International Computer Symposium 2002, Workshop on Computer Systems

# Embedding Edge-Disjoint Spanning Trees on the Alternating Group Graphs

Chin-Tsai Lin (林錦財)*

Assistant Professor
Department of Information Management,
Kun-Shan University of Technology

**_Abstract_**

*In this paper we construct a graph that consists of the maximum number of directed edge-disjoint spanning trees on the alternating group graph. The paths that route from the common root node to any given node through different spanning trees are node-disjoint. This graph can be used to derive fault tolerant algorithms for the broadcasting and scattering problems under the all port communication model.*

Keywords: Interconnection networks, alternating group graphs, node-disjoint paths, edge-disjoint spanning trees

| | |
|---|---|
| *Correspondance: | 台南縣永康市 710 忠義街 10 巷 1 號, Taiwan |
| Tel.: | (06) 311-2107 (0921) 217-922 |
| Fax.: | (06) 273-2726 |
| E-mail: | ctlin@mail.ksut.edu.tw |
| | jench1@seed.net.tw |

# Embedding Edge-Disjoint Spanning Trees on the Alternating Group Graphs

Chin-Tsai Lin

Department of Information Management, Kun-Shan University of Technology

Taiwan, Republic of China

***Abstract***

*In this paper we construct a graph that consists of the maximum number of directed edge-disjoint spanning trees on the alternating group graph. The paths that route from the common root node to any given node through different spanning trees are node-disjoint. This graph can be used to derive fault tolerant algorithms for the broadcasting and scattering problems under the all port communication model.*

*Keywords: Interconnection networks, alternating group graphs, node-disjoint paths, edge-disjoint spanning trees*

## 1. Introduction

Jwo, Lakshmivarahan and Dhall [18] proposed Cayley graphs based on the alternating groups as a topology for interconnecting processors in parallel computers. With respect to contention problem for general routing, it has been shown that alternating group graphs perform better than the star graphs and are close to the hypercube.

The alternating group graphs are defined as follows. Let $g_i^+$ denote the permutation (1 2 $i$), $g_i^-$ denote (1 $i$ 2). Let $\Omega=\{g_i^+ \mid 3 \le i \le n\} \cup \{g_i^- \mid 3 \le i \le n\}$. It is well known that $\Omega$ is a generator set for the set of all even permutations of $n$ symbols, denoted by $A_n$.

**Definition 1.** An alternating group graph on $n$ symbols, denoted by $AG_n = (V_n, E_n)$, is an undirected graph with $n!/2$ nodes, in which $V_n$ is the set of $n!/2$ even permutations. $E_n = \{(p, q) | p, q \in A_n, q = p \cdot h$, for $h \in \Omega\}$.

Fig. 1 shows the $AG_4$ graph. It can be verified that $g_i^+ \cdot g_i^+ = g_i^-$, $g_i^- \cdot g_i^- = g_i^+$, $(g_i^+)^{-1} = g_i^-$ and $(1\ 2) \cdot g_i^+ \cdot (1\ 2) = g_i^-$, for $3 \le i \le n$. The diameter of $AG_n$ is $\lfloor 3(n-2)/2 \rfloor$ and the connectivity is optimal, i.e., equal to the degree $2(n-2)$.

Alternating group graphs have received considerable attention. In [19], Lai and Tsay presented algorithm for all-port all-to-all broadcasting and single-node scattering whose time performances are one step more than the lower bounds. Hung and Huang [15] also introduced an optimal one-port one-to-all broadcasting algorithm on alternating group graphs. Yang et al. [23] introduced a method to explore a set of edge-disjoint paths between any two nodes. In [20], it has been shown that the fault diameter of an alternating group graph is no more than the original diameter by one. Cheng and Lipman [5] have shown that the alternating group graphs are a subclass of arrangement graphs. Day and Tripathi [9] introduced schemes for constructing node-to-node disjoint paths in the arrangement graphs. Another scheme for constructing node-to-node disjoint paths is given in [21].

In this paper, we introduce a scheme for constructing on $AG_n$ $2(n-2)$ *directed edge-disjoint spanning trees* (*EDSTs*). This is the maximum number of edge-disjoint spanning trees that can be constructed on $AG_n$, since the degree of $AG_n$ is $2(n-2)$. The depth of the

EDSTs graph we construct differs from the minimum possible depth by a small constant. The root node of EDSTs has $2(n-2)$ node-disjoint paths to each one of the other nodes, one path through each of the $2(n-2)$ edge-disjoint spanning trees. Similar graphs have been previously constructed on other interconnection networks, such as the binary hypercube [16], the cube-connected-cycles (CCC) [13], and the star graphs [12]. The construction of the EDSTs graph is equivalent to the problem of exploiting the disjoint paths between a node s and all the other $n!/2–1$ nodes of $AG_n$. Using the EDSTs graph we can derive fault tolerant algorithms for the single-node and multimode broadcasting, and for the single-node and multinode scattering problem on $AG_n$ as in [12].

The remaining of the paper is organized as follows. In the next section, we will introduce the notations and definitions that are throughout the paper. Section 3 presents the scheme of embedding $2(n-2)$ directed edge-disjoint spanning trees. Conclusions are finally drawn in section 4.

## 2. Notations and Definitions

We now define the rotation operation that is important for constructing the $2(n–2)$ directed edge-disjoint spanning trees.

**Definition 2.** Let us define a bijection $r$ from the set $\{1, 2, \ldots, n\}$ to itself as follows:

$$r(i) = \begin{cases} i, & \text{if } i = 1 \text{ or } 2 \\ ((i-2) \bmod (n-2)) + 3, & \text{otherwise,} \end{cases}$$

(notice that $r$ maps symbols 1 and 2 to themselves and the remaining symbols as follows: $3 \rightarrow$

$4 \rightarrow \ldots \rightarrow n-1 \rightarrow n \rightarrow 3$). The rotation of a node $p$ of $AG_n$ is defined to be the node:

$$R(p) = r(p_1)r(p_2)r(p_n)r(p_3)\ldots r(p_{n-1})$$

or equivalently $R(p) = p'$ so that $p'_{r(i)} = r(p_i)$, $1 \le i \le n$.

This means that the position of each symbol of $p$ and the symbol itself are mapped through $r$. For example, $R(4132) = 3124$ and $R(4321) = 3412$. Notice that $R(g_i^+) = g^+_{((i-2) \bmod (n-2))+3}$ and $R(g_i^-) = g^-_{((i-2) \bmod (n-2))+3}$.

By rotation of a network we mean that rotation is applied to each node of the network. The rotation operation is an automorphism on $AG_n$ that possesses the following properties.

1. It maps the identity node to itself. As an extension to this, nodes $p$ and $R(p)$ are always at the same distance from the identity node.
2. Let $(v, u)$ be an edge and $u = v \cdot g_i^\sigma$. Then $(R(v), R(u))$ is an edge of generator $g^\pm_{((i-2) \bmod (n-2))+3}$. ($\sigma$ denotes the *sign* + or -.)

The conjugation operation is also important for constructing the $2(n-2)$ directed edge-disjoint spanning trees.

**Definition 3.** The conjugation of a node $p$ of $AG_n$ is defined to be the node $(1\ 2) \cdot p \cdot (1\ 2)$.

## 3. The edge-disjoint spanning trees

In this section we construct the $2(n-2)$ directed edge-disjoint spanning trees graph, rooted at the identity node of $AG_n$. The spanning tree rooted at the neighbor of the identity node over generator $g_i^\sigma$, node $g_i^\sigma$, is denoted by $T_i^\sigma$. Each spanning tree includes all nodes of $AG_n$ except the identity node. The reverse-direction spanning tree of $T_i^\sigma$ is a rendezvous result of the disjoint paths whose last generator is $(g_i^{\sigma})^{-1}$ from each one of the other nodes to the identity node. Since the alternating group graphs are symmetric, the EDSTs graph can be easily translated into that rooted at any other node.

Before proceeding to description of the spanning trees algorithms, we need the following definition.

**Definition 4.** For each node $p$ (excluding the identity node and its neighbor), we define $w_1$ and $w_2$ as follows:

1. If the cycle structure of $p$ is $(x_1\ x_2 \ldots x_t\ 1)(2)\ldots$ or $(x_1\ x_2 \ldots x_t\ 1\ 2)\ldots$, $t > 1$, we define $w_1 = x_1$. For example, for node $p = 42351 = (4\ 5\ 1)(2)(3)$ of $AG_5$, we have $w_1 = 4$.
2. If the cycle structure of $p$ is $(x_1\ 1)(2)\ldots$ or $(x_1\ 1\ 2)\ldots$, we define $w_1$ to be the first position in $p$, among $x_1$+1, …, $n$, 3, …, $x_1$–1 that does not include its correct symbol. If $w_1$ cannot be found, we let $w_1$= 0 and $p_{w_1} = 0$. For example, for node $p = 52431 = (5\ 1)(2)(3\ 4)$ of $AG_5$, we have $p_{w_1} = p_3 = 4$. For node $25341 = (5\ 1\ 2)(3)(4)$ of $AG_5$, we have $w_1 = p_{w_1} = 0$.
3. If the cycle structure of $p$ is $(y_1\ y_2 \ldots y_s\ 2)(1)\ldots$ or $(y_1\ y_2 \ldots y_s\ 2\ 1)\ldots$, $s > 1$, we define $w_2 = y_1$. For example, for node $p = 31452 = (3\ 4\ 5\ 2\ 1)$ of $AG_5$, we have $w_2 = 3$.
4. If the cycle structure of $p$ is $(y_1\ 2)(1)\ldots$ or $(y_1\ 2\ 1)\ldots$, we define $w_2$ to be the first position in $p$, among $y_1$+1, …, $n$, 3, …, $y_1$–1 that does not include its correct symbol. If $w_2$ cannot be found, we let $w_2$= 0 and $p_{w_2} = 0$. For example, for node $p = 13254$ of $AG_5$, we have $p_{w_2} = p_4 = 5$. For node $41325 = (4\ 2\ 1)(3)(5)$ of $AG_5$, we have $w_2 = p_{w_2} = 0$.
5. If the cycle structure of $p$ is $(x_1\ x_2 \ldots x_t\ 1)(y_1\ y_2 \ldots y_s\ 2)\ldots$ or $(x_1\ x_2 \ldots x_t\ 1\ y_1\ y_2 \ldots y_s\ 2)\ldots$, $s \geq$

1, $t \geq 1$, we define $w_1 = x_1$ and $w_2 = y_1$. For example, for node $p$ = 43521= (3 5 1 4 2) of $AG_5$, we have $w_1 = 3$ and $w_2 = 4$.

We now describe an algorithm, Parent($p$, $T_i^\sigma$), that for given node $p$ (excluding the identity node and its neighbors) computes the parent node of $p$ in each one of the spanning tree $T_i^\sigma$, $3 \leq i \leq n$. In what follows, by $f_i(p)$ we denote the parent of node $p$ in spanning tree $T_i^\sigma$. Since the first two symbols is determinate if the other symbols are known, an even permutation $p= p_1p_2\ldots p_n$ can be represented by $\{p_1, p_2\}p_3\ldots p_n$ or $\{p_2, p_1\}p_3\ldots p_n$ without ambiguity.

**Algorithm** Parent($p$, $T_i^\sigma$) {

if (σ==+) { // 1 is at position $i$ right before reaching the identity node

(1) if ($p$=={1, $z$}$p_3\ldots p_n$) {$f_i(p)$ ={$z$, $p_i$}$p_3\ldots p_{i-1}1p_{i+1}\ldots p_n$;} // $z$ may be 2 or $y_1$

else if ($p$=={2, $x_1$}$p_3\ldots p_n$) {

$k = p^{-1}(1)$;

(2) if ($i \neq k$ and $i \neq p_{w_1}$ and $i \neq x_1$) {$f_i(p)$ ={$i$, 2}$p_3\ldots p_n$;}

(3) else if ($i$==$k$) {$f_i(p)$ ={$p_{w_1}$, 2}$p_3\ldots p_{w_1-1}x_1p_{w_1+1}\ldots p_n$;}

(4) else if ($i$==$x_1$) {$f_i(p)$ ={1, 2}$p_3\ldots p_{k-1}x_1p_{k+1}\ldots p_n$;}

else if ($i$==$p_{w_1}$) {

(5) if ($x_1$==$k$) {$f_i(p)$ ={1, 2}$p_3\ldots p_{k-1}x_1p_{k+1}\ldots p_n$;}

(6) else {$f_i(p)$ = {$k$, 2}$p_3\ldots p_n$;}

} // end of if ($p$=={2, $x_1$}$p_3\ldots p_n$)

else if ($p==\{x_1, y_1\}p_3\ldots p_n$) {    // $x_1$, $y_1$ are as those stated above

$k_1 = p^{-1}(1)$; $k_2 = p^{-1}(2)$;

(2’) if ($i \notin \{k_1, p_{w_1}, x_1\}$ and $i \notin \{y_1, y_2, \ldots, y_s\}$) {$f_i(p) = \{y_1, i\}p_3\ldots p_n$;}

(3’) if ($i==k_1$) {$f_i(p) = \{p_{w_2}, x_1\}p_3\ldots p_{w_2-1}y_1p_{w_2+1}\ldots p_n$;}

(4’) else if ($i==x_1$) {$f_i(p) = \{y_1, 1\}p_3\ldots p_{k_1-1}x_1p_{k_1+1}\ldots p_n$;}

(6’) else if ($i==p_{w_1}$) {$f_i(p) = \{y_1, k_1\}p_3\ldots p_n$;}

(7) else if ($i==k_2$) {$f_i(p) = \{1, x_1\}p_3\ldots p_{k_1-1}y_1p_{k_1+1}\ldots p_n$;}

(8) else if ($i==y_1$) { $f_i(p) = \{y_1, k_2\}p_3\ldots p_n$;}

(9) else {$f_i(p) = \{y_1, i\}p_3\ldots p_n$;}

} // end of if ($p==\{x_1, y_1\}p_3\ldots p_n$)

} // end of if ($\sigma$==+)

if ($\sigma$==–) { // 2 is at position $i$ right before reaching the identity node;

// It is the conjugation of the case $\sigma$==+.

(1) if ($p==\{2, z\}p_3\ldots p_n$) {$f_i(p) = \{z, p_i\}p_3\ldots p_{i-1}2p_{i+1}\ldots p_n$;}    // $z$ may be 1 or $x_1$

else if ($p==\{1, y_1\}p_3\ldots p_n$) {

$k = p^{-1}(2)$;

(2) if ($i \neq k$ and $i \neq p_{w_2}$ and $i \neq y_1$) {$f_i(p) = \{i, 1\}p_3\ldots p_n$;}

(3) else if ($i==k$) {$f_i(p) = \{p_{w_2}, 1\}p_3\ldots p_{w_2-1}y_1p_{w_2+1}\ldots p_n$;}

(4) else if ($i==y_1$) {$f_i(p) = \{1, 2\}p_3\ldots p_{k-1}y_1p_{k+1}\ldots p_n$;}

else if ($i==p_{w_2}$) {

(5) if ($y_1$==$k$) {$f_i(p)$ ={1, 2}$p_3\ldots p_{k-1}y_1p_{k+1}\ldots p_n$;}

(6) else {$f_i(p)$ = {$k$, 1}$p_3\ldots p_n$;}

} // end of if ($p$=={1, $y_1$}$p_3\ldots p_n$)

else if ($p$=={$x_1$, $y_1$}$p_3\ldots p_n$) { // $x_1$, $y_1$ are as those stated above

$k_1 = p^{-1}(1)$; $k_2 = p^{-1}(2)$;

(2’) if ($i \notin \{k_2, p_{w_2}, y_1\}$ and $i \notin \{x_1, x_2, \ldots, x_t\}$) {$f_i(p)$ ={$x_1$, $i$}$p_3\ldots p_n$;}

(3’) else if ($i$==$k_2$) {$f_i(p)$ ={$p_{w_1}$, $y_1$}$p_3\ldots p_{w_1-1}x_1p_{w_1+1}\ldots p_n$;}

(4’) else if ($i$==$y_1$) {$f_i(p)$ ={$x_1$, 2}$p_3\ldots p_{k_2-1}y_1p_{k_2+1}\ldots p_n$;}

(6’) else if ($i$==$p_{w_2}$) {$f_i(p)$ ={$x_1$, $k_2$}$p_3\ldots p_n$;}

(7) else if ($i$==$k_1$) {$f_i(p)$ ={2, $y_1$}$p_3\ldots p_{k_2-1}x_1p_{k_2+1}\ldots p_n$;}

(8) else if ($i$==$x_1$) {$f_i(p)$ ={$x_1$, $k_1$}$p_3\ldots p_n$;}

(9) else {$f_i(p)$ ={$x_1$, $i$}$p_3\ldots p_n$;}

} // end of if ($p$=={ $x_1$, $y_1$}$p_3\ldots p_n$)

} // end of if (σ==–)

} // end of algorithm

For example, the EDSTs graph on $AG_5$ are shown in Fig.2. The following lines illustrate the usage of rules (6’) and (9) in the case $p$ = 43156287 = (3 1 4 5 6 2) (7 8) through $T_5^-$ and $T_5^+$ respectively on $AG_8$:

$T_5^-$: 43156287 $\xrightarrow{(6')}$ 36154287 $\xrightarrow{(2')}$ 53164287 $\xrightarrow{(4')}$ 32164587 $\xrightarrow{(1)}$ 43162587→…

$T_5^+$: 43156287 $\xrightarrow{(9)}$ 54136287 $\xrightarrow{(8)}$ 65134287 $\xrightarrow{(4')}$ 16534287 $\xrightarrow{(1)}$ 64531287→… .

**Theorem.** The Parent($p$, $T_i^\sigma$) algorithm defines a spanning tree, rooted at the node $g_i^\sigma$. The 2($n$–2) spanning trees constructed by the Parent algorithm possess the following properties:

1) If all the edges of the spanning trees are directed from parent to children nodes, these spanning trees are edge-disjoint. Consequently, the EDSTs graph with node $12\ldots n$ as the common root contains all edges of $AG_n$ except those edges that are directed towards node $12\ldots n$.
2) The identity node has 2($n$–2) disjoint paths to each other node of $AG_n$, one path through each one of the 2($n$–2) spanning trees. The lengths of these paths differ from the shortest possible lengths by a small additive constant.
3) The depth of the EDSTs graph is less than or equal to $\lfloor 3(n-2)/2 \rfloor + 4$, which is optimal to within a small additive constant. This constant is less than or equal to 3.
4) Each spanning tree can be derived from its preceding one, by the application of a rotation or conjugation. From the properties of the rotation and conjugation operations, we conclude that all the spanning trees are isomorphic.

**Proof:** We prove each of the properties separately.

1) It is sufficient to prove that the parent of a node $p$ in each one of the spanning trees, is a different one of its neighbor nodes. Four cases of $p$ are distinguished:
   i) $p = \{1, 2\}p_3\ldots p_n$. Its parent in $T_i^+$, $f_i(p) = \{2, p_i\}p_3\ldots p_{i-1}1p_{i+1}\ldots p_n$, is derived from moving 1 to position $i$ and its parent in $T_i^-$, $f_i(p) = \{1, p_i\}p_3\ldots p_{i-1}2p_{i+1}\ldots p_n$, is derived from moving 2 to position $i$, for $3 \le i \le n$. Clearly, the parent of the node in each one

of the $2(n-2)$ spanning trees is a different one of its neighbor nodes.

ii) $p = \{1, y_1\}p_3 \ldots p_n$, where $y_1 \neq 2$. Its *parent* in $T_i^+$, $f_i(p) = \{y_1, p_i\}p_3 \ldots p_{i-1}1p_{i+1} \ldots p_n$, is derived from moving 1 to position $i$, while its parent in $T_i^-$ has symbol 1 at one of the first two positions and $y_1$ at some position $j$, $3 \leq j \leq n$. Clearly, the parent nodes in $T_i^+$, are different. The parents in $T_i^-$ are again different neighbors of $p$ because the $j$'s are different.

| | $p$: $y_1 = p^{-1}(2)$ | $p$: $p_{w_2} = p^{-1}(2)$ | $p$: $p_{w_2} \neq p^{-1}(2)$ and $y_1 \neq p^{-1}(2)$ |
|---|---|---|---|
| $T_i^-$ : $i = p^{-1}(2)$ | Rule (3); $j = p^{-1}(p_{w_2})$ | | |
| $T_i^-$ : $i = y_1$ | Rule (3) | Rule (4); $j = p^{-1}(2)$ | |
| $T_i^-$ : $i = p_{w_2}$ | Rule (5); $j = p^{-1}(2)$ | Rule (3) | Rule (6); $j = p^{-1}(p^{-1}(2))$ |
| $T_i^-$ : $i$ = other | Rule (2); $j = p^{-1}(i)$ | | |

iii) $p = \{x_1, 2\}p_3 \ldots p_n$, where $x_1 \neq 1$.The proof is similar to the case b) except by conjugation.

iv) $p = \{x_1, y_1\}p_3 \ldots p_n$, where $\{x_1, y_1\} \neq \{1, 2\}$. It can be verified that the parent nodes in $T_i^+$ by rules (2'), (4'), (6'), (8) and (9), and those in $T_i^-$ by rules (3') and (7), $y_1$ is at one of the first two positions and the positions of symbol $x_1$ are all different. Similarly, the parent nodes in $T_i^-$ by rules (2'), (4'), (6'), (8) and (9), and those in $T_i^+$ by rules (3') and (7), $x_1$ is at one of the first two positions and the positions of symbol $y_1$ are all different.

2) We explain how the path from each node $p$ to the identity node is created through each one of the spanning trees. Let us first consider $T_i^+$. If $p_i = 1$, the parent is that derived by rule (3) or (3'), which has the properties:

i) 1 is fixed at position $i$. Hence, the parent of $f_i(p)$ is also derived by applying rule (3) or

(3') if it exists.

ii) If 2 is at one of the first two positions of $p$, 2 is also at one of the first two positions of $f_i(p)$.

iii) $f_i(p)$ is closer the identity node than $p$, that is, $p$ has a path of minimum length to $12\ldots n$ through $T_i^+$.

In case $p_1 = 1$ or $p_2 = 1$, the parent of $p$ is that derived by rule (1), in which 1 is at position $i$. If $1 \notin \{p_1, p_2, p_i\}$, we first apply a sequence of rule(s) to move 1 to one of the first two positions, and then apply rule (1) to move 1 to position $i$. By a careful trace of the Parent algorithm, the possible sequences of rules applied before symbol 1 has moved to one of the first two positions are as follows:

$$\{2, x_1\}p_3\ldots p_n \xrightarrow{(2)} \{i, 2\}p_3\ldots p_n \xrightarrow{(4)} \{1, 2\}\, p_3\ldots p_n$$

$$\{2, x_1\}p_3\ldots p_n \xrightarrow{(4)} \{1, 2\}p_3\ldots p_{k-1}\, x_1\, p_{k+1}\ldots p_n$$

$$\{2, x_1\}p_3\ldots p_n \xrightarrow{(5)} \{1, 2\}p_3\ldots p_{k-1}\, x_1\, p_{k+1}\ldots p_n$$

$$\{2, x_1\}p_3\ldots p_n \xrightarrow{(6)} \{k, 2\}\, p_3\ldots p_n \xrightarrow{(5)} \{1, 2\}p_3\ldots p_n$$

$$\{2, x_1\}p_3\ldots p_n \xrightarrow{(6)} \{k, 2\}p_3\ldots p_n \xrightarrow{(2)} \{i, 2\}p_3\ldots p_n \xrightarrow{(4)} \{1, 2\}p_3\ldots p_n$$

$$\{x_1, y_1\}p_3\ldots p_n \xrightarrow{(2')} \{y_1, i\}p_3\ldots p_n \xrightarrow{(4')} \{1, y_1\}p_3\ldots p_n$$

$$\{x_1, y_1\}p_3\ldots p_n \xrightarrow{(4')} \{y_1, 1\}p_3\ldots p_{k_2-1}\, x_1\, p_{k_2+1}\ldots p_n$$

$$\{x_1, y_1\}p_3\ldots p_n \xrightarrow{(6')} \{y_1, k_1\}p_3\ldots p_n \xrightarrow{(2')} \{i, y_1\}p_3\ldots p_n \xrightarrow{(4')} \{y_1, 1\}p_3\ldots p_n$$

$$\{x_1, y_1\}p_3\ldots p_n \xrightarrow{(7)} \{1, x_1\}p_3\ldots p_{k_1-1}\, y_1\, p_{k_1+1}\ldots p_n$$

$$\{x_1, y_1\}p_3\ldots p_n \xrightarrow{(8)} \{y_1, k_2\}p_3\ldots p_n \xrightarrow{(4')} \{k_2, 1\}p_3\ldots p_n$$

$$\{x_1, y_1\}p_3\ldots p_n \xrightarrow{(9)} \{y_1, i\}p_3\ldots p_n \xrightarrow{(8)} \{i, k_2\}p_3\ldots p_n \xrightarrow{(4')} \{k_2, 1\}p_3\ldots p_n$$

It can be observed that for $p = \{2, x_1\}p_3\ldots p_n$ symbol 2 is always at one of the first two positions through this part of paths, and for $p = \{x_1, y_1\}p_3\ldots p_n$ symbol $y_1$ is always at one of the first two positions through the sequences if the first rule is (2'), (4') or (6'). We have similar properties for $T_i^-$ if we exchange the roles of 1 and 2 and the roles of $x$ and $y$.

To prove the paths are node-disjoint, we now distinguish among different of nodes.

a) $p = \{1, 2\}p_3\ldots p_n$. The paths are node-disjoint since that through $T_i^+$, 1 is at position $i$ while 2 is at one of the first two positions, and that through $T_i^-$, 2 is at position $i$ while 1 is at one of the first two positions.

b) $p = \{1, y_1\}p_3\ldots p_n$, where $y_1 \neq 2$. Node $p$ has a path to $12\ldots n$ through $T_i^+$ in which any node has 1 fixed at position $i$, and if the node has 2 at one of the first two positions, its parent node has 2 at one of the first two positions, too. Thus, these paths through $T_i^+$'s are node-disjoint to one another. They are node-disjoint to the paths through $T_i^-$'s since in those any node has symbol 1 at one of the first two positions. We have to prove that the paths through $T_i^-$'s, are also node-disjoint to one another. Through $T_i^-$, symbol 2 is moved first to one of the first two positions by applying a sequence of rule(s) unless 2 is initially at position $i$. Thereafter, 2 is fixed at position $i$ and 1 is always at one of the first two positions, so the last parts of these paths are node-disjoint. Therefore, we have to prove that the first parts of the paths are also node-disjoint. The possible sequence of rules applied are as follows:

$\{1, y_1\}p_3\ldots p_n \xrightarrow{(2)} \{i, 1\}p_3\ldots p_n \xrightarrow{(4)} \{1, 2\}\, p_3\ldots p_n$

$\{1, y_1\}p_3\ldots p_n \xrightarrow{(4)} \{1, 2\}p_3\ldots p_{k-1}\, y_1\, p_{k+1}\ldots p_n$

$\{1, y_1\}p_3\ldots p_n \xrightarrow{(5)} \{1, 2\}p_3\ldots p_{k-1}\, y_1\, p_{k+1}\ldots p_n$

$\{1, y_1\}p_3\ldots p_n \xrightarrow{(6)} \{k, 1\}\, p_3\ldots p_n \xrightarrow{(5)} \{1, 2\}p_3\ldots p_n$

$\{1, y_1\}p_3\ldots p_n \xrightarrow{(6)} \{k, 1\}p_3\ldots p_n \xrightarrow{(2)} \{i, 1\}p_3\ldots p_n \xrightarrow{(4)} \{1, 2\}p_3\ldots p_n$

These paths also start at different neighbors of node *p*. In $AG_n$ there are no paths of length 3 or less that start at different neighbors of a node of the form {1, *}*, end at a node of the form {1, 2}*, have 1 always at one of the first two positions, and are not node-disjoint. ({1, 2}* denotes a node of $AG_n$ whose set of the first two symbols is {1, 2}, and {1, *}* denotes a node whose set of the first two symbols contains 1.) Furthermore, the lengths of these paths differ from the minimum possible length by a small additive constant because the part of each path from node {1, 2}* to 12…*n* are shortest paths through the subgraph of $AG_n$ that fixes 1 at position *i*.

c) $p = \{x_1, 2\}p_3\ldots p_n$, where $x_1 \neq 1$. The proof is similar to case b) except by conjugation.

d) $p = \{x_1, y_1\}p_3\ldots p_n$, where $\{x_1, y_1\} \neq \{1, 2\}$. The last parts of these paths are node-disjoint, since symbol 1 (symbol 2) is fixed at position *i* and if symbol 2 (symbol 1) is at one of the first two positions, symbol 2 (symbol 1) is also at one of the first two positions of the parent node. (Either the position of 1 or the position of 2 is different.)

We have to prove that the first parts of the paths are also node-disjoint. The paths that first apply rule (2'), (4') or (6') can be classified into the following two kinds:

1. Those have $y_1$ always at one of the first two positions in the first part of the path;
2. Those have $x_1$ always at one of the first two positions in the first part of the path.

Since $x_1$ or $y_1$ does not appear at the first two positions in the counterpart, two paths of different kinds are node-disjoint in the first part of the paths. These paths also start at different neighbors of node $p$. In $AG_n$ there are no paths of length 3 or less that start at different neighbors of a node of the form $\{*, y_1\}*$, end at a node of the form $\{1, y_1\}*$, have $y_1$ always at one of the first two positions, and are not node-disjoint. Hence, the paths of the first kind are node-disjoint to each other. Similarly, the paths of the second kind are node-disjoint to each other. The paths that first apply rule (8) are node-disjoint to the other paths because they distinguish themselves by moving symbol $k_1$ or $k_2$. The paths that first apply rule (9) are node-disjoint to the other paths because they distinguish themselves by moving symbol $i$ through $T_i^+$ or $T_i^-$. It can be verified that the parent nodes of $p$ by rule (7) do not happen to be one of the nodes in the other paths. Therefore, they are all node-disjoint.

3) From part 2 of this lemma, we notice the depth of each spanning tree (from node $g_i^\sigma$) is less than or equal to the diameter of $AG_{n-1}$ plus 4, $\lfloor 3(n-3)/2 \rfloor+4 \leq \lfloor 3(n-2)/2 \rfloor+3$. Consequently, the depth of the EDSTs graph (from node $12\ldots n$) is less than or equal to $\lfloor 3(n-2)/2 \rfloor+4$, which is the diameter of $AG_n$ plus 4. A lower bound for the depth of the

EDSTs graph is posed by the fault diameter of $AG_n$. The fault diameter of $AG_n$ is the diameter of the remaining graph when an arbitrary set of $2(n–2)–1$ nodes are removed from $AG_n$ and has been shown to be one more than the fault free diameter of $AG_n$, $\lfloor 3(n–2)/2 \rfloor +1$. Therefore the depth of the EDSTs graph is optimal to within a small constant. This constant is less than or equal to 3.

4) According to the definition of the rotation operation, when we say that each spanning tree can be obtained as a rotation of its preceding one cyclically, it is equivalent to saying that spanning tree $T_{r(i)}^{\sigma}$ can be obtained as a rotation of spanning tree $T_i^{\sigma}$, $3 \le i \le n$ and $\sigma = +$ or $–$. According to the definition of the conjugation operation, when we say that each spanning tree can be obtained as a conjugation of its preceding one, it is equivalent to saying that spanning tree $T_i^{+}$ ($T_i^{-}$) can be obtained as a conjugation of spanning tree $T_i^{-}$ ($T_i^{+}$), $3 \le i \le n$.

We first prove that edge $(p, f_i(p))$ belongs to spanning tree $T_i^{+}$ if and only if edge $((1\ 2)\cdot p\cdot(1\ 2), (1\ 2)\cdot f_i(p)\cdot(1\ 2))$ belongs to spanning tree $T_i^{-}$. From the properties of the conjugation operation, $(1\ 2)\cdot p\cdot(1\ 2)$ preserves the cycle structure of $p$ except 1 is replaced by 2 and vice versus. Consequently, from the definition of $p_{w_1}$ ($p_{w_2}$) it can be verified that the role of $x$ is replaced by $y$, the role of $w_1$ is replaced by $w_2$, and vice versus. From these we conclude that node $p$ derives its parent in $T_i^{\sigma}$ from a specific rule (1) to (9) of the Parent($p$, $T_i^{+}$) algorithm if and only if node $(1\ 2)\cdot p\cdot(1\ 2)$ derives its parent from the same statement of the Parent($(1\ 2)\cdot p\cdot(1\ 2)$, $T_i^{-}$) algorithm.

We will prove that if edge $(p, f_i(p))$ belongs to spanning tree $T_i^{\sigma}$, then edge $(R(p), R(f_i(p)))$ belongs to spanning tree $T_{r(i)}{}^{\sigma}$. From the properties of the rotation operation, the rotation of a node $p$ is node $R(p) = p'$, so that $p'_{r(i)} = r(p_i)$. If $p_i$ = 1 or 2 then node $p'_{r(i)}$ = 1 or 2. Furthermore, $p'_1 = r(p_1)$, $p'_2 = r(p_2)$ and from the definition of $p_{w_1}$ ($p_{w_2}$) it can be verified that $p'_{w_1} = r(p_{w_1})$ (similarly, $p'_{w_2} = r(p_{w_2})$). From these we conclude that if node $p$ derives its parent in $T_i^{\sigma}$ by a specific rule (1) to (9) of the Parent($p$, $T_i^{\sigma}$) algorithm, then node $p'$ derives its parent by the same rule of the Parent($p'$, $T_{r(i)}{}^{\sigma}$) algorithm.

It can be verified for nodes that derive their parents through each different rule of the Parent algorithm, that if node $p$ is connected to its parent in $T_i^{\sigma}$ through dimension $j$ then node $p'$ is connected to its parent in $T_{r(i)}{}^{\sigma}$ through dimension $r(j)$.

Since the rotation operation is an automorphism on $AG_n$, all spanning trees $T_i^{\sigma}$, $3 \le i \le n$, are isomorphic.     Q.E.D.

## 4. Concluding remarks

In this paper, we have introduced a scheme to construct $2(n–2)$ edge-disjoint trees in $AG_n$. Our scheme is different from that developed by Chen et al.[4] since $AG_n$ is isomorphic to the arrangement graph $A_{n,n-2}$, not $A_{n,2}$. As in [12], these spanning trees can be used to derive fault tolerant algorithms for the broadcasting and scattering problems under the all port communication model.

## References

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

1987.

[2] S. B. Akers and B. Krishnamurthy. "Group graphs as interconnection networks." In *Proceedings of the 14th International Conference on Fault Tolerant Computing*, pp. 422–427, 1984.

[3] Y.-S. Chen, T.-Y. Juang, and E.-H. Tseng. "Congestion-free embedding of multiple spanning trees in an arrangement graph." In *Proceedings of the 1998 International Conference on Parallel and Distributed Systems*, pp.260-366, 1998.

[4] Y.-S. Chen, T.-Y. Juang, and Y.-Y. Shen. "Congestion-free embedding of 2($n$–$k$) spanning trees in an arrangement graph." *Journal of Systems Architecture*, Vol. 47, pp. 73-86, 2001.

[5] E. Cheng and M. J. Lipman. "Orienting split-stars and alternating group graphs." *Networks*, Vol. 35(2), pp. 139-144, 2000.

[6] W.-K. Chiang and R.-J. Chen, "On the arrangement graph." *Information Processing Letters*, Vol. 66, pp. 215-219, 1998.

[7] S. P. Dandamudi. "On hypercube-based hierarchical interconnection network design." *Journal of Parallel and Distributed Computing*, Vol. 12, No. 3, pp.283–289, 1991.

[8] K. Day and A. Tripathi. "Arrangement graphs: a class of generalized star graphs." *Information Processing Letters*, Vol. 42, pp. 235-241, 1992.

[9] K. Day, A. Tripathi. "Characterization of node disjoint path in arrangement graphs," Technical Report TR91-43, Computer Science Department, University of Minnesota, 1991.

[10] Q.-P. Gu and S. Peng. "Node-to-set disjoint paths problem in star graphs." *Information Processing Letters*, Vol. 62, pp.201-207, 1997.

[11] P. Fragopoulou and S. G. Akl. "Fault tolerance communication algorithms on the star network using disjoint paths." In *Proceedings of the 28th Annual Hawaii International Conference on System Sciences*, pp.4-13, 1995.

[12] P. Fragopoulou and S. G. Akl. "Edge-disjoint spanning trees on the star network with applications to fault tolerance." *IEEE Trans on Computers*, Vol.45, No.2, pp.174-185, 1996.

[13] P. Fraigniaud and C. T. Ho. "Arc-disjoint spanning trees on the cube connected cycles network," In *Proceedings of the International Conference on Parallel Processing*, Vol.I, pp.225-229, St.

Charles, III., 1991.

[14] D. F. Hsu. "On Container Width and Length in Graphs, Group and Networks." *IEICE Trans. Fundamentals Electronics, Information and Computer Science,* E77-A, Vol.4, pp.668-680, 1994.

[15] H.-S Hung and H-L Huang. "Broadcasting on the alternating group graph." In *Proceedings of the 2001 National Computer Symposium*, PCCU, Taipei, Taiwan, Dec. 2001, A085.

[16] S.L. Johnson and C. T. Ho. "Optimal broadcasting and personalized communication in hypercubes," *IEEE Trans. on Computers*, Vol.38, No.9, pp.1249-1268, 1989.

[17] J.-S. Jwo, S. Lakshmivarahan, and S. K. Dhall. "Characterization of node disjoint (parallel) paths in star graphs." In *Proceedings of the 5th International Parallel Processing Symposium*, pp.404–409, April 1991.

[18] J.-S. Jwo, S. Lakshmivarahan, and S. K. Dhall. "A new class of interconnection networks based on the alternating group." *Networks*, Vol.23, No.4, pp.315–326, July 1993.

[19] C.-M. Lai and J.-J. Tsay. "Communication algorithms on alternating group graph." In *Proceedings of the Second Aizu International Symposium, Parallel Algorithms/Architecture Synthesis*, pp.104-110, 1997.

[20] C.-T. Lin. "Fault Diameter of the Cayley graphs based on the alternating group." In *Proceedings of the 2000 International Computer Symposium, Workshop on Computer Architecture,* National Chung Cheng University, Chiayi, Taiwan, pp.40-45, Dec. 6-8, 2000.

[21] C.-T. Lin and W.-C. Chiu. "A routing scheme for constructing node-to-node disjoint paths in alternating group graphs." in *Proceedings of the 19th Workshop on Combinatorial Mathematics and Computation Theory* (Algo2002), National Sun Yat-Sen University, Kaoshiung, Taiwan, Mar.29-30, 2002, pp.89-99.

[22] Y.-C. Tseng and J.-P. Sheu. "Toward optimal broadcast in a star graph using multiple spanning trees." *IEEE Trans. on Computers*, Vol.46, No.5, pp.593-599, 1997.

[23] J-S Yang, T-Y Ho, Y-L Wang, and M-T Ko. "Applying Latin square on parallel routing of alternating group graphs." In *Proceedings of the 2000 International Computer Symposium, Workshop on Algorithm,* National Chung Cheng University, Chiayi, Taiwan, Dec. 6-8, 2000.

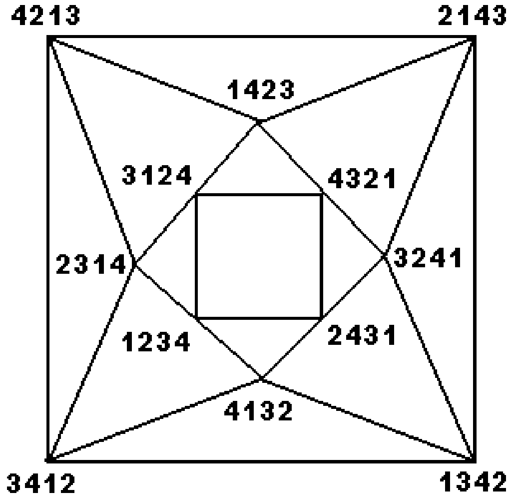


Fig. 1 $AG_4$

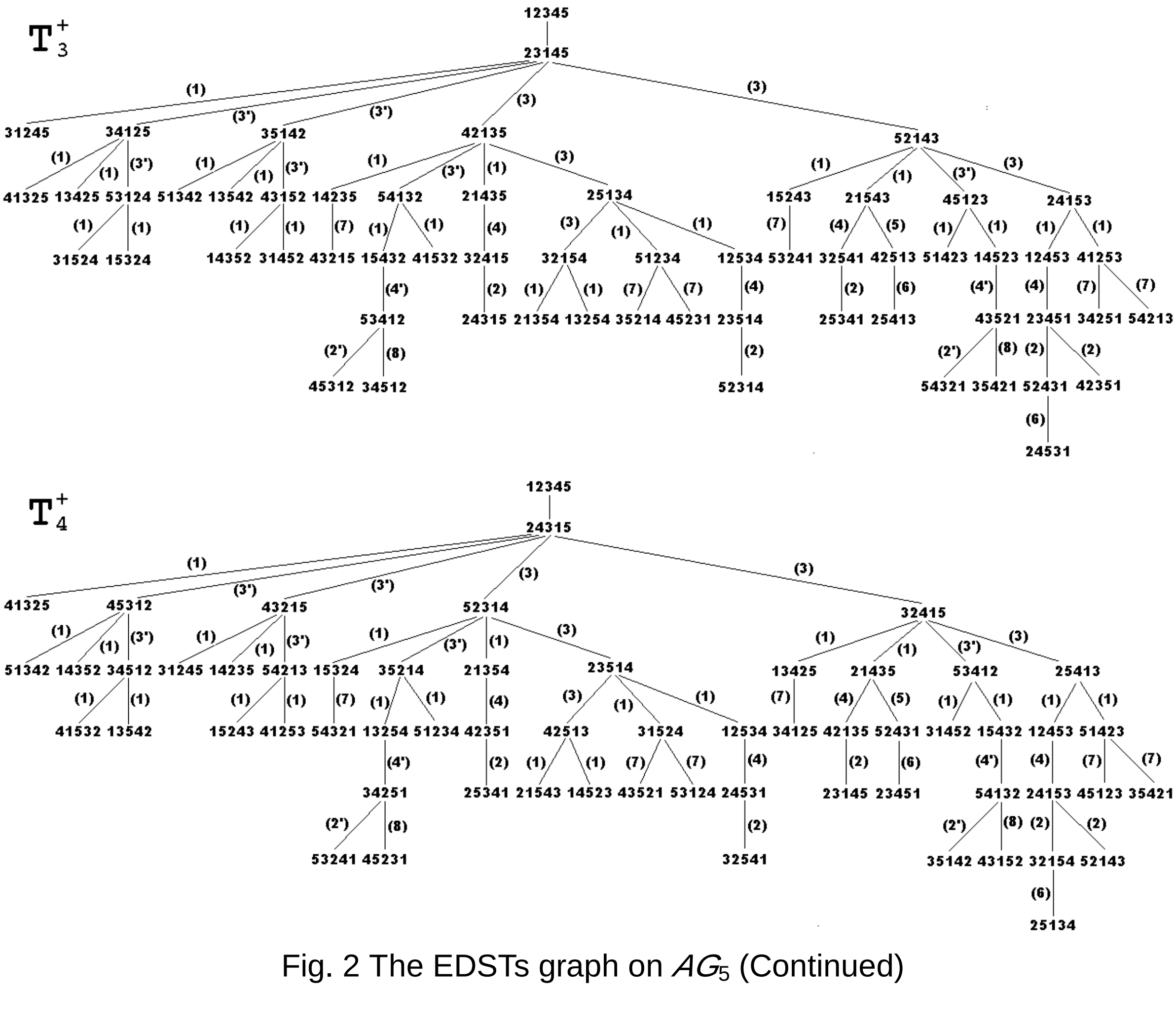


Fig. 2 The EDSTs graph on $AG_5$ (Continued)

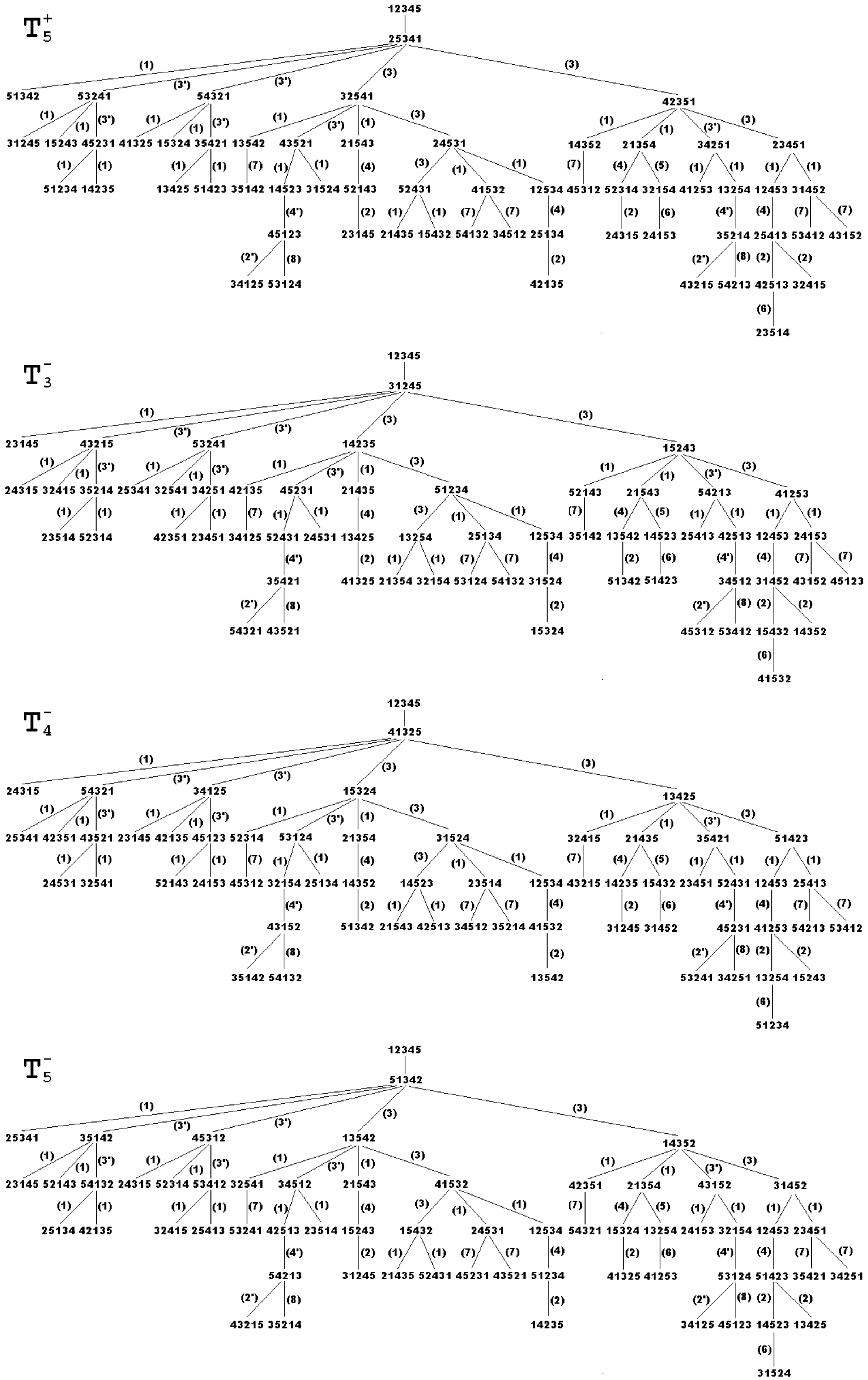


Fig. 2 The EDSTs graph on $AG_5$