2012

Permutations with Extremal Routings on Cycles

Luis Alejandro Valentin
College of William and Mary

Permutations with Extremal Routings on Cycles

A thesis submitted in partial fulfillment of the requirement for the degree of Bachelor of Science in Mathematics from The College of William and Mary

by

Luis Alejandro Valentin

Accepted for Honors

Professor Gexin Yu, advisor

Professor Ryan Vinroot

Professor Virginia Torczon

Williamsburg, VA
April 30, 2012

Acknowledgements

I would like first to thank my advisor, Professor Gexin Yu for his guidance and confidence in me. I want also to thank Professors Ryan Vinroot and Virginia Torczon for serving on my defense committee. I want to thank the NSF CSUMS program for providing funding for this research. Finally, I want to thank my good friends Kevin Ji and Jeff Soosiah for helping me get through many frustratingly long nights.

I would like to dedicate this thesis to a wonderful mentor, Mr. Jim Allison.

Abstract

Let G be a graph on n vertices, labeled $v_1, \ldots, v_n$. Suppose that on each vertex there is a pebble, p_j which has a destination of v_j. During each step, a disjoint set of edges is selected and the pebbles on an edge are swapped. The routing problem, $rt(G, \pi)$, asks what the minimum number of steps necessary for any permutation of the pebbles to be routed so that for each pebble, p_i is on v_i.

Li, Lu, and Yang prove that the routing number of a cycle of n vertices is equal to $n-1$. They conjecture that for $n \geq 5$, if $rt(C_n, \pi) = n-1$, then $\pi = (123 \cdots n)$ or its inverse. They show that the conjecture holds true for values of n less than 8. We prove here that the conjecture holds for all even n.

Contents

Chapter 0

Background in Graph Theory

For completeness, we include here a brief introduction to graph theory. Standard graph theory terminology is presented, which will be used throughout.

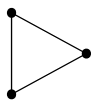
0.1 Graphs

In graph theory, a **graph** consists of a set of **vertices** and a set of edges, where an **edge** is a pair of vertices. Generally, the number of vertices and edges are denoted n and m, respectively. When drawn, vertices are represented by dots, while edges are represented by lines connecting two dots. Edges can either be **undirected** or **directed**, in which case the edge is an ordered pair. An edge is called a **loop** if it begins and ends at the same vertex. A **simple graph** allows at most one edge between any two vertices and no loops. **Multi-graphs** allow for more than one edge between any two vertices and loops. The **degree** of a vertex is equal to the number of edges adjacent to that vertex.

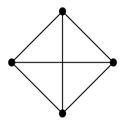
0.2 Classes of Graphs

Graphs with certain common properties are characterized into classes. Here we describe a few classes that are of interest.

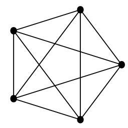
First, a graph with n vertices and all possible edges is known as a **complete graph**, denoted K_n. The number of edges equals $\binom{n}{2} = \frac{n(n+1)}{2}$. Below are the graphs K_n for n from three to six.



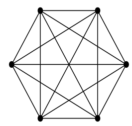
[K_3]



[K_4]

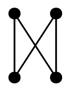


[K_5]

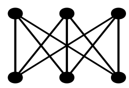


[K_6]

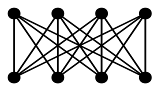
A **bipartite graph** is one where the vertices can be partitioned into two sets, A and B, such that each edge has one endpoint in A and the other in B. A **complete bipartite graph** is a bipartite graph with every edge possible. Such a graph is denoted $K_{m,n}$, where the sizes of each partition is m and n and the number of vertices is $m + n$. The number of edges in $K_{m,n}$ is mn. Below are the first few balanced bipartite graphs.



[$K_{2,2}$]



[$K_{3,3}$]



[$K_{4,4}$]

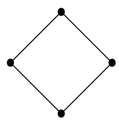
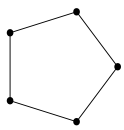
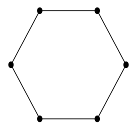
Suppose we place n vertices in a line and connect any vertices directly next to each other. We have $n-1$ edges that form the class P_n, the path of length n. Each vertex has degree two except for the end vertices, which have degree one. Below are a few examples.

[P_3]

[P_4]

[P_5]

If n vertices are placed in a circle and neighbor vertices are connected by an edge, we get what is known as a cycle graph, C_n. Cycles have the same number of vertices and edges. Below are a few examples.


[C_4]

[C_5]

[C_6]

0.3 Properties of Graphs

There exist many classes of graphs because one can consider the existence and non-existence of any of many properties of graphs. One property which is of interest is a **matching**. A matching H is a subgraph of G where the degree of each vertex in H is at most one. That is, a matching is the union of some disjoint edges in G. If every vertex of a matching has degree one, the matching is known as a **perfect matching**.

A **maximum matching** is a matching of G with the most number of edges possible. A **maximal matching** is a matching such that no edge can be added and maintain the matching property.

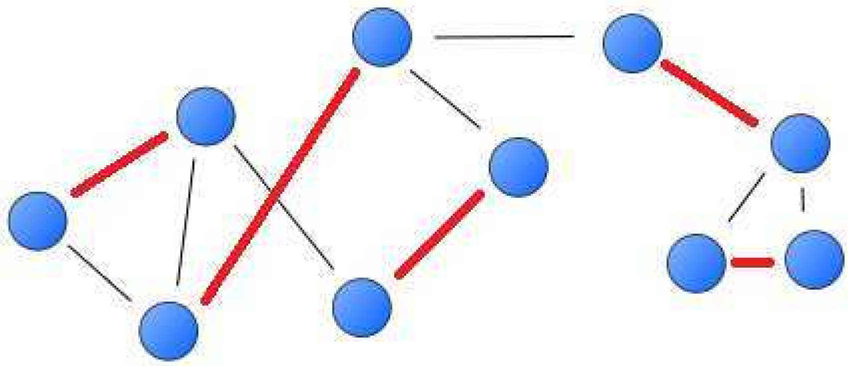


Figure 1: An example of a matching. The marked edges form a perfect matching, which, by definition, is maximum and maximal.

Chapter 1

Introduction

1.1 Graph Routings

Routing problems occur in many areas of computer science. Sorting a list involves routing each element to the proper location. Communication across a network involves routing messages through appropriate intermediaries. Message passing between multiprocessors requires the routing of signals to correct processors.

In each case, one would like the routing to be done as quickly as possible. We will use a routing model first described by Alon in 1994 [2]. Let $G = \{V, E\}$ be a graph. Label the vertices from v_1 to v_n. Place at each vertex a pebble p_i ($1 \leq i \leq n$). Let π be the permutation corresponding to the location of each pebble. That is, p_i is placed at $v_{\pi(i)}$. We wish to move each p_i to its destination vertex v_i. To do so, we repeatedly apply the following action until p_i is at v_i for all i: select a matching of G and swap the pebbles at the endpoints of each edge.

Let $rt(G, \pi)$ denote the minimum number of steps necessary to route π on G. Then, the **routing number** of G is defined as:

$$rt(G) = \max_{\pi} rt(G, \pi),$$

where π is any permutation of the vertices of G.

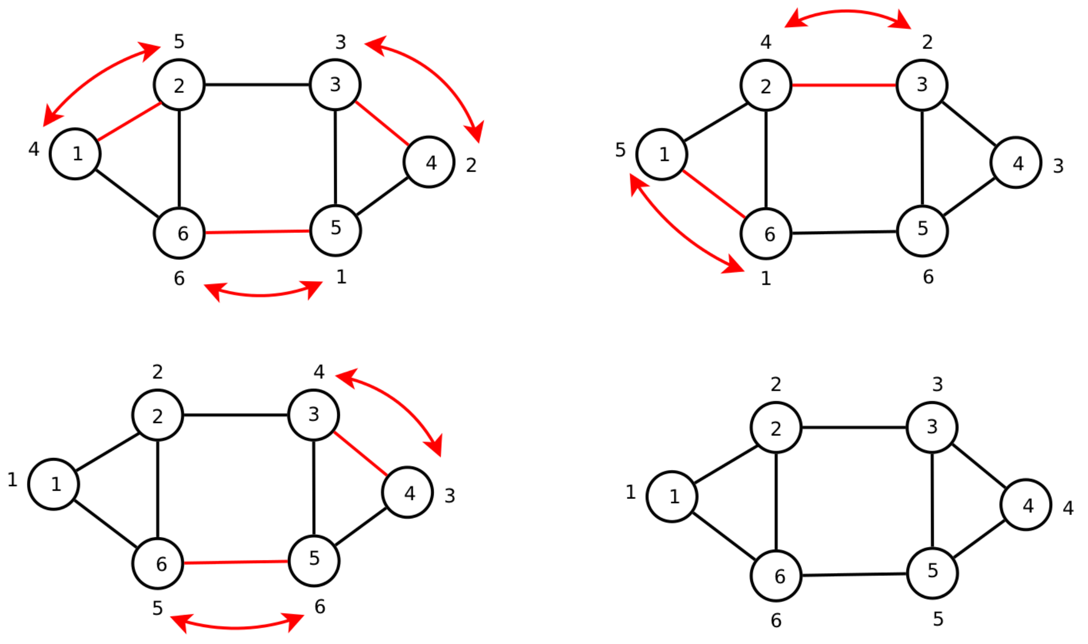


Figure 1.1: A routing taking 3 steps

1.2 Previous Results

As the routing problem occurs in problems in computer science, some of the first bounds shown are consequences of computer science algorithms. The odd-even transposition sort [4] and Benes network [3] show

$$rt(P_n) = n \text{ and } rt(Q_n) \leq n - 1$$

respectively.

Very few results are known for the exact values of the routing numbers of graphs. Alon, Chung, and Graham [2] prove

$$rt(K_n) = 2, \text{ and } rt(K_{n,n}) = 4$$

For a general graph G, they present two lower bounds:

$$rt(G) \geq diam(G), \text{ and}$$

$$rt(G) \geq \frac{2}{|C|} \min\{|A|, |B|\},$$

where *diam* is the diameter of G and C is a set that cuts G into parts A and B. They also present two upper bounds:

$$rt(G) \leq rt(H), \text{ and}$$

$$rt(T_n) < 3n$$

where H is a spanning subgraph of G and T_n is a tree on n vertices. They show that for Cartesian product graphs

$$rt(G_1 \times G_2) \leq 2rt(G_1) + rt(G_2),$$

and for the hypercube graphs,

$$n \leq rt(Q_n) \leq 2n - 1.$$

Zhang [6] improves their bound on trees, showing $rt(T_n) \leq \left\lfloor \frac{3n}{2} \right\rfloor + O(\log n)$.

Li, Lu, and Yang [5] show $n+1 \leq rt(Q_n) \leq 2n-2$, improving both the previous upper and lower bounds on hypercubes. They also present new bounds on cycles:

$$rt(C_n) = n - 1$$

They make a conjecture about the permutations that require this extremal amount of steps to route, which is the focus of this thesis.

1.3 New Results

Li, Lu, and Yang [5] made the following conjecture on the permutations that exhibit the worst case routing on the cycle.

Conjecture 1.3.1. *For $n \geq 5$, if $rt(C_n, \pi) = n - 1$, then π is the rotation $(123 \cdots n)$ or its inverse.*

The rotation permutations mentioned in the conjecture are exactly the (1)-rotation and the (-1)-rotation. (See Figure 1.2.) The conjecture does not hold for $n = 4$; the permutation that transposes two non-adjacent vertices and fixes the other two serves as a counterexample. The conjecture hints towards a very counter-intuitive idea, that the worst case permutation on the cycle is one where each pebble is only a distance of one away from its destination.

Using ideas from Albert et al. [1], we give a proof of the conjecture for when n is even. After selecting a fixed routing algorithm, we count the number of steps any one pebble needs to arrive at its destination. We then show that for a routing of length $n - 1$, no configuration is possible except for the rotation $(123 \cdots n)$. Thus, we have:

Theorem 1.3.2. *For even $n \geq 6$, if $rt(C_n, \pi) = n - 1$, then π is the rotation $(123 \cdots n)$ or its inverse.*

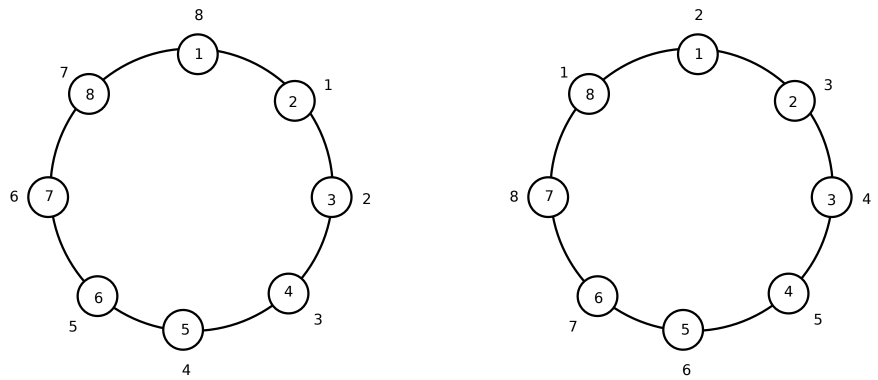


Figure 1.2: The $(123 \cdots 8)$ permutation and its inverse on C_8

Chapter 2

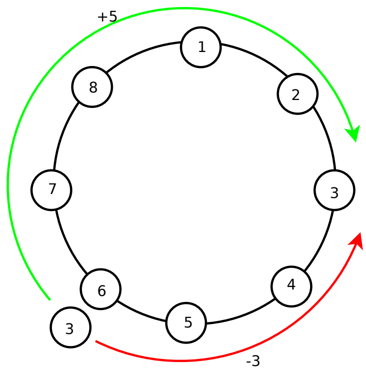
Proof

2.1 Spins and Disbursements

Let $G = C_n$ and label the vertices of C_n as $v_1, v_2, \ldots, v_n$ in a clockwise manner. Let the clockwise and counterclockwise direction be denoted as the positive and negative direction, respectively. There are exactly two paths for p_i to reach its destination vertex, either by traveling in the positive or negative direction. Let $d^+(v_i, v_j)$ denote the distance from v_i to v_j when traveling along the cycle in the positive direction and $d^-(v_i, v_j)$ the distance when traveling in the negative direction. Note that $d^+(v_i, v_j) + d^-(v_i, v_j)$ equals n when $i \neq j$ and 0 when $i = j$.

Consider a (not necessarily optimal) routing of π on C_n. For each pebble p_i, let $s(p_i)$, the **spin** of p_i, represent the displacement experienced by p_i. So, $s(p_i) \in \{d^+(p_i, v_i), -d^-(p_i, v_i)\}$. Let $B = \{s(p_i) | p_i \in P\}$ be a **disbursement** of π. The disbursement describes the direction in which each pebble will move in the

Figure 2.1: The spins of a pebble on C_8



routing. Not all 2^n possible combinations of spins produce valid disbursements. The following lemma describes which disbursements can correspond to routings.

Lemma 2.1.1. *A disbursement B is valid if and only if $\sum_{b \in B} b = |B| = 0$.*

Proof. The total displacement must equal zero, since, for every edge that is swapped, one pebble moves forward one unit and one pebble moves back one unit. So, if $\sum_{b \in B} b \neq 0$, the disbursement cannot possibly be valid. Now, if $\sum_{b \in B} b = 0$, we can first route p_1 to v_1 in the direction described by B. Then, for $2 \leq i \leq n$, we route p_i into position. Before routing p_{i+1}, we correct the routing of $p_1, \ldots, p_{i+1}$, if needed. □

From this lemma we know that there is at least one pebble p_i with positive spin and one pebble p_j with negative spin in a valid disbursement. If we change the spins of p_i and p_j so that they move in the opposite directions, the new disbursement is still valid. We say that we **flip** the spins of p_i and p_j when we apply this change.

Given C_n and a particular valid B, the maximum absolute value of the spins in B is a trivial lower bound to the routing of C_n using B. We also have as a lower bound $\frac{|B|}{2\lfloor \frac{n}{2} \rfloor}$, since the denominator represents the maximum total distance change during one step using a maximum matching. Because of these bounds, we want to minimize $|B|$. From this point forward, we assume B is valid and $|B|$ is minimized, unless otherwise noted.

Lemma 2.1.2. *There exists a routing where for each pair of distinct pebbles, p_i and p_j, they swap with each other at most once.*

Proof. If two pebbles p_i and p_j swap twice in opposite directions, neither swap is necessary. (Note that this does not change $|B|$.) If they swap twice in the same direction, then $|s(p_i)| + |s(p_j)| > n$. Their spins can be flipped to decrease the absolute sum of B. □

Since any two pebbles swap with each other at most once and that swap has direction, there is some sense of ordering to the pebbles.

Definition 2.1.3. *Given a disbursement, B, define the partial ordering '$\prec$' by $p_i \prec p_j$ if p_i and p_j swap with p_i moving in the negative direction and p_j in the positive direction. Equivalently, $p_i \prec p_j \Leftrightarrow s(p_j) - s(p_i) > d^+(p_j, p_i)$*

This partial ordering on the pebbles happens to be transitive.

Proof. Suppose $p_i \prec p_j$ and $p_j \prec p_k$. Then, by definition, we have

$$s(p_j) - s(p_i) > d^+(p_i, p_j), \text{ and}$$
$$s(p_k) - s(p_j) > d^+(p_j, p_k).$$

Now, either $s(p_k) - s(p_i) = d^+(p_i, p_j) + d^+(p_j, p_k) = d^+(p_i, p_k)$ or $n + d^+(p_i, p_k)$. Suppose $s(p_k) - s(p_i) > n$. Then, $s(p_k) > 0$ and $s(p_i) < 0$, since $-n < s(p_a) < n$ for all a. Flipping the spins of p_i and p_j gives a smaller value for $|B|$, a contradiction. Therefore, $s(p_k) - s(p_i) = d^+(p_i, p_k) \Leftrightarrow p_i \prec p_k$. Thus, the partial ordering is transitive. □

2.2 The Odd-Even Routing Algorithm

The results on the routing number of P_n were shown using what is known as the **odd-even transposition sort**. First we describe the odd-even routing algorithm on the path. Label the vertices of P_n as $v_1, v_2, \ldots, v_n$. We say an edge $e = v_i v_{i+1}$ is an odd edge if i is odd; otherwise i is even and e is an even edge. Note that the set of odd edges and even edges partition P_n into two maximal matchings. During the first step and every other odd step of the routing process, we consider only the odd edges. We select a subset of the odd edges and swap the pebbles on the endpoints. During the even steps of the routing process we consider only the even edges and act similarly. During each step, the edges that are selected are those where swapping the pebbles take them closer to their destinations.

We can generalize this algorithm to the class of graphs C_{2n}, the cycles of even length. Label the edges as even and odd like above. Given a particular disbursement B, each vertex is given a particular spin. During odd steps we select odd edges e_i to swap only if the spin of the pebble at vertex v_i is greater than the spin of the pebble at vertex v_{i+1}. During even steps we do the same using only even edges. Because the even edges and odd edges are indistinguishable on C_{2n}, we actually have two algorithms. We can choose to start with the odd edges, like in the path case. We can also choose to start with the even edges, though. We will call this the **even-odd transposition sort**.

Note that this algorithm is not defined on cycles of odd length since the edges that would be labeled as odd edges do not form a matching.

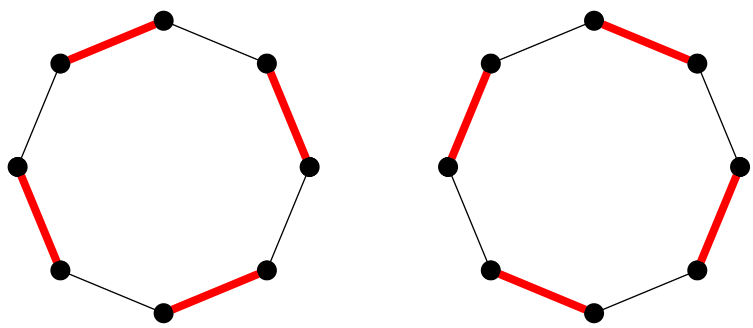


Figure 2.2: The partition of C_8 into two matchings.

2.3 Rotation Permutations

As Conjecture 1.3.1 explicitly mentions two rotations, one might suspect that rotations provide a source of information about the routing number of cycles. In fact, we give the exact value for the routing number of rotations.

Lemma 2.3.1 (Rotation Lemma)**.** *Suppose π is a rotation permutation such that $\pi(a) = a + q(\mod n)$ for some integer q, where $-\frac{n}{2} < q \leq \frac{n}{2}$. Then, $rt(C_n, \pi) = n - |q|$.*

Proof. Without loss of generality, assume $q > 0$ (since the q-rotation and the $(-q)$-rotation are isomorphic). We first show $rt(C_n, \pi) \geq n - q$. For each pebble, p, the spin of p is either $n - q$ or $-q$. Since the sum of spins is zero there must be exactly q pebbles with positive spin and $n - q$ pebbles with negative spin. So, $n - q \leq rt(C_n, \pi)$.

Now, we show $rt(C_n, \pi) \leq n - q$. Since $q \leq \frac{n}{2}$, we flip the spins of pebbles so that q pebbles all on odd edges have positive spin. All other pebbles have negative spin. The partial ordering is then equivalent to

$$p_i \prec p_j \Leftrightarrow s(p_i) < s(p_j).$$

That is, every swap made will occur between pebbles with opposite parity spins. So, a pebble moving in the positive direction will reach its destination by swapping with the $n - q$ pebbles moving in the negative direction and vice-versa. Using the following step we achieve this goal in $n-q$ steps. During each step all pebbles with positive spin will be moved. In no subsequent step will two pebbles with positive spins be adjacent. Therefore, all positive spin pebbles will always move and reach their destinations in $n-q$ steps. Since no swaps occur between two pebbles with negative spins, all such pebbles must also be at their destinations.

Therefore, $rt(C_n, \pi) \leq n - q$. Thus, $rt(C_n, \pi) = n - |q|$. □

2.4 The Window of a Pebble

Given a minimized disbursement and the use of the odd-even routing algorithm, we count the number of steps necessary for each pebble to reach (and stay at) its destination vertex. The maximum of these values is an upper bound on $rt(C_n, \pi)$.

We know that once p_i has swapped positions with all of the pebbles larger and smaller than it, p_i will have reached its destination and will not move anymore. If this is the case for every pebble, then the routing is complete. Consider an arbitrary pebble, A. Let p be the number of pebbles greater than A and q be the number of pebbles less than A in the partial ordering. We label the larger pebbles $v_1, \ldots, v_q$ going in the positive direction starting at A. Similarly, we label the smaller pebbles $u_1, \ldots, u_p$ going in the negative direction starting at A. Since the partial ordering is transitive when the total spin is minimized, we have $d^-(A, u_p) + d^+(A, v_q) < n$. Let $x_1, \ldots, x_s$ denote the pebbles unrelated (by the partial ordering) to A and between A and v_q. Let $y_1, \ldots, y_t$ denote the pebbles unrelated to A and between A and u_p.

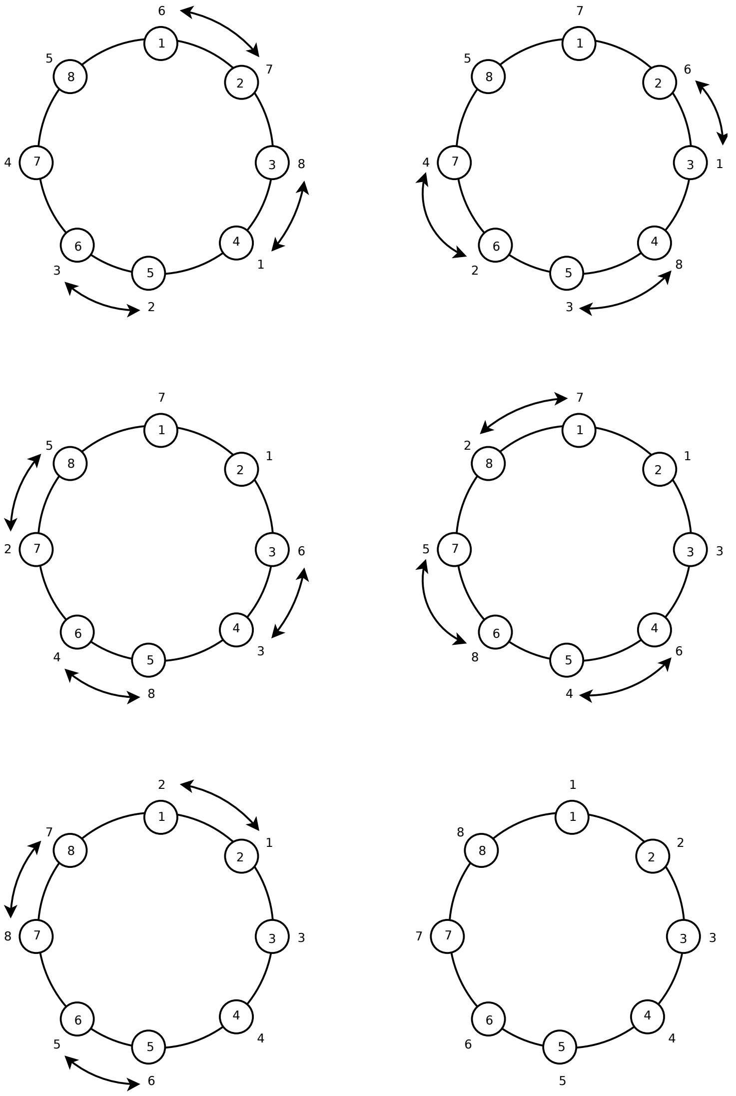


Figure 2.3: The routing of the 3-rotation on C_8

Let U_i, Y_i, X_i, and V_i represent a sequence of u, y, x, and v pebbles, respectively. For example, we have:

$$\underbrace{u_p, u_{p-1}}_{U_k}, Y_k, \ldots \underbrace{y_3}_{Y_2}, \underbrace{u_3, u_2, u_1}_{U_1}, \underbrace{y_2, y_1}_{Y_1}, A, \underbrace{x_1}_{X_1}, \underbrace{v_1, v_2}_{V_1}, \underbrace{x_2, x_3}_{X_2}, \underbrace{v_3, v_4}_{V_2}, \ldots, X_l, \underbrace{v_{q-1}, v_q}_{V_l}$$

The notation allows for two ways to describe one pebble. Here, v_4 refers to the fourth pebble from A that is larger than A. We refer to the pebbles from u_p to v_q as the **window** of A. By transitivity, we know that $x_i \prec u_j$ and $v_i \prec x_j$ and $v_i \prec u_j$ for all i and j.

Now, we wish to count the number of steps required for A to cross all pebbles related to it. During each step either A is being swapped or not. As A swaps with $p+q$ pebbles, A is moving in $p+q$ steps. Now we need to calculate the number of steps in which A does not move.

Lemma 2.4.1. *During the routing, there is at most one y_j between two consecutive u's in U_i.*

Proof. Initially, there are no y's between any consecutive u's in U_i. Suppose there existed a step where two y's were between consecutive u's. Let step s be the first step where this occurs, giving:

$$u_a, y_b, y_{b-1}, u_{a-1}.$$

Since this is the first step with two y's between consecutive a's, the previous step has $\underline{u_a, y_b}$ as an edge to be swapped. But that edge would have swapped, giving $y_b, u_a, y_{b-1}, u_{a-1}$, in step s, a contradiction. □

Corollary 2.4.2. *Suppose A swaps with the first pebble (rightmost) of U_i in step s. Then, in the next $|U_i| - 1$ steps, A will swap with all other u in U_i.*

Proof. If there is no y between u_i and u_{i+1} in step s, then A will swap with u_{i+1} in step $s+1$. Otherwise, there is one y between u_i and u_{i+1}. In this case, u_i and y will swap in step s, allowing A to swap with u_i in step $s+1$. □

So, we consider in what step A swaps with the first pebbles of each sequence U_i and V_i. Let $\{Z_i\} = \{U_i\} \cup \{V_i\}$. Let Z_i be the the ith sequence with which A swaps. From Corollary (2.4.2), we know that any steps where A does not move occurs between swapping with pebbles in Z_i and Z_{i+1}. Let w_i be the number of steps A waits between swapping with the last pebble of Z_{i-1} and the first pebble of Z_i. This gives $w_0 = \min\{|X_1|, |Y_1|\}$.

To calculate w_i, we consider Z_i. Assuming $Z_i \in \{U_i\}$, we have that, in order for Z_i to swap with A, Z_i must wait for all $Z_j(0 < j < i)$ to swap with A, Z_i to swap with $Z_j(Z_j \in \{V_i\}, 0 < j < i)$ and $y_j(0 < j < i)$. But, as $\sum_{j=1}^{i-1} |Z_j| + \sum_{j=0}^{i-1} w_i$ steps have passed, the wait between Z_{i-1} and Z_i is the difference of these terms. If negative, the wait is zero. Therefore, we have:

$$w_i = w_{Z_i} = \max\left\{0, \sum_{j=1}^{i-1} |Y_j| - \left(\sum_{j=0}^{i-1} w_j + \sum_{j=1, Z_j \in \{U_i\}}^{i-1} |Z_j|\right)\right\} \text{ for } Z_i \in \{U_i\}$$

$$w_i = w_{Z_i} = \max\left\{0, \sum_{j=1}^{i-1} |X_j| - \left(\sum_{j=0}^{i-1} w_j + \sum_{j=1, Z_j \in \{V_i\}}^{i-1} |Z_j|\right)\right\} \text{ for } Z_i \in \{V_i\}$$

These waits assume that all edges $\underline{u_i, y_j}$ and $\underline{x_i, v_j}$ are swapped during the first step of the routing process. Because we use the even-odd sorting algorithm, this may not be the case. But, if such an edge is not swapped during the first step, it must be swapped in the second step. So, to each w_i, we add δ_i, where $\delta_i = 0$ if the first pebble of Z_i moves during the first step and $\delta_i = 1$ otherwise.

Now, the total number of steps needed for A to be in position is $p+q+\sum_{j=1}^{k+l} w_j$. Consider the sum of waiting times. Suppose a is the largest index such that w_a is not zero. The summation of wait times then breaks down to

$$\sum_{j=1}^{k+l} w_j = \sum_{j=1}^{i-1} |X_j| - \sum_{j=1, Z_j \in \{V_i\}}^{i-1} |Z_j| + \delta_a.$$

So, the number of steps required by A is

$$\sum_{j=1}^{k} |U_j| + \sum_{j=1}^{l} |V_j| + \left(\sum_{j=1}^{a-1} |X_j| - \sum_{j=1, Z_j \in \{V_a\}}^{a-1} |Z_j| + \delta_a \right).$$

Let O_{pq} be the number of pebbles outside the range of u_p and v_q. Since $\sum_{j=1}^{k} |U_j| + |Y_j| + \sum_{j=1}^{l} |V_j| + |X_j| \leq n-1$, we have:

$$\sum_{j=1}^{k} |U_j| + \sum_{j=1}^{l} |V_j| + \left(\sum_{j=1}^{a} |X_j| - \sum_{j=1, Z_j \in \{V_a\}}^{a-1} |Z_j| + \delta_a \right) \leq n-1, \text{ and}$$

$$(n-1) - \sum_{j=1}^{k} |Y_j| - \sum_{j=1}^{a-1} |Z_j| - \sum_{j=a+1}^{l} |X_j| + \delta_a - O_{pq} \leq n-1.$$

Every permutation that takes $n-1$ steps to route must contain a pebble A

such that

$$\sum_{j=1}^{k} |Y_j| + \sum_{j=1}^{a-1} |Z_j| + \sum_{j=a+1}^{l} |X_j| + O_{pq} = \delta_a \tag{2.1}$$

2.5 The Extremal Windows

In equation (2.1), δ_a equals either zero or one. When $\delta_a = 1$, there are two possible values for O_{pq}, also zero or one. We divide the permutations that satisfy equation (2.1) into three cases.

2.5.1 Case 1: $\delta_a = 0$

When $\delta_a = 0$, each term in equation (2.1) must be zero. This leaves the following window for A:

$$UAXV, \text{ where } X \text{ can be empty.}$$

Since $O_{pq} = 0$, either $p = 0$ or $q = 0$.

Proof. Suppose neither U nor V is empty. We know that $s(u_p) \geq +(1+q), s(v_p) \leq -(n-q)$, and $|s(u_p)|+|s(v_q)| \geq n+1$. Since their signs are different, the spins of u_p and v_q can be flipped. When that happens, we get $s(u_p) \geq -(n-(1+q)), s(v_p) \leq q$, and $|s(u_p)| + |s(v_q)| \leq n - 1$. Flipping the spins of u_p and v_q has decreased $|B|$ by at least 2, a contradiction. Therefore, either U or V is empty. □

This leaves only three possible configurations for the window of A:

$$UA \text{ or } AV \text{ or } AXV$$

The first case is isomorphic to the second case and the second case is a special instance of the third case, when $|X| = 0$. So, we consider only the third case. We show the only possible permutation that satifies this formation is a rotation permutation.

Lemma 2.5.1. *Suppose an arbitrary pebble A of π has a window of AXV. Then π is a rotation.*

Proof. As the spin of A is positive and the spin of v_q is negative, we can flip their spins. Initially, we have $s(A) = +q$, $s(v_q) \leq -(n-q)$, and $|s(A)| + |s(v_q)| \geq q + (n-q) = n$. When the spins are flipped, we get $s(A) = -(n-q)$, $s(v_q) \leq +q$, and $|s(v_q)| + |s(A)| \leq n$. Since $|B|$ is minimized, we must have equality. This gives $s(v_q) = -(n-q)$.

Now, by flipping the spins of v_q and x_1, we get the same result of $s(x_1) = +q$. By induction, we flip the spins of each v_i and x_i to show that $s(v_i) = -(n-q)$ for all i and that $s(x_i) = +q$ for all i. Since the positive spin of each pebble equals q, we have a rotation permutation, $\pi = (q, q+1, \ldots, n, 1, 2, \ldots, q-1)$. By the Rotation Lemma (2.3.1), the only rotations that require $n-1$ steps to route are π_0 and π_0^{-1}. □

2.5.2 Case 2: $\delta_a = 1, O_{pq} = 0$

When $\delta_a = 1$, exactly one term in equation (2.1) is equal to one, while the others are all equal to zero. If $O_{pq} = 0$ then three cases remain. The possible windows of

A are as follows:

- U, y, A or U, y, U, A when $\sum_{j=1}^{k} |Y_j| = 1$
- A, X, V, x, V or A, V, x, V when $\sum_{j=a+1}^{l} |X_j| = 1$
- A, X, v, X, V or A, v, X, V when $\sum_{j=1}^{a-1} |Z_j| = 1$

The first possibility, where the window is U, y, A, means the permutation is a rotation, as shown in Case 1. The five other cases all follow the pattern A, X_1, V_1, X_2, V_2, where X_1 can be empty and either X_2 or V_1 has a length of 1.

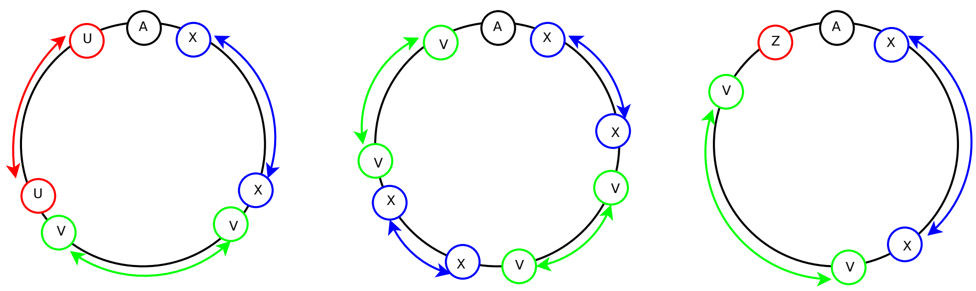


Figure 2.4: The extremal windows corresponding to Cases 1, 2, and 3.

Consider the spins of A and v_q. We have $s(A) = q$ and $s(v_q) \leq -(n-q)$. Since $|B|$ is minimized, we must have $s(v_q) = -(n-q)$. Otherwise, flipping the spins of A and v_q would cause a contradiction. The means the only pebbles that v_q swaps with are A and all x's. Since v_{q-1} does not cross with v_q, we get $s(v_{q-1}) \leq -(n-q)$. For the same reason, we have $s(v_{q-1}) = -(n-q)$. By induction, we have $s(v_i) = -(n-q)$ for all $v_i \in V_2$. Similarly, by comparing the spins of pebbles in X_1 to that of v_q, we have $s(x_i) = q$ for all $x_i \in X_1$. Since $s(A)$ is the same as the pebbles in X_1, let $X_1' = X_1 \cup \{A\}$. Now the cycle has been divided into four segments, X_1', V_1, X_2, and V_2, where no segment is empty, either $|X_2| = 1$ or $|V_1| = 1$, $s(X_1') = q$, and $s(V_2) = -(n-q)$. From this point we assume $|X_2| = 1$. Let x_2 represent this pebble. (The case when $|V_1| = 1$ is symmetric by swapping the labels X_1 and V_2 and swapping the labels V_1 and X_2.)

We know that the sum of all spins equals 0. This gives us

$$\sum_{p_i \in X_1'} s(p_i) + \sum_{p_i \in V_1} s(p_i) + \sum_{p_i \in X_2} s(p_i) + \sum_{p_i \in V_2} s(p_i) = 0$$

We have $(n-q) = |X_1'| + |X_2| = |X_1'| + 1$ and $q = |V_1| + |V_2|$. So, the above becomes

$$|X_1'|(|V_1| + |V_2|) + \sum_{p_i \in V_1} s(p_i) + s(x_2) - |V_2|(|X_1'| + 1) = 0$$

$$|X_1'||V_1| + \sum_{p_i \in V_1} s(p_i) + s(x_2) - |V_2| = 0$$

$$\sum_{p_i \in V_1} (s(p_i) + n - q) = q - s(x_2)$$

Since $\sum_{i=1}^{n} |s(p_i)|$ is minimized, we have the following for all $v_i \in V_1$

$$s(A) - s(v_i) \leq n \text{ implies } s(v_i) + (n-q) \geq 0.$$

The two results above mean that there are at most $q - s(x_2)$ pebbles in V_1 such that $s(v_i) + n - q > 0$. This statement is equivalent to saying there are at least $|V_1| - q + s(x_2)$ pebbles in V_1 such that $s(v_i) + n - q = 0$ implies $s(v_i) = -(n - q)$. We know that x_2 must cross all pebbles in V_2. So, the number of pebbles in V_1 that cross with x_2 is $s(x_2) - |V_2| = (q - t) - (q - |V_1|) = |V_1| - q + s(x_2)$.

Now we consider the ordering of only those pebbles in V_1. We show the structure of the pebbles in V_1. Take the last pebble v_i, in V_1 (next to x_2). We have

$$-(|X_1'| + 1) \leq s(v_i) \leq -(|X_1'|).$$

The lower bound comes from the fact that the total spin is minimized. The upper bound comes from the fact that no pebble is to the positive direction of v_i for it to move forward. If $s(v_i) = -(|X_1'|)$, then v_i cannot cross with any pebble other than those in X_1'. We then look at v_{i-1}, which must have the property that $-(|X_1'| + 1) \leq s(v_{i-1}) \leq -(|X_1'|)$. Otherwise, v_i either crosses with x_2 or crosses with exactly one pebble in V_1. If v_i crosses with x_2, then every pebble in V_1 between v_i and x_2 (which is all of V_1) must also cross with x_2. This means no two pebbles in V_1 can cross each other, since the lower bound $-(|X_1'| + 1) \leq s(v_i)$ holds for all pebbles in V_1. Otherwise, suppose v_i does not cross with x_2 but crosses with some v_k in V_1. Then, every pebble between v_i and v_k must swap with either v_i or v_k. Since no pebble from V_1 other than v_k swaps with v_i, all pebbles in between must swap with v_k. Now, pebble v_{i-1} cannot swap with any other pebble in V_1, otherwise $-(|X_1'|+1) \geq s(v_{i-1})$. In particular, v_{i-1} does not swap with v_{i-2}. By induction, we have that no pebble between v_i and v_k swaps with any pebble in V_1 other than v_k. Then we consider the segment V_1 $\{v_k, \ldots, v_i\}$ and repeat this process for identifying the structure from v_{k-1}, which now holds the property that $-(|X_1'|+1) \leq s(v_i) \leq -(|X_1'|)$.

Now, V_1 has been partitioned into segments of the following type: a "block" where the only swaps that occur are the head pebble swapping with all other pebbles in the block, "isolated" pebbles that do not swap with either x_2 or any pebble in V_1, and an "end block", where all the pebbles swap with x_2 but with no pebble in V_1. Note that there can only be one "end block" and it must contain the first pebbles of V_1, those next to X_1'. Let $V_{1,1}$ denote this "end block".

The segments V_2, X_1' and $V_{1,1}$ are all displaced by the same amount. Therefore, they look similar to a rotation. In fact, we use a similar algorithm to route this permutation that we used to route rotations.

Suppose $q > \frac{n}{2}$. Then, the number of pebbles with positive spin is less than the number with negative spin. We flip the spins of the pebbles in the segments V_2, X_1' and $V_{1,1}$ so that no two pebbles with positive spin are adjacent. Then, our routing goes as follows- while the heads of the "blocks" in V_1 have not reached the end of their block, we treat them like positive spin pebbles, moving them toward the end of their block. Afterwards, we treat these head pebbles like negative spin pebbles. In the first $s(x_2)$ steps, we swap the positive spin pebbles except for x_2 toward the positive direction. In the last $q - s(x_2)$ steps, we swap all pebbles with positive spin toward the positive direction. This process successfully routes this permutation for the same reasons used in the Rotation Lemma: no two negative pebbles cross each other and all the positive pebbles have reached their destinations.

Now, the only rotations that require $n-1$ steps to route are the 1-rotation and the (-1)-rotation. In each of these cases, either the number of pebbles with positive spin is 1 or the number with negative spin is 1. But, $|X_1'| + |X_2| \geq 2$ and $|V_1|+|V_2| \geq 2$. So, this permutation cannot be the 1-rotation nor the (-1)-rotation and, therefore, can be sorted in less than $n-1$ steps.

2.5.3 Case 3: $\delta_a = 1, O_{pq} = 1$

Now, when $\delta_a = 1$ and $O_{pq} = 1$, all other terms in equation (2.1) are zero. We are left with the case where our window looks like:

$$z, U, A, X, V.$$

The spin of u_p is at least $1 + |V|$. The spin of v_q is at least $-(|U| + 1 + |X|)$. Since the total spin is minimized, we must have equality in both cases. As above, by induction, we have that $s(u_i) = 1+|V|$ for all $u_i \in U$ and $s(v_i) = -(|U|+1+|X|)$ for all $v_i \in V$. Now, we try to determine the structure of the spins in segment X. Notice that the pebbles in X either swap with z or do not. This situation is analogous to Case 2 above if you let X be V_1 from above and z be x_2. So, X is made up of "blocks", "isolated" pebbles, and an "end block". Again, we have segments U, V, and the "end block" of X all have the same displacement. Similar to Case 2, we flip the spins of the pebbles in U, V, and the end block of X so that the spins of those pebbles are alternating between positive and negative. Now, in the routing, we first treat the heads of the blocks in X as positive pebbles, treat A as a negative pebble, and move all positive pebbles. As the heads of the blocks of X finish crossing with all the pebbles in the block, we treat the head as a negative pebble. Similarly, once A has moved in the negative direction the appropriate amount, we treat A as a positive pebble. After these changes have occurred, we satisfy the generalized rotation lemma. As there are at least two pebbles with positive spin ($|U| + |X|$) and two pebbles with negative spin ($|V| + |Z|$), this permutation can be sorted in less than $n - 1$ steps.

This, though, assumes that neither U nor V is empty. But, if U is empty, we have

$$A, X, V, z,$$

and if V is empty, X must also be empty and we have

$$z, U, A.$$

In these two cases, recall that the routing of A takes $n-1$ steps becasue $\delta_a = 1$. In the first case, v_1 and the neighboring x need to swap but are delayed because they lie on an even edge. In the second case it is A and u_1 that are connected by an even edge. What we do is switch from the odd-even transposition sort to the even-odd transposition sort. This switch means v_1 and the last x will swap during the first step of the routing (in the first case) and u_1 and A will swap in the first step of the routing (in the second case). Now A has one fewer wait step, arriving at its destination in $n-2$ steps.

What we must check now, though, is that no other pebble needs $n-1$ steps to route because of this change. When the window is zUA and u_1 and A swap during the first step, the only way for the routing to take $n-1$ steps is if z swaps with every u. This configuration means that π must be a rotation, specifically the 2 or -2 rotation. Therefore, this permutation can be routed in at most $n-2$ steps.

When the window structure is $AXVz$, suppose the window structure of z or some v or x is extremal after the change to the even-odd algorithm. This extremal window must fall under Case 3; call it $A'X'V'z'$. (If the extremal pebble is z, by transitivity it must swap with v's only or x's only. Furthermore, z must swap first with either v_1 or the last x since the sum of the spins is minimized.) Because all x's cross with all v's, it must be that the edge between v_1 and the last x is the same edge as the one between v_1' and the last x'. Since $\delta_a = 1$ for both extremal pebbles, this edge cannot be odd and cannot be even, a contradiction. Therefore, no v, x, or z is extremal after the change to the even-odd algorithm. Therefore, all such permutations can be routed in at most $n-2$ steps.

This concludes the proof of Theorem 1.3.2

Chapter 3

Concluding Remarks

3.1 Summary

This research gives insight into what makes a routing on a cycle difficult. We began by solidifying the concept of displacement into spins. Then we use the indirect verification of arriving at a destination by looking at one's neighbors. Finally, we introduce the idea of a window– the set of neighbors that affect and are affected by any single pebble. All of these tools come together to prove a property that is somewhat surprising. The worst case routing seems to be that where each pebble is only a distance of one away from its destination. This property is quite intriguing, as this setup is the easiest to route on the path. We hope these results may assist in determining bounds on graphs with cycles, as trees have already been studied.

3.2 Future Work

Much research in routing numbers has yet to be done. The next step is to tackle the case when n is odd, where standard use of an odd-even transposition sort is not as well-defined. Another reasonable problem to consider is the worst case routings on P_n.

Other types of routings also need to be looked into. Fractional routing, where a pebble can be broken into pieces and swapped across many edges, is at most equal to the routing number, but can bring improvements in some cases. Is the cycle one of those cases? The banded routing number, where the distance to the destination of each pebble is bounded, has been studied. Although, this research shows that the banded routing number of graphs with cycles may not be of interest, since the worst case routing of a cycle has a bandwidth of 1. Other routing models, where more than one pebble is allowed to reside at one vertex, for example, can be studied. And finally, determining whether there exist efficients ways of calculating the routing number of a graph is of interest.

Bibliography

[1] Chase Albert, Chi-Kwong Li, Gilbert Strang, and Gexin Yu. Permutations as product of parallel transpositions. *SIAM J. Discrete Math*, 25(3):1412–1417, 2011.

[2] N. Alon, F. R. K. Chung, and R. L. Graham. Routing permutations on graphs via matchings. *SIAM J. Discrete Math*, 7(3):513–530, August 1994.

[3] V.E. Benes. *Mathematical Theory of Connecting Networks*. Academic Press, New York, 1965.

[4] Donald E. Knuth. *The Art of Computer Programming, Volume 3: (2nd ed.) Sorting and Searching*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1998.

[5] Wei-Tian Li, Linyuan Lu, and Yiting Yang. Routing numbers of cycles, complete bipartite graphs, and hypercubes. *SIAM J. Discrete Math*, 24(4):1482–1494, 2010.

[6] L. Zhang. Optimal bounds for matching routing on trees. *SIAM J. Discrete Math*, 12:64–77, 1999.