Applying Latin Square on Parallel Routing of Alternating Group Graphs

Jinn-Shyong Yang[1,2] , Ting-Yem Ho[1,2] , Yue-Li Wang[1,*], Ming-Tat Ko[3]

[1] Department of Information Management, National Taiwan University of Science and Technology, Taipei, Taiwan, ROC

[2] Department of Information Management, National Taipei College of Business, Taipei, Taiwan, ROC

[3] Institute of Information Science Academia Sinica Taipei, Taiwan, ROC

Abstract

A major topic of studying communication networks is to find an appropriate route for message transmission. In particular, the design of parallel routing can be used to transmit multiple packets efficiently from a source node to a destination node simultaneously. In this paper, we consider the problem of constructing parallel routes in an alternating group graph, which has recently been developed as a new model of the interconnection topology for parallel and distributed computing systems. An n-alternating group graph contains $(n!)/2$ nodes and is a regular graph with degree $2(n-2)$ for each node. The aim of our work is to provide an algorithm for constructing $2(n-2)$ edge-disjoint paths for any pair of nodes in an n-alternating group graph for a special case. Furthermore, we show that the lengths of all paths are shortest.

Keywords: Interconnection networks, Alternating group graphs, Parallel routing, Latin square

*Corresponding author: ylwang@cs.ntust.edu.tw.

1. Introduction

In this paper, we consider the problem of constructing parallel routes in an alternating group graph, which has recently been developed as a new model of the interconnection topology for parallel and distributed computing systems [3, 4, 7, 9]. The alternating group graphs, like the well-known star graphs and the n-cube, belong to a class of graphs called the *Cayley graphs*, a family of graphs that possesses group theoretic properties [1]. Furthermore, It has been shown in [4] that a class of generalized star graphs called the *arrangement graphs* also contains alternating group graphs as members. Indeed, a proof given in that paper showed that the n-alternating group graph AG_n is isomorphic to the $(n, n-2)$-arrangement graph $A_{n,n-2}$.

The arrangement graph has been shown to be vertex and edge symmetric, strongly hierarchical, maximally fault tolerant, and strongly resilient [5], and thus of the alternating group graph. A comprehensive analysis of symme-

try in a wide variety of the Cayley graphs of permutation groups can also be found in the survey paper [9]. Jwo et al. [7] proposed a simple shortest-path routing algorithm for alternating group graphs. According to their results, there exist efficient schemes to embed two-dimensional grids and arbitrary cycles on alternating group graphs. Moreover, they also provided an algorithm for broadcasting messages by using a spanning tree. Recently, Cheng and Lipman [3] proposed an assignment of directions to the edges of the alternating group graphs and showed that the resulting directed graphs are not only strongly connected, but, in fact, they have maximal arc-fault tolerance and a small diameter.

In this paper, we will present an algorithm for constructing $2(n-2)$ edge-disjoint paths between every pair of vertices in an n-alternating group graph under a special case. Thus, the resulting parallel paths can be used to transmit multiple packets efficiently from a source node to a destination node simultaneously. Moreover, we show that the lengths of all the routing paths are shortest.

2. Preliminaries

Let $p = p_1 p_2 \cdots p_n$ be a permutation of $\{1, 2, \ldots, n\}$, where p_i is the element at position i for $1 \leq i \leq n$. Also, let p_i^{-1} be the position where the element i can be found. If $p_i = i$, then the element i is said to be *at the right place*; otherwise i is *out of place*. A convenient way to denote a permutation p is by using its cycle representation $C(p) = c_1 c_2 \cdots c_k e_1 e_2 \cdots e_l$, where c_i is a cycle of length $|c_i| \geq 2$ for $1 \leq i \leq k$ and e_i is an *invariant* (i.e., an element which has been at the right place) for $1 \leq i \leq l$. For simplicity, all invariants are suppressed in the representation. As an example, if $p = 16543827$, then $C(p) = (2687)(53)$. Note that the set of all permutations of n elements is called the symmetric group of degree n and is denoted by S_n. A pair (i, j) in a permutation is called an *inversion* if $i < j$ and $p_i > p_j$. A permutation is said to be *even* (respectively *odd*) if its parity (i.e., the number of inversions) is even (respectively odd). The *alternating group* A_n is a subgroup of S_n consisting of all even permutations. For fundamental definitions of group theory, please refer to [2, 10].

A construction of graphs built on alternating groups was originally introduced by Jwo et al. [7]. Let $\mathrm{g}_i^+ = (12i) = \mathrm{T}_{12} \cdot \mathrm{T}_{2i}$ and $\mathrm{g}_i^- = (1i2) = \mathrm{T}_{2i} \cdot \mathrm{T}_{12}$ for $3 \leq i \leq n$, where T_{ij} denotes that the elements in positions i and j are exchanged and $T_{ij} \cdot T_{st}$ means exchanging positions i and j then exchanging positions s and t. It can be shown that $\Omega = \{\mathrm{g}_i^+ \mid 3 \leq i \leq n\} \cup \{\mathrm{g}_i^- \mid 3 \leq i \leq n\}$ is a generator set for A_n [10]. Define an *n-alternating group graph* $AG_n = (V_n, E_n)$ as follows: the vertex set V_n is the set of all even permutations (i.e., $V_n = A_n$) and the edge set $E_n = \{(p, q) \mid p, q \in V_n \text{ and } q = p \cdot \mathrm{g} \text{ for some } \mathrm{g} \in \Omega\}$, where $p \cdot \mathrm{g}$ means applying operation g on p and we neglect "$\cdot$" when it will not make. Note that $q = p \cdot \mathrm{g}_i^+$ if and only if $p = q \cdot \mathrm{g}_i^-$. For example, $31546827 = 16543827 \cdot \mathrm{g}_5^-$ and $63541827 = 16543827 \cdot \mathrm{g}_5^+$. Also, it is easy to check that AG_n is a regular graph with degree $2(n-2)$ and $|V_n| = (n!)/2$. Figure 1 depicts an example of AG_n for $n = 4$.

Suppose P is a path of length h with the starting vertex p and the terminal vertex q in AG_n, then $q = p \cdot \mathrm{g}_1 \cdot \mathrm{g}_2 \cdots \mathrm{g}_h$ for some $\mathrm{g}_1, \mathrm{g}_2, \ldots, \mathrm{g}_h \in \Omega$. Let $r_i = p \cdot \mathrm{g}_1 \cdot \mathrm{g}_2 \cdots \mathrm{g}_i$,

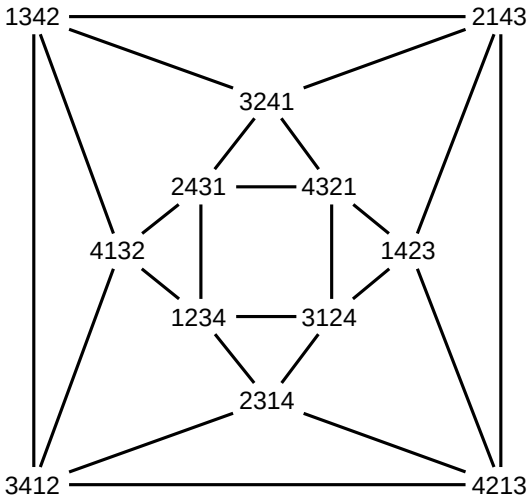

Figure 1: A 4-alternating group graph AG_4.

$i = 1, \ldots, h-1$, be the internal vertices of P. We shall use $[p]\ \mathrm{g}_1[r_1]\ \mathrm{g}_2[r_2] \cdots \mathrm{g}_{h-1}[r_{h-1}]\ \mathrm{g}_h[q]$ (or simply write $[p]\ \mathrm{g}_1\ \mathrm{g}_2 \cdots \mathrm{g}_{h-1}\ \mathrm{g}_h[q]$) to represent the path P. In particular, if there are no internal vertices $r_i = r_j$ with $i \neq j$, then P is called a *simple path*. Note that every shortest path must be a simple path. In what follows, we only consider the simple path produced by the routing algorithm.

Due to the vertex symmetry of alternating group graphs, the problem of routing between two arbitrary vertices in AG_n can be reduced to the problem of routing from an arbitrary vertex p to the identity vertex $I_n = 12 \cdots n$. We now describe the shortest-path routing algorithm proposed in [7]. Suppose $p \neq I_n$ and let $r = r_1 r_2 \cdots r_n$ be a vertex in the routing path from p to I_n. Then, the immediate successor r' of r in the path can be determined as follows.

Algorithm SPR-1 (Shortest-Path Routing)

Input: An alternating group graph AG_n and a vertex p.

Output: A shortest routing path from p to I_n.

1. Initially, $r = p$.
2. Repeat the following steps until the identity vertex is reached (i.e., $r = I_n$).
3. If $r_1, r_2 \in \{1, 2\}$ then
   select any element $i \notin \{1, 2\}$ that is out of place and let $r' = r \cdot \mathrm{g}_i^*$ where $* \in \{+, -\}$
   Let $r = r'$.
4. If $r_1 \notin \{1, 2\}$ then
   let $i = r_1$ and $r' = r \cdot \mathrm{g}_i^+$
   else if $r_2 \notin \{1, 2\}$ then
   let $i = r_2$ and $r' = r \cdot \mathrm{g}_i^-$
   Let $r = r'$.

Since every vertex p in AG_n corresponds to an even permutation, the elements 1 and 2 will automatically be at the right place in the algorithm when all the other elements are at the right place. Furthermore, it can be shown that the shortest-path routing algorithm corresponds to an optimal sorting by using the operations in Ω of the permutation p. Consequently, Algorithm SPR-1 can produce a shortest path from an arbitrary vertex to the identity vertex in AG_n. In addition, the routing distance can be computed as follows.

Lemma 1 [4, 7] *Suppose that vertex p has the representation $C(p) = c_1 c_2 \cdots c_k e_1 e_2 \cdots e_l$. Then the distance D_p from p to I_n in AG_n is:*

$$D_p = \begin{cases} n+k-l, & \text{if } p_1 = 1, p_2 = 2 \\ n+k-l-2, & \text{if } (p_1 \neq 1, p_2 = 2) \text{or } (p_1 = 1, p_2 \neq 2) \\ n+k-l-3, & \text{if } 1, 2 \in c_i \text{ for } 1 \leq i \leq k \\ n+k-l-4, & \text{if } 1 \in c_i, 2 \in c_j \text{ for } 1 \leq i \neq j \leq k \end{cases}$$

For example, consider a vertex p with the cycle representation (23)(54)(876). The following is a shortest path P produced by the routing algorithm and the distance $D_p = 8+3-1-2 = 8$.

P : [13254867] g_3^- [21354867] g_4^+ [15324867] g_5^-
[41325867] g_4^+ [12345867] g_6^+ [28345167] g_8^-
[72345168] g_7^+ [26345178] g_6^- [12345678].

Let P be a routing path from vertex $p(\neq I_n)$ to the identity. We say that P is *predominant* if every non-invariant of p, except 1 and 2, will never be out of place again throughout the path after it has been at the right place. It is clear that every shortest path from p to I_n is predominant. A sequence consisting of all non-invariants of p, except 1 and 2, is called a *conveying sequence* if a predominant path can be generated according to the sequence. For example, in the preceding instance, predominant path generated by $3, 5, 4, 8, 7, 6$ is a conveying sequence of vertex $p = 13254867$.

3. The parallel routing algorithm

In this section, we present a parallel routing algorithm from an arbitrary vertex $p = p_1p_2\cdots p_n$ to the identity vertex in AG_n where $p_1p_2 = 12$ or $p_1p_2 = 21$ and no $p_i = i$, $3 \leq i \leq n$. The construction scheme is based on the characterization of Latin square which is different from the one in the previous section. first of all, we define a sequence from a cycle representation, which is helpful for describing our shortest-path routing algorithm.

Assume $C(p) = c_1c_2\cdots c_k$. If element 1 is a non-invariant, without loss of generality we may assume that it is the first element in the representation of a cycle. Moreover, if element 2 is also a non-invariant and 1,2 are contained in distinct cycles, we assume that element 2 is the first element in the representation of the cycle that contains it. Otherwise, we choose arbitrarily an element of a cycle as the first element. With respect to a permutation $p(\neq I_n)$, the *elementary cycle sequence* of p, denoted by $ecs(p)$, is a sequence consisting of all non-invariants, except 1 and 2, such that for every pair of elements s and t, s precedes t in the sequence if $s \in c_i$, $t \in c_j$ where $1 \leq i < j \leq k$, or $s, t \in c_i$ and s comes before t in the representation of the cycle. From the definition, it is easy to see that all elements of a cycle appear consecutively in the sequence. For example, if $C(p) = (14)(235)(67)$, then $ecs(p) = 4, 3, 5, 6, 7$.

In the following, we present a shortest-path routing algorithm that all non-invariants of p become invariants according to the sequence $ecs(p)$.

Algorithm SPR-2^+ (Shortest-Path Routing)
Input: $C(p) = c_1c_2\cdots c_k$.
Output: A shortest routing path from p to I_n.

1. Let $ecs(p) = s_1, s_2, \ldots, s_r$ where r is the number of non-invariants of p, except 1 and 2.
2. For $i = 1$ to r do
2.1 Let $h = p_{s_i}^{-1}$.
2.2 If $h = 1$ then $f = 1$
else if $h = 2$ and 1,2 are contained in distinct cycles then $f = 2$
else if s_i is the first element in the representation of a cycle then perform g_h^+ and let $f = 2$.
2.3 If $f = 1$ then perform $g_{s_i}^+$ and let $f = 2$
else perform $g_{s_i}^-$ and let $f = 1$.

For example, $p = 1623745$. Then the corresponding cycle of p is $(2643)(57)$ where $c_1 = (2643)$, $c_2 = (57)$ and $ecs(p) = 6, 4, 3, 5, 7$. When $i = 1$, the operations performed in Step 2 are as follows. In Step 2.1, $h = p_6^{-1} = 2$. In Step 2.2, $f = 2$. Step 2.3 performs g_6^- and let $f = 1$. Thus, the operations performed for p are g_6^-, g_4^+, g_3^-, g_7^+, g_5^- and g_7^+.

In the above algorithm, f denotes the position 1 or 2 that the element in the position will be moved into the right place in the next step. Also, we observe that for two elements $s, t \in c_i$ appear consecutively in $ecs(p)$ and s precedes t, if s is moved into the right place by some operations, then t must be at the position 1 or 2. We then carry out the operation g_t^+ or g_t^- and thus t will be at the right place in the next step. Continue this work for each next element of the cycle. Therefore, all elements of c_i should be at the right place. Thus, we can proceed to the elements of the next cycle. As a result, the algorithm produces a routing path. Indeed, the length of the derived path equals to D_p.

Note that the algorithm possesses the following interesting property. Let $r = r_1 r_2 \cdots r_n$ be a vertex in the derived path which is processing. If we carry out operation g_h^- (respectively g_h^+) in step 2.2, then r_1 (respectively r_2) always retains its position at 1 or 2 during the course that all elements of the cycle are moved into the right place. In particular, r_1 and r_2 will back in the position 1 and 2 if all other elements of the cycle have been arranged at the right place. Besides, we can change the operation g_h^- instead of g_h^+ and let $f = 1$ in step 2.2. Then the resulting path is still a shortest routing path. Specifically, we call such a modified algorithm as SPR-2^-.

We now introduce the notion about Latin square which is helpful for the constructing of parallel routing. Assume $|ecs(p)| = d$. For each $i = 1, \ldots, d$, we construct a new sequence from $ecs(p)$ by regularly moving circularly the elements with i positions. Combining these d sequences to form a matrix results in a Latin square matrix; i.e., none of the elements occurring twice within any row or column in the matrix. For instance, if $ecs(p) = 4, 3, 5, 6, 7$, we can obtain the following matrix:

$$\begin{bmatrix} 4 & 3 & 5 & 6 & 7 \\ 3 & 5 & 6 & 7 & 4 \\ 5 & 6 & 7 & 4 & 3 \\ 6 & 7 & 4 & 3 & 5 \\ 7 & 4 & 3 & 5 & 6 \end{bmatrix}$$

Applying each row of the $d \times d$ Latin square as a conveying sequence, it can be shown that the derived paths are node-disjoint (by the similar technique used in [6] and [8] on star graphs and recursive circulant network respectively). Let $ecs_i(p)$ be the conveying sequence obtained from row i, $i = 1, 2, \cdots, d$. In what follows, we will show that for each $ecs_i(p)$ we can construct two corresponding edge-disjoint paths. Thus, we have totally $2d$ edge-disjoint routing paths. Moreover, we will prove that all of the paths are shortest.

3.1. The case where P contains no invariant

In this subsection, we consider the case that p contains no invariant, i.e, $p_i \neq i$ for all $i \geq 3$.

Our algorithm need the cycle representation $C(p) = c_1 c_2 \cdots c_k$ as the input. The following algorithm can obtain $2(n-2)$ edge-disjoint paths and all derived paths are the shortest paths.

Algorithm A (Parallel Routing)

Input: $C(p) = c_1c_2\cdots c_k$, and $p_1, p_2 \in \{1,2\}$, and $p_i \neq i$ for $i = 3,4,\ldots,n$.

Output: $2(n-2)$ edge-disjoint paths from p to I_n.

1. Get $ecs(p) = s_1, s_2, \ldots, s_{n-2}$.

2. For $i = 1$ to $n-2$ do

   $ecs_i(p) = s_i, \ldots, s_{(n-2)}, s_1, \ldots, s_{(i-1)}$.

   Assume s_i is contained in cycle c and s_j is the last element of c.

2.1 Let $h = p_{s_i}^{-1}$.

2.2 Let $p' = p\ \mathrm{g}_h^+\ \mathrm{g}_{s_i}^-\ \mathrm{g}_{s_{(i+1)}}^+ \cdots \mathrm{g}_{s_j}^*$, where $*$ is "$-$" if $(j-i)$ is even, otherwise $*$ is "$+$".
The subsequence $s_{j+1}, \ldots, s_{n-2}, s_1, s_2, \ldots, s_{i-1}$ is the conveying sequence of p'.
Call SPR-2^+ with $s_{j+1}, \ldots, s_{n-2}, s_1, s_2, \ldots, s_{i-1}$ as the input to obtain the routing path between p' and I_n.

2.3 Let $p'' = p\ \mathrm{g}_h^-\ \mathrm{g}_{s_i}^+\ \mathrm{g}_{s_{(i+1)}}^- \cdots \mathrm{g}_{s_j}^*$, where $*$ is "$+$" if $(j-i)$ is even, otherwise $*$ is "$-$".
The subsequence $s_{j+1}, \ldots, s_{n-2}, s_1, s_2, \ldots, s_{i-1}$ is the conveying sequence of p''.
Call SPR-2^- with $s_{j+1}, \ldots, s_{n-2}, s_1, s_2, \ldots, s_{i-1}$ as the input to obtain the routing path between p'' and I_n.

For example, let $p = 12537846$. Then, $c_1 = (3574)$, $c_2 = (68)$ and $ecs(p) = 3,5,7,4,6,8$. When $i = 1$, the operations performed in Step 2 are as follows. In Step 2.1, $h = p_3^{-1} = 4$. In Steps 2.2 and 2.3, $p' = p\ \mathrm{g}_4^+\ \mathrm{g}_3^-\ \mathrm{g}_5^+\ \mathrm{g}_7^-\ \mathrm{g}_4^+ = 21345876$ and $p'' = p\ \mathrm{g}_4^-\ \mathrm{g}_3^+\ \mathrm{g}_5^-\ \mathrm{g}_7^+\ \mathrm{g}_4^- = 21345876$, respectively. In Step 2.2 (respectively Step 2.3), Algorithm SPR-2^+ (respectively SPR-2^-) will be invoked to find the routing path from p' (respectively p'') to I_n.

Lemma 2 *Let $p = p_1p_2\cdots p_n$ be a vertex in AG_n. If $p_1, p_2 \in \{1,2\}$ and $p_i \neq i$ for $3 \leq i \leq n$, then the derived path in Algorithm A corresponding to the conveying sequence $ecs_i(p)$, $i = 1,2,\ldots,n-2$, is a shortest path.*

Proof. We only prove the case $p_1 = 1$ and $p_2 = 2$. For the case where $p_1 = 2$ and $p_2 = 1$, it can be proved by a similar way. Without loss of generality, we may assume that the derived path is produced by Step 2.2 of Algorithm A. Clearly, it requires $(j-i+1)+1$ operations to move the elements $s_i, \ldots, s_j$ into their right places. We first consider $j-i+1$ to be an odd integer. Let $p' = p\ \mathrm{g}_h^+\ \mathrm{g}_{s_i}^-\ \mathrm{g}_{s_{i+1}}^+\ \mathrm{g}_{s_{i+2}}^- \cdots \mathrm{g}_{s_j}^-$ where $h = p_{s_i}^{-1}$. That is, p' is the first vertex in the derived path from p to I_n such that the elements $s_i, \ldots, s_j$ are in their right place. Applying Algorithm SPR-2^+ to construct a routing path from p' to I_n, $D_{p'} = n+k-(j-i+1)-3$, where k is the number of cycles in $C(p)$. Therefore, the derived path has length $D_p = D_{p'} + (j-i+1)+1 = n+k-2$, and by Lemma 1, it is a shortest path . For the case where $j-i+1$ is an even integer, by a similar argument as the preceding case, we can show that the derived path is also a shortest path. □

Theorem 3 *Algorithm* A *produces* $2(n-2)$ *edge-disjoint shortest paths from vertex p to I_n.*

Proof. According to the result of Lemma 1 and the conveying sequences from Latin square, we only need to show that the two derived paths produced by Steps 2.2 and 2.3 in Algorithm A are edge-disjoint. Due to the fact that performing the operation g_h^+ (respectively g_h^-) will cause 2 (respectively 1) retaining at the starting two positions (where h is the position that the first element of a cycle can be found), this shows that the two subpaths (not including the source vertex and destination vertex) generated by the elements of a cycle in the corresponding conveying sequences are node-disjoint. Thus, these two derived paths from p to I_n are edge-disjoint. □

3.2. The case where p contains invariants

In this sbusection, we consider the case that p contains l invariants, and 1 and 2 are not as a part of the invariants. In this case, $|ecs(p)| = n - l - 2$ where l is the number of invariants contained in p. Due to the result of the routing paths produced by the Algorithm A in Section 3.1, we can obtain $2(n-l-2)$ edge-disjoint paths from p to I_n. To complete the design of parallel routing algorithm, we need to construct another $2l$ edge-disjoint paths from p to I_n.

In this case, either $p_1 = 1$, $p_2 = 2$ or $p_1 = 2$, $p_2 = 1$. Without loss of generality, we consider $p_1 = 2$ and $p_2 = 1$. Let $p = p_1p_2p_3 \cdots p_n$ with cycle representation $c_1c_2 \cdots c_k$. Assume that p_t is an invariant of p. We now operate $p \cdot g_t^+$ and $p \cdot g_t^-$. That is, $p' = p\ g_t^+ = p_2tp_3 \cdots p_{(t-1)}p_1p_{(t+1)} \cdots p_n$ and $p'' = p\ g_t^- = tp_1p_3 \cdots p_{(t-1)}p_2p_{(t+1)} \cdots p_n$. Therefore, we have $C(p') = (2t)c_1c_2 \cdots c_k$ and $C(p'') = (1t)c_1c_2 \cdots c_k$. For example, consider $p = 21435$. Then $p' = p\ g_5^+ = 15432$ and $p'' = p\ g_5^- = 52431$. The cycle representations are $C(p') = (34)(25)$ and $C(p'') = (34)(15)$.

Algorithm B (Parallel Routing)

Input: $p = p_1p_2 \cdots p_n = c_1c_2 \cdots c_ke_1e_2 \cdots e_l$.

Output: $2l$ node-disjoint paths from p to I_n.

For $i = 1$ to l do

Let $t = e_i$.

1. Let $p' = p\ g_t^+ = p_2tp_3 \cdots p_{(t-1)}p_1p_{(t+1)} \cdots p_n$. Use $c_1c_2 \cdots c_k \cdot t$ as the conveying sequence to produce a path from p' to I_n.
2. Let $p'' = p\ g_t^- = tp_1p_3 \cdots p_{(t-1)}p_2p_{(t+1)} \cdots p_n$. Use $c_1c_2 \cdots c_k \cdot t$ as the conveying sequence to produce a path from p'' to I_n.

Theorem 4 *Let $p = p_1p_2 \cdots p_n$ be a vertex in AG_n. If $p_1, p_2 \in \{1, 2\}$, then the paths from p to I_n found by Algorithm B are node-disjoint. In particular, all the paths have length $D_p + 2$.*

Proof. Suppose that $C(p) = c_1c_2 \cdots c_k$ and there is an invariant at position t (i.e., $p_t = t$). Let $p' = p\ g_t^+$ and $p'' = p\ g_t^-$. Then $C(p') = c_1c_2 \cdots c_k(2t)$ and $C(p'') = c_1c_2 \cdots c_k(1t)$. Since p' and p'' are the first vertices, respectively, in the two derived paths from p to I_n and the algorithm uses the same conveying sequence to produce the paths, it guarantees that the two derived paths are node-disjoint. Moreover, since p contains l invariants and each invariant t always retains its position at 1 or 2, the algorithm can produce $2l$ node-disjoint paths. Also, it is easy to see that the length of each path produced by the algorithm is $D_p + 2$.

□

4. Conclusions

In this paper, we propose an algorithm that generates $2(n-2)$ edge-disjoint paths on alternating group graphs under a special case. All of the derived paths are shortest. Now, we are trying to solve the problem for the general case.

References

[1] S. B. Akers and B. Krishnamurthy, A group theoretic model for symmetric interconnection networks, *IEEE Transsactions on Computers*, C-38 (1989) 555–566.

[2] C. Berge, *Principles of Combinatorics* (Academic Press, New York, 1971).

[3] E. Cheng and M. J. Lipman, Orienting split-stars and alternating group graphs, *Networks*, 35 (2000), 139–144.

[4] W. K. Chiang and R. J. Chen, On the arrangement graph, *Information Processing Letters*, 66 (1998) 215–219.

[5] K. Day and A. Tripathi, arrangement graphs: A class of generalized star graphs, *Information Processing Letters*, 42 (1992) 235–241.

[6] K. Day and A. Tripathi, A comparative study of topological properties of hypercubes and star graphs, *IEEE Transactions on Parallel and Distributed Systems*, 5 (1994) 31–38.

[7] J. S. Jwo, S. Lakshmivarahan, and S. K. Dhall, A new class of interconnection networks based on the alternating group, *Networks*, 23 (1993), 315–326.

[8] S. Kim and I. Chung, Application of the special Latin square to a parallel routing algorithm on a recursive circulant network, *Information Processing Letters*, 66 (1998) 141–147.

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

[10] W. Ledermann, *Introduction to the Theory of Finite Groups* (Oliver and Boyd, London, 1949).