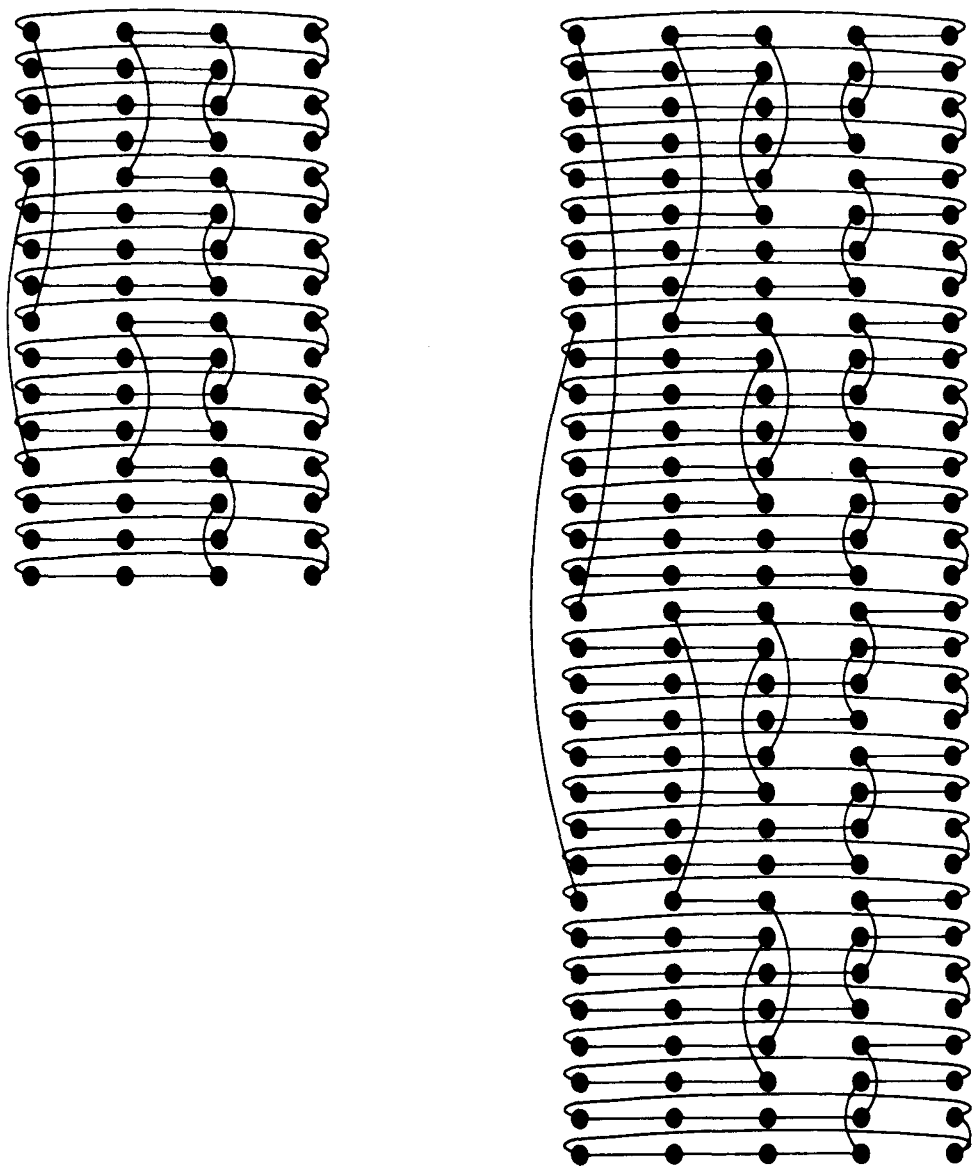


Figure 7.6: Hamilton Cycles in the 4- and 5-dimensional Cube-connected-cycles

and k are both even. If v is in the form $\langle a_1 \cdots a_{i-1} x y \underbrace{0 \cdots 0}_{s}, i \rangle$, s odd, and $xy = 01$ or $xy = 10$, then again we flip the ith bit according to the recursive construction. Hence, we flip the ith bit when $i \geq l - 1$, and i and k are both odd. We have the following algorithm for the even case.

Algorithm 7.3: Algorithm for generating a Hamilton cycle of CCC_r when r is even.

The level indices are reduced modulo r and in the range from 1 to r.

1. $\langle w_0, i_0 \rangle \leftarrow \langle 00 \cdots 0, r \rangle$.
2. For $p = 1$ to $r2^r$

    let $l = \begin{cases} 0 & \text{if } w_{p-1} = 00 \cdots 0 \\ max\{j : j\text{th bit of } w_{p-1} \text{ is } 1\} & \text{otherwise,} \end{cases}$

    let k = number of 1's in w_{p-1}.

    If k is even then

      if $i_{p-1} \geq l$ then

        if i_{p-1} is even then

          $w_p \leftarrow flip(w_{p-1}, i_{p-1})$, $i_p \leftarrow i_{p-1}$,

        else

          $w_p \leftarrow w_{p-1}$, $i_p \leftarrow i_{p-1} - 1$,

      else

        $w_p \leftarrow w_{p-1}$, $i_p \leftarrow i_{p-1} - 1$,

    else

      if $i_{p-1} \geq l - 1$ then

        if i_{p-1} is odd then

          $w_p \leftarrow flip(w_{p-1}, i_{p-1})$, $i_p \leftarrow i_{p-1}$,

        else

          $w_p \leftarrow w_{p-1}$, $i_p \leftarrow i_{p-1} + 1$,

      else

        $w_p \leftarrow w_{p-1}$, $i_p \leftarrow i_{p-1} + 1$.

3. The Hamilton cycle is

$$\langle w_0, i_0\rangle\langle w_1, i_1\rangle \cdots \langle w_{r2^r}, i_{r2^r}\rangle.$$

The odd case is basically the same as the even case. Since the recursive construction starts from R_{r-3}, some modifications are needed.

Algorithm 7.3: Algorithm for generating a Hamilton cycle of CCC_r when r is odd.

The level indices are reduced modulo r and in the range from 1 to r.

1. $\langle w_0, i_0\rangle \leftarrow \langle 00\cdots 0, r\rangle$

2. For $p = 1$ to $r2^r$

   let $l = \begin{cases} 0 & \text{if } w_{p-1} = 00\cdots 0 \\ max\{j : j\text{th bit of } w_{p-1} \text{ is } 1\} & \text{otherwise,} \end{cases}$

   let k = number of 1's in w_{p-1}.

   If k is even then

     if $a_{r-1} = 1$ then

       if $i_{p-1} = r - 1$ then

         $w_p \leftarrow flip(w_{p-1}, i_{p-1})$, $i_p \leftarrow i_{p-1}$,

       else

         $w_p \leftarrow w_{p-1}$, $i_p \leftarrow i_{p-1} + 1$,

     else

       if $l \geq r - 1$ then

         if $i_{p-1} = r - 1$ then

           $w_p \leftarrow flip(w_{p-1}, i_{p-1})$, $i_p \leftarrow i_{p-1}$,

         else

           $w_p \leftarrow w_{p-1}$, $i_p \leftarrow i_{p-1} - 1$,

       else

         if $i \geq l$ and i is even then

           $w_p \leftarrow flip(w_{p-1}, i_{p-1})$, $i_p \leftarrow i_{p-1}$, else

$w_p \leftarrow w_{p-1}$, $i_p \leftarrow i_{p-1} - 1$,

else

if $a_{r-1} = 1$ then

if $i_{p-1} = r$ then

$w_p \leftarrow flip(w_{p-1}, i_{p-1})$, $i_p \leftarrow i_{p-1}$,

else

$w_p \leftarrow w_{p-1}$, $i_p \leftarrow i_{p-1} - 1$,

else

if $l \geq r - 2$ then

if $i_{p-1} = r - 2$ then

$w_p \leftarrow flip(w_{p-1}, i_{p-1})$, $i_p \leftarrow i_{p-1}$,

else

$w_p \leftarrow w_{p-1}$, $i_p \leftarrow i_{p-1} + 1$,

else

if $l - 1 \leq i \leq r - 1$ and i is odd then

$w_p \leftarrow flip(w_{p-1}, i_{p-1})$, $i_p \leftarrow i_{p-1}$,

else

$w_p \leftarrow w_{p-1}$, $i_p \leftarrow i_{p-1} + 1$.

3. The Hamilton cycle is

$$\langle w_0, i_0\rangle\langle w_1, i_1\rangle \cdots \langle w_{r2^r}, i_{r2^r}\rangle.$$

7.5 Connectivity

Since the cube-connected-cycles has the bounded degree, the connectivity is bounded. Using the fact that the cube-connected-cycles is hamiltonian, we have the following result.

Theorem 7.8 *The connectivity of the r-dimensional cube-connected-cycles is 3.*

Proof: Let u, v be any two vertices in CCC_r. Since CCC_r is vertex-transitive, we can assume that $u = \langle 0 \cdots 0, r \rangle$. Using the Algorithm 7.3, we can get a Hamilton cycle $\langle 0 \cdots 0, r \rangle A v B \langle 0 \cdots 0, r \rangle$. If A or B is empty, then $CCC_r - \{\langle 0 \cdots 0, r \rangle, v\}$ is connected. We assume that both A and B are non-empty. Then $\langle 0 \cdots 01, r \rangle \in A$ and $\langle 0 \cdots 0, 1 \rangle \in B$. Consider the paths

$P_1 : \langle 0 \cdots 01, r \rangle \langle 0 \cdots 01, 1 \rangle \langle 10 \cdots 01, 1 \rangle \langle 10 \cdots 01, r \rangle \langle 10 \cdots 0, r \rangle$
$\langle 10 \cdots 0, 1 \rangle \langle 0 \cdots 0, 1 \rangle$.

$P_2 : \langle 0 \cdots 01, r \rangle \langle 0 \cdots 01, r-1 \rangle \cdots \langle 0 \cdots 01, 2 \rangle \langle 010 \cdots 01, 2 \rangle$
$\langle 010 \cdots 01, 3 \rangle \cdots \langle 010 \cdots 01, r \rangle \langle 010 \cdots 01, r \rangle \langle 010 \cdots 0, r \rangle$
$\langle 010 \cdots 0, r-1 \rangle \langle 010 \cdots 0, 2 \rangle \langle 0 \cdots 0, 2 \rangle \langle 0 \cdots 0, 1 \rangle$.

These are the vertex-disjoint paths from $\langle 0 \cdots 01, r \rangle$ to $\langle 0 \cdots 0, 1 \rangle$ without the vertex $\langle 0 \cdots 0, r \rangle$. Since v can only disconnect one of the paths, There must exist an edge from A to B. That is, $CCC_r - \{\langle 0 \cdots 0, r \rangle, v\}$ is connected. Since CCC_r is 3-regular, the result follows. □

Corollary 7.9 *The edge-connectivity of the r-dimensional cube-connected-cycles is 3.*

Proof: The r-dimensional cube-connected-cycles is 3-regular and has connectivity 3. The result follows. □

7.6 Summary

Before finishing the thesis, we make a little summary of the results that we get in Table 7.1.

Table 7.1: Summary of the Results

	Q_r	B_r	CCC_r
Cayley Graph	Yes	Yes	Yes
Group	$\langle (1\ 2), (3\ 4), \ldots$ $(r\text{-}1\ r) \rangle$	the group defined in Proposition 6.3	the group defined in Proposition 6.3
Symbol Set	$\{(1\ 2), (3\ 4), \ldots$ $(r\text{-}1\ r)\}$	$\{(e,1), (e, r-1),$ $((1\ 2), 1),$ $((2r\text{-}1\ 2r), r\text{-}1)\}$	$\{(e,1), (e, r-1),$ $((1\ 2), 0)\}$
Transitivity	distance-transitive	vertex-transitive	vertex-transitive
Degree	r	4	3
Girth	4	4 if $r \geq 4$ 3 if $r = 3$	$\min\{8, r\}$ r
Diameter	r	$\lfloor \frac{3r}{2} \rfloor$	$\lfloor \frac{5r-4}{2} \rfloor$
Hamiltonian	Yes	Yes	Yes
Connectivity	r	4	3
Edge-connectivity	r	4	3
Bipartite	all r	r even	r even
Hamilton-laceable/ connected	Yes	Yes	†

† As far as the author knows, the 3-dimensional cube-connected-cycles is hamilton-connected and 4-dimensional cube-connected-cycles is hamilton-laceable.

Bibliography

[1] S. B. Akers and B. Krishnamurthy. *A group-theoretic model for symmetric interconnection networks.* IEEE Transactions on Computers, **38**, No. 4 (1989), 555–566.

[2] N. Biggs. *Algebraic Graph Theory.* Cambridge University Press, Cambridge, 2nd edition, 1993.

[3] Bélá Bollobás. *Graph Theory: A Introduction Course.* Spring-Verlag New York Inc., New York, 1979.

[4] J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications.* American Elsevier Publisher Co., Inc., New York, 1976.

[5] J. B. Fraleigh. *A First Course in Abstract Algebra.* Addison-Wesley Publishing Company, Ontario, 4th ed. edition, 1989.

[6] M. Hall. *The Theory of Groups.* The Macmillan Company, New York, 1968.

[7] M. S. Krishnamoorthy and B. Krishnamurthy. *Fault diameter of interconnection networks.* Comput. Math. Applic., **13**, No. 5/6 (1987), 577–582.

[8] S. Lakshmivarahan, Jung-Sing Jwo, and S. K. Dhall. *Symmetry in interconnection networks based on Cayley graphs of permutation groups: A survey.* Parallel Computing, **19** (1993), 361–407.

[9] F. T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays·Trees·Hypercubes.* Morgan Kaufmann Publishers, Inc., California, 1992.

[10] J. J. Rotman. *The Theory of Groups: An Introduction.* Allyn and Bacon, Inc., Boston, 1966.

[11] R. Stong. *On hamiltonian cycles in Cayley graphs of wreath products.* Discrete Math., **65**, No. 1 (1987), 75–80.

[12] S. A. Wong. *Hamilton cycles and paths in butterfly graphs*, March 1995.

[13] H. P. Yap. *Some Topics in Graph Theory.* Cambridge University Press, Cambridge, 1986.