AN ALGEBRAIC APPROACH FOR FINDING DISJOINT PATHS IN THE ALTERNATING GROUP GRAPH

JEFFE BOATS, LAZAROS KIKAS, JOHN OLEKSIK

ABSTRACT. For the purpose of large scale computing,we are interested in linking computers into large interconnection networks. In order for these networks to be useful,the underlying graph must possess desirable properties such as a large number of vertices, high connectivity and small diameter. In this paper, we are interested in the alternating group graph, as an interconnection network, and the k-Disjoint Path Problem. In 2005, Cheng, Kikas and Kruk showed that the alternating group graph, AG_n has the $(n-2)$-Disjoint Path Property. However, their proof was an existence proof only. They did not show how to actually construct the $(n-2)$ disjoint paths. In 2006, Boats, Kikas and Oleksik developed an algorithm for constructing the three disjoint paths in the graph AG_5. Their algorithm exploited the hierarchal structure of AG_n to construct the paths. In this paper we develop a purely algebraic algorithm that constructs the $(n-2)$ disjoint paths from scratch. This algebraic approach can be used for other Cayley graphs such as the split-star and the star graphs. Indeed, we believe that our approach can be used for any Cayley graph. We close with remarks on possible research directions stemming from this work.

Keywords: Interconnection networks, graphs, vertex disjoint paths

1. INTRODUCTION

For the purposes of large scale computing,we are interested in connecting together a large number of processors/computers. In order for these networks to be useful,the underlying graph structure should have properties such as vertex symmetry, high connectivity, low diameter and have a large number of vertices. Please see [1, 2].

Recall that connectivity refers to the number of vertices one can delete from a graph without disconnecting it. Deletion of a vertex can be considered as processor failure in a computer network. We want our network to be able to handle a large number of processor failure and still have a functioning network. That is, we want our graph to remain connected. Thus, we want our graph to have high connectivity.

Since these networks will be used for very large computations involving large data sets, it is desirable for our graph structure to have a large number

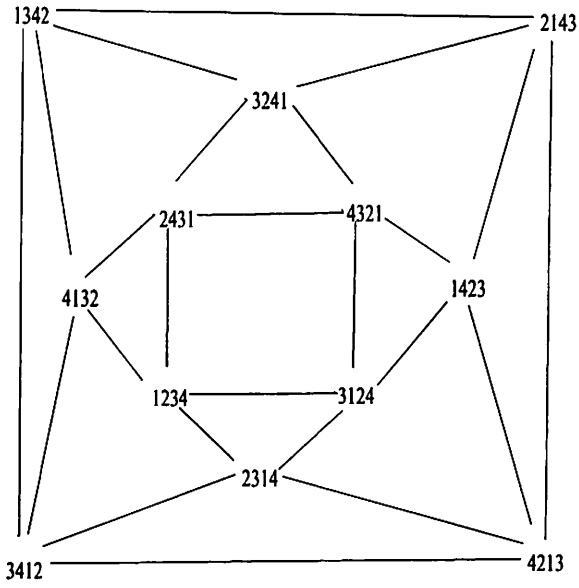

FIGURE 1. AG_4

of vertices. We also want, relative to the number of vertices, for the graph to have low diameter. This reduces communication delay within the network.

The study of interconnection networks and their underlying graph topologies has been the subject of much research. For a very long time the n-cube was the network topology of choice. In 1988, the star graph was introduced as a competitive alternative to the n-cube [1, 2]. In the 1990's the split star and its companion graph was introduced as interconnection network topologies [3, 6].

In this paper we study the alternating group graph AG_n. Let A_n be the alternating group: the group of even permutations on the symbols $1, 2, 3, ..., n$. The vertex set of AG_n is the set A_n. Two vertices of AG_n are adjacent if and only if one can get from one vertex to the other via a 3-rotation. A 3-rotation is accomplished by rotating elements in the first, second, and kth position where $k \in \{3, 4, 5, ..., n\}$. There are two types of 3-rotations: a left rotation and right rotation. As an example consider the permutation $(12345) \in AG_5$. Then both the permutations (51342) and

(25431) are adjacent to (12345). Figure 1 displays the alternating group graph AG_4.

A summary of some of the basic properties of AG_n can be found in [6]. A few worth noting are that the alternating group graph AG_n is a regular graph with degree $2(n-2)$ and it has $\frac{n!(n-2)}{2}$ edges. It is also vertex symmetric and has a hierarchal structure. Consider the subgraph of AG_n induced by the vertices where 1 is fixed in the nth position. It is clear our subgraph will be isomorphic to AG_{n-1}. Since, we can do this for all n symbols it is clear that AG_n is made up of n copies of AG_{n-1}. These properties make the alternating group graph AG_n a novel topology in the design of parallel networks.

Interconnection networks of interest such as the split-star, the star graph and the alternating group graph, belong to a class of graphs called **Cayley graphs**. Suppose that G is a finite group with "multiplication" as its binary operation. A Cayley graph has as its vertex set the group elements of G or subgroup of G. Two vertices $a, b \in G$ are adjacent if and only if there exists a generator $g \in G$ such that $b = ga$. The split-star and the star graph has as its vertex set the symmetric group S_n. Cayley graphs are not just interesting in their own right -they serve well as interconnection networks.

Suppose that we have four computers in an interconnection network and denote them by A, B, C and D. Suppose we want computer A to communicate with B and C with D simultaneously. Communication between two processors is accomplished by sending the message across a path in the network. Suppose the path of communication between A and B share a computer with the path between C and D,then during the simultaneous communication a resource has to be shared. This sharing is called a **signal collision** and can cause communication delay.

If many signal collisions occur,the performance of the network is affected. Hence, signal collision is a major factor in the performance of parallel networks. The question is: Given an interconnection network, how many simultaneous signals can be routed through a particular network topology while avoiding signal collisions?

In graph theoretic terms we want to study the following problem: Given k pairs of distinct nodes $(s_1, t_1), (s_2, t_2), ..., (s_k, t_k)$ do there exist k-disjoint paths, one connection each pair? This problem is called the **k-Disjoint Path Problem**,and has generated much research. If for a graph G we can do this for any selection of k pairs of distinct nodes,then G is said to have the **k-Disjoint Path Property**.

Much has been studied about this problem. It has been shown for $k \geq 3$ the problem of finding k-disjoint paths is NP-hard. Watkins in [8] showed that if a graph G has the k-Disjoint Path Property then it must be $(2k-1)$-connected. Past work done by Cheng and Lipman shows that the split star graph, S_n^2 has the $(n-1)$-Disjoint Path Property [3]. In 2005, Cheng, Kikas,

and Kruk showed that the alternating group graph AG_n has the $(n-2)$-Disjoint Path Property [4, 7]. Both these proofs for the split star and the alternating group are existence proofs; they do not provide an algorithm for the construction of these paths.

In 2006, Boats, Kikas and Oleksik developed an algebraic/geometric algorithm that took advantage of the hierarchal structure of AG_5 to demonstrate the construction of the 3-disjoint paths [5]. In this paper, we provide a purely algebraic algorithm for finding the $(n-2)$-disjoint paths in AG_n. We believe that our approach can be used for any type of Cayley graph where the finite group is some permutation group. We also believe that we can extend our approach for other type of Cayley groups other than permutation groups. These claims remain to be proved.

2. Some Main Ideas for our Approach

There are advantages to routing within a Cayley graph as opposed to other graphs. Principally, path routing is a matter of algebraic factorization. From each vertex, the possible movements to neighboring vertices are determined by the group generators and their inverses. That is, for each generator or inverse, there exists an edge from said vertex.

Distance is measured in terms of the number of factors. For example, if we want to route a path from v_1 to v_2, we can examine the algebraic relationship between these two vertices. Suppose our set of generators for our Cayley graph is $\{g_1, g_2, g_3, ..., g_n\}$. If $v_1 = g_1 g_2 g_3 v_2$ then the path from v_1 to v_2 uses three edges. However, if another factorization gives $v_1 = g_4 g_5 v_2$ then we have found a shorter path. The distance between two vertices is defined as the least number of generators(or their inverses) necessary to create a path.

Our algorithm begins at s_1 and considers the generators and inverses of generators as possible movements. It measures each possible movement in terms of criteria for optimality, and then selects the direction deemed best. Meanwhile, the other movements' evaluations are kept for later use should there be a need for "backtracking".

Backtracking occurs when past options, once overlooked, turn out to have better characteristics than the options available later down the path currently believed best. It also occurs if later paths, say the (s_3, t_3) path, is unroutable due to blockages from previous paths.

A key question is: How does one approach optimality? What makes one movement from a given vertex better than another, and what makes one completed path better than another? A future consideration is whether an approach can be formulated which is applicable to all Cayley graphs. For now, we have an approach which works(with very minor adjustments) for any Cayley graph whose vertex set is a permutation group or subgroup of such.

In this paper, we address our approach in terms of the alternating group graph AG_5. Let H_i be the substar of AG_5 induced by the elements of A_5 with i fixed in the 5th place. If we consider the elements of A_5 then the elements of H_5 forms a subgroup of A_5 with cosets formed by H_1, H_2, H_3 and H_4. Clearly, each H_i is isomorphic to AG_4 for $i = 1, 2, 3, 4, 5$. It is an elementary exercise in algebra to show that the group A_n is generated by the three cycles $(12k)$ and $(1k2)$ for $k \in \{3, 4, 5, .., n\}$. A path in AG_n from vertex α to vertex β is accomplished by performing a sequence of 3-rotations. In algebraic terms, a path from α to β can be found if one can express $\beta = g_1 g_2 ... g_s \alpha$ where each g_i for $i = 1, 2, ..., s$ is either a $(12k)$ or a $(1k2)$ for $k = 3, 4, 5, ..., n$.

As an example suppose that $\alpha = 1234$ and $\beta = 2143$. Consider the path $\alpha, 2431, 3241, \beta$. This is equivalent to the following factorization of β. That is, $\beta = \alpha(124)(132)(124)$. Thus finding a path in AG_n is equivalent to factoring elements of A_n over its generators. Finding alternate factorization allows us to route around blocks. The idea of factoring over generators is what lead to the approach found in [5]

3. The Algorithm

To find a path in AG_n from α to β we saw that it is equivalent to factoring the element β over the generators $(12k)$ and $(1k2)$ for $k = 3, 4, 5, ..., n$. For the sake of simplicity let us restrict our discussion to the graph AG_5. The description we give can be generalized to AG_n.

We denote $a = (123)$, $a^2 = (132)$, $b = (124)$, $b^2 = (142)$, $c = (125)$ and $c^2 = (152)$. Note that if $\alpha \in H_i$ for some $i \in \{1, 2, 3, 4, 5\}$ then αa, αa^2, αb and αb^2 are all in H_i. But αc and αc^2 are elements of H_k where $k \neq i$.

Let α be any permutation from A_5. The first two positions are called the **Vestibule** and the last three positions are called the **Locked** positions. Say we wish to route from α to β in AG_5. We need to apply a sequence of 3-rotations to transform α to β. As soon as the elements that need to be in the lock position are locked, the two positions in the vestibule are automatically satisfied, since α is an even permutation.

Suppose we wish for $\alpha = 14235$ to be routed to $\beta = 34125$. We call β the **target** permutation. Let L be the number of elements locked. In the target permutation 1 needs to be in the third place, 2 in the fourth place, and 5 in the fifth place. In α, 5 is in the correct position. Hence, $L = 1$. We let V be the number of elements in the vestibule waiting to be locked. The elements 1 and 2 need to be locked. In α the element 1 in the vestibule and 2 is out of the vestibule but not in the correct position. So $V = 1$. Suppose that we are routing from s_1 to t_1. We define the stepcount parameter s for some vertex v to be the number of steps that it takes to route from s_1 to v. Obviously, the vertex s_1 has stepcount value $s = 0$. We also compute the stepcount for when routing from s_2 to t_2 and from s_3 to t_3.

When we apply the operators a, a^2, b, b^2, c and c^2 we move elements in and out of the vestibule. For each move we make we compute and store the parameters V and L and s.

At the outset we start with the pairs (s_1, t_1), (s_2, t_2) and (s_3, t_3). We first route s_1 to t_1 then s_2 to t_2 and finally s_3 to t_3. When routing from s_1 to t_1 we note the blocks s_2, t_2, s_3 and t_3. When routing from s_2 to t_2 we note the path from s_1 to t_1 that was computed and the vertices s_3 and t_3. When routing from s_3 to t_3 the elements of the paths from s_1 to t_1 and from s_2 to t_2 are our blocks. All these blocks are stored.

For each move we make we desire to increase L. However, this is not always possible due to blocks. If we are forced to backtrack in our search the we look for the next best L and then the next best V.

Formally, we define the following parameters for the algorithm.

Definition 3.1. *Let v be a vertex and $i \in \{1, 2, 3\}$ is fixed. The parameter* **Stepcount** *s computes the number of edges in a route from s_i to the vertex v.*

Definition 3.2. *A* **BLOCK** *is a vertex that has been used in a previously computed route or is one of the vertices $\{s_1, t_1, s_2, t_2, s_3, t_3\}$*

Definition 3.3. *A vertex is classified as an* **OVERSTEP** *if there exists a shorter path from the source to that vertex.*

Definition 3.4. *A vertex is said to be* **ACTIVE** *if it is potentially a vertex that can be used as part of some route.*

The algorithm may be described,formally, with the following steps.

ALGEBRAIC ALGORITHM

(1) $i = 0$
(2) Input $(s_1, t_1), (s_2, t_2), (s_3, t_3)$.
(3) $i = i + 1$. Stop when $i = 4$
(4) Initialize routing for (s_i, t_i) for current value of i
(5) Apply the operators a, a^2, b, b^2, c and c^2. Compute L, V, s.
(6) Characterize each result as either as a BLOCK, OVERSTEP or as ACTIVE.
(7) Choose best ACTIVE point. If none exist goto step 9
(8) If done goto step 3 and store computed route. Else goto 5.
(9) Backtrack to the last step with remaining ACTIVE point available and goto step 5.

In [4,7] the following theorem was proved.

Theorem 3.1. *Let AG_n be the alternating group graph and let $n \geq 5$. Then AG_n has the $(n-2)$-Disjoint Path Property.*

Our algorithm is a semi-brute force algorithm. It searches for paths in an organized manner so that not ever possible path must be computed. Given $(s_1, t_1), (s_2, t_2), (s_3, t_3)$ our algorithm is guaranteed to find the three disjoint paths because Theorem 3.1 guarantees that these three disjoint paths exist.

4. A Worked out Example

We present in this section a worked out example illustrating the algorithm as outlined in the previous section. In our example let $s_1 = 14235$, $t_1 = 34125$,$s_2 = 21435$, $t_2 = 23145$, $s_3 = 42135$ and $t_3 = 12345$. We first route from s_1 to t_1, and then from s_2 to t_2 and then from s_3 to t_3. As we try to route we keep track of blocks and the parameters V, L, s. For s_1 we have $V = 1$,$L = 1$ and $s = 0$. From each ACTIVE node we apply the generators a, a^2, b, b^2 and c^2. When we do this at s_1 we get the following:

(1) 21435 BLOCKED.
(2) 42135 BLOCKED
(3) 31245 $L = 1$, $V = 1$, $s = 1$, ACTIVE
(4) 43215 $L = 1$, $V = 0$, $s = 1$, ACTIVE
(5) 51234 $L = 0$, $V = 2$, $s = 1$, ACTIVE
(6) 45231 $L = 0$, $V = 2$, $s = 1$, ACTIVE

Note that when we applied the operator b to s_1 we got the vertex 31245. This node has $L = 1$ and $V = 1$. Looking at the above table this vertex has the best values of L and V. So we continue our route from 31245 we get. We apply the operators again and we get

(1) 23145 BLOCKED.
(2) 12345 BLOCKED.
(3) 43215 OVERSTEP
(4) 14235 OVERSTEP
(5) 53241 $L = 0$, $V = 1$, $s = 2$ ACTIVE
(6) 15243 $L = 0$, $V = 2$, $s = 2$ ACTIVE

At this point we route from 15243 since it has a higher V value than 53241. We do this and we obtain:

(1) 21543 $L = 0$,$V = 2$,$s = 3$ ACTIVE
(2) 52143 $L = 1$,$V = 2$,$s = 3$, ACTIVE
(3) 41253 $L = 0$,$V = 1$,$s = 3$, ACTIVE
(4) 54213 $L = 0$,$V = 1$,$s = 3$, ACTIVE
(5) 31245 OVERSTEP
(6) 53241 OVERTSTEP

The vertex with the highest L value is 52143. Hence, we continue our route from there. We do this and we get:

(1) 15243 OVERSTEP
(2) 21543 OVERSTEP

(3) 45123 $L = 2$,$V = 1$,$s = 4$ ACTIVE
(4) 24153 $L = 1$, $V = 1$, $s = 4$ ACTIVE
(5) 35142 $L = 1$, $V = 1$, $s = 4$ ACTIVE
(6) 23145 $L = 2$, $V = 1$, $s = 4$, ACTIVE

Note, that at this point both vertices 45123 and 23145 have the value $L = 2$. This means that both these vertices are one step away from t_1. Choose one, say 45123 and by applying the operator c we complete the route. Our route from s_1 to t_1 is,therefore, $s_1 \Rightarrow 31245 \Rightarrow 15243 \Rightarrow 52143 \Rightarrow 45123 \Rightarrow t_1$. This route is now stored and its vertices are now considered potential blocks when routing the other pairs of vertices.

For route 2 we have $s_2 = 21435$ and $t_2 = 23145$. For s_2 we have that $L = 1$,$V = 1$ and $s = 0$. Routing from s_2 gives us:

(1) 42135 BLOCKED
(2) 14235 BLOCKED
(3) 32415 $L = 1$, $V = 0$, $s = 1$ ACTIVE
(4) 13425 $L = 1$, $V = 1$, $s = 1$ ACTIVE
(5) 52431 $L = 0$, $V = 1$, $s = 1$ ACTIVE
(6) 15432 $L = 0$, $V = 2$, $s = 1$ ACTIVE

The vertex with the best value is 13425. Routing from there we get:

(1) 41325 $L = 1$, $V = 2$, $s = 2$ ACTIVE
(2) 34125 BLOCK.
(3) 21435 OVERSTEP
(4) 32415 OVERSTEP
(5) 51423 $L = 0$, $V = 2$, $s = 2$ ACTIVE
(6) 35421 $L = 0$, $V = 1$, $s = 2$ ACTIVE

The node 41325 has the best L value. We continue our search from there. We get:

(1) 34125 BLOCK
(2) 13425 OVERSTEP
(3) 24315 $L = 1$, $V = 1$, $s = 3$ ACTIVE
(4) 12345 BLOCK
(5) 54321 $L = 0$, $V = 2$, $s = 3$ ACTIVE
(6) 15324 $L = 0$, $V = 2$, $s = 3$ ACTIVE

The node 24315 has the best L value. So routing from there gives us:

(1) 32415 OVERSTEP
(2) 43215 $L = 1$, $V = 1$, $s = 4$ ACTIVE
(3) 12345 BLOCK
(4) 41325 OVERSTEP
(5) 52314 $L = 0$, $V = 1$, $s = 4$ ACTIVE
(6) 45312 $L = 0$, $V = 2$, $s = 4$ ACTIVE

The node 43215 has the best L value. Routing from there we obtain:

(1) 24315 OVERSTEP
(2) 32415 OVERSTEP
(3) 14235 BLOCK
(4) 31245 BLOCK
(5) 54213 $L = 0$, $V = 2$, $s = 5$
(6) 35214 $L = 0$, $V = 1$, $s = 5$

Note that 54213 and 35214 tie with $L = 0$. But 54213 has a better V value. Hence, we route from there. Doing this we obtain:

(1) 25413 $L = 0$, $V = 1$, $s = 6$ ACTIVE
(2) 42513 $L = 0$, $V = 1$, $s = 6$ ACTIVE
(3) 15243 BLOCK
(4) 41253 $L = 0$, $V = 2$, $s = 6$ ACTIVE
(5) 35214 OVERSTEP
(6) 43215 OVERSTEP

The nodes 25413, 42513 and 41253 have the same L values. But 41253 has the best V value. So we continue our route from there:

(1) 24153 $L = 1$, $V = 1$, $s = 7$ ACTIVE
(2) 12453 $L = 0$, $V = 1$, $s = 7$ ACTIVE
(3) 54213 OVERSTEP
(4) 15243 BLOCK
(5) 34251 $L = 0$, $V = 1$, $s = 7$ ACTIVE
(6) 13254 $L = 0$, $V = 1$, $s = 7$ ACTIVE

The node 24153 has the highest L value. Hence, we route from there. This gives us:

(1) 12453 $L = 0$, $V = 1$, $s = 8$ ACTIVE
(2) 41253 OVERSTEP
(3) 52143 BLOCK
(4) 45123 BLOCK
(5) 32154 $L = 1$, $V = 0$, $s = 8$ ACTIVE
(6) 43152 $L = 1$, $V = 1$, $s = 8$ ACTIVE

The nodes 32154 and 43152 have the same L value. But 43152 has the better V value. Thus, we continue our route from there. Doing this we get:

(1) 14352 $L = 0$, $V = 2$, $s = 9$ ACTIVE
(2) 31452 $L = 0$, $V = 1$, $s = 9$ ACTIVE
(3) 54132 $L = 1$, $V = 2$, $s = 9$ ACTIVE
(4) 35142 $L = 2$, $V = 1$, $s = 9$ ACTIVE
(5) 24153 OVERSTEP
(6) 32154 OVERSTEP

Note the node 35142 has $L = 2$. We are, thus, one step away from the target. We complete the route to t_2 by applying the operator c. Our

route for s_2 to t_2 is therefore: $s_2 \Rightarrow 13425 \Rightarrow 41325 \Rightarrow 24315 \Rightarrow 43215 \Rightarrow 54213 \Rightarrow 41253 \Rightarrow 24153 \Rightarrow 43153 \Rightarrow t_2$.

Again this route along with the route from s_1 to t_1 are stored as potential blocks when we route from s_3 to t_3. We have that $s_3 = 42135$ and $t_3 = 12345$. Hence, for s_3 we have that $L = 1$, $V = 1$ and $s = 0$. We start our route from s_3. Doing this we get

(1) 14235 BLOCK
(2) 21435 BLOCK
(3) 34125 BLOCK
(4) 23145 BLOCK
(5) 54132 $L = 0$, $V = 2$, $s = 1$ ACTIVE
(6) 25134 $L = 0$, $V = 1$, $s = 1$ ACTIVE

The nodes 54132 and 25134 have the same L values. But 54132 has the better V value. So we route from there. Doing this we obtain:

(1) 15432 $L = 0$, $V = 1$, $s = 2$ ACTIVE
(2) 41532 $L = 0$, $V = 1$, $s = 2$ ACTIVE
(3) 35142 BLOCK
(4) 43152 BLOCK
(5) 25134 OVERSTEP
(6) 42135 OVERSTEP

Here we have that 15432 and 41532 are tied in terms of their L and V values. So, at this point we choose the first node, 15432. Routing from there we get:

(1) 41532 OVERSTEP
(2) 54132 OVERSTEP
(3) 31452 $L = 0$, $V = 1$, $s = 3$ ACTIVE
(4) 53412 $L = 0$, $V = 2$, $s = 3$ ACTIVE
(5) 21435 BLOCK
(6) 52431 $L = 0$, $V = 1$, $s = 3$ ACTIVE

The nodes 31452, 53412 and 52431 all have the same L values. But. 53412 has the best V value. So we route from that vertex. Doing this we get:

(1) 45312 $L = 1$, $V = 2$, $s = 4$ ACTIVE
(2) 34512 $L = 0$, $V = 2$, $s = 4$ ACTIVE
(3) 15432 OVERSTEP
(4) 31452 OVERSTEP
(5) 25413 $L = 0$, $V = 1$, $s = 4$ ACTIVE
(6) 32415 $L = 1$, $V = 1$, $s = 4$ ACTIVE

The nodes 45312 and 32415 have the same L value. But, 45312 has the better V value. So, we route from there. We get:

(1) 34512 OVERSTEP

(2) 53412 OVERSTEP
(3) 14352 $L = 1$, $V = 1$, $s = 5$ ACTIVE
(4) 51342 $L = 2$, $V = 1$, $s = 5$ ACTIVE
(5) 24315 BLOCK
(6) 52314 $L = 1$, $V = 1$, $s = 5$ ACTIVE

At this point we have the vertex 51342 with $L = 2$. Hence, we are now one step away from t_3. Applying the operator c^2 completes the route. Our route from s_3 to t_3 is therefore $s_3 \Rightarrow 54132 \Rightarrow 15432 \Rightarrow 53412 \Rightarrow 45312 \Rightarrow 51342 \Rightarrow t_3$. We, thus have our three disjoint paths.

5. Conclusions

The purpose of this paper was to give an algebraic algorithm that constructs the $(n-2)$- Disjoint Paths for AG_n. We claim our approach works for other Cayley graphs. This remains to be shown.

References

[1] S. B. Akers, D. Harel, and B. Kirshnamurthy. The star graph: An attractive alternative to the n-cube. *Proceedings of the International Conference on Parallel Processing*, pages 393–400, 1987.

[2] S. B. Akers and B. Kirshnamurthy. A group theoretic model for symmetric interconnection networks. *IEEE Transactions on Computers*, 38(4):555–566, 1989.

[3] E. Cheng and M. Lipman. Disjoint paths in split-stars. *Congressus Numerantium*, 137:47–63, 1999.

[4] E.Cheng, L.D.Kikas, and S.Kruk. A disjoint path problem in the alternating group graph. *Congressus Numerantium*, 175:117–159, 2005.

[5] J.Boats, L.D.Kikas, and J.Oleksik. Algorithm for finding disjoint paths in the alternating group graph. *Congressus Numerantium*, 181:97–109, 2006.

[6] J. S. Jwo, S. Lakshimivarahan, and S. K. Dhall. A new class of interconnection network based on the alternating group. *Networks*, 23:315–325, 1993.

[7] Lazaros D. Kikas. *Interconnection Networks and the k-Disjoint Path Problem.* PhD thesis, Oakland University, 2004.

[8] M. E. Watkins. On the existence of certain disjoint arcs in graphs. *Duke Mathematics Journal*, 35:231–246, 1968.

Jeffe Boats, Department of Mathematics and Computer Science, University of Detroit Mercy, Detroit, MI, 48221, USA
E-mail address: boatsjj@udmercy.edu

Lazaros Kikas, Corresponding Author, Department of Mathematics and Computer Science, University of Detroit Mercy, Detroit, MI, 48221, USA
E-mail address: kikasld@udmercy.edu

John Oleksik, Department of Mathematics and Computer Science, University of Detroit Mercy, Detroit, MI, 48221, USA
E-mail address: oleksijj@udmercy.edu